

THE SUBMERGED HERITAGE: A VIRTUAL JOURNEY IN OUR SEABED

V. Barrile, A. Fotia*, E. Bernardo

DICEAM –Department of Civil, Energy, Environmental and Material Engineering - Mediterranean University - Località
Feo Di Vito 89124 Reggio Calabria (Italy) - antonino.fotia@unirc.it

Commission II, WGII/9

KEY WORDS: Rov, 3D models, virtual reality

ABSTRACT:

On 21 and 22 August 1972, a young diver near the coast of Riace Marina (South Italy) found two bronze statues representing one of the highest moments of the sculptural production and the most important archaeological discovery of the last century: the Riace Bronzes. The Geomatics Laboratory of the Mediterranean University of Reggio Calabria sets a goal to relive and re-propose the experience of a real immersion and the exciting moment of the discovery: an immersive journey in the interest of interactivity in virtual reality. Immersive Virtual Reality tries to combine the real and the virtual world, even involving our senses. In other words, it allows us to enter into the scenery like the virtual reality one, but without being just simple spectators. We become protagonists and we can participate and decide what to do and how to do it. Unity 3D is the multi-platform development environment used. The scenarios and the objects included in the scene from three-dimensional models were realized through photogrammetric techniques (seabed and bronzes). Remote Operated Vehicles (ROVs) have been widely used in recent times by researchers to explore underwater environments, both in shallow and deep water, for different types of studies. The seabed was detected through the aid of an experimental ROV (realized in a broader project of agreement with the IPCF CNR Messina). The use of a grid of known dimensions assisted the ROV's use in order to correct the effects of distortion effects obtained in water. The proposed app is still being perfected and completed.

1. INTRODUCTION

On 21 and 22 August 1972, a young diver near the coast of Riace Marina (South Italy) found two bronze statues representing one of the highest moments of the sculptural production and the most important archaeological discovery of the last century: the Riace Bronzes.

The Riace Bronzes are two full-size bronzes cast about V century BC, located at the Museo Archeologico Nazionale della Magna Grecia in Reggio Calabria. Discovered in the coast of Riace Marina (RC) in 1972, they actually represent significant sculptural masterpieces of the Greek art in the world, thanks to their outstanding manufacturer.

The Geomatics Laboratory of the Mediterranean University of Reggio Calabria sets a goal to relive and re-propose the experience of a real immersion and the exciting moment of the discovery: an immersive journey in the interest of interactivity in virtual reality.

Immersive Virtual Reality tries to combine the real and the virtual world, even involving our senses. In other words, it allows us to enter into the scenery like the virtual reality one, but without being just simple spectators. We become protagonists and we can participate and decide what to do and how to do it. Unity 3D is the multi-platform development environment used.

In this application, we obtained the scenarios and the objects included in the scene from three-dimensional models realized through photogrammetric techniques (seabed and bronzes). The 3D modelling is based on the use of imaging techniques, as digital photogrammetry and computer vision.

The use of photogrammetry and computer vision offers high accuracy, the ability to measure more details, less contact with the object and the construction of 3D textured models. As known, the direct underwater survey carried out by a human operator through trilateration methodology (objective difficulty of the diver in maintaining a stable attitude) involves the use of a large amount of time (which is always limited in this environment) and an accentuated inaccuracy due to human errors. Moreover, the Remote Operated Vehicles (ROVs) have been widely used in recent times by researchers to explore underwater environments, both in shallow and deep water, for different types of studies.

The most innovative technologies of recent years have led improving the quality of underwater surveys. They support the work of archaeologists even at depths where divers can work easily but with reduced immersion times.

The ROVs, if assembled with cameras, could make a survey photogrammetric in a single dive thanks to the high battery's life. For this reason, the seabed was detected through the aid of an experimental ROV.

* Corresponding author

A grid of known dimensions assisted the ROV's use in order to correct the effects of distortion effects obtained in water.

2. CASE STUDY

As part of broader activities concerning the control and monitoring of coastlines and seabed developments, in collaboration with the IPCF CNR of Messina, the Geomatics laboratory of Mediterranean University of Reggio Calabria is using an experimental ROV (submarine drone) to acquire underwater images.

