

MARKER-LESS MOBILE AUGMENTED REALITY APPLICATION FOR MASSIVE 3D POINT CLOUDS AND SEMANTICS

A.Kharroubi^{1,*}, R.Billen¹, F.Poux¹

¹ Geomatics Unit, University of Liège, Allée du six Août, 19, 4000 Liège, Belgium - (akharroubi, rbillen, fpoux)@uliege.be

Commission II, WG II/3

KEY WORDS: Augmented Reality, Web-AR, Marker-Less AR, 3D Point cloud, Semantics, Classification, Virtual environment

ABSTRACT:

Mobile Augmented Reality (MAR) attracts significant research and development efforts from both the industry and academia, but rarely integrate massive 3D dataset's interactions. The emergence of dedicated AR devices and powerful Software Development Kit (ARCore for android and ARKit for iOS) improves performance on mobile devices (Smartphones and tablets). This is aided by new sensor integration and advances in computer vision that fuels the development of MAR. In this paper, we propose a direct integration of massive 3D point clouds with semantics in a web-based marker-less mobile Augmented Reality (AR) application for real-time visualization. We specifically investigate challenges linked to point cloud data structure and semantic injection. Our solution consolidates some of the overarching principles of AR, of which pose estimation, registration and 3D tracking. The developed AR system is tested on mobile phones web-browsers providing clear insights on the performance of the system. Promising results highlight a number of frame per second varying between 27 and 60 for a real-time point budget of 4.3 million points. The point cloud tested is composed of 29 million points and shows how our indexation strategy permits the integration of massive point clouds aiming at the point budget. The results also gives research directions concerning the dependence and delay related to the quality of the network connection, and the battery consumption since portable sensors are used all the time.

1. INTRODUCTION

Mobile augmented reality essentially consists of superimposing virtual elements over real objects on the screen, to give added information and enrich the interaction with reality. It improves how we acquire, understand, and display information and geo-data without distracting us from the real world. Recently, with the emergence of advanced camera systems and more precise sensors in mobile devices, augmented reality has completed the transition from image-based or QR code activations to marker-less AR experience (Fraga-Lamas et al., 2018; Shih et al., 2019). Consequently, several AR applications exploiting 3D data are appearing such as the projection of 3D mesh catalogues of furniture in real set ups (Khairnar et al., 2015; Şahin and Togay, 2016). However, these 3D models are often very constrained (availability, complexity, time for production) to permit an experience that depends on the available resources of the used device. Moreover, the semantic information - essential to allow pure geometries to be usable by decision-making systems (Poux et al., 2018; Poux and Billen, 2019) - is often inexistent or unused.

To open on new AR applications, our paper investigates new ways to better integrate massive 3D datasets and semantics through a marker-less mobile AR application. We propose to incorporate unstructured 3D data such as 3D point clouds directly within AR frameworks. This type of data has several unique properties worth mentioning. First, they look real to even the most casual observer because of their 3D nature. Secondly, they permit to geometrically describe objects of any shape and scale

with an adapted resolution and point density. Finally, point clouds are generated by a wide variety of sensors permitting applications spanning from robotics (Parra et al., 2020) architecture, engineering and construction (Stojanovic et al., 2019) to cultural heritage (Pierdicca et al., 2020). For these unique properties, point clouds increasingly constitute a very important support for decision-making and support for semantic integration (Poux et al., 2017). Therefore, ways to do this through an adapted data structure and web mobile augmented reality set-ups are explored. While some works investigate point clouds integration with semantics in virtual reality (VR) (Fraiss, 2017; Kharroubi et al., 2019; Mures et al., 2018), to the best of our knowledge there exist no Web-AR frameworks to manage point clouds efficiently.

The rest of the paper is structured as follows: in section 2, we discuss the principals of Mobile AR and its mechanisms, and different Web-AR implementation approaches. Section 3 mainly discusses the methodology and gives details on the segmentation, classification and structuration approach for real-time semantically rich AR experiences. In section 4, we develop the main components of AR applications using web frameworks and investigate several visualization methods that leverage point clouds and semantics. A discussion of the achieved results is given in section 5, and finally the conclusion of the work and the presentation of future works are shared, highlighting interesting research directions to identified challenges.

* Corresponding author

2. AR TECHNOLOGY OVERVIEW

2.1 Brief history of Augmented Reality

The phenomenal growth of AR over years attracted significant research and development efforts (Khoshelham et al., 2019; Qiao et al., 2019). By integrating virtual contents with the real world, AR makes possible to provide users with a sensory experience beyond reality. AR is today used in many domains such as entertainment (Paavilainen et al., 2017), advertisement (de Ruyter et al., 2020), education (Karakus et al., 2019), navigation (Hübner et al., 2020), building maintenance (Codina et al., 2019), and Building Information Modeling (Raimbaud et al., 2019). In fact, both AR and Virtual Reality (VR) can alter the perception of our presence in the world. While VR transposes our presence from reality to virtuality, AR allows users to be present in the real world and simply “augments” their perception of the world by adding the ability to enhance user’s immediate context through digital data.

