Creating AR Applications for the IOT : a New Pipeline

Barnabé Edoh Barnabé Soedji, Jérémy Lacoche, Eric Villain

Creating AR Applications for the IOT : a New Pipeline

Ablam Edoh Barnabé Soedji
Orange Labs
Rennes, France
barnabe.soedji@orange.com

Jérémy Lacoche
Orange Labs
Rennes, France
jeremy.lacoche@orange.com

Eric Villain
Orange Labs
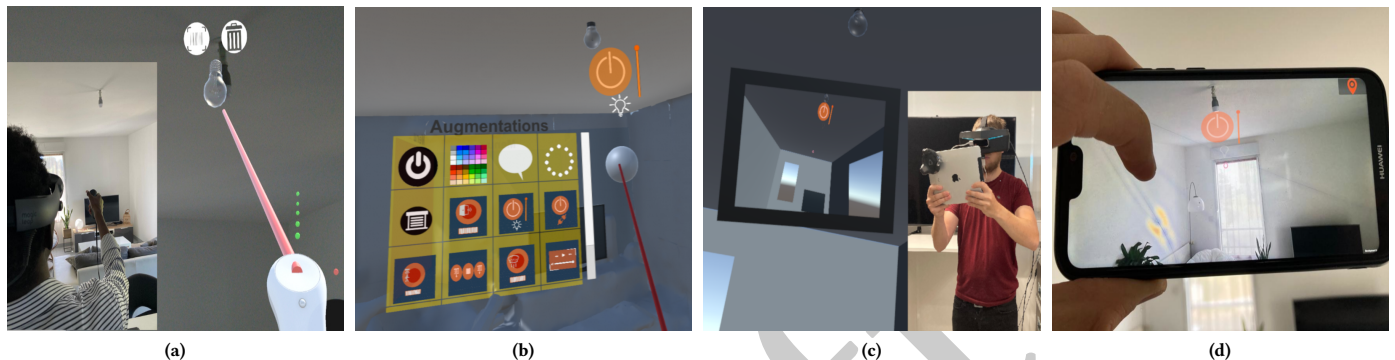Rennes, France
eric.villain@orange.com

Figure 1: The creation pipeline is composed of 4 steps: example of an augmented light bulb (a) A digital twin of the smart environment is captured (b) Augmentations are configured in the digital twin with VR interactions (c) The created application can be tested with an AR in VR simulator (d) The application can be deployed on AR devices.

ABSTRACT

Prototyping Augmented Reality (AR) applications for smart environments is still a difficult task. Therefore, we propose a pipeline to help designers and developers to create AR applications for monitoring and controlling indoor environments equipped with connected objects. This pipeline starts with the capture (geometry and objects) of the real environment with an AR device. Then, it proposes a Virtual Reality (VR) tool to configure augmentations in this captured environment. This tool includes a feature to simulate AR devices to help anticipate the application's rendering on real devices. The created application can then be seamlessly deployed on various AR devices including smartphones, tablets and glasses.

CCS CONCEPTS

• **Human-centered computing** → **Interactive systems and tools**; *HCI theory, concepts and models*; *Virtual reality*.

KEYWORDS

IoT, Augmented Reality, Virtual Reality

1 INTRODUCTION

The Internet of Things (IoT) is the network of connected objects: sensors and actuators that surround us in our daily lives. It is the core component of smart environments we all tend to live in: smart-homes, smart-buildings, smart cities. In the IoT context, Augmented Reality (AR) can be seen as a powerful tool to monitor and control smart-environments [1]. However, developing such applications is still a difficult task as it raises various issues such as context acquisition, 3D registration according to the connected objects, content adaptation for various AR devices, etc.

Pfeiffer et al. [3] propose to prototype such applications in VR thanks to 360 panorama captures. The result is assessed with an AR simulation tool in VR. However, 360 panorama lacks depth information to correctly create a 3D layout of augmentations. Moreover, it only proposes a static view of an environment and cannot simulate how connected objects would react to interactions such as turning on a light. Likewise, we introduced in a previous work [2] a real-virtual continuum for the creation of automation behaviors in smart-environments. It relies on a 3D capture of the real environment and it introduces a Presentation-Abstraction-Control (PAC) software-based model to manage seamlessly simulated and real connected objects. However, this solution does not target AR.

We then propose to extend this previous work to support the creation of AR applications for indoor smart environments by improving the approach proposed in [3]. The result is a four-step pipeline that includes 3D capture of the environment, prototyping of the augmentations in VR, AR simulation in VR, and deployment of the created application to various AR devices.

2 CREATION PIPELINE

Our pipeline is developed within Unity and relies on the PAC model proposed in [2] in order to ease the transition from simulation to deployment. Our PAC-based SDK includes a collection of existing connected objects such as doors and motion sensors, appliances switches, shutters, light bulbs, etc. Our current VR platform for prototyping and simulation is composed of a Pimax 5K, HTC Vive controllers and trackers, and a leap motion. Target AR devices include ARCore smartphones and tablets, Hololens and Magic Leap.

