Thesis Title： VRSimulator: connect real and virtual IoT devices

List of your readings and references：

Number of references：12

Main references：

[1] Myeong-in Choi, Lee Won Park, Sanghoon Lee, Jun Yeon Hwang and Sehyun Park, "Design and implementation of Hyper-connected IoT-VR Platform for customizable and intuitive remote services," 2017 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, 2017, pp. 396-397, doi: 10.1109/ICCE.2017.7889368.

[2] Simiscuka, Anderson Augusto & Muntean, Gabriel-Miro. (2018). Synchronisation Between Real and Virtual-World Devices in a VR-IoT Environment. 10.1109/BMSB.2018.8436742. – Their framework is simple (what is good), but there is no good realization of it. The future work is continued in the article of the next year by same author.

[3] Simiscuka, Anderson Augusto & Markande, Tejas & Muntean, Gabriel-Miro. (2019). Real-Virtual World Device Synchronization in a Cloud-enabled Social Virtual Reality IoT Network. IEEE Access. 7. 106588-106599. 10.1109/ACCESS.2019.2933014. – This is an extension of the previous article with a better explanation of the platform. The main problem the authors tried to solve was how to control IoT device as if they are located far from the owner. The framework provided is not complex, with a virtual device structure containing just a several fields. In the demo of the project too few devices are used, but the authors are planning to add more types of the IoT devices supported in the next work, which I hope to be published in 2020. The creating of the room in Unity and adding the devices in it with a functionality of controlling them is a part of this research goals, so this article is considered to be useful as a reference.

[4] Begault, Durand. (2001). 3-D Sound For Virtual Reality and Multimedia. – In this paper by National Aeronautics and Space Administration (NASA) worker HRTF (Head-related transfer function) magnitude data in numerical form for 43 frequencies between 0.2---12 kHz is given. Although this research is pretty complex and consists of 247 pages, it seems to contain much information needed to build a feasible sound integration in the VR platform proposed in this research.

[5] Mohapatra, Sukant & Bhuyan, Jay. (2016). A Solution Framework for Managing Internet of Things (IOT). International journal of Computer Networks & Communications. 8. 73-87. 10.5121/ijcnc.2016.8606. – Having a system architecture based on cloud technology which provides security, privacy and scalability in managing IoT is one of the goals of creating the platform. The architecture model proposed in the article may be useful in developing this topic platform.

[6] Onoriode, Uviase & Kotonya, Gerald. (2018). IoT Architectural Framework: Connection and Integration Framework for IoT Systems. Electronic Proceedings in Theoretical Computer Science. 264. 1-17. 10.4204/EPTCS.264.1. – This paper reviews existing architectural frameworks for integrating IoT devices and identifies the key areas that require further research improvements. This is very useful for the research and helped in finding the existing implementations of IoT in AR environment (but not VR). Secondly, the paper concludes by proposing a possible solution based on

microservice. The solution is not implemented and the author has not published other paper on the future work, but the adapted SOA paradigm he focused on will be considered in this research.

[7] Dongsik Jo & Gerard Jounghyun Kim (2016): ARIoT: scalable augmented reality framework for interacting with Internet of Things appliances everywhere. IEEE Transactions on Consumer Electronics 62(3), pp. 334– 340, doi:10.1109/tce.2016.7613201. – The AR in IoT framework the authors developed looks interesting and should be further considered, although the way it works is different from the platform proposed in this thesis. Using the mobile device as an AR client is similar to using VR glasses.

[8] https://www.youtube.com/watch?v=Y7YyMK7r7a4 – This is footage of a VR prototype. In this video person activates and displays real world sensor data from the IOT devices. Although the author could not manage to find the source code or any further explanation of the technology used (besides the brief video description), and thus assumes that the video may have been edited post-factum, it is still a good example of how the visualization proposed in this thesis can be implemented.

[9] https://www.softwareag.com/en_corporate/platform/iot.html – One of the biggest providers of service of IoT integration into business (as written on their website). Using their platform, a big amount of data from the connected IoT devices can be analyzed and then turned into actionable insights. Parts of the architecture and design in this project may be useful in implementing the proposed thesis topic. Since it is a corporate product, it will be more difficult to find demonstration, screenshots and examples of using this platform, compared to the articles, but author hopes to retrieve such information in the future.