The study area is located in the Riace area (South Italy), about 30 m from the shoreline, a place of discovery of the Riace Bronzes.

2.1 ROV

The ROV used to carry out the survey consists of two watertight chambers, the first of which is placed at the bow of the ROV containing both the control electronics and a full HD camera that can be used both for driving the device by the operator placed on the surface both for shooting and taking photographs of the backdrop; the second room instead is placed aft and contains the battery supply. The control electronics consists of a Raspberry PI 3 that deals with the communication with the surface, the video streaming and the automatic calibration operations. The ROV is characterized by various onboard sensors, which are a nine-axis IMU (three accelerometers, three gyroscopes, and three magnetometers), a pressure sensor (located outside the sealed chamber) and an integrated odometer in the tail thruster.

Thanks to the use of this experimental system, a photogrammetric survey of the backdrop was performed, displaying it in real time on a PC (placed on the boat) connected to the ROV by means of a suitable cable (Fig.1) (Barrile et al., 2018).



Figure 1. Experimental ROV used in the survey

2.2 Data acquisition

The image acquisition takes place under low natural lighting conditions. Additional light sources were used (ROV's light). Due to a number of difficulties during the acquisition process leading to low quality images (mainly motion blur), a selection process is applied in order to select the best images to ensure a most accurate result. A grid of known dimensions assisted the ROV in order to correct the effects of distortion obtained in water and also was used as GCPs in order to orient and scale the model in a reference system. (Skarlatos et al., 2012)

2.3 Image enhancement

As know, the underwater environment influences the way in which images are created, due to the water characteristic that deviates the optical rays an unpredictable way. (Henderson et al., 2013).

Moreover the underwater colours light depends on factors as the depth, the season, the water surface conditions, the time of the day, the cloudiness of the sky, the presence of marine vegetation or the soil type.

So the sunlight that penetrates from the air-water interface into the water is altered from several causes and more depth increases, more the lower temperature colours disappear.

The reflections coming from the surface create light games that alter the colours. The framed scene is generally flattened, not balanced and slightly contrasted. These effects are accentuated above all as the distance between the sensor and the object increases.

Digital cameras are able to balance colour images under a wide variety of illumination conditions by internal processes but these settings often fall in underwater environments (Lavest et al., 2003).

In order to improve accuracy, the estimated interior camera parameters (horizontal and vertical focal lengths, the coordinates of the principal point, the radial and tangential lens distortions in the Brown's model and the skew coefficient) were used as initial parameters in Agisoft alignment process to better perform the system convergence.

Moreover, the effect of refraction (depending on a number of parameters such as depth, temperature and salinity) can cause unstable modelling effects, so the intrinsic parameters of a submerged camera change significantly in comparison to the primary values.

In our case studies, clear and shallow water, good light conditions and a high-quality white balance allowed a good chromatic result.

In this article, two simple image enhancement techniques have been combined: improved overall contrast and adaptive equalization of the colour histogram (CLAHE). The first algorithm used for image enhancement is the contrast mask. The following algorithm is based on a photographic technique that allows the reduction of image blur and a consequent increase in detail. To improve the effect of the application of a contrast mask, the CIELAB colour space is used, which allows representing every colour perceivable by the human eye while maintaining the luminosity value (channel L), channel up, on a

single channel. The mask is applied to. From the L channel the starting image (L_{input}) is processed in order to obtain a more blurred version (L_{blur}). The two images are then combined according to the formulas:

$$L_{blur} = \text{medianFilter}(L_{input}) \quad (2.1)$$

$$L_{result} = 1.5 \cdot L_{input} - 0.5 \cdot L_{blur} \quad (2.2)$$

In 2.2 the weights of the images are empirical values extracted through experimental checks. The value of the original L channel is replaced with the result to recompose the image.

The CLAHE application was performed on all three channels of the RGB image separately and it allowed to improve the local contrast and add more detail. In fact, as known, the effect of CLAHE is to increase the contrast between colours in the areas of the image with different shades. (The adaptive method differs from the ordinary equalization of the histogram since it calculates different histograms, each corresponding to a distinct section of the image, and uses them to distribute the brightness values).