2.1 Digital Twin Capture

We capture a digital twin of the environment with a Magic Leap AR headset. First, we capture the geometry of the environment. Two options are given to the end-user. First, it can be captured with the same approach as in [2] by pointing the walls and the openings, using the plane detection and ray-casting features of the device. The second approach proposes to use the triangulated point cloud captured by the device. While the first approach is fast and provides semantic information, the second takes more time but provides a more complete result. Second, the user places the connected objects in the room thanks to a 3D menu and a 3D ray-based interaction technique manipulated with the Magic Leap controller. We chose this interaction technique as it allows users to place out of reach objects such as a lightbulb on a ceiling as shown in Figure 1a. We also use a virtual keyboard to assign a unique identifier to each object to be able to communicate with it at runtime. Third, 2D image markers in the room are identified by the user. They are used for relocalization during the deployment step.

2.2 Augmentations Prototyping in VR

We then propose a VR editor to associate each connected object with a 3D augmentation as seen in Figure 1b. Contrary to directly prototyping in AR, using the digital twin in VR allows a developer to remotely create and iterate on his application while keeping in mind the 3D physical constraints of the real environment. Our VR-based tool also allows us to develop for different kinds of AR devices. Augmentations need to be previously created with the Unity User Interface (UI) system and exported as a Unity AssetBundle. Thanks to our previously developed PAC-model, these UIs can seamlessly communicate with simulated and real connected objects. The VR editor imports the digital twin captured during the first step and displays the augmentations loaded from the AssetBundle in a 3D menu. The user can then use a 3D ray-based interaction technique to add the augmentations to the scene and associate them with each object. They can be precisely positioned, rotated and scaled. A billboard mode can also be enabled per augmentation.

2.3 AR Simulation in VR

As in [3], we then propose an AR simulation mode inside VR to assess the result. For instance, a designer could realize that his/her augmentations are too large regarding the target device field of view before deploying its application.

Binocular AR glasses (Hololens, Magic Leap, etc.) can be simulated as well as handheld devices (smartphones and tablets). Simulation parameters such as field of view, tracking latency and occlusion management (On or Off) can be edited at runtime. The Pimax VR headset that we use has the advantage of providing a large field of view in order to tend towards the aspect of a real AR experience. As shown in Figure 1c, a real tablet can be co-localized with the one in the virtual world in order to provide tangible tactile interactions. We provide state-of-the-art AR interaction techniques to interact with augmentations including head pointing, ray-casting (controller or hand-based thanks to leap motion), tactile interactions as well as vocal commands. These interactions are configured in an XML file and we use our own device abstraction layer to make them compatible with VR and AR devices. Thanks to the PAC software model, these interactions trigger the same behaviors on the simulated connected objects as they would on the real ones.

2.4 AR Deployment

In order to deploy the application to AR devices, we developed several AR players that can take into account: the AssetBundle of augmentations, configuration files of the digital twin, of the augmentations and of the interactions. These players are available for ARCore devices as shown in Figure 1d, Magic Leap and Hololens. All include a relocalization step where the user needs to target a 2D image marker identified during the capture in order to correctly place the 3D elements. Then, it deploys the PAC component of each object configured to communicate with the real objects as described in [2]. To continue, it instantiates the augmentations configured with the parameters edited in VR. Last, interaction techniques are provided to the end user in order to interact with the augmentations. They are chosen by the developer before deployment as some of them are not compatible with all platforms. For instance, tactile interaction is only available on handheld devices while hand-based selection is only available on AR headsets.

3 CONCLUSION AND FUTURE WORK

We introduce an end-to-end solution for the creation of Augmented Reality Applications for supervising and controlling indoor smart-environments. It includes a capture step in AR, a prototyping loop in VR and a deployment step for targeting various AR devices.

Future improvements include the capacity to deal with other types of augmentation than Unity UI objects such as HTML elements. The relocalization step could also benefit from more recent work on camera localization [4] in order to get rid of the 2D image markers. Two user experiments are planned regarding this work. First, we aim to evaluate if our solution fits the application developers' and designers' needs. Second, we aim to determine if the end-users would give the same feedbacks on an application in our VR simulator and on a real AR device.

REFERENCES

[1] Dongsik Jo and Gerard Jounghyun Kim. 2016. ARIoT: scalable augmented reality framework for interacting with Internet of Things appliances everywhere. *IEEE Transactions on Consumer Electronics* 62, 3 (2016), 334–340.
[2] Jérémy Lacoche, Morgan Le Chenechal, E Villain, and A Foulonneau. [n.d.]. Model and Tools for Integrating IoT into Mixed Reality Environments: Towards a Virtual-Real Seamless Continuum. In *ICAT-EGVE 2019*. Tokyo, Japan.
[3] Thies Pfeiffer and Nadine Pfeiffer-Leßmann. 2018. Virtual Prototyping of Mixed Reality Interfaces with Internet of Things (IoT) Connectivity. *i-com* 17, 2 (2018), 179–186.
[4] Nathan Piasco, Désiré Sidibé, Cédric Demonceaux, and Valérie Gouet-Brunet. 2018. A survey on visual-based localization: On the benefit of heterogeneous data. *Pattern Recognition* 74 (2018), 90–109.