Immersive Virtual Reality Training Tool for IoT Device Placement

Yukun Jia
Beijing Dublin International College
Beijing Shi, Chaoyang Qu, China
georgina.jia@outlook.com

Abraham G. Campbell
University College Dublin
Belfield, Dublin, Ireland
abey.campbell@ucd.ie

ABSTRACT

The Internet of Things[1] promises to revolutionise our lives but for all the large-scale research into the field, one area has been neglected. This is the simple task of training staff to correctly place devices in an environment to create an ad-hoc wireless network that these devices require to be connected.

This training must include reference to obstacles and signals propagation models. Give the user a solid knowledge base by visually showing how the interconnectivity can take place and finally model aspects of the propagation including reflection and diffraction. This paper will outline the results of the creation of VR simulation of a generic space where IoT devices can be placed, and then evaluated in real time to test if their placement will be a viable network. To achieve this, the UNITY game engine is used for developing and HTC VIVE is used for testing.

Three models with different modes were successfully built in this project starting with a simple 'Virtual Room' case and followed by the 3D model of our university. The last model is a more realistic one with consideration of obstacles and signals propagation models.

CCS CONCEPTS

• **Human-centered Computing → Interaction paradigms**; *Virtual Reality*; Interactive systems and tools

KEYWORDS

Virtual Reality, Internet of Things, Training

ACM Reference format:

Y. Jia and A.G. Campbell "Immersive Virtual Reality Training Tool for IoT Device Placement " In Proceedings of SIMUTOOLS '17, September 11–13, 2017, Hong Kong, China, 6 pages.
https://doi.org/10.1145/3173519.3173521

1 INTRODUCTION

Recently, there has been considerable interest in the Internet of Things (IoT) technologies which always related to the deployment of wireless networks that consist of many nodes to gather information from the environment and communicate with other nodes. Nowadays sophisticated engineers who have a solid knowledge base of electronic engineering are responsible for the simulation and the placement of a network. But the booming of IoT technology leads to the need of a training tool which can help unprofessional workers carry out the same job.

This work explores how general training for IoT workers could be achieved through the creation of an Immersive Virtual Reality Training tool for IoT device placement. This allows untrained staff to gain intuitive knowledge into how they can correctly place devices in a building to create either a mesh network or fit into an existing wireless network infrastructure. There are three components of this work that will be discussed in this paper.: Virtual Room Design; UCD-Model Design; Real-life Scenario Design. To achieve these aims, UNITY game engine is used for developing and HTC VIVE is used for testing the system.

For this tool each node acts as a 'signal generator' and generates a 3D bubble which is a sphere centered at itself with a fixed radius. All the nodes have identical capabilities for communication, sensing, and mobility. The specific wireless protocols were not considered here, neither is energy efficiency of the network

SIMUTOOLS '17, September 11–13, 2017, Hong Kong, China

ACM ISBN 978-1-4503-6388-4/17/09...$15.00
https://doi.org/10.1145/3173519.3173521
†

simulated as this is only the beginning of the prototype training tool. This tool is a novelty due to its interface in an Immersive Virtual Reality environment, other 2D approaches will be discussed in the related work section, followed by a discussion of three use cases for the tool, and finally future work will be discussed.

2 RELATED WORK

Previous research has been extensive into simulating Wireless Signal propagation [1] [3][4][6][7] and generating training tools [8][9][10][11]. Automatic discovery of IoT devices through a Middleware such as Sixth[2] can allow for the IoT devices with sensors to become a sensor Web[16]. All this research has not explored the combination of Immersive virtual Environments [20], although they have become cheaper in the last decade [17], it has only been the introduction of the Oculus and HTC VIVE in the last year that has resulted in this technology becomi affordable for large scale deployment in training.