[10] https://internetofthingsagenda.techtarget.com/feature/Understand-how-to-apply-SOA-for-IoT - I am using SOA definition taken from this article as if it is easy to understand, but later I will use scientific articles, not just a webpage.

[11] Mark Richards (2016): Microservices vs. service-oriented architecture. O'Reilly Media

[12] Jeong, Yuna et al. "AVIoT: web-based interactive authoring and visualization of indoor internet of things." IEEE Transactions on Consumer Electronics 61 (2015): 295-301.

[13] https://circuitstream.com/blog/unity-vs-unreal/

[14] https://www.techtimes.com/articles/248296/20200324/video-oculus-vr-for-all-this-developer-built-real-life-virtual-reality-room-while-stuck-at-home.html

Thesis Work Schedule

Outline of the thesis and schedule of thesis writing.
The importance & impact of the chosen topic, Motivations and objectives of my Thesis, expected outcomes of my project, my approach for conducting this research:

Virtual reality is used in many fields: from healthcare to games, from microbiology to the space exploration. It is used in education and in business, in engineering and in sales. Many people have VR devices, such as Google Cardboard and its analogues, gaming VR headsets at homes. But even if they don't, most of them have smartphones, which have accelerometers and gyroscope and can be alternatives to VR headsets.

While using IoT is already considered profitable by many businesses, and automated control saves time and money for companies, the problem of scalability and connection between heterogeneous IoT devices are the most important problems in this field, and the second one is not solved yet.

Several companies attempted to create a unification for IoT devices, but still the problem of connectivity is not solved.

The frameworks existing nowadays which try to solve this issue seem to be usually not user-friendly and not easy-to-use. Some of them use AR and mobile phone as a client, some of them just provide a 2D-interface, but the problem exists: there is no solution which can be user-friendly and easy to understand to a regular person, who does not know about IoT much.

Let's provide an example: a person bought a new apartment, and now wants to make his home smart. There are three possible solutions:

- The first one is to buy a set of IoT devices provided by one company. In this case the person first needs to wait for the devices to arrive, then needs time to install them, and after, if he/she is not satisfied with the quality of these IoT devices, he/she will return them back to the store and start over again.
- The second one is to use one of the existing frameworks. Since they are not user-friendly, more probably the person will decide to wait for a better framework to be created.
- The third one is this thesis' solution: in ideal state, the person scans his apartment, the 3d-model of his apartment is created in Unity automatically. Then the person can place virtual IoT devices inside the apartment, and try to achieve the best configuration according to his/her needs.

Another example: The person has already bought several IoT devices, for example: smart speaker as a hub for these devices, smart scaler which is used on daily basis, smart toothbrush, smart vacuum cleaner and smart air conditioning. There are two rooms in the apartment, and the person wants to buy a TV to watch it in the living room. For example, some TVs such as Huawei's new TVs can perform a function of smart hub and be the center of smart home. For this configuration, the responsibilities of being the center of the ecosystem can be shared between TV and smart speaker. The smart speaker is in the hall and cannot understand what command the person gives if he/she is located in another room, and the person can not hear the feedback of the speaker. The solution is: in the VR environment person adds a TV in the living room, and can try to test the functions of a smart TV as a second smart hub. For example, he can see how the TV shows an info of the toothbrushing session after the person comes back to the room or how the person sets a reminder using the smart speaker which then is visualized on the TV screen (like a TO-DO task).

There are many other examples of how to use the framework, but the result is the same: the solution seems promising, and after the framework and platform are created, and the user study proves how useful it is, then it can be used by regular customers and businesses and become a standard.

Further the possible implementation will be discussed, and since the developing of the whole platform is a complex task and can not be performed in a short period of time, it will be first implemented on the level of proof-of-concept for several tasks, such as: creating 3D-model of apartment and adding the models of devices (which is done in [3], although no source code has been provided), adding a basic framework for adding the virtual devices and their interconnection, extending the devices structure to use physical devices, adding a server to store the state of the IoT ecosystem, and, finally, creating the VR IoT framework.

To start with, the architecture of the platform basically consists of three parts [3]:

- Abstraction of Sensors: each of the sensors is defined but the attributes in the structure, such as position in the world space, real or virtual, the type and its state.
- The integration platform: this is a framework to support the connection between all of the sensors.
- The visualization platform: it is used to show the devices in the Virtual Reality 3D-space.

