

New Approaches in Training for in situ Visualization of Ionizing Radiation Measurements

by

Joseph Chaput, MAsC, BEng

**A thesis submitted to the
School of Graduate and Postdoctoral Studies in partial
fulfillment of the requirements for the degree of**

Doctor of Philosophy

in

Nuclear Engineering

Faculty of Energy Systems and Nuclear Science

University of Ontario Institute of Technology

Oshawa, Ontario, Canada

August, 2022

©Joseph Chaput, 2022

New Approaches in Training for in situ Visualization of Ionizing Radiation Measurements

Submitted by Joseph Chaput

Doctor of Philosophy in Nuclear Engineering

An oral defence of this thesis took place on 17 June, 2022 in front of the following examining committee:

Examining Committee

Chair of Examining Committee:	Dr. Jennifer McKellar
Research Supervisor:	Dr. Edward Waller
Examining Committee Member:	Dr. Anthony Waker
Examining Committee Member	Dr. Eleodor Nichita
University Examiner:	Dr. Diana Petrarca
External Examiner:	Dr. Sanjoy Mukhopadhyay

The above committee determined that the thesis is acceptable in form and content and that a satisfactory knowledge of the field covered by the thesis was demonstrated by the candidate during an oral examination. A signed copy of the Certificate of Approval is available from the School of Graduate and Postdoctoral Studies.

ABSTRACT

Surveying an environment for sources of ionizing radiation requires the use of measurement tools and skills developed from training in order to understand and analyze measurements. Recent advances in technology allow for new approaches to be made in the radiation survey methodology generally used with the incorporation of augmented reality (AR) technology to improve real-time awareness *in situ* and virtual reality (VR) technology to better develop the skill set of the surveyor in realistic virtual environments beforehand. This thesis investigates and develops a novel process to display real-time measured radiation monitoring data in AR to support a radiation surveyor during a search of an environment for hazardous sources of radiation. This AR process is then modified to show how it can be used with virtual radiation sources to allow a radiation surveyor to practice with a digital twin of a radiation field using a virtual source. This process is then modified further and shown how it can be adapted and used to develop VR training scenarios to teach the skill sets needed to assess potential hazards (radiological and non-radiological) and identify sources of radiation. Finally, an approach using reinforcement learning methods is developed and applied to demonstrate a strategy to localize a single radiation source leveraging the real-time measurement data taken in AR.

Keywords: Virtual reality training; augmented reality; radiation measurement visualization; reinforcement learning source localization;

DECLARATION OF AUTHORSHIP

I hereby declare that this thesis consists of original work of which I have authored. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize the University of Ontario Institute of Technology (Ontario Tech University) to lend this thesis to other institutions or individuals for the purpose of scholarly research. I further authorize University of Ontario Institute of Technology (Ontario Tech University) to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research. I understand that my thesis will be made electronically available to the public.

Joseph Chaput

STATEMENT OF CONTRIBUTIONS

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication.

The generic radiation field visualization description provided in Chapter 3 and the included figures are based on the cited work originally written by the author in their Masters of Applied Science in Nuclear Engineering thesis. The Unity Point Kernel (UPK) radiation transport code developed and described in Chapter 4 was further expanded on by University of Ontario Institute of Technology student Mark Breitingner for the purpose of creating training scenarios for x-ray non-destructive investigation purposes. UPK was described, focusing on his specific contributions and applications in his thesis (2021). University of Ontario Institute of Technology employee Callan Brown provided support in helping to connect to the radiation detector via Bluetooth as described in Chapter 4. Stefan Ecker and Mark Breitingner provided support conducting the virtual reality demonstrations and helping to gather the survey results discussed in Chapter 7.

I have used standard referencing practices to acknowledge ideas, research techniques, or other materials such as figures that belong to others.

Outside of what is described in this section, I hereby certify that I am the sole source of the creative works and/or inventive knowledge described in this thesis.

ACKNOWLEDGEMENTS

Funding for this research was provided by the Natural Sciences and Engineering Research Council of Canada (NSERC) and the University Network of Excellence in Nuclear Engineering (UNENE).

TABLE OF CONTENTS

ABSTRACT	ii
DECLARATION OF AUTHORSHIP.....	iii
STATEMENT OF CONTRIBUTIONS.....	iv
ACKNOWLEDGEMENTS	v
TABLE OF CONTENTS.....	1
LIST OF FIGURES.....	6
LIST OF TABLES	19
LIST OF ABBREVIATIONS AND SYMBOLS	20
Chapter 1: Introduction	21
1.1. Background.....	21
1.2. Problem description	24
1.3. Research objectives.....	25
1.4. Structure of this thesis.....	27
Chapter 2: Literature review.....	29
2.1. Radiation surveying.....	29
2.2. Visualization of ionizing radiation fields.....	36
2.2.1. <i>Practical techniques</i>	36

2.2.2. <i>Simulated and augmented data</i>	39
2.3. Source localization	42
2.4. Virtual environments for training.....	45
Chapter 3: Theoretical background	48
3.1. Generic approach to radiation field visualization	48
3.2. Requirements for a voxelization approach to be viable as a real-time display technique	53
3.3. Virtual environment for training and testing.....	54
3.3.1. <i>Emergency response exercises</i>	54
3.3.2. <i>Virtual reality methodology</i>	59
3.3.3. <i>Moving in the virtual environment and user comfort</i>	65
Chapter 4: Radiation Measurement visualization	70
4.1. Development of a flexible radiation transport model for virtual and augmented reality simulations	70
4.1.1. <i>Radiation modelling</i>	70
4.1.2. <i>Creating the model in Unity – the Unity Point Kernel (UPK) approach</i>	77
4.1.3. <i>Complex source distributions and other UPK applications</i>	82
4.1.4. <i>Validation of the UPK model</i>	85
4.1.5. <i>UPK Performance</i>	88
4.1.6. <i>Alternative approaches</i>	91

4.2.	Displaying a radiation field in augmented reality	94
4.2.1.	<i>Initial approach</i>	94
4.2.2.	<i>Advanced approach for visualization of radiation measurement information (Google Project Tango)</i>	102
4.2.3.	<i>Advanced approach for visualization of radiation measurement information (Unity AR Foundations and Google ARCore)</i>	110
4.3.	Development of a real-time capability to display radiation measurement information in augmented reality	111
4.4.	Development of a virtual reality environment for the realistic training of radiation detection concepts	115
Chapter 5: Reinforcement learning approach to radiation source localization Support		120
5.1.	Reinforcement learning	120
5.2.	Unity ML Agents.....	121
5.3.	Design of the reinforcement learning scenarios and agent training.....	125
5.3.1.	<i>Episode Selection</i>	130
5.4.	Selection of an agent training process	134
Chapter 6: Synthesis of radiation measurement visualization, source localization and augmented reality training approaches.....		141
6.1.	Radiation measurement visualization.....	141