The defect found by the application of this methodology has been the local introduction of small artifacts in the image, without however affecting the performance of the final application. The two algorithms were applied in cascade (the first contrast mask and the CLAHE subsequently), providing, despite their simplicity, seemingly satisfying results for the type of application (Fig. 2, 3).



Figure 2 - Example of the application of a contrast mask. On the left the original image. On the right the result of the elaboration.

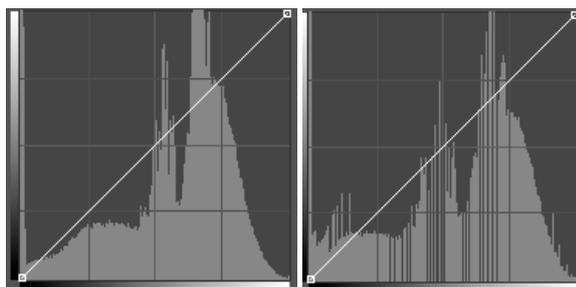


Figure 3: Histograms of the RGB channels. On the left the histogram of the original image in an underwater environment. On the right the histogram of the result of the elaboration.

2.4 Data processing

The acquired photographs were processed using the Agisoft Photoscan software and 3Dflow Zephyr, respectively for the realization of the 3D model of the seabed and for the bronzes models (Fig. 4, 5).

The workflow is completely automatic both as regards the orientation of the images and for the generation and reconstruction of the model. This condition led to an

optimization of processing times ensuring good performance of the machine/software complex (Balletti et al, 2015) (Canciani et al., 2003).

The phases of the elaboration were the following:

1. Align photos (photo alignment); This is the most important phase of the entire photogrammetric process. In this phase, the software aligns our photographs with each other, calculating their position in space and reconstructing the so-called gripping geometry. Then, through a process of geometric triangulation, calculate the position in space of the elements present in the photographs. It follows that the quality of this alignment derives the goodness of the final 3D model. The result of this phase is a sparse point cloud. Following the alignment, measurement of the control points takes place, to register the block in real world coordinates, in the reference system defined by the control points.
2. Build Dense Cloud; Through this phase, a dense cloud is constructed using dense image matching algorithms. These are subdivided into algorithms that use a stereo pair to find matches and those that identify them in multiple images;
3. Extraction mesh; it consists in generating a polygonal model based on the newly created dense cloud. The mesh is a subdivision of a solid into smaller solids of a polyhedral shape;
4. Build texture; it instead allows obtaining the 3D representation of the work under investigation.

The last step is the scaling of the model, that is the assignment of the correct metric dimension to the model, in order to make precision measurements on it (Remondino et al, 2012).

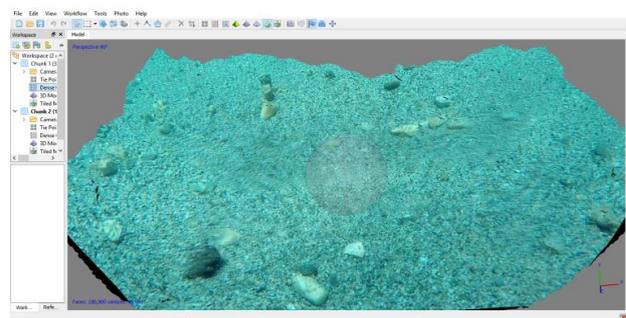


Figure 4. 3D model of Seabed



Figure 5. 3D model of Riace Bronzes

3. THE APP

3.1 Virtual Reality

The possibilities open with the use of mobile devices connected to the Internet as means of information available from anywhere and at any time have generated great interest (given the many possible applications) around the Virtual Tours and immersive experiences.

The advancement of information technology allows navigating in photorealistic settings in real time, interacting with the objects present in them.

Theoretically, the virtual reality is immersive virtual reality. In current use, however, this definition is applied to any type of virtual simulation; therefore, from the technological point of view, it is currently possible to distinguish two levels. The first (immersive) gives a measure of "perception" of the digital environment as existing (the user, equipped with appropriate peripheral devices, becomes the main interface with which to manipulate the 3D environment and experiences a sense of "belonging" to it); the second (interactive) introduces stringent requirements of feedback in real time, from a measure of "realism" of the simulation (the user has the impression of seeing the 3D world and the possibility of modifying it through an external interface) (Liarokapis et al., 2017).