Beginning from the first time Sportvision applied AR to live TV, and then with the first dedicated AR device (Google Glass, 2012) and smartphones (Google Project Tango, 2014), on to the first phenomenal AR App (Pokémon GO, 2016), it has become clear that both AR technologies and devices tend to be powerful, mobile, and lightweight. However, the current mobile augmented reality hardware and operating systems (e.g., Android and iOS) present a complex diversity. Most mobile AR applications or solutions are designed based on a specific platform and lack cross-platform support.

To reach more users, an AR application needs to go through repeated development cycles to accommodate different platforms, which undoubtedly increases the cost of development and deployment. Although there are already, some preliminary attempts toward web-based mobile AR (Web-AR), the limited networking and computing capability greatly hinder its practical application. Since 2017, the Web-AR provisioning solution has gradually attracted developers’ attention again due to the ever-increasing development of user device and mobile network and has emerged as a promising direction for Mobile AR. The invention of the World Wide Web marked the beginning of this new era, with a Web-based service-provisioning paradigm. Mobile web-AR deploys the sufficient computational resources of a server and the GPU-accelerated rendering of WebGL API to accommodate its services to user’s devices. For more of the steps of the augmented reality evolution, a brief history of AR is shown in Figure 1.

Although the technology of the Web offers a promising approach for the cross-platform, lightweight, and pervasive service provisioning of Mobile Web-AR, there are still several challenges to applying Web-AR in real cases. Computational efficiency of browsers, energy efficiency of the mobile devices, and networking are three important challenges. AR is a computation- and data-intensive application. The limited computing and rendering capabilities on the Web make it more challenging to achieve a high-performance and energy-efficient Web-AR (Qiao et al., 2019).