6.2.	Radiation source localization support using ML	151
6.3.	Virtual reality training approach developed into scenarios	158
6.3.1.	<i>Transport accident with a radiological source in a vehicle (scenario 1) ...</i>	158
6.3.2.	<i>Transport accident with radiological material in severe weather (scenario 2)</i> 162	
6.3.3.	<i>Transport accident with a sea container carrying radiological material at a port (scenario 3)</i>	165
6.3.4.	<i>Transport accident with radiological material on a train in a rural area (scenario 4)</i>	168
6.4.	Mixed reality training approach	171
Chapter 7:	Results	176
7.1.	Results from the use of augmented reality to visualize a radiation field	176
7.1.1.	<i>Results of simulated radiation field visualization using Layer Augmented reality web browser</i>	176
7.1.2.	<i>Results of simulated radiation measurement visualization using Google Project Tango and AR Foundations with AR Core</i>	177
7.2.	Results from the use of augmented reality to visualize radiation measurements 185	
7.3.	Results from the use of machine learning approach to localizing radiation sources	189
7.4.	Results from the use of virtual reality technology for the conduct of emergency	

response training	201
7.4.1. <i>Evolution</i>	201
7.4.2. <i>Feedback</i>	202
7.5. Specific considerations for the use of VR technology under the circumstances of the Covid-19 pandemic	212
Chapter 8: Conclusions and recommendations.....	214
8.1. Conclusions.....	214
8.2. Recommendations for future research	216
References	219
Annex I: UPK calculation code.....	238
Annex II: Augmented reality visualization code	255
Annex III: Reinforcement learning code.....	258
Annex IV: Dose conversion factors	280
Annex V: Subjective lessons learned during the development and deployment of VR training scenarios	281
Annex VI: Virtual Reality exercise feedback survey.....	304

LIST OF FIGURES

Figure 1 - Example of results from a radiation survey of the Unit 3 Reactor Building First Floor at the Fukushima Daiichi NPS[27]	31
Figure 2 - Example of robot survey with gamma camera survey taken inside Reactor Building for Unit 1 at the Fukushima Daiichi NPS[28]	31
Figure 3 - Example of a series of handheld radiation surveys taken around the Fukushima Daiichi NPS[29]	32
Figure 4 - Example of a series of vehicle borne survey taken outside of the Fukushima Daiichi NPS[30]	33
Figure 5 - Example of interpolated results from an aerial survey in the region around the Fukushima Daiichi NPS (with the flight path indicated on the right)[30]	33
Figure 6 - Example of a radiosensitive polymer which can be used to visually display the direction of radiation field intensity[36]	38
Figure 7 - Elements types 1, 2, 3 and 4 (from left to right)	48
Figure 8 - Construction of the 7x7x7 field model sequence with radiation source at the center element	49
Figure 9 - Image from (previous figure) with one material made completely transparent and the others at 20% transparency producing the visualization effect	49
Figure 10 - Point source on a virtual table with scale (isometric perspective) and 3d visualized radiation field (higher fidelity model from a 25x25x25 grid with the source at the center)	50

Figure 11 - Point source on a table with scale (isometric perspective) and 3d visualized radiation field	53
Figure 12 - Progressive approach to training and exercises[73]	56
Figure 13 - HTC Vive VR hardware (controllers, headset and headphones).....	60
Figure 14 - HTC Vive base station mounted on camera stand (only one visible) and close-up	61
Figure 15 - A user in a VR environment	62
Figure 16 - A user in a VR environment demonstrating how interaction requires physical movement and engagement with the synthetic environment	63
Figure 17 - Labeled buttons on the Vive controller	64
Figure 18 - Summary of forwards backwards slide locomotion	67
Figure 19 - Example of forwards slide locomotion.....	68
Figure 20 - Conversion for Air Kerma Free in Air per Fluence (pGy cm ²)[96]	76
Figure 21 - Conversion for H*(10) from Gy[96].....	77
Figure 22 - Example of Shield source script configuration (note the visible attenuation values that have been linearly interpolated for Co-60).....	81
Figure 23 - Example of Radiation source script configuration for Co-60	81
Figure 24 - Illustration of how the unity raycast is used to retrieve the hit points on the shield object needed to calculate the penetration distanced needed for taking into the account the shielding	82

Figure 25 - Example of point kernel approach for measuring larger sources from close distances	83
Figure 26 - Example of point kernel approach for an arbitrary ground contamination scenario	84
Figure 27 - Results from Unity for dose rate from 1 Ci activity Co-60 source at 1 meter	86
Figure 28 - Results from Unity for dose rate from 1 Ci activity Co-60 source at 1 meter with 1cm of lead shielding between source and detector.....	87
Figure 29 - Screenshot of the Unity profiler being used to investigate performance of UPK	89
Figure 30 - UPK execution time (ms) vs. number of game objects (sources, shields and detectors).....	90
Figure 31 - Example of the use of UPK adapted to generate an approximation of an X-Ray visualization of an explosive device[104].....	92
Figure 32 - Example of the use of UPK adapted to visualize a simulated x-ray of a human body	93
Figure 33 - Visualization of point source intensity data from Microsoft excel (using conditional formatting)	96
Figure 34 - Visualization of imported point source intensity data in Google SketchUp 7	97
Figure 35 - Colored surface mesh of point source intensity data in Google SketchUp 7	

.....	98
Figure 36 - Real-time visualization of point source 3d mesh in AR interface via Layer Reality Browser (seen from distance).....	99
Figure 37 - Real-time visualization of point source 3d mesh in AR interface via Layer Reality Browser (seen as approaching).....	99
Figure 38 - Visualization of several point sources showing intensity data from Microsoft excel (using conditional formatting)	100
Figure 39 - Colored surface mesh of several point sources intensity data in SketchUp 7	101
Figure 40 - Real-time visualization of several point sources intensity 3d mesh in AR interface via Layer Reality Browser (seen as approaching).....	101
Figure 41 - System diagraph for the Google Tango Tablet[110]	103
Figure 42 - Preliminary environmental scan taken with included Tango constructor application (approaching staircase).....	104
Figure 43 - Preliminary environmental scan taken with included Tango constructor application (after traversing staircase).....	104
Figure 44 - Preliminary environmental scan taken with included Tango constructor application (showing several stories and the inside of an apartment).....	105
Figure 45 - Preliminary environmental scan taken with included Tango constructor application (highlighting detail captured in close range)	106
Figure 46 - Preliminary radiation visualization results (1).....	107

Figure 47 - Preliminary radiation visualization results (2).....	108
Figure 48 - Preliminary radiation visualization results (3).....	108
Figure 49 - Preliminary radiation visualization results without environmental meshing	110
Figure 50 - Side view of the Abacus Radiation Detector	112
Figure 51 - Top view of the Abacus Radiation Detector	113
Figure 52 - Bottom view of the Abacus Radiation Detector	113
Figure 53 - Screenshot of tutorial scene starting point	117
Figure 54 - Screenshot of user learning basic interactions at starting table in tutorial scene	117
Figure 55 - Screenshot of user approaching radiation demonstration table in tutorial scene	118
Figure 56 - Screenshot of user holding radiation detector and learning to perform a measurement.....	118
Figure 57 - Screenshot of user measuring the training sources and successfully building a small castle around the sources to see the shielding effect.....	118
Figure 58 - Reinforcement learning generalized cycle	125
Figure 59 - Screenshot of the agents during training (1)	128
Figure 60 - Screenshot of the agents during training (2)	129
Figure 61 - Summary description of reinforcement learning for finding radiation source based on random set of measurements	130