Virtual reality systems are generated by the combination of the real scene seen by the user and the pre-registered digital virtual scene, which can be significantly influenced by the behaviour of the active subject. The subject can "assist" representation in a real space, or he can "explore" the real environment where, in any case, virtual elements are superimposed.

The distinctions between the two systems refer to the level of interaction of the subject with the real space is exploring; moreover, the real space can be related to the use or not of devices that act as an interface. (Drap et al., 2007)

When the user does not have any device, substantially he attends a representation, however, influenced also by his body presence only. Otherwise, the user uses one or more devices through which he explores the real space, receives the additional digital information on this, and realizes the display of augmented reality.

This can be realized through monitors (that can be semi-mobile, furniture, or even inserted into a pair of glasses—video see-through display—which worn, allow the user a sort of immersion) or glasses (optical see-through display) on whose transparent lenses are projected the virtual scene.

The main technical and technological problems are therefore referable to the integration between real natural and virtual vision, to the frequency of updating of the virtual scene, as well as, obviously, to the visual accuracy of this, which, in turn, must be realized in perfect adherence to the real space. (Fuchs et al., 2011)

3.2 Unity 3D and Vuforia

Unity is a cross-platform tool for creating interactive 3D content, such as architectural visualizations or 3D animations in real time, which uses two different programming languages (customized JavaScript based on NET and C ++ languages).

Unity's development environment works with both Microsoft Windows and MacOS and the apps produced can also run on Android mobile devices.

From an operational point of view, Unity's programming work is based on the use of objects: the so-called GameObjects. To these elements, which may or may not have a graphical representation, scripts can be associated, which allow defining their behaviour using particular event functions.

The integration with Vuforia has led this software to be much simpler and more complete to use for the implementation of augmented reality applications. The integration between real images and virtual information, based on artificial vision techniques, has as its aim the unification of the two reference systems taken into consideration: the virtual one and the real one.

3.3 The app created

The app, developed in Unity environment, is an academic tourist app for mobile device. Particularity of the app is that it allows the user in real time, to perform a virtual tour (with a viewer or simply through the screen of the device) having as main scene a 3D model (seabed) and then the user is able to explore it and interacting with the surrounding environment and other models (change the position of the models of the Bronzes to view the details).

Moreover, the app allows having additional information (AR) on the area of investigation (Historical information, multimedia...). It is an innovative way to bring out the archaeological artefacts and to place them into the discovered places (Barrile et al., 2018).

Users can interact with suspended hotspots to learn information in the surrounding environment or to move between the tours scenes.

In figure 6 the main phases, that led to the creation of the app, are reported and highlights, with particular reference to those where we operated in a non-standard way in order

- To select a 3d model as "main scene" (3d Seabed model)
- To move the 3d immersed (3d Bronzes) with natural movements (SCRIPT ASSOCIATION)
- To identify the user's position within the 3d model (3d seabed) (TRACKING PHASE)
- To Overlay multimedia contents with real objects (Replacing the real topographic coordinates of the model with those of the application's reference area).

The project consists of three scenes: an initial, loading and a scene for exploration, all managed by the SceneManagerScript script to perform initialization operations and the SceneLoader script that controls the transition from one scene to another.

-

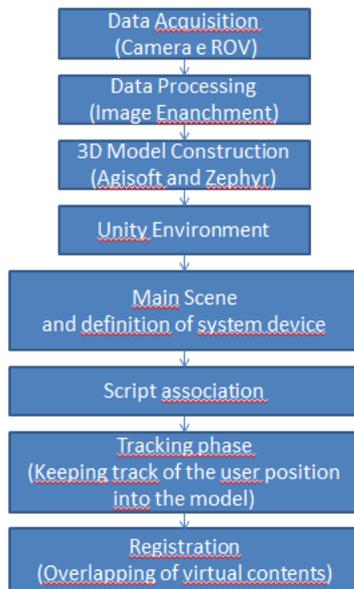


Figure 6. Flow chart app

3.3.1 Main SCENE

The first part of the project is focused on the creation of a single scene, the basic elements of a virtual tour. It contains all the information of an immersive environment. In fact, a virtual tour can be thought as a sequence of scenes.