2.1 Network Modelling

Unlike [1] previous work in this field the training simulator only used a 'Brute Force' search to make sure the nodes connected, but this proved sufficient for evaluating the tool. A Mesh network was chosen as the primary training target. This network topology is based on each node taking part in both receiving and transmitting data, as said by Eiman Alotaibi in [1], where nodes support one another in conveying a data packet through wireless media. An example is shown in Fig. 1.

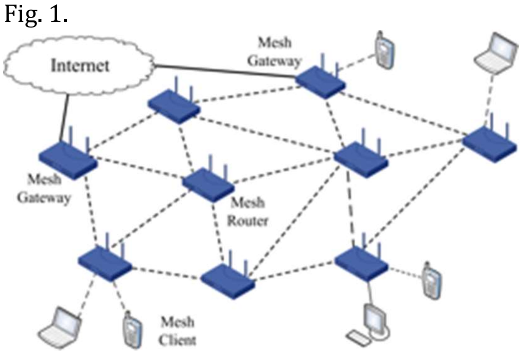

Figure 1 : Mesh Network

To model the Wireless signal, the simulation signals are said to be able to get to almost everywhere in a room because of its three characters: reflection, diffraction, scattering In combination with a simplified wireless signal propagation, different Wireless Deployment strategies were explored. The first one examined was a Hybrid network which consists of both stationary and mobile nodes. In this kind of network, the static nodes are usually deployed first and then some strategies are used to place mobile ones to ensure a better coverage. There are two ways to make the decision:

One is introduced by Bang Wang in [3] to using mobile nodes to heal coverage holes after the initial deployment to achieve the maximum coverage with minimum moving cost. This task contains the following issues: 1. Hole detection and hole size estimation; 2. Destination selection; 3. Moving strategy. The other is discussed in the work Shu Zou has done [4], the primary work is to use programming to find the optimal set of groups of nodes (islands) to be connected. Then placing mobile nodes along a line connecting two islands, finally using a 'virtual forces algorithm' to spread them out to reduce overlap.

The mobile network was also modelled, this contains only mobile nodes, the main task of it is to leverage mobile nodes to get an optimized result. There are four approaches studied:

(1) Coverage Pattern Based

Bang Wang had brought up another idea [3], in which the target locations for mobile nodes are computed based on a predefined pattern that can meet both coverage and connectivity requirements. It describes one commonly used pattern—local hexagon coverage pattern [5] which works as follows: one node is selected as a seed, it computes six hexagonal locations to form a regular hexagon surrounding itself.

(2) Virtual Forced Based

In this case, nodes can be seen as electromagnetic particles. Just like electrostatic force, there can be a virtual force between two nodes. In Bang Wang's paper [3], it comes up with a way to use this virtual force to move nodes.

The formula used here is from Zou and Chakrabarty [6]. A node can experience two types of force: positive and negative which determines their actions.

(3) Grid Quorum Based

In this case, the deployment problem is viewed as a load balancing problem. The whole area is divided into many small grid cells, and the number of the nodes in each cell is seen as the burden of the cell.

(4) Energy-Efficient Deployment

In the literature from Nojeong Heo and Pramod K. Varshney [7], a combination of cluster structuring and a peer-to-peer deployment scheme is used to get an energy-efficient deployment. This helps to improve the energy consumption.

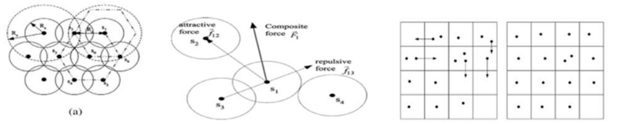

Figure 2. Illustration of the models talked above: coverage based, virtual force based, grid quorum based

2.2 Simulation Tools

Alberto Puggelli and Luciano Lavagno [8] present a graphical tool to assist the design exploration and synthesis of network topology.

After collecting this data, the tool can guide users toward an optimal placement of network relay nodes (e.g., routers). The GUI of the tool is showing in Fig. 3.