The IoT basic architecture consists of three layers [6]: Physical Sensing Layer, IoT Middleware and Application Layer. The sensors gathering real world data are a part of the “Thing”. The embedded devices make use of these sensors and through the gateway mechanism send this data to the second layer. In the second layer the data is analyzed and sent to the application layer (and in the backward direction as well – from application layer to the Physical sensing layer). In the application layer the commands are sent to the real world using the apps on the mobile phones, smart speakers, personal computers, servers and smart hubs.

But setting up the sensors in a traditional architecture model brings less advantages, compared to if service-oriented architecture was used. Service-oriented architecture for IoT is subtly different from the basic architecture and focuses on software functionality that can be reused for different tasks. In the SOA it is first defined how to the IoT application should interact with the outside world, compared to the technical aspects of the basic model. For example, a vehicle location system in a vast yard facility should locate vehicles, not read sensors. “The request for vehicle location is then pushed down toward the hardware. The vehicle has a position and the things in the yard have a position. After the hardware identifies the positions, the application architects focus on how they know it. In this case, the sensors establish those positions.

The IoT service is then designed to deliver functionalities that fulfill the requirements; in this case, the positions. The service's outputs - their API definitions - fulfill the original mission, but also related missions that are derived from the same information. The devices themselves are invisible, which gives this model some increased benefits.”.[10].

Using SOA for IoT, the services will present harmonic view of how sensors and controllers work, so differences in the devices or changes in them won’t affect the logic of the system.

Let’s explain it using the same situation: the person has already bought several IoT devices, for example: smart speaker as a hub for these devices, smart scaler which is used on daily basis, smart toothbrush, smart vacuum cleaner and smart air conditioning. The next thing to buy is smart TV. If we use SOA architecture, connecting smart TV as a smart hub will not be a difficult task as if there is already a task defined “if Smart TV (smart hub) added, then activate new device connection”. The service will do the rest.

Expected difficulties encountered in your project and the methods you intend to adopt to overcome those obstacles:

The main problem with SOA is that it is not scalable. So, having many services may slightly slow down the performance if the number of devices in the system is relatively high. Defining the architecture for IoT in VR this disadvantage of SOA for IoT has to be also considered.

Other disadvantages to be considered are: lack of standardized description of services, poor context-awareness of services, poor device service classification and poor information visualization and analysis [6]. In ideal state, the VR IoT platform is supposed to solve every of these issues.

Also, the VR IoT platform should follow these requirements:

- Contract decoupling: it is an ability of service customers and service producers to independently evolve without terminating the contract between them [11]. If the output file format needs to be changed, the platform must be able to handle it.
- Scalability
- Ease of testing. Putting the helmet on and off every time to test the platform does not sound relaxing
- Fault tolerance: framework should effectively handle faults.
- Platform should run smoothly.

There are other requirements, proposed in [6], but following them is not the aim of the VR IoT platform.

Methods/Tools/Implementations used in my research and arguments:

In the next paragraph the author will try to discuss the vision of how the platform will look like (very basic explanation):

The architecture created for the device must be an extension of the basic architecture with added services and a layer for VR integration.

The VR visualization should be a different layer since it is a heavy instrument and the performance of it must not affect the performance of IoT system. To decrease the latency, several methods are proposed in [3], which will be considered in future work.

- 3D Engine: from the variety of 3D engines presented on the market, there seem to be only two suitable for the realization of platform (apart from creating our own): Unity3D and Unreal Engine. 60 percent of AR/VR content is made with Unity3D [13], so the first engine is more popular, but the second should be considered as an alternative as if the latency may be reduced. Although authors of [3] also use Unity3D as part of their platform.
- Room creation: There are several methods of creating a room by scanning it with depth and RGB cameras, but they still don't provide competitive results. The author has worked with Kinect API before and to use such functionality of representing a room by scanning it does not provide acceptable results (at least for these years). But at the same time, the scan can be useful in creating a 3D-model of the room manually (by recreating it following the created 3D-mesh).
- Another problem is how to interact with the IoT devices. In [3] a good method was not proposed, as if they just use a pointer for turning the light switcher on and off. But since IoT device provide many ways of interacting with them, such as touchscreen, voice, tactile

buttons, smell, heat, humidity etc., for each of the sensors the simulation of it in VR needs to be defined and implemented.