Figure 62 - Image of training environment as seen from top-down perspective	131
Figure 63 - Image of training environment as seen from side perspective	131
Figure 64 - Example of data being tracked during the training process showing positions of the measurements in the room, the measurements and the normalized measurements	133
Figure 65 - Plotted reward function	136
Figure 66 - Agents response to reward function during training (at step ~1.5 million). 137	
Figure 67 - Visualization of agent response during training with the most accurate 'guesses' circles in red (at step ~1.6 million)	138
Figure 68 - Histogram plot of the frequency of reward vs. time during training (at step ~1.8 million)	138
Figure 69 - Visualization of agent response during training (at step ~4 million)	139
Figure 70 - Fiestaware dish 1	142
Figure 71 - Fiestaware dish 2	142
Figure 72 - Thorium mantle	143
Figure 73 - Three sources of radiation placed in an indoor environment	144
Figure 74 - Radiation detector and phone apparatus setup for surveying (front view) 145	
Figure 75 - Radiation detector and phone apparatus setup for surveying (front view) 146	
Figure 76 - Augmented reality radiation measurement real-time visualization screenshot (seen from first person perspective) (1)	147

Figure 77 - Augmented reality radiation measurement real-time visualization screenshot (seen from first person perspective) (2)	148
Figure 78 - Augmented reality radiation measurement real-time visualization screenshot (seen from top-down perspective) (3)	149
Figure 79 - Augmented reality radiation measurement real-time visualization screenshot (seen from third-person perspective with environmental meshing) (4)	150
Figure 80 - Augmented reality radiation measurement real-time visualization screenshot (seen from third-person perspective with environmental meshing) (5)	151
Figure 81 - Example primary interface for the application with and without the placement of a source of radiation	153
Figure 82 - Example of three view points (3 rd person, top down and first person) in the example application interface measuring a single source.....	154
Figure 83 - Example of the measurement of 4 sources of radiation in the example interface from first and 3 rd person viewpoints	155
Figure 84 - Example of the measurement of a single source of radiation with the environmental meshing activated	156
Figure 85 - Example of field guidance for first responders provided by IAEA Portable Digital Tool for Assisting First Responders to a Radiological Emergency.....	157
Figure 86 - Screenshot from the transport accident with a radiological source in a vehicle showing the general environment of the scene	160
Figure 87 - Screenshot from the transport accident with a radiological source in a	

vehicle showing the user approaching the accident.....	161
Figure 88 - Screenshot from the transport accident with a radiological source in a vehicle showing the user investigating the transport package	162
Figure 89 - Screenshot from the transport accident with radiological material in severe weather with showing the general dark environment and severe rain (not visible is severe wind causing the trees to shake).....	163
Figure 90 - Screenshot from the transport accident with radiological material in severe weather with showing the user investigating a drum, visually inspecting its signage, conditions and taking radiation measurements.....	164
Figure 91 - Screenshot from the transport accident with radiological material in severe weather with showing a user investigating a damaged drum requiring them to crouch down	165
Figure 92 - Screenshot from the transport accident with a sea container carrying radiological material at a port showing the general environment	166
Figure 93 - Screenshot from the transport accident with a sea container carrying radiological material at a port showing a user investigating the scene and trying to identify which packages carry radiological material	167
Figure 94 - Screenshot from the transport accident with a sea container carrying radiological material at a port showing a user investigating the scene and trying to identify which packages carry radiological material	168
Figure 95 - Screenshot from the transport accident with radiological material on a train in a remote community showing the train from close perspective.....	169

Figure 96 - Screenshot from the transport accident with radiological material on a train in a remote community showing the train from the viewpoint of the user..... 170

Figure 97 - Screenshot from the transport accident with radiological material on a train in a remote community showing the user using a scope to investigate the train and read signage and assess hazards from a distance..... 171

Figure 98 - A space setup for mixed reality filming using a green screen 172

Figure 99 - User beginning a scenario picking up a radiation detector from a table and examining its response with a check source (filmed in mixed reality) 173

Figure 100 - User approaching the scene of a traffic accident involving the transport of a radiological source (filmed in mixed reality) 173

Figure 101 - User visually examining and measuring a package with a radiological source (filmed in mixed reality) 174

Figure 102 - User retrieving a package with a radiological source from a traffic accident (filmed in mixed reality)..... 174

Figure 103 - User putting a package with a radiological source into a concrete container for temporary storage (filmed in mixed reality)..... 175

Figure 104 - Series of images from the same location showing the radiation visualization failing to 'lock in place' 177

Figure 105 - Example of advanced AR approach where data remains fixed in the environment for the user to analyze..... 179

Figure 106 - Example of advanced AR approach showing the benefits of real-time

meshing to help contextually understand where measurements are being taken in the environment.....	180
Figure 107 - Example of advanced AR approach showing that even when reentry into the same environment is done the original results remain fixed in the environment	181
Figure 108 - Example of advanced AR approach showing a top down perspective helping the user understand where the source may be located.....	181
Figure 109 - Example of advanced AR approach showing a top down perspective helping the user understand where the source may be located with a modified scale to more clearly indicate the highest locations	182
Figure 110 - Example of advanced AR approach showing virtual measurements taken in real-time locked in the environment from two different angles	183
Figure 111 - Example of advanced AR approach showing virtual measurements taken in real-time locked in the environment and viewed from top down, 3 rd person and first person perspectives.....	184
Figure 112 - Example of advanced AR approach showing measurements of three sources from top-down angle with environmental meshing	186
Figure 113 - Example of advanced AR approach showing measurements of three sources from first person perspective	187
Figure 114 - Example of influence of integration period on measurement placement in augmented reality	188
Figure 115 - Example of advanced AR approach showing measurements of three	

sources from 3 rd person perspective with environmental meshing	189
Figure 116 - Third person and first person perspective of measurements and a guess (white sphere) 2.01 m (blue line) from actual source (yellow sphere).....	191
Figure 117 - Third person, Top down and first person perspective of measurements and a guess (white sphere) 1.56 m (blue line) from actual source (yellow sphere)	192
Figure 118 - Third person perspective of measurements and a guess (white sphere) 2.20 m (blue line) from actual source (yellow sphere)	193
Figure 119 - Third person and first person perspective of measurements and a guess (white sphere) 1.44 m (blue line) from actual source (yellow sphere).....	194
Figure 120 - Third person and first person perspective of measurements and a guess (white sphere) 1.44 m (blue line) from actual source (yellow sphere).....	195
Figure 121 - Distance guessed location from source for an AI agent in 25000 simulations (linear scale)	197
Figure 122 - Distance guessed location from source for an AI agent in 25000 simulations (log scale)	197
Figure 123 - Example of 80 different training sessions cumulative reward progress compared (note horizontal axis represents steps into the training process)	199
Figure 124 - Example of 80 different training sessions cumulative reward progress compared (note horizontal axis represents total time spent in the training process)	200
Figure 125 - Survey results question 1 (n=46) (pre-conduct)	204