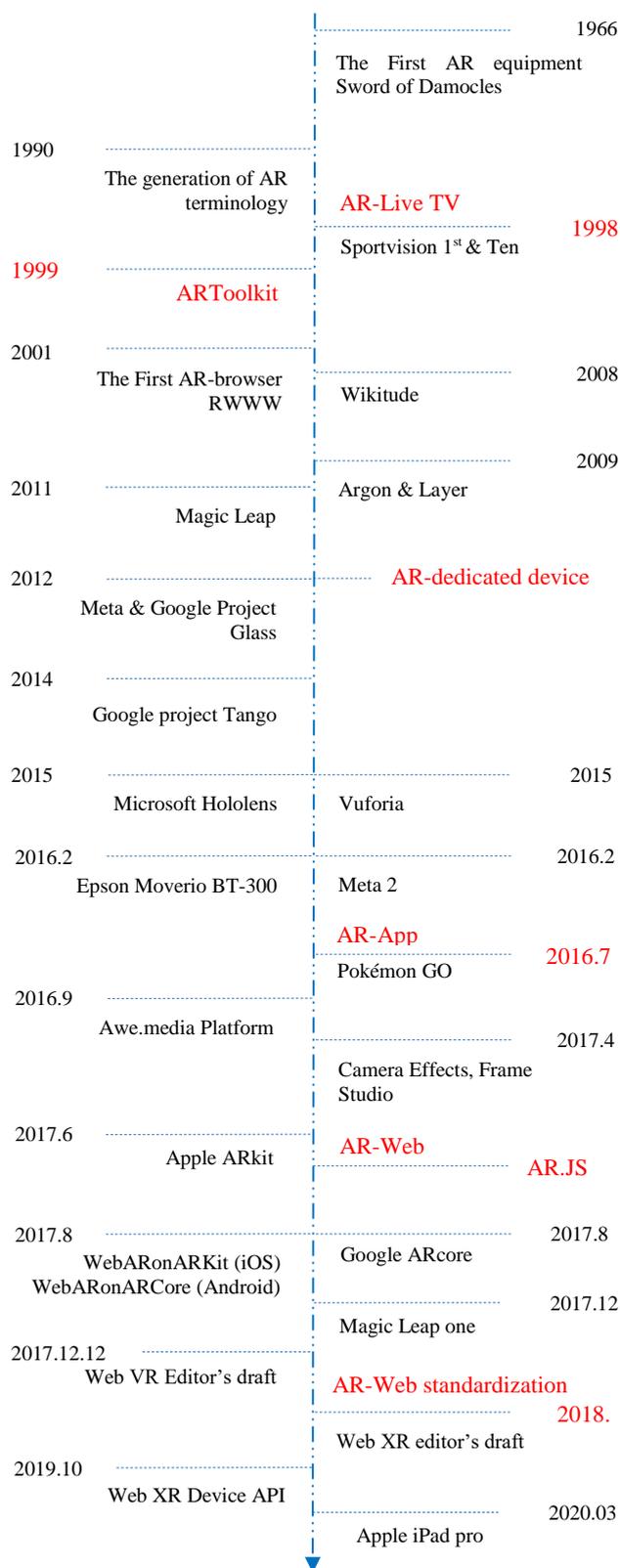


Figure 1. Brief history of AR

2.2 Principals of mobile AR

A typical AR process is shown in Figure 2. The camera and other types of sensors are used to continually gather user ambient information. The environment perception based on computer vision and machine-learning algorithms analyses the captured

information (e.g., image, location, and orientation) for real-world recognition and perception. In the meantime, the user's interaction information is additionally gathered by the sensors then analysed for tracking objects. Both the results of the perception of the environment and the interaction are used for a seamless integration of virtual contents with the real world, i.e., a rendering operation is performed, after which the AR is presented to the user.

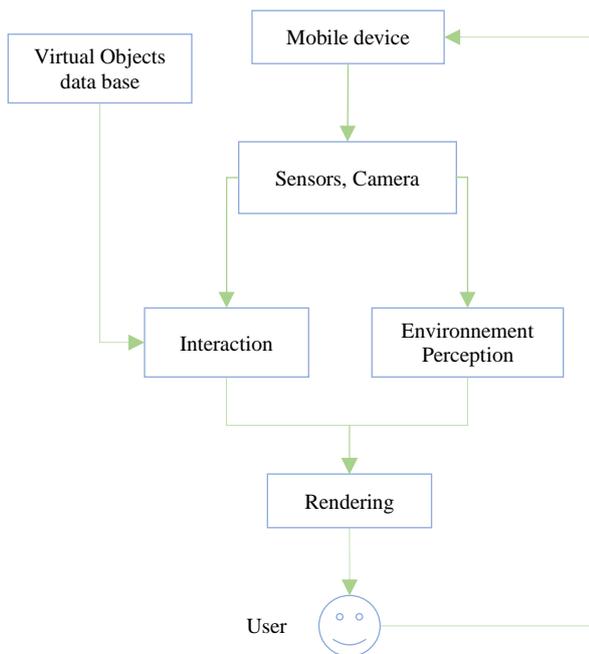


Figure 2. Augmented reality simplified process

2.3 Typical Implementation Mechanisms

We divide the typical MAR implementation mechanisms in three aspects, that is, sensor-based, vision-based, and hybrid methods. These implementation mechanisms differ in terms of computing, networking, and storage. First, the sensor-based method could be a relatively lightweight Mobile AR implementation approach, while the vision-based approach places high demands on the computing and storage capabilities of the runtime platform, furthermore as network capability. The hybrid tracking mechanism is clearly a compromise solution between both the first and the second solution.

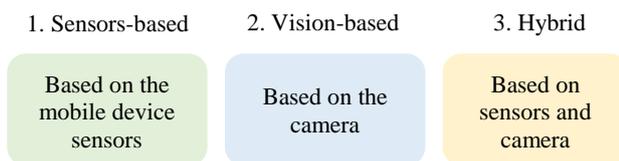


Figure 3. MAR implementation mechanisms

Each of these approaches in Figure 3 is detailed below:

1) Sensor-based mechanism: Mobile devices nowadays already support a variety of sensors, such as GPS, gyroscopes, magnetometers, compasses, accelerometers, and so on. Note that the camera can be enabled to capture the surrounding environment, but only for the display of the environment as the background. In addition to Mobile AR implementation mechanisms based just on a single sensor, combining different

sensors (two or more) allows many applications to achieve more accurate tracking results. The increasing of sensor category, as well as the continuous enhancement of sensor functionality, provides the basis and opportunities for the diversification of AR applications (as location based augmented reality application). Considering the complexity of computation, storage, and networking, this lightweight MAR implementation mechanism is currently the lowest option for users to get started. However, this method works in an open-loop way, which will result in an unavoidable cumulative error, since the tracking error cannot be evaluated and corrected in real time (an inside-out tracking).

2) Vision-based mechanism: Similarly, to (1), the camera on the device captures the surrounding environment, but it further provides the basis for vision-based object recognition, feature detection, and tracking. This type of mechanism uses feature correspondences to estimate pose information to align the virtual content with real-world objects and is analogous to a closed-loop system. This frame-by-frame tracking approach avoids the mentioned error accumulation. However, it introduces heavy computational pressure on mobile devices, especially for natural feature tracking methods. Depending on different features, it can be divided into the following methods:

- The marker-based method uses a predefined marker to satisfy the tracking requirement.
- The markerless method detects and understands real-world environment (e.g., the locations of walls), no pre-knowledge of the environment is required, which will efficiently promote large-scale Mobile AR.