In the initial scene the user can observe the study area in its entirety and select some explorable areas available to visit (seabed scene).

For each area that can be visited, a "Button" object is added to the exact position in which the corresponding background is located, in terms of latitude and longitude.

Then, there is a phase of the scene's initialization where at the start of the program, some preliminary operations are carried out:

1. Generation of the object Background: loaded at runtime using the Resources.Load function.
 2. Reading DTM files (3D seabed): for each area that can be visited, the corresponding DTM file is kept in the StreamingAssets project folder.
 3. Generation of the selection keys: for each file read a Button object is loaded and positioned on the backdrop to the corresponding latitude and longitude. To these is assigned a Listener that, once triggered the event, allows to perform the function that changes the scene.
- Moreover, in order to allow the user to move inside the scene, we built a 'point of view'.

Created the Point of View, we proceeded to create a panoramic photo (Blue Sea, and other elements in order to correct some defectiveness of the seabed model) that had to wrap the user 360 degrees in the scene. This photo had to come through a request to the server (RESTful service), because previously there is nothing saved on the device (Forte M, Kurillo G 2010). Once the image has been downloaded, the Skybox object will be used for displaying it (A Skybox is a container around the entire development scene).

3.3.2 Script association

Script association is the fundamental part of Unity3D and the work of the thesis project. Through JavaScript or C # is possible to execute both everything the software user interface offers (such as adding lights, changing the geometric transformations of a GameObject, importing a 3D model or associating a textures with a material), both to program new features and interactions between two GameObjects: i.e. the user can make a script to grab an object when you click on it and drag it while he is moving into the scene; he can create new shaders and extend the Editor. The main functions that can be used are: - the Update function, invoked to each frame; - the Start function invoked only to the first frame; - the Awake function invoked when the script instance is loaded (before Start). They are containers of important code that allow to choose when to execute that code.

When exploring an area it is important to be able to define some relevant points. A special class has been created that defines the so-called POIs: PointOfInterest; the properties that are accessible via this class are briefly described below:

- ID: a unique and ever-growing whole that identifies the POI
- Lat: latitude of the point on the terrain identified by the POI
- Lon: longitude of the point on the terrain identified by the POI
- PoiMarkerGameObject: object used as a marker of the POI to display it on the ground
- PoiMarkerImage: image associated with the POI marker
- PoiMarkerText: text associated with the POI marker
- PoiTextInfo: text containing the information associated with the POI
- Position: position in the Unity coordinates of the POI

The POIs are managed using a global variable called poiList () (Fig.7), that allows to associate each PointOfInterest object with its own ID, a number that is incremented each time a POI is defined. This data structure allows to keep an always updated list of the defined points of interest and to be able to access them using the point identifier as a key, facilitating its use.

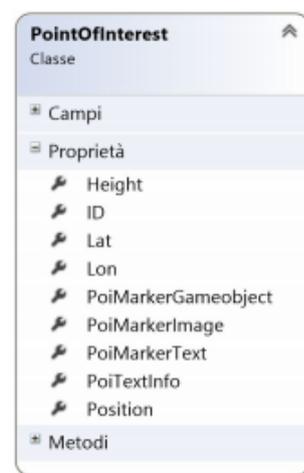


Figure 7. POI Propriety

So, obtained a scene with a totally explorable 3D model of the seabed, the necessity to create a viewfinder at the 'center of the eye' was born in order to point object and be able to start the Actions (Fig.8).

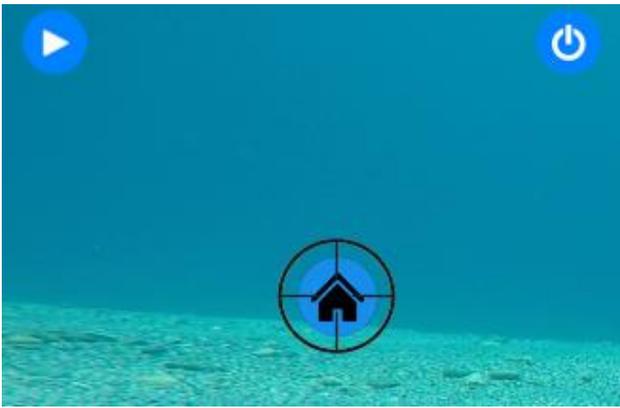


Figure 8. Viewfinder

The management of this part has been entrusted to the *Reticle.cs* script present in *Assets / Standard Assets / General Scripts*, which displays on the scene prefabs in the form of a crosshair built ad hoc. The programming work inside Unity is based on the use of objects: the so-called Game Objects.