Figure 126 - Survey results question 2 (n=46) (pre-conduct)	204
Figure 127 - Survey results question 3 (n=46) (pre-conduct)	205
Figure 128 - Survey results question 4 (n=46) (pre-conduct)	205
Figure 129 - Survey results question 5 (n=46) (pre-conduct)	206
Figure 130 - Survey results question 6 (n=44) (post-conduct)	207
Figure 131 - Survey results question 7 (n=44) (post-conduct)	207
Figure 132 - Survey results question 8 (n=44) (post-conduct)	208
Figure 133 - Survey results question 9 (n=44) (post-conduct)	208
Figure 134 - Survey results question 10 (n=44) (post-conduct).....	209
Figure 135 - Survey results question 11 (n=44) (post-conduct).....	209
Figure 136 - Survey results question 12 (n=44) (post-conduct).....	210
Figure 137 - Survey results preference of VR scenario (n=25)	210
Figure 138 - Example environment for scanning into 3d	286
Figure 139 - Results from Constructor Developer Tool	287
Figure 140 - Results of mesh capturing highlighting imperfections in the mesh.....	288
Figure 141 - Results of mesh repairing in MeshLab afterwards	289
Figure 142 - Overhead crane still representing a hazard in the scene	302
Figure 143 - VR transport survey feedback survey page 1	304
Figure 144 - VR transport survey feedback survey page 2	305

Figure 145 - VR transport survey feedback survey page 3 306

Figure 146 - VR transport survey feedback survey page 4 307

LIST OF TABLES

Table 1 - Specifications of the Abacus Radiation Detector[121].....	114
Table 2 - Example hyperparameter file for input to train a network (called RadiationSourceLocalizing)	127
Table 3 - Tabled reward function (provided when value is less than or equal to).....	136
Table 4 - RadiationSource.cs C# code	240
Table 5 - RadiationShield.cs C# code	242
Table 6 - RadiationCalculation.cs C# code	246
Table 7 - C# code in Unity used to query and receive data from the Abacus Radiation Detector	255
Table 8 - GuessingAgent.cs C# code	260
Table 9 - RadiationArea3d.cs C# code.....	266
Table 10 - RadiationCalculationForML.cs C# code	267
Table 11 - Dose conversion factors for photon fluence to Gy to H*(10)	280

LIST OF ABBREVIATIONS AND SYMBOLS

AI	Artificial Intelligence	MCNP	Monte Carlo N-Particle
ALARA	As Low As Reasonably Achievable	MLE	Maximum Likelihood Estimator
AR	Augmented Reality	NPP	Nuclear Power Plant
CCD	Charge-Coupled Device	RDD	Radiological Dispersal Device
CMOS	Complementary Metal-Oxide- Semiconductor	RL	Reinforcement Learning
EKF	Extended Kalman Filter	TEPCO	Tokyo Electric Power Company
EPR	Emergency Preparedness and Response	UKF	Unscented Kalman Filter
GIS	Geographic Information System	UPK	Unity Point Kernel
GPS	Global Positioning System	VR	Virtual Reality
GPT	Google Project Tango		

CHAPTER 1: INTRODUCTION

1.1. Background

Exposure to ionizing radiation can have negative consequences to persons and the environment. Over time, the nuclear industry has developed international safety standards[1], national regulatory requirements[2], technological tools and design strategies[3] to reduce, as much as practical, these negative consequences. In nuclear power plants and other industrialized environments with nuclear material, workers are supported with radiation detection equipment to measure and inform them of the local environment, protective gear to keep material off their skin and prevent accidental ingestion, and advanced occupational training programs to give them the skills they need to operate safely in their work environment. Teams of engineers anticipate, plan, design engineering controls and work to ensure the protection of workers, the public and the environment from any potential consequences from the nuclear and radiological material that is being used. However, accidents involving exposure to ionizing radiation have taken place[4][5].

Accidents in the nuclear industry have been shown to have the potential for large scale negative consequences[6][7]. This is not exclusive to accidents involving nuclear power plants (NPPs), as the mishandling of radiological material can also have large scale (impacting a large area / many people)[8][9][10] and local scale (impacting a small area / few people)[11][12][13][14] negative consequences.

Workers frequently using or exposed to radiological material are given specialized training on the safe use and handling of that material which is commensurate with the

potential consequences of postulated accidents. Depending on the scope of the responsibilities of the worker, that training includes emergency response procedures[15] for tasks such as conducting a radiological survey during an emergency. Workers have access to an array of radiation detection devices and personal protective equipment, further helping them manage the situation safely during an emergency. Emergency response training can extend beyond staff employed at a specific industrial facility and may be provided to local municipalities and their first responders who could be involved in the response to a larger scale accident. Under the context of the post September 11, 2001 attack in the United States, the need for first responders to have robust capabilities to mitigate and respond to hazards of all natures (chemical, biological, radiological, nuclear and explosive) remains relevant 20 years later[16].

Nuclear and radiological emergencies have the potential to happen anywhere, including in rural communities without nuclear or radiological industries[5][17]. In these communities, general first responder training may include some elements of nuclear and radiological hazard recognition and initial response procedures to build up a core skill set to respond. Compared to their counterparts in larger communities with facilities such as NPPs this training will be infrequent. The skills that are developed during this training may diminish if they are not regularly practiced. Practicing these skills requires access to sources of ionizing radiation which a third party trainer would need to provide during the conduct to have realistic scenarios to train with.

Nuclear and radiological materials are used widely in nuclear fuel cycle facilities, NPPs, and in practices involving nuclear sources. The potential consequences from an accident involving nuclear or radiological material can be severe and widespread and

can potentially happen anywhere. Regardless, it is necessary to have trained responders available who can assess, understand and localize potential hazards. Training is often limited due to restraints related to time, money or local/regional prioritization with other hazards.

As stated, in Canada, and internationally, it has been recognized there is always a potential for nuclear or radiological emergencies to occur[6][7][8][9][10][11][12][13][14]. An emergency could be triggered by events such as the discovery of an abandoned source of radiological material, an accident at a nuclear power plant, spacecraft powered by nuclear materials re-entering the atmosphere, or even from a terrorist attack.

During an emergency outside of a fixed facility, first responders may have the responsibility for the initial management of the scene, isolation of the hazard, and protection of those in the area. In communities with minimal nuclear industry, the training provided to those responders for nuclear and radiological emergencies will be limited and they may have a general fear of radiation that has manifested from highly publicized emergencies (such as Chernobyl, Fukushima and Three Mile Island)[18]. Even if radiation detection equipment is available, tasks such as locating and isolating radiological material may be challenging, as the first responders may have had very few, if any, professional opportunities to practice this skill or to revisit this skill after learning it previously[18].