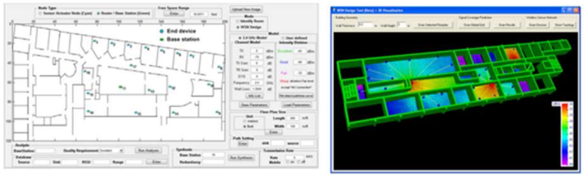

Figure 3. Illustration of the GUI & results

The graph on the right is the result generated by Antony Guinard's tool [9]

Simulation tools must also consider obstacles, in [10], they have small regions and large regions. In the small region, they find bisectors and then deploy a sequence of nodes along the bisector to satisfy the coverage and connectivity. And in a large region, some area near boundaries or obstacles may be left uncovered, this is solved by sequentially placing sensors along the borders of the area and obstacles.

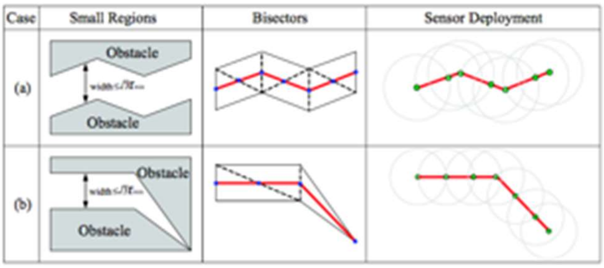

Figure 4. Deploying sensors with obstacles considered

In [11], they designed an algorithm for decreasing the impact of obstacles. In this scheme, a mobile anchor node cooperates with static sensor nodes and moves actively to refine location performance, and it is based on nodes' communication.

3 DESIGN AND IMPLEMENTATION

3.1 Specification and Design

This project took place in the UCD visualization cave, and the software used is 'Unity 5.5', which is an ultimate game development platform. It is used to develop quality 3D scenarios and to support VR equipment. For testing HTC VIVE is used. The language to be used is C#.

The whole project is divided into three case studies, and in each case, there are two modes. A simple illustration of three cases are shown below:

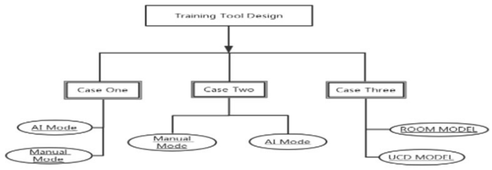

Figure 5. Structure of the project

3.2 Realization of Virtualization

The principle behind simulating a VR world is the use of polarizing 3D image-forming. The theory is if people can see a 3D object, it is because the pictures shown to each eye is different, which is caused by the distance of two eyes. As in present stereoscopic images and films, two images are projected superimposed onto the same screen or display through different polarizing filters.

3.3 User-interface Design

3.3.1. Main Menu. At the beginning of each case, there is the main menu to let users choose the mode they want to go into. This menu serves as a user interface to let players quickly pick one mode, so buttons are used to represent the modes.

The interaction is supported by the script from the toolkit VRTK; this includes support of: 'UI_Interactions.cs.' The button will 'listen' to the events, and if things happen, the corresponding functions will be called. See 'On Click' part in the inspector. The interface does require the user to move to the relevant location but potentially this added effort can be seen as exercise [18] for the real life task which will involve a large amount of movement around a building , this can be seen as a form of gamification of the task [18].

3.3.2. Changing Scenes. In each scene, there is a primary function allowing users to go back to the main menu,

3.4 Background Design

In case 1, the background is just a resizable room with four signal generators randomly placed at the beginning of the scene. This is done by using ‘Random.Range()’ and ‘transform. localScale’ In case two, a map of UCD is provided as a generic place to be positioned by devices. To simulate this scenario, we need to identify which area needs to be covered, i.e. determine the buildings. Here cubes are used to represent buildings and the place where buildings stand; the signal generator must cover it. In case three, one background is a floor with three rooms with obstacles. The floors and obstacles are all colliders to make sure that the range of a signal generator can go through it and have the corresponding reaction. Another mode uses the same background as in case two.