Each object in the scene is a *GameObject*. It can ideally be thought of as an empty "container" within which the user can insert components (components) that will characterize it. Scripts associated to the "Game Objects" are extensions of the "MonoBehaviour" base class, which allow to define their behavior thanks to the use of particular functions called "event handlers".

These procedures are called automatically when certain circumstances arise, for example in the event of a collision with another object (*OnCollisionEnter*) or at the beginning of the life cycle of the object (*Start*), allowing the programmer to specify all the functions that they want to give to the item (Fig.9).

A collision calculation tool (*Collider*) is a component that can be associated with a *GameObject* (Unity elementary primitive or 3D model imported into Unity) that already has a *Rigidbody* component allowing it to collide: if two *GameObjects* collide between of them, and they both have the *Collider* and *Rigidbody* component, the physical engine will calculate the collision (Fig.10).

The Script "*ObjectInquadrato.cs*" was used to activate the *GameObjects*. This script is assigned to the observer element and has the function of "shooting the ray" of *Raycast*, starting from the object itself allowing to recognize to which Layer belongs the object that has been intersected by the latter. Based on the outcome of the operations described, the script will execute specific commands defined for each category (Layer).

The script is divided into two main methods that derive from the *MonoBehaviour* class and are *Start ()* and *Update ()*. The first one is executed only once during the execution of the application, more precisely when it starts, the second one is executed continuously during the execution of the application. For each frame the ray is "fired" which starts from the *transform.position* point (user position in a virtual environment) and proceeds for the direction of observation *forward*.

The *transform.position* value represents the scene coordinates of the *GameObject* to which the script is assigned, so in this case "*CenterEyeAnchor*" (the point at the center of the two cameras

representing the two ocular views). If the *Raycast* operation (Boolean operation) is successful, proceed to check which Layer the affected object belongs to.

When the "ObjectInteractive" layer and the *OVRPlayerController* layer are not moving, the *Update ()* method is entered. In addition to the various controls, the main task of this last piece of code (*Update ()*) is to check if the ray intersects the same object for more than three seconds. If this happens, the script will send a message to a script attached to the hit object (Fig.11).

When doing this, a timer must also be managed, which must be reset every time the beam no longer hits the object. The choice was made to spend three seconds before sending the message to the script that opens the description panel because, otherwise, every time the look crosses with some interactive object, the description panel would open immediately even if the user does not want it.

When the user wants to activate a new object, all the properties of the previous object must be automatically deactivated. In the event that the *Raycast* operation intersects the same object for multiple consecutive frames, messages are sent to the script attached to the hit object to deactivate the load animation and then continue with the operations in progress.

When the *Raycast* ray intersects with button to close the panel, the commands provided to execute the "Exit" Layer assigned. So through the code written with this conditions it is possible to deactivate and to activate interactive object, to close the description panel, to stop the sounds emitted and the animations associated with them.

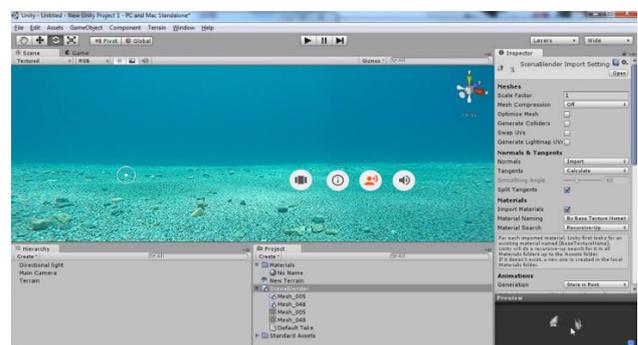


Figure 9 – unity environment



Figure 10 – Integration among the three 3d models in Unity environment



Figure 11 – Integration among 3d models in Unity environment.