Advanced technologies for the localization and mapping of radiological consequences (such as robot and aerial based mapping solutions) may not be available immediately to those first responders, if at all. These responders will face challenges to understand the

extent and scope of the hazardous area despite the fact that they may be required to identify and isolate dangerous sources of radiological material (either through retrieval or the establishment of a cordon) with only the use of a handheld radiation detector providing a numeric value of its measurement and responding through an audible alarm and/or vibration when pre-set dose rate or accumulated dose alarm thresholds are exceeded. Although Canada is internationally recognized as having a well-developed and mature nuclear and radiological emergency preparedness and response (EPR) system across all levels of Government[19], the provision of better real-time support to first responders on threats and hazards continues to be a significant area of interest in activities such as the Canadian Next Generation First Responder Initiative[20].

1.2. Problem description

There are situations where someone (e.g. a first responder), who is not a specialist in radiation surveying, may need to survey an environment for sources of ionizing radiation. Advanced tools such as drones or GPS linked systems that provide a mapping capability are unlikely to be available in all situations due to their rarity and/or may not work in the specific environment (i.e. indoors). The surveyor can only reliably expect to have their handheld instrumentation with a display and will be reliant on skills developed during specialized training which may not be frequently reinforced in their routine training cycle leading to those skills becoming diminished.

During a radiation survey a lack of awareness of the radiation field distribution in the environment will cause a surveyor to spend more time searching for any sources of radiation. A lack of recent training opportunities to develop and reinforce skills to conduct a radiological survey may exacerbate the situation and further increase the time

it takes to finish their task. Under the basic principles of radiation protection using time, distance and shielding, any increase in time in a radiation field will result in an increase to their total exposure.

The nuclear industry operates under the concept of keeping exposures to ionizing radiation As Low As Reasonably Achievable (ALARA). In line with this, practical new technology and methodologies to help minimize exposure during such surveys are always needed to be researched, investigated and developed to increase user awareness during the conduct of a radiological survey and to facilitate the development and reinforcement of the skills needed during training activities. Achieving new solutions in this area will support adherence to the principals of ALARA.

1.3. Research objectives

The research described in this thesis demonstrates an approach to address the issue of reducing exposure to ionizing radiation to prospective radiation surveyors in two directions. The first direction addresses this issue directly by demonstrating an approach to provide enhanced real-time awareness to a user conducting a radiation survey without requiring expensive specialized equipment and in a form factor that is deployable in most environments (indoors and outdoors). The second direction addresses this issue by demonstrating an approach to facilitate the development and reinforcement of the skills needed to conduct a survey in a practical manner that could be deployed anywhere. By demonstrating both approaches to address this issue the current research contributes to the scientific toolset of the nuclear industry and outlines new strategies that can be used to better adhere to the principals of ALARA in some circumstances.

Related to the topic of increasing real-time awareness, this research demonstrates an approach to visualize radiation measurements in augmented reality (AR) (the merging of fabricated digital elements, such as 3d models directly onto live camera video footage enhancing awareness of the environment thereby augmenting reality) on a handheld cellular phone. A cellular phone is a standard piece of personal equipment that has a high degree of likelihood for being available to any radiological surveyor and therefore using this as a platform for visualizing radiation measurements demonstrates a readily available and practical solution to this issue. Furthermore, this research provides an example of how the use of the cellular phone to visualize the measurements taken opens up the possibility for new approaches for real-time analysis. An example of how modern reinforcement learning (RL) techniques can be used to provide artificial intelligence (AI) based advice to a surveyor in real-time further increasing their awareness of the situation and therefore reducing their time surveying is given.

Related to the topic of improving skills through training, this research demonstrates an approach to provide training to prospective radiation surveyors using modern virtual reality (VR) hardware that allows for realistic environments to teach and reinforce skills while also being highly portable, cost effective and requiring no sources of ionizing radiation (providing realistic fully digital environments where interaction is conducted with realism focused head mounted displays and 3d room scale movement). Furthermore, this research demonstrates the use of the AR technology developed as a tool to simulate the measurement of a source of radiation, essentially creating a digital twin of a hypothetical radiation field, demonstrating a potential new tool to train radiation surveyors which could be deployed anywhere.

1.4. Structure of this thesis

This thesis has the following structure:

Chapter 1: Introduction

- This chapter introduces the material and provides the problem statement for this research.

Chapter 2: Literature review

- This chapter provides a review of the available literature, focusing on real-time visualization solutions for displaying radiation measurements, practical and theoretical techniques, some approaches for radiological source localization, and a discussion of virtual environments for training.

Chapter 3: Theoretical background

- This chapter describes the theory behind the radiation measurement visualization, overviews the virtual reality hardware, discusses the use of virtual environments for training and overviews user comfort concerns in virtual environments

Chapter 4: Radiation field measurement visualization

- This chapter describes the technique that was developed to simulate the radiation field measurements, and how to provide real-time visualization in augmented reality of radiation measurements from simulated environments and live measurement data.

Chapter 5: Reinforcement learning approach to radiation source localization support

- This chapter describes an approach to further utilize the radiation measurement and augmented reality information to train a virtual agent using reinforcement learning techniques to provide real-time advice on the location of a radiation source based on augmented reality measurement data.

Chapter 6: Synthesis of radiation measurement visualization, source localization and augmented reality training approaches

- This chapter brings the previously developed techniques for radiation measurement visualization in augmented reality, source localization and virtual reality together and how they were deployed in experiments.

Chapter 7: Results

- This chapter discusses the results of the augmented reality approach to radiation measurement visualization, the results of the machine learning approach to source localization and the results and feedback from the use of the virtual reality training scenarios.

Chapter 8: Summary and conclusions

- This chapter summarizes the research conclusions and suggests further research development.

Annexes

- Several annexes are provided which include the C# code that was used in this research, relevant dose conversion factors that were used and a summary of some lessons that were learned that were of a subjective nature related to the use of virtual reality for training radiation surveyors.

CHAPTER 2: LITERATURE REVIEW

2.1. Radiation surveying

As a general description, to characterize an ionizing radiation field¹ within an environment (also known as conducting a radiation survey), a user takes a radiation measurement tool and moves this tool around the environment (either personally or via a remote device such as a robot) to ascertain the radiation field in the area(s) of interest or concern. This results in a data set of measurements taken with the measurement tool at discrete locations in the environment. The coordinated data set becomes the characterization of the radiation field in that environment and each unique measurement can be associated with a specific location. This dataset can be stored in a computer system, written down on paper or just kept in the mind of the surveyor(s) who will make decisions on where in the environment to explore further and what tasks to complete based on this data.

Many factors impact the shape and distribution of a radiation field. This includes the specific activity and type of material releasing radiation, the type of radiation being emitted (e.g., alpha, beta, gamma, neutron), the transport pathway of the radiation through the environment, and the amount and configuration of any material that interacts with (i.e. shields) the radiation in that environment. Ultimately, the entire field distribution can be considered to be a complex interaction of all these factors which if proper analysis is performed can be deterministically modelled or stochastically simulated.

¹ For the remainder of this thesis, any reference to radiation is in reference to ionizing radiation unless specifically stated otherwise.