3.5 Nodes Design

3.5.1. Moving & Placing. In case one, the main idea of AI mode is to have some searching algorithm to let the signal generators move by themselves and get a placement plan. The trajectory used in this case is a spiral-up fashion, to understand this approach here is one of the formula’s used to achieve this spiral movement.

$$\text{Spiral}\,(q, r, h) = \{(q_x + r\sin t, q_y + r\cos t, h\sin t):\ 0 <= t <= 2*\text{pi}\}$$

In case two, the ‘routers’ movement follows an algorithm called ‘coverage based’. This includes creating a list to store all the positions, then creating ‘routers’ in specific positions.

In case three, the range of the signal generator is no longer presented by a simple bubble. Instead, a series of spheres are used to indicate the range a node can cover. Spheres are still propagating in all directions which create a great visual demonstration of how signals transmit. What’s more, the real-life scenario is simulated here as the signal will not only go through the wall, but also bounce back, which makes great sense as a real signal will have reflection and diffraction.

3.5.2. Interacting Part. The ‘Is Grabble Scripts’ provided in VRTK Toolkits. Creating More nodes is done by imitating how VRTK fires a virtual gun. The creation of a bullet is the same as the generation of nodes. Changing Color to Indicate Connection uses the fact that the object is a rigid body, and it must be assigned a trigger collider to enable its ability to be woken up when there is a collision. When there are collisions, the corresponding function will be called.

3.6 End Devices Design

In case one and case three-mode 1, the end devices are the same: just a cube to stand for any IoT end device like a sensor.

4 RESULTS AND DISCUSSION

4.1 Results

In case one, a virtual room is designed to show an idealistic scene of this idea. Two modes are available in this case: AI Mode & Manual Mode: AI Mode is a self-searching mode where the ‘routers’ will be moving around and following some predefined trail, and the Manual Mode is to let human place ‘routers’ in a room with the controller.

Fig. 6 shows some sample scenes in case one, first two are in manual mode where a laser is used control the ‘routers’; and the third is showing that all ‘routers’ are moving around to found positions.

This case is an absolute demonstration of the general idea of how ‘routers’ and its range can be shown in a 3D model, and the basic interaction between human and ‘routers’ in the scene lays a foundation for later implementation.

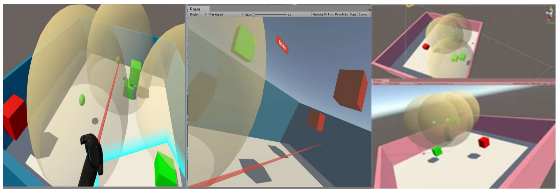

Figure 6. Case One Sample Scenes

Case two extends case one to the UCD model. The general space is bigger and the task is to simulate the placement of ‘routers’ in UCD. In the AI mode, the ‘routers’ are initialized every second and they follow a pattern according to the coverage-based algorithm mentioned

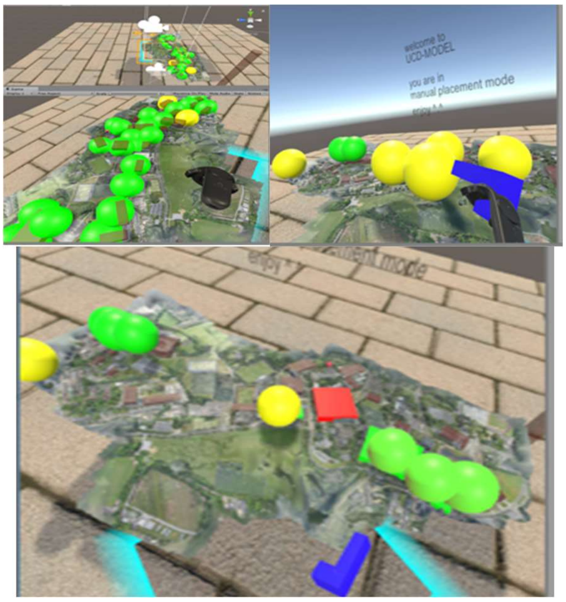

Figure 7. Case Two Sample Scenes

Fig. 7 shows sample scenes where the first one is the predefined positions and the other two are human-placement. This case gives a more picturesque description of how the tool can be used as it extends the first case to a real map. The algorithm to find the placement plan in AI mode can be improved.