3) The hybrid tracking mechanism: This is a compromised solution. It overcomes the weaknesses and limitations of the individual methods mentioned earlier by combining different methods. It not only provides MAR applications with convincing precise and robust results, but also reduces the computational complexity. Considering the limited computing capability and network performance, this hybrid scheme will play an important role in the availability of MAR on a larger scale, and we will base our approach on this implementation mechanism.

2.4 Different Web-AR implementation approaches

Web-AR, as a branch of MAR, has recently attracted a great deal of attention due to its lightweight and cross-platform features. Depending on the different computing paradigms, we can classify the Web-AR implementation approaches into two types as follows.

- Self-contained method: it executes all tasks on the mobile device locally. The advantage of this method is that it is less dependent on mobile networks, so the real-time tracking performance will not be degraded by additional communication delay. However, the inefficient computing capability of the mobile device becomes its fatal flaw; current mobile devices still cannot carry out these tasks very well.
- Computation outsourcing method: it leverages the computation and storage capabilities of the cloud servers, and it can usually provide a better user experience than the mentioned self-contained one. However, this method has a strong dependence on the mobile networks, and therefore, the performance of Web-MAR applications is easily affected by network conditions, as we will show in this work.

3. DEVELOPED METHODOLOGY

Our methodology is subdivided into three main parts, each of them covering several sub-parts as illustrated in Figure 4. The first part covers the point cloud processing, starting with segmentation (3.1), then classification into specific classes (3.2), and then structuration in an octree format for the purpose of real time visualization (3.3). The second part concerns the creation of an immersive session; it begins with the verification of the compatibility of the device with the specifications of augmented reality, then the estimation of the pose of the device based on the sensors. Then the detection of characteristic points that constitutes a planar surface is used for anchoring the point cloud. The last one is the visualization and interaction step, it consists of an iterative verification of the different nodes of the octree which are present in the user's field of view and sends a request to the remote server for their loading and anchoring in the anchor point chosen by the user. This complete workflow is summarized in the following figure and implemented in Section 4.

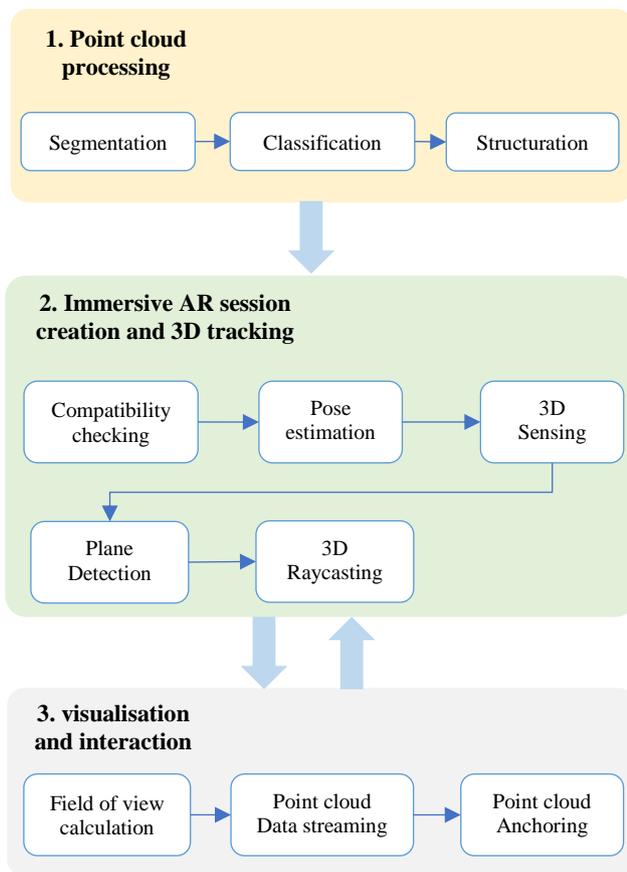


Figure 4. A general workflow of the followed methodology

3.1 Segmentation

Segmentation is the process of labelling each point of the point cloud, so that these points belonging to a particular surface or region have the same label. The segmentation is done under CloudCompare software (CloudCompare, 2019). We started by segmenting the point cloud into subspaces as defined in (Poux et al., 2017) (e.g. rooms for building), then segmenting each subspace into smaller significant segments corresponding to classes (walls, doors, windows, chairs, offices, closets, etc.) as in Figure 5. The segmentation is done semi-automatically with the selection tools available on CloudCompare, assisted by the

following automatic plugins: RANSAC Shape Detection (Wahl and Klein, 2007), CSF based on the cloth simulation filter developed by (Zhang et al., 2016), Histogram filtering, Label Connected Components, and CANUPO.