Having methods to verify and quantify the radiation field within an environment is important for the nuclear industry. From decommissioning operations to ensuring an area is below legal limits before it is released for public use[21], to emergency response operations to determine if public protective actions need to be implemented[22][16], radiation surveying remains a critical function in these processes. Radiation surveyors are often challenged to understand and communicate the coordinated data set of radiation measurements collected during a survey. Methods have been devised to communicate radiation field distribution in an intuitive manner[23][24]. Ultimately the method of communication chosen directly impacts the understanding of the measured data and therefore influences their understanding of the radiation field distribution in that environment[25][26].

The accident at Tokyo Electric Power Company's (TEPCO) Fukushima Daiichi Nuclear Power Station in March, 2011[7] has resulted in an enormous volume of radiation survey data. Surveys have been performed by aerial, vehicular, robotic and human mounted technologies using a wide array of detector types and subsequent methodologies to communicate the results. The following figures provide several different examples illustrating the different types of communication products that were developed by responders to share the results from radiation surveys that have been taken in and around the Fukushima Daiichi Nuclear Power Station after the accident in March, 2011.

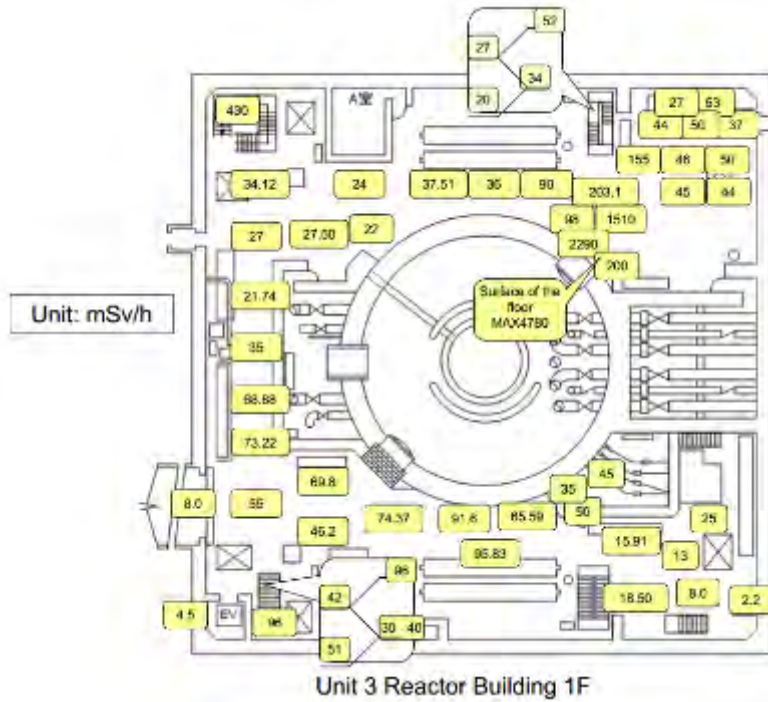


Figure 1 - Example of results from a radiation survey of the Unit 3 Reactor Building First Floor at the Fukushima Daiichi NPS[27]



Figure 2 - Example of robot survey with gamma camera survey taken inside Reactor Building for Unit 1 at the Fukushima Daiichi NPS[28]

Survey Map of the Entire Fukushima Daiichi Nuclear Power Station
 (Used in the Measurement Performed on May 7-10, 2013)

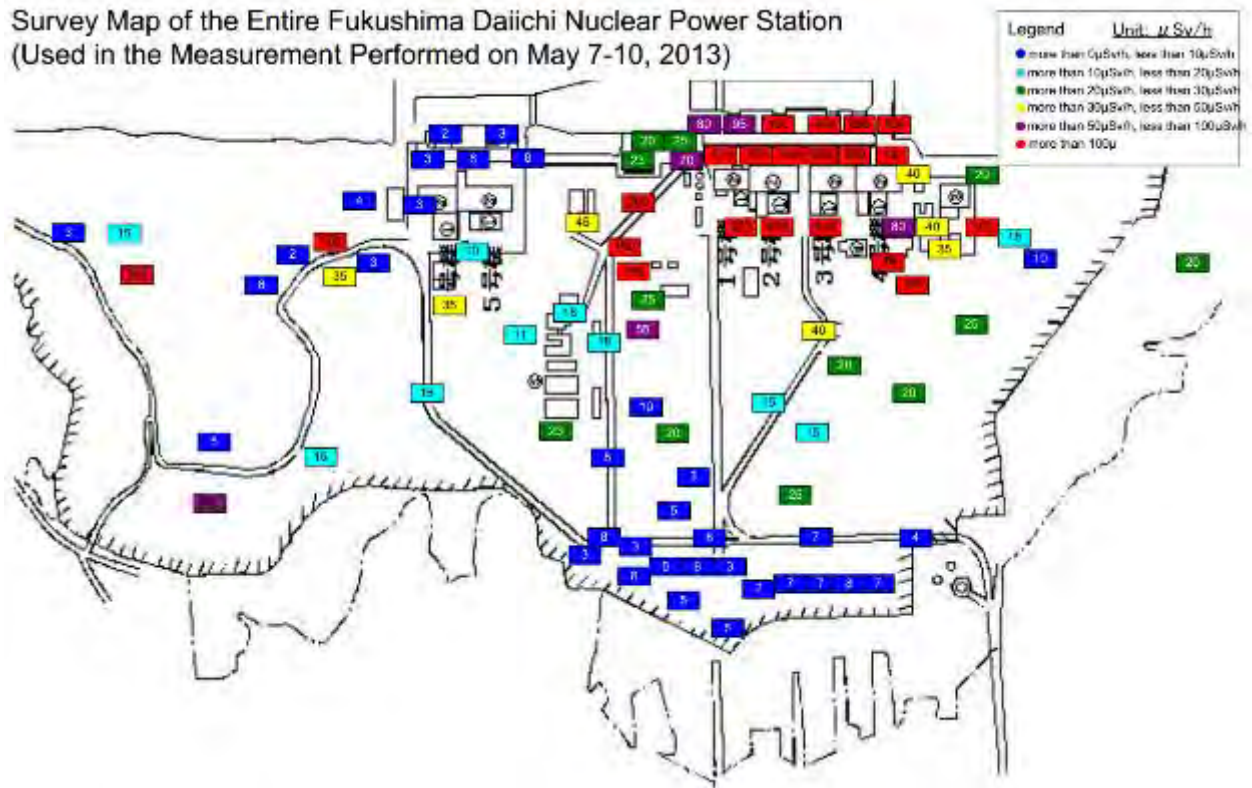


Figure 3 - Example of a series of handheld radiation surveys taken around the Fukushima Daiichi NPS[29]