Case three counts in obstacles and a real signal propagation model. In the virtual scene, a three-floor room with walls and constraints are created to indicate an actual room situation. There are two 'routers' and nine end devices. Users can see the real dynamic 3D range of routers directly. When the user uses the controller to move the 'router', the previous range will disappear, and a new range will start showing up. In the UCD scene, the real propagation model is introduced into the UCD Model.

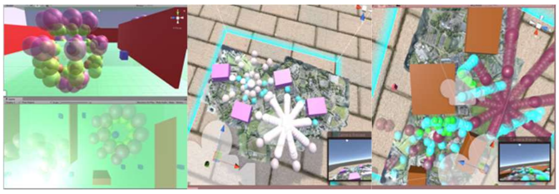

Figure 8. Case Three Sample Scenes

Fig. 10 shows the sample scenes in case three: the first one is the virtual room version where the user can see the model of a router in a new way; and the other two show the reflection and diffraction of the signal.

4.2 Discussion

In Case One only a change in color is used to indicate the connection. This will cause problems like when two or more of them cover the same device at the same time, so from a design perspective multiple colors could be used but a blending shader would be needed to make sense of the additional colors. Similar to research discussed in Section 2.2, this work can offer a better way to convey information and let the user clearly choose different configurations. Case Two allows for the AI mode, the number of 'routers' is four which may be not enough for a real-life case. Also. there are still some algorithm problems to be optimized. The final case gives a vivid demonstration of what this tool will be like in the end. The advantage is clear as the user can understand the real signal propagation from a lively 3D environment. The real model is not precisely designed, though. There will be some drawbacks of this tool as it's not perfect and not be as sophisticated.as previous work. Compared to the works done by Damien B. Jourdan in [13] and E. Amaldi done in [14], which have the coverage represented in the way shown in Fig. 9, the graphs are clear, but it's a two-dimension model. In this project, however, the connectivity and coverage are shown in a straightforward way in which a human can check the range of the 'signal generator' in a 3D environment and even be in it. This is this tools contribution, conveying the profound professional knowledge in a lucid way. also note that a render graph like Fig. 9 can be used when representing propagations, this is also a different point.

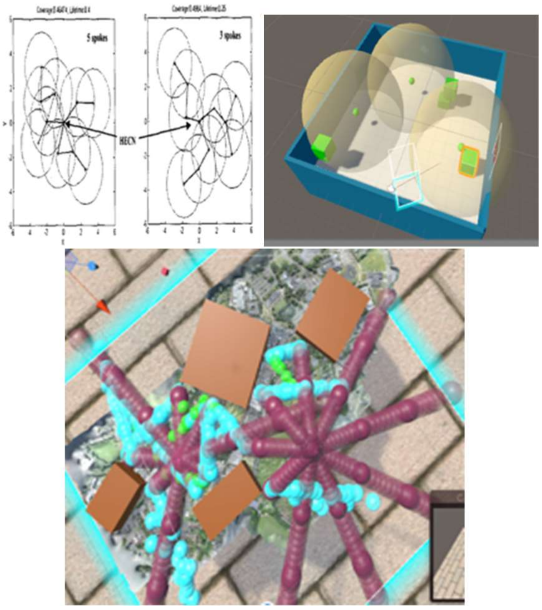

Figure 9. Coverage Representations

This system can be tested by letting people with no experiences in VR and who have no knowledge base of electronic engineering to place the 'routers' in a room. And then compare to the results from complicated algorithms and advanced simulator.