3.2 Classification

In this step, we assign for each created segment a specific class pointer. First, a classification list is established regrouping several indoor classes and outdoor classes that holds a great potential for several applications. Then, a new scalar field with a constant integer value is created named Classification, in order to respect the specifications of the export format that is the “.las” format. This point cloud format supports this attribute by default. A class number is assigned to this created classification field.

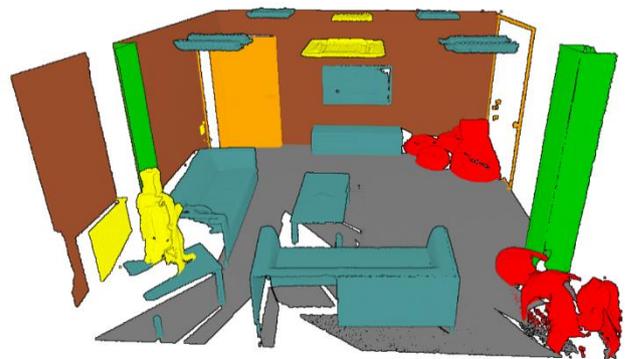


Figure 5. Classification results: a piece with different objects.

3.3 Structuration in a Potree's octree structure

The main purpose of a data structure is to permit an easier/better access to the underlying data.

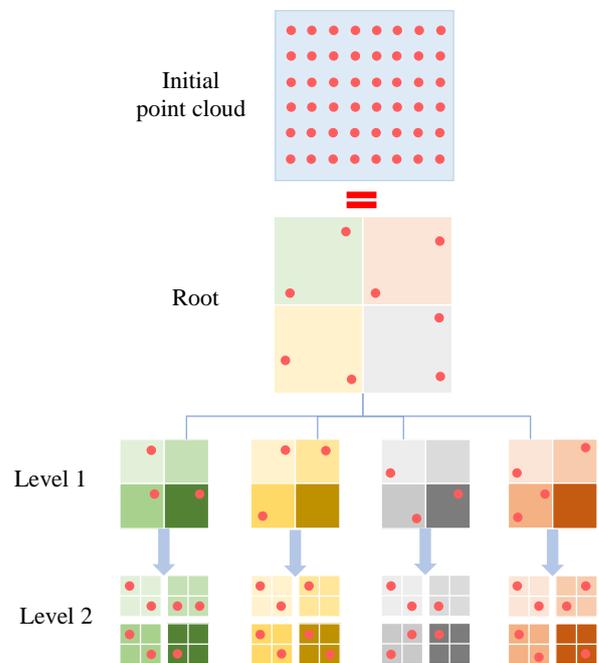


Figure 6. Potree's octree structure explained in 2D.

As such, we orient ourselves by spatially indexing massive point clouds based on a Potree's octree data structure for real time

purposes. Potree uses a variation of the modifiable nested octree structure with a different subsampling method and a partition of the hierarchy into smaller, quickly streamable, chunks (Martinez-rubi et al., 2015). The point cloud is iteratively subdivided into eight cubic parts (nodes in a octree) by starting from an initial bounding box with an initial spacing. Then each part is subdivided in the same way and the spacing is halved to increase the density. The resolution of a node is defined by the spacing property, which specifies the minimum distance between points. Thus, as the depth of the tree increases, the spacing decreases and the level of detail increases, as shown in Figure 6.

4. IMPLEMENTATION

We classify the point cloud and we structure it in Potree's Octree structure using Potree Converter 1.7 (Potree, 2020). The next step is to proceed to the web-based augmented reality implementation with the JavaScript Application Programming Interface (API) Three.js and by exploiting the WebXR API.

- Three.js is a cross browser JavaScript library used to create and display animated 3D computer graphics in a web browser; it is built upon WebGL standard and has a decent documentation and examples for getting started in WebXR. This library provides three mains things: scene, camera, and a renderer function.
- WebXR device API is the successor to webVR API, the acronym 'XR' stands as a placeholder for saying 'VR and AR', using 'X' as a sort of variable. WebXR provides access to input and output capabilities commonly associated with virtual reality and augmented reality devices. Its goal is to detect if XR capabilities are available, query the XR device capabilities, poll the XR device and associated input device state, and display imagery on the XR device at the appropriate frame rate.

Our solution is based on a hybrid tracking mechanism: it exploits the sensors present in the mobile device and the camera for the feature detection. Its advantage is the use of a computation outsourcing approach that takes advantage of storage and processing resources on cloud servers. The second big advantage of this solution is the storage of the structured point cloud on a remote server. It allows finding the nodes that are present in the user's field of view through queries on the server. Therefore, there is no need to store massive point clouds on the mobile device and only display the part of the point cloud that interests us.