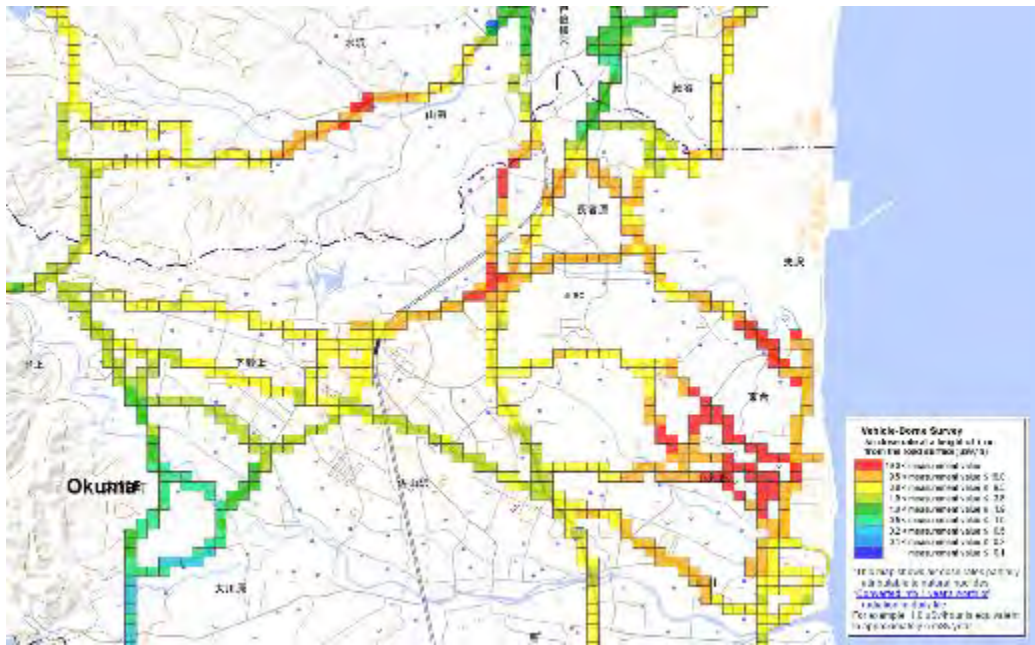


Figure 4 - Example of a series of vehicle borne survey taken outside of the Fukushima Daiichi NPS[30]

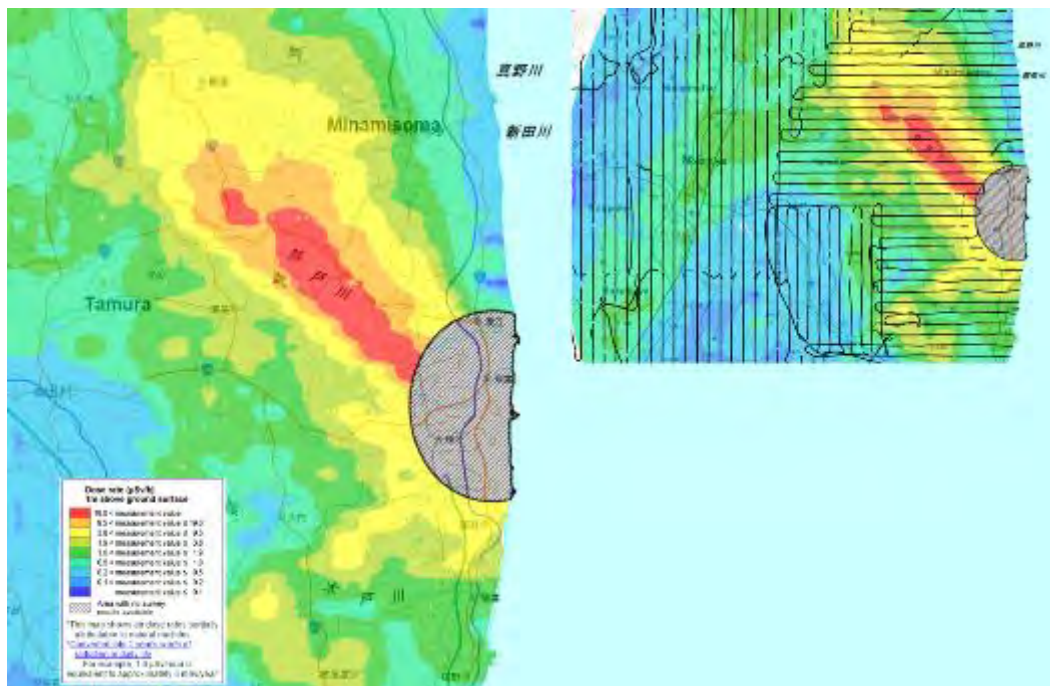


Figure 5 - Example of interpolated results from an aerial survey in the region around the Fukushima Daiichi NPS (with the flight path indicated on the right)[30]

These figures provide just a few examples of common surveying techniques used and the resultant types of measurement products created every day in the nuclear industry. Figure 1 and Figure 2 highlight surveys taken inside a building. The approach of the survey in Figure 1 was to have a small team with handheld equipment walk around a building and manually record the results on a map (measuring at a fixed height to make all results easily comparable) which were used to create a visual image afterwards, with a singular measurement of a contact dose rate on the floor being highlighted as an outlier result of interest for further investigation. Figure 2 relied on the deployment of a gamma camera which provides field intensity information in two special dimensions which can be overlaid on an image to show different regions of intensity of a radiation field as they related to the physical environment.

Figure 3, Figure 4, and Figure 5 show the results of surveys taken outdoors. The process used to produce the map in Figure 3 is identically the same as the process used for Figure 1 inside the building, namely, using manual location data. Figure 4 and Figure 5 highlight the usefulness of having positional data automatically associated with the detection equipment. Measurements in both these examples was taken and coordinated with geographical information systems (GIS) data from GPS that were used to automatically create maps colorized by the intensity of the measurement in a given location. Figure 5 demonstrates the effectiveness of interpolating the space between measurements to fill in areas which were not measured by the discrete measurements to provide improved situational awareness.

All of these surveying results have a common quality - the results only indicate the relative radiation field measured in reference to a single plane of spatial orientation.

This is a very common quality in maps of radiation surveying results. The exception is Figure 1 which includes a specific measurement taken on contact with the floor implying the surveyor was investigating the environment not just at one fixed height, but rather at many heights in three dimensions but chose only to include a singular point of interest at a different height than the rest of the measurements. The spatial plane of reference in Figure 2 is at a different orientation than the others but it still provides only the data measured by the sensor at the depth it was in reference to the source of radiation in the image. The results from Figure 5 were taken from an aerial monitoring system hundreds of meters off of the ground but have been adjusted and calibrated to display the results in reference to 1 meter off the ground, which is a common standard for aerial surveying².

In practice, radiation surveyors almost always navigate an environment in all available dimensions by moving their detection equipment to investigate areas of interest above, below and laterally. It would be expected for a trained survey team to move their equipment closer to any instances of elevated readings to locate and better characterize such points. Although radiation transport is a process which occurs in 3 dimensions, the results from radiation surveys teams are almost always shown and communicated in reference to only one plane of reference (generally height). For emergency response monitoring a recent paper provides an overview of gamma radiation detection systems in use for emergency radiation monitoring[31].