5 CONCLUSION AND FUTURE WORK

With the completion of final case study, a prototype tool was successfully developed. The first stage of this project was a simple implementation of the core idea of VR simulator while the next stage extended it to a more real scenario. This practical tool enables users to examine in real time the network topology and make decisions of placement plans based on that. It is different from the previous work discussed in this paper as it provides a chance for the non-expert to be able to visually understand the complex algorithms and placement patterns. The simulation of a wireless network is no longer limited to a 2-D model where only professional workers can analyze it. This tool can possibly solve the problem of training stuff to place the IoT devices in an uncomplicated way.

As mentioned in Section 1, the use of the game engine and virtual reality technology enables people to perform a challenging task in an intuitive fashion. This tool is efficient considering both time and computational complexity even with its brute force approach in establishing connections. In future, more efficient algorithms will need to be used to simulate large scale training with 20 or more nodes. Future work on the tool can be explored from five different perspectives Energy, More Complex Models, Number of Devices, AI Algorithms, User interfaces The placement algorithm does not consider the energy expenditure. Previous research such as Shnayder et al [15] will need to be examined where a scalable simulation environment for WSN that provides an estimation of power consumption had been brought up. This tool had the ability to indicate energy levels in a user-friendly way, colors, for example. As the 'routers' moving around or as the end devices constantly communicate with 'routers', their energy will decrease which can be shown by a fading in their color.

The tool does not include any protocol-related knowledge which is obviously far from how the real-world communication works. So more complex models can be tried in the future. As in Ruzzelli et al [13], an energy-efficient and low latency routing protocol for WSN is explored with OmNet++ simulator used.

In this case , the number of devices/'routers' are fixed, and it is not enough for a real-life scenario. Case three, for example, as shown in Section 4.2.3, only contains two 'routers', which is far from the reality. To consider more sophisticated algorithms will significantly improved my work. So, future work can focus on figuring out how to deal with the interactions between 'routers' or end devices, create a more efficient way to cover more devices, considering that there are such complicated relations between devices even only two of them, their signals will suffer overlapping or canceling. In this work, the user interface is limited to buttons and labels which offer a limited user experience. A more dynamic user interface can be designed to give the users a more vivid feel of this tool. User-interface in Google's tilt Brush [19] is an excellent example which I want to learn from in which the controller becomes a powerful control device containing many functions. I want to add more biological information into this tool, for instance, if a human can just wave their hands to change a mode just as they modify the screen of their phone, then it will become very easy to check an AI result and make some comparison.

ACKNOWLEDGMENTS

The equipment for this project in the UCD Visualisation CAVE was in part funded by the UCD centenary seed funding project for Virtual Sacksville street.

REFERENCES

[1] Eiman Alotaibi and Biswanath Mukherjee. 2012. A survey on routing algorithms for wireless Ad-Hoc and mesh networks Department of Computer Science (Kemper Hall), University of California, Davis, CA 95616, USA.

[2] Enabling a mobile, dynamic and heterogeneous discovery service in a sensor web by using AndroSIXTH
L Görgü, B Kroon, AG Campbell, GMP O'Hare 2013
International Joint Conference on Ambient Intelligence, 287-292
Dublin,Ireland

[3] Bang Wang *, Hock Beng Lim and Di Ma, 2009. A survey of movement strategies for improving network coverage in wireless sensor networks.

[4] Shu Zhou, Min-You Wu, and Wei Shu, 2004. Finding Optimal Placements for Mobile Sensors: Wireless Sensor Network Topology Adjustment. Department of Electrical and Computer Engineering, The University of New Mexico.

[5] P.-C. Wang, T.-W. Hou and R.-H. Yan, 2006. Maintaining coverage by progressive crystal- lattice permutation in mobile wireless sensor networks, IEEE International Conference on Systems and Networks Communication (ICSNC).