3.3.3 Tracking processes and Registration in VR

Through the functionalities (location-based technologies like RFID and NFC, Nearby, means for the geolocation of the device (GPS) and QR code reader) that the portable devices intended for the app generally got, it's possible receive and transmit localized and contextualized information and it's possible also support users preferences by transmitting their personalized and specific information.

Compared to virtual environments, AR localization devices must be more precise, and are usually more complex. Tracking accuracy depends not only on the geometric model but also on the distance at which the object is located.

The localization technologies of the device are very useful for example when the exposure is very dense and there are many objects close together, in these situations the points of interest are many and without a solid hierarchy of information the visit would become confusing and inaccurate (Barrile et al, 2019).

Tracking processes (user trace into the model) and Registration (overlapping of virtual contents) exploited both the presence of markers (such as the logo of the municipality or the entrance sign to the beach) (Fig.12) in order to view the 3D model (or of its interior) with additional information, both the link between the coordinates in the real detected through the device GPS, and the coordinates of the 3D georeferenced model.



Figure 12 – example of association of virtual elements with real elements

Using the GPS coordinate, clearly, the virtual tour cannot be start inside the water so we choose a different area on the beach exchanging the real coordinates with virtual ones.

The link between real coordinates and virtual coordinates makes, for example, that once the app is started, whenever the user is within a pre-established range of the identified area, he will be informed immediately, by notification that he can make the virtual tour accompanied by a virtual guide that provides information. Specifically, the difficulty in developing the app was to ensure that the 3D model detected coincided with the development environment. In the Visualization phase, since the orientation of the axes of the whole model can vary from device to device, a control on the type of device used has been inserted in the code, which allows realignment of the axes in order to obtain, for example, a visualization of the scene congruent with reality.

A problem encountered is linked to the insertion of the coordinates (x, y), in fact in unity the coordinate system used is expressed in floating points and therefore the coordinates will have a limitation of 7 significant digits. This means that the more we move away from the source the more inaccuracy increases.

Regarding the possible actions that you want to make to a model in relation to the surrounding environments (for example you want to "take" the model of Bronzi and orientate it to visualize the details) it was important to note that objects that were moved position or rotated (using the Transform function) interpenetrated the other models (collider) as if no surface existed. To overcome this problem, it was necessary to add a Rigidbody component or move the object in another way with rigidbody.ApplyForce or rigidbody.ApplyTorque.

4. CONCLUSION

4.1 Future Development

One of the main problems that the realized app showed was the size in terms of MB. In fact, the excellent resolution of DTM files and 3D models of objects in the scenes, allows the generation of very detailed models, consisting of a large number of vertices and edges. It is precisely the quantity of vertices that need to be elaborated to increase the complexity of the application in terms of computational calculation; also negatively impacting the memory used. What is most influential in the management of "land" in Unity, however, are not the models in themselves, as much as the colliders that need to be added in order to detect collisions between objects from the physical engine.

Developing the app itself in a progressive web application (PWA) could be a great way to address this problem. We refer to a new structure or technology. In other words, this is a set of practical improvements to make a web app work similar to a desktop or mobile application. The goal is to have an experience similar or equal to that of a native mobile application, so that the user does not realize the difference that exists between a native application and a PWA.

PWAs can be defined as a kind of mix between web apps and native applications, allowing the first to install the application from a navigation window without having to enter the store and offering many benefits for both the developer and users .

As known, PWAs generally work faster than a website:

- They work without needing an internet connection.
- They are "responsive".
- the icon appears on the phone as in native apps

- allow the sending of push notifications and have access to some features of the telephone (GPS - Nearby).
- PWAs are indexed in search engines
- No need for installation from the store.
- They do not take up space in the memory of the mobile device since it is not necessary to download them.

4.2 Conclusion

The use of virtual tours associated with elements of geomatics in addition to the ability to better emphasize the particularities of a place and capture the visitor's attention (who will be fully involved in the experience and will stop and have time to deepen the exploration) it can be considered a valid support for the dissemination of the exploitation and dissemination of archaeological / cultural assets allowing to stimulate more the curiosity of the user or the visitor compared to a simple web page containing texts, images or videos.