² This is common as the implementation of many important public protective actions during emergencies, such as evacuation, sheltering and temporary relocation, is normally intended to be based off of measurements taken at 1 meter in height[22]

Outside of emergency response radiation surveying, other types of monitoring such as surveying and mapping using gamma ray spectroscopy information have been shown to be successfully used in industries such as uranium mining for many years. This is a well-studied field with techniques and tools to help understand and visualize that data post surveying activities[32]. However, the specific focus of this research will be to use techniques that focus on managing singular dose rate data related to a measurement taken at a specific point rather than binned spectral energy data as this would be the most likely measurement able to be taken from emergency instrumentation not belonging to a specialized team.

2.2. Visualization of ionizing radiation fields

2.2.1. Practical techniques

There have been a number of approaches over the years to develop techniques to visualize radiation fields. Attempts to develop low cost solutions using cellular phones and fixed camera systems have been developed as early as 2011[33][34]. This approach uses the camera sensor in these devices which, if blocked from receiving visible light, can still produce a measurable visual signal seen as static noise due to interactions between the charged-coupled device (CCD) or the complementary metal oxide semiconductor (CMOS) sensors and the ionizing radiation. If recorded over a period of time and evaluated, the level of static noise seen in the recording can be used to try and quantify the radiation exposure. While this approach is an interesting use of this physical effect to attempt to quantify ionizing radiation, its use for real-time visualization of radiation as desired in this research would be limited as measurements

are slow to perform in low exposure fields or require operation in environments not suitable for hand held equipment (but suitable for robotic mounted equipment).

Another novel technique to visualize a radiation field employed radiophotoluminescence photography where pulverized glass particles were placed into polystyrene balls that were moved into a grid formation in a high radiation field area. Post exposure, these balls would react with an ultraviolet light and 'glow' which could be picked up with sensitive camera equipment or the naked eye at large doses (>5Gy) and be used to attribute the dose each ball received[35]. This approach required post-exposure processing and is not suitable for real-time visualization.

In 2012 an application was developed using a radiosensitive polymer formed into a sphere that was placed into a tungsten collimator and put into a high radiation field environment [36]. The analysis post exposure using an optical computer tomography scanner allowed for the 3d characterization of sources of radiation in that environment. An example of the apparatus is provided in shown Figure 6. Similar to the previous approach, this approach does not provide real time visualization.

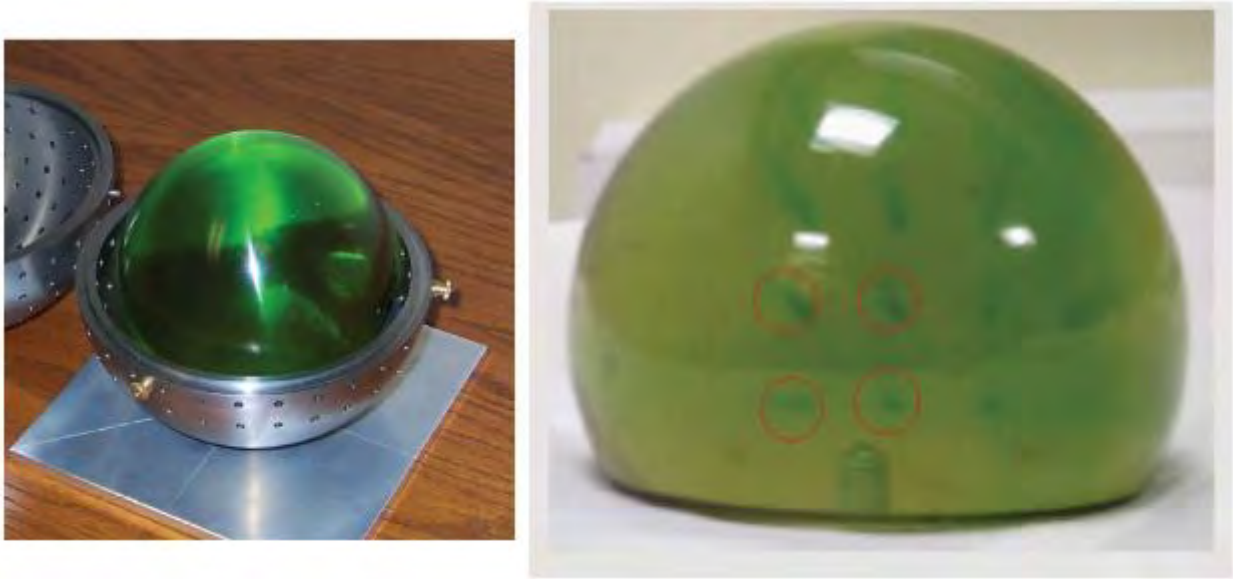


Figure 6 - Example of a radiosensitive polymer which can be used to visually display the direction of radiation field intensity[36]

While this research is focused on measuring ionizing radiation fields in a large environment (e.g. room scale / building scale), a similar and parallel scientific field that has been challenged with visualizing large volumes of 3d measurement data can be found in the medical area. An in depth discussion of volume rendering techniques and how they have been applied to medical data within the body of a patient outlines many of the different techniques used in this field[37]. These techniques are interesting and useful for their specific needs; however having the need to visualize such mass quantities of measurements from a handheld survey is unlikely.

Progress continues to be made with technology to support the development of portable gamma-ray imaging instrumentation that can provide radiation visualization paired with live photographic imagery taken from a traditional camera. In 2018, a demonstration of an approach showing the potential of using multi-model data fusion for radiological

localization and mapping problems was made where gamma-ray imaging data was combined with contextual data from portable platforms[38]. This is an area where a large amount of research is currently focused in developing more improved and sensitive detectors. The research in this thesis seeks to visualize dose rate measurement information using a more simplistic approach, only requiring a modern cellular phone and a dose rate meter.

The visualization of alpha and beta sources which generally do not have the energy to reach the detector apparatus held at a typical survey height of 1m has resulted in techniques that focus on visualizing the secondary effects such as ultraviolet fluorescence from the radiation interactions with air[39].

2.2.2. Simulated and augmented data

There have been novel augmented reality approaches to visualizing radiation fields in nuclear engineering and science[40][41]. An approach was taken in 2002 with some early technology that provided positive indications that showed augmented reality radiation readings did not negatively impact the performance of workers[42]. This original research was promising but also required the use of specialized prototype equipment instead of something more commonly available, such as a cellular phone, which is a priority of the research in this thesis.

In many ways the medical field has been at the forefront for the use of augmented reality tools as making medical practitioners more aware of the local environment when performing surgeries and providing treatments would result in better outcomes for patients. An approach in 2012 involved the visualization of magnetic resonance scan

data over a patient during a surgical procedure to support the surgeon's awareness during an operation[43]. Developments in the medical field use many of the same approaches and technology that is used in the industrial fields however they take a smaller scale approach focusing on improving treatment during surgery, rehabilitation, education and training[44].

The visualization of radiation fields has been used to help better understand the radiation distribution in the Experimental Advanced Superconducting Tokamak reactor to improve the efficiency of maintenance work and decrease exposure of personnel in those areas in 2010[45]. This approach has also been used for path planning of maintenance workers during simulations of maintenance activities at nuclear power plants, including showing visualizations of the radiation fields to the workers prior to their activities in 2013[46] leading to improved performance as they had better understanding of the