The code is done in JavaScript based on several scripts, which allow reading the file cloud.js, which contains the metadata, the .hrc files containing the hierarchy, and the .bin files which contains the nodes in binary format. Indeed, this process begins with the opening of an immersive session in augmented reality via an ARButton which allows to launch the camera to reconstruct the surrounding environment (which must be a heterogeneous surface to detect the features points), and to obtain information from the sensors of the mobile device for the estimation of camera pose. The environmental understanding is achieved via an inside-out tracking, where sensors on the tracked device itself (cameras and other sensors) are used to determine the device's position and orientation.

Then, a Hit-test script is launched to allow making a raycast with the reconstructed surface to find a horizontal surface to anchor the 3D point clouds. After that, a search is done in the Octree to find the nodes, which are in the field of view of the camera, so that these nodes are loaded progressively from the hosting server. Each point is associated with a PointCloudMaterial that can

either be its value in RGB, the intensity, or a color that corresponds to the identifier of the class to which it belongs. Moreover, to have a shape more visible to the user, each point is made in the form of either square or circle with a variable pixel size according to the choice of the user.

5. RESULTS

The results of this work is a workflow allowing optimized management and display of massive 3D point clouds on a web based AR without having either to install an application or having to store the 3D point clouds on your mobile device (Figure 7). Admittedly, web-based augmented reality remains very dependent on the internet, but it allows outsourcing of the computation that are on a remote server, and of data that is also stored on a server.

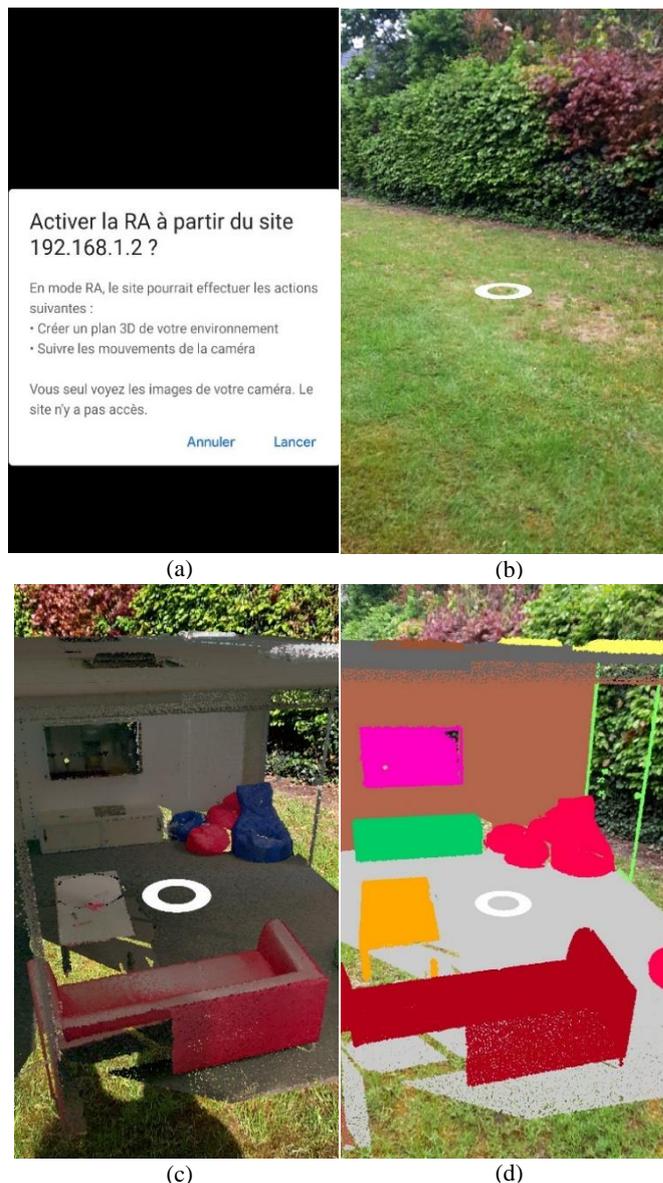


Figure 7. Result of a test with a point cloud; (a) creation of augmented reality session, (b) Hit-test to choose the anchor point, (c) point cloud in real color, (d) display in classes colors.

The evaluation is carried out on a device with the technical characteristics, which are summarized in Table 1:

Mobile	Exploitation system	Navigator version	RAM	Battery Capacity
Samsung Galaxy A7(2018)	Android 9	Chrome Beta 83.0.4103.34	4Go	3300mAh

Table 1. Technical characteristic of the device used

The application shows decent performance in terms of network delay that does not exceed hundred milliseconds. The number of images per second exceeds 27, but from time to time decreases even if the graphics are rendering at close to 60 frames per second. We noticed a high consumption of the battery with a drop exceeding 10% per hour, for a usage of 10h maximum on the tested phone. In addition, we note a larger delay (100 ms) at the start when loading the point cloud for the first time, which could be optimized.