[6] Y. Zou, K. Chakrabarty, 2004. Sensor deployment and target localization in distributed sensor networks, ACM Transactions on Embedded Computing Systems.

[7] Nojeong Heo and Pramod K. Varshney. 2003. An Intelligent Deployment and Clustering Algorithm for a Distributed Mobile Sensor Network*. Department of Electrical Engineering and Computer Science Syracuse University, Syracuse, NY 13244;

[8] Alberto Puggelli, Student Member, IEEE, Mohammad Mostafizur Rahman Mozumdar, Member, IEEE, Luciano Lavagno, Member, IEEE, and Alberto L. Sangiovanni-Vincentelli, Fellow, 2015. Routing-Aware Design of Indoor Wireless Sensor Networks Using an Interactive Tool, IEEE.

[9] Antony Guinard and Alan Mc Gibney, A Wireless Sensor Network Design Tool to Support Building Energy Management. 2009. Dirk Pesch Centre for Adaptive Wireless Systems Cork Institute of Technology Cork, Ireland,

[10] You-Chiun Wang, Chun-Chi Hu, and Yu-Chee Tseng∗. 2004. Efficient Deployment Algorithms for Ensuring Coverage and Connectivity of Wireless Sensor Networks. Department of Computer Science and Information Engineering National Chiao Tung University, Hsin-Chu, 30050, Taiwan.

[11] Hongyang Chen, Qingjiang Shi, Rui Tan, H. Vincent Poor, Fellow, IEEE, and Kaoru Sezaki, 2010. Mobile Element Assisted Cooperative Localization for Wireless Sensor Networks with Obstacles.

[12] NeXuS: delivering behavioural realism through intentional agents GMP OHare, AG Campbell, JW Stafford
Active Media Technology, 2005.(AMT 2005). Proceedings of the 2005 takamatsu, Japan

[13] Damien B. Jourdan, Olivier L. de Weck. 2004. Layout Optimization for a Wireless Sensor Network Using a Multi-Objective Genetic Algorithm, Dept. of Aeronautics and Astronautics, Massachusetts Institute of Technologyx.

[14] E. Amaldi, A. Capone, M. Cesana *, I. Filippini, F. Malucelli Optimization models and methods for planning wireless mesh networks. Dipartimento Elettronica e Informazione, Politecnico di Milano, Piazza Leonardo da Vinci 32, 20133 Milano, Italy

[15] Victor Shnayder, Mark Hempstead, Bor-rong Chen, Geoff Werner Allen, and Matt Welsh. 2004. Simulating the Power Consumption of Large-Scale Sensor Network Applications. Division of Engineering and Applied Sciences Harvard University

[16] Giving mobile devices a SIXTH sense: Introducing the SIXTH middleware for Augmented Reality applications,
AG Campbell, L Gorgu, B Kroon, D Lillis, D Carr, GMP O'Hare, 2013 pages 245-46,Mixed and Augmented Reality (ISMAR), 2013 IEEE International Symposium, Adelaide, Australia

[17] The lair: Lightweight affordable immersion room
B Denby, AG Campbell, H Carr, GMP O'Hare
Presence: Teleoperators and Virtual Environments 18 (5), 409-411 2009

[18] Exergaming: a future of mixing entertainment and exercise assisted by mixed reality agents L Görgü, A Campbell, M Dragone, GMP O'Hare Computers in Entertainment (CIE) 8 (4), 27, 2010

[19] Machkovech, Sam. "Learning how to VR with Tilt Brush, HTC Vive's killer app." Ars Technica 5 (2016).

[20] Cruz-Neira, Carolina, Daniel J. Sandin, and Thomas A. DeFanti. "Surround-screen projection-based virtual reality: the design and implementation of the CAVE." Proceedings of the 20th annual conference on Computer graphics and interactive techniques. ACM, 1993.