- The location service will be responsible for real-time coordinates mapping of the IoT device in real time to the world coordinates in the virtual reality.
- The interaction service will be responsible for running the sensing in real and virtual worlds.

Location and interaction services:

One of the possible uses of the platform is the research of the best placement of IoT devices in the room based on several factors, such as: voice input / output signal level, WIFI / 5G network connection stability etc. In this case, the location mapping of real IoT devices from the real world to the virtual world should be done smartly and with the lowest error. Suppose:

- The real-world room is fully represented in the virtual reality (like in [14])
- The 3D-meshes of the IoT devices are collected, and the physics is defined

If the system stays stationary, then the location mapping of real-world devices to virtual reality can be done without an error (the distance error is << smaller than the sizes of the objects). But if the research requires moving the devices in real and virtual worlds, the enhancement of location retrieval needs to be provided. How the updating of the position of the devices in the real world should be handled? One of the solutions is using cameras to locate the objects movement in the real world (other possible solutions will be proposed after further brain-storming and research). So, we can suppose that:

- The mapping of real-world devices to the virtual reality (coordinates + features) performed correctly

For example, we want to know, where to place the smart speaker (as a hub) in the room, so the WIFI-connected virtual devices signal remains acceptable and the speech recognition can work in any part of the room. The microphone could be either emulated in VR, or be used in the real device. Emulating microphone in VR requires adding the sound physics: for example, sound reflection from the surfaces requires defining the surfaces materials, which is quite difficult to implement. So, the possible solution in this case is to use real-world devices' sensors for microphones.

One of the disadvantages of such approach is that the sensors from the real world can't "sense" virtual world and vice-versa. What if the smart speaker from the previous example is not real but virtual? How can the user test the speaker and mic input/output?

Microphones are of course not the only sensors. But for a lot of other sensors, the idea remains the same: implementing the physics for their sensing of objects may be too difficult task, so the virtual sensors should be used in addition to the real ones.

An example:

	Real world sensor physics support	Virtual reality sensor physics support
Real device 1	+	+
Real device 2	+	-
Virtual device 1	-	+
Virtual device 2	-	+

Because virtual devices of course cannot support real world sensor physics, virtual reality sensor physics support for real device 1 may need to be implemented (if the Real device 2 uses Virtual devices output).

Picture of IoT:

The IoT hardware can be:

- Hubs: Brand, Universal, Raspberry Pi, …
- Smart assistants: Siri, Alexa, Google Assistant, DuerOS, Xiaowei, AliGenie, …
- Gadgets: Chinese market (dominating), IKEA and others

The IoT software:

- Protocols: ZigBee + Z-Wave, Wifi
- Open-Source: HomeBridge, OpenHab, HomeAssistant, …
- SDK: HomeKit, OneNet, …

VRPlatform will be based on one of the Open-Source solutions at first, to concentrate more on the location and interaction services.

Conclusion / Summarizing the ideas

Before scheduling the research, let's summarize what was said previously:

1. The Virtual Reality room is created by user and requires to be a copy of the real room;
2. The sensing is run in the real world by default, but the physics support (such as sound physics) can be added to the platform if needed for the system to function;
3. Operating with devices in the virtual world is a part of my research
4. If the system is not stationary (devices are moving), then their position needs to be retrieved (by camera sensing or other methods) and then updated in the virtual world
5. The platform needs to follow the requirements (defined above) providing the best user experience and functionality

Not considering difficult tasks such as recreating a 3D room using camera as well as updating the position of the objects by camera, *I plan to mostly focus on implementing ways of interaction with IoT devices in VR and adding the physics support for them (apart from developing the platform prototype)*.

Detailed schedule of the project:

During the next year of the research, I will mainly focus on the development of the prototype of platform and interaction with IoT devices. The schedule depends on the possibility of returning to campus in September.

September 2020 – December 2020: creating 3D-model of apartment / building and adding the models of devices to it, adding a basic framework for adding the virtual devices and their interconnection

December 2020 – January 2020: extending the devices structure to use physical devices

January 2020 – March 2020: adding a server to store the state of the IoT ecosystem

March 2020 – April 2020: prototype of VR IoT framework.

If possible, I will continue this research in PhD later.