The applied image processing technique, despite its simplicity, has allowed us to obtain acceptable results in relation to the purposes of the specific application made.

It is not the same thing to read about a book as exceptional as that of the discovery of the Bronzes or to relive and explore the places through a virtual visit. We would go from being spectators in virtual reality to protagonists thanks to the immersive virtual reality!

REFERENCES

- Balletti, C., Guerra, F., Scocca, V., and Gottardi, C., 2015, 3d integrated methodologies for the documentation and the virtual reconstruction of an archaeological site, *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, XL-5/W4, 215-222, doi:10.5194/isprsarchives-XL-5-W4-215-2015, 2015.
- Barrile V., Fotia A., Bilotta G., 2018, Geomatics and augmented reality experiments for the cultural heritage in *Applied Geomatics* Vol. 10, Issue 4, pp. 569-578
- Barrile, V., Fotia, A., Bilotta, G., & De Carlo, D. 2019 Jan 28. Integration of geomatics methodologies and creation of a cultural heritage app using augmented reality. *Virtual Archaeology Review*, 10 (20), pp. 40-51.
- Barrile V, Fotia A, Siclari M, Aliotta F, Ponterio R ., 2018 Metodologie della geomatica per l'analisi e lo studio degli interventi di salvaguardia delle linee di costa. In: Atti della XXII Conferenza Nazionale ASITA. ATTI CONFERENZA NAZIONALE ASITA, p. 93-100, Confederazione ASITA, ISBN: 978-88-941232-1-0, Bolzano (Italy), 27-29 Novembre 2018
- Canciani, M., Gambogi, P., Romano, F.G., Cannata, G. and Drap, P., 2003, Low Cost Digital Photogrammetry for Underwater Archaeological Site Survey And Artifact Insertion. The Case Study of the Dolia Wreck In Secche Della Meloria Livorno-Italia. *Int. Arch. Photogram. Remote Sens. Spatial Inf. Sci.*, XXXIV- 5/W12, 95-100.
- Drap, P., Seinturier, J., Scaradozzi, D., Gambdogi, P., Long L., Gauch, F. 2007. Photogrammetry for virtual exploration of underwater archeological sites. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, XXXVI- 5/C53.
- Forte M, Kurillo G (2010) Cyberarchaeology: experimenting with teleimmersive archaeology. In *Virtual systems and multimedia* (VSMM 2010), Seoul
- Fuchs P, Moreau G, Guitton P (2011) Virtual reality: concepts and technologies. CRC Press, Inc., Boca Raton
- Henderson J., Pizarro O., Johnsons-Roberston M. and Mahon I., 2013, Mapping Submerged Archaeological Sites using StereoVision Photogrammetry, *The International Journal of Nautical Archaeology*, 42.2, 243-56.
- Lavest, J. M., Rives, G., Lapreste, J. T., 2003, Dry camera calibration for underwater applications, in *Machine Vision and Applications*, Vol.:13, pp.245-253.
- Liarokapis, F., Kouřil, P., Agrafiotis, P., Demesticha, S., Chmelík, J., and Skarlatos, D., 2017 3D MODELLING AND MAPPING FOR VIRTUAL EXPLORATION OF UNDERWATER ARCHAEOLOGY ASSETS, *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, XLII-2/W3, 425-431, <https://doi.org/10.5194/isprs-archives-XLII-2-W3-425-2017>.
- Remondino, F., Del Pizzo, S., Kersten, P.T., Troisi, S., 2012. Low-cost and open-source solutions for automate image orientation - a critical overview. *Lect. Notes Comput. Sci.* 7616:40–54.
- Skarlatos, D., Demestihia, S., Kiparissi, S., 2012, An "open" methods for 3D modelling and mapping underwater archaeological sites. *International Journal of Heritage in the digital era*. Vol. 1, no. 1, March 2012. DOI: 10.1260/2047-4970.1.1.1. Available from: <http://multiscience.metapress.com/content/q603163q0k861431/>. [Accessed: 17th April 2014]