6. CONCLUSION AND FUTURE WORKS

In this paper, we have reviewed the story of augmented reality and its mechanisms while showing different Web-AR implementation approaches. We propose a methodology that starts with a semantic injection step through semi-supervised segmentation and classification for the definition of different elements present in the point cloud with different algorithms. Then, to permit a real-time interaction on mobile devices, we structure massive point clouds through an adapted hierarchical Potree's Octree structure. Finally, we define the main components of AR applications with Three.js and WebXR and investigate several visualisations that leverage point clouds and semantics, opening on a new human-machine interface. The evaluation was made with a point cloud of around twenty-nine million points, but the application can support much bigger datasets as it is linked to the number of points displayed on the screen (point budget), and not the size of the original point cloud.

In future work, we will investigate a more interactive User Interface (UI) for better interactions and manipulations on point clouds and semantics. This UI can for example manage searching queries based on displayed semantics and better integrate natural language processing. In addition, we will focus on the major challenges that arise when using web-based augmented reality, such as the Network Delay (we will study the effect of 5G on its improvement and minimization), with higher bandwidth (0.1~1 Gb/s) and lower network delay (1~10 ms), which can certainly improve the data transmission on mobile networks.

ACKNOWLEDGEMENTS

The authors are thankful to the developers of CloudCompare, Potree, Three.js and WebXR API. We also thank a lot the users of the Github forum, especially the authors of three.js that helped us in several steps of this research.

REFERENCES

CloudCompare, 2019. CloudCompare 3D point cloud and mesh processing software Open Source Project www.cloudcompare.org (25 december 19).

Codina, Castells-Rufas, Carrabina, Salmon, Ayuso, Guerendiain, Alvarez, 2019. Augmented Reality for Emergency Situations in Buildings with the Support of Indoor Localization. *Proceedings* 31, 76. <https://doi.org/10.3390/proceedings2019031076>

de Ruyter, K., Heller, J., Hilken, T., Chylinski, M., Keeling, D.I.,

Mahr, D., 2020. Seeing with the Customer's Eye: Exploring the Challenges and Opportunities of AR Advertising. *J. Advert.* 0, 1–16. <https://doi.org/10.1080/00913367.2020.1740123>

Fraga-Lamas, P., Fernandez-Carames, T.M., Blanco-Novoa, O., Vilar-Montesinos, M.A., 2018. A Review on Industrial Augmented Reality Systems for the Industry 4.0 Shipyard. *IEEE Access* 6, 13358–13375. <https://doi.org/10.1109/ACCESS.2018.2808326>

Fraiss, S.M., 2017. Rendering Large Point Clouds in Unity. <https://www.cg.tuwien.ac.at/research/publications/2017/FRAIS-S-2017-PCU/>

Hübner, P., Clintworth, K., Liu, Q., Weinmann, M., Wursthorn, S., 2020. Evaluation of hololens tracking and depth sensing for indoor mapping applications. *Sensors (Switzerland)* 20. <https://doi.org/10.3390/s20041021>

Karakus, M., Ersozlu, A., Clark, A.C., 2019. Augmented reality research in education: A bibliometric study. *Eurasia J. Math. Sci. Technol. Educ.* 15. <https://doi.org/10.29333/ejmste/103904>

Khairnar, Khushal, Khairnar, Kamleshwar, Mane, S., Chaudhari, R., 2015. Furniture Layout Application Based on Marker Detection and Using Augmented Reality. *Int. Res. J. Eng. Technol.* 2, 540–544. <https://www.irjet.net/archives/V2/i7/IRJET-V2I780.pdf>

Kharroubi, A., Hajji, R., Billen, R., Poux, F., 2019. Classification and Integration of Massive 3D Points Clouds in a Virtual Reality (Vr) Environment. *ISPRS - Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* XLII-2/W17, 165–171. <https://doi.org/10.5194/isprs-archives-XLII-2-W17-165-2019>

Khoshelham, K., Tran, H., Acharya, D., 2019. Indoor mapping eyewear: Geometric evaluation of spatial mapping capability of hololens. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. - ISPRS Arch.* 42, 805–810. <https://doi.org/10.5194/isprs-archives-XLII-2-W13-805-2019>

Martinez-rubi, O., Verhoeven, S., Meersbergen, M. Van, Schuetz, M., van Oosterom, P., Gonclves, R., Tijssen, T., 2015. Taming the beast : Free and open-source massive point cloud web visualization. *Capturing Real. Forum* 2015 23–25. <https://doi.org/10.13140/RG.2.1.1731.4326>

Mures, O.A., Jaspe, A., Padrón, E.J., Rabuñal, J.R., 2018. Virtual reality and point-Based rendering in architecture and heritage. *Virtual Augment. Real. Concepts, Methodol. Tools, Appl.* 1, 549–565. <https://doi.org/10.4018/978-1-5225-5469-1.ch026>

Paavilainen, J., Korhonen, H., Alha, K., Stenros, J., Koskinen, E., Mäyrä, F., 2017. The Pokémon go experience: A location-based augmented reality mobile game goes mainstream. *Conf. Hum. Factors Comput. Syst. - Proc.* 2017-May, 2493–2498. <https://doi.org/10.1145/3025453.3025871>

Parra, C., Cebollada, S., Paya, L., Holloway, M., Reinoso, O., 2020. A novel method to estimate the position of a mobile robot in underfloor environments using RGB-D point clouds. *IEEE Access* 8, 9084–9101. <https://doi.org/10.1109/ACCESS.2020.2964317>

Pierdicca, R., Paolanti, M., Matrone, F., Martini, M., Morbidoni, C., Malinverni, E.S., Frontoni, E., Lingua, A.M., 2020. Point cloud semantic segmentation using a deep learning framework

for cultural heritage. *Remote Sens.* 12, 1–23.
<https://doi.org/10.3390/rs12061005>

Potree, 2019. PotreeConverter.
<https://github.com/potree/PotreeConverter> (22 December 19).

Poux, F., Billen, R., 2019. Voxel-based 3D Point Cloud Semantic Segmentation: Unsupervised Geometric and Relationship Featuring vs Deep Learning Methods. *ISPRS Int. J. Geo-Information* 8, 213. <https://doi.org/10.3390/ijgi8050213>

Poux, F., Neuville, R., Hallot, P., Billen, R., 2017. MODEL FOR SEMANTICALLY RICH POINT CLOUD DATA. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* IV-4/W5, 107–115.
<https://doi.org/10.5194/isprs-annals-IV-4-W5-107-2017>

Poux, F., Neuville, R., Nys, G.-A., Billen, R., 2018. 3D Point Cloud Semantic Modelling: Integrated Framework for Indoor Spaces and Furniture. *Remote Sens.* 10, 1412.
<https://doi.org/10.3390/rs10091412>

Poux, Florent, Neuville, R., Van Wersch, L., Nys, G.-A., Billen, R., 2017. 3D Point Clouds in Archaeology: Advances in Acquisition, Processing and Knowledge Integration Applied to Quasi-Planar Objects. *Geosciences* 7, 96.
<https://doi.org/10.3390/geosciences7040096>

Qiao, X., Ren, P., Dustdar, S., Liu, L., Ma, H., Chen, J., 2019. Web AR: A Promising Future for Mobile Augmented Reality-State of the Art, Challenges, and Insights. *Proc. IEEE* 107, 651–666. <https://doi.org/10.1109/JPROC.2019.2895105>

Raimbaud, P., Lou, R., Merienne, F., Danglade, F., Figueroa, P., Hernandez, J.T., 2019. BIM-based mixed reality application for supervision of construction. 26th IEEE Conf. Virtual Real. 3D User Interfaces, VR 2019 - *Proc.* 1903–1907.
<https://doi.org/10.1109/VR.2019.8797784>

Şahin, D., Togay, A., 2016. Augmented reality applications in product design process. *Glob. J. Humanit. Soc. Sci.*
<https://doi.org/10.18844/gjhss.v0i0.288>

Shih, N.J., Diao, P.H., Chen, Y., 2019. ARTS, an AR tourism system, for the integration of 3D scanning and smartphone AR in cultural heritage tourism and pedagogy. *Sensors (Switzerland)* 19. <https://doi.org/10.3390/s19173725>

Stojanovic, V., Trapp, M., Richter, R., Döllner, J., 2019. Generation of approximate 2D and 3D floor plans from 3D point clouds. *VISIGRAPP 2019 - Proc. 14th Int. Jt. Conf. Comput. Vision, Imaging Comput. Graph. Theory Appl.* 1, 177–184.
<https://doi.org/10.5220/0007247601770184>

Wahl, R., Klein, R., 2007. Efficient RANSAC for Point-Cloud Shape Detection 0, 1–12. <https://doi.org/10.1111/j.1467-8659.2007.01016.x>

Zhang, W., Qi, J., Wan, P., Wang, H., Xie, D., Wang, X., Yan, G., 2016. An easy-to-use airborne LiDAR data filtering method based on cloth simulation. *Remote Sens.* 8, 1–22.
<https://doi.org/10.3390/rs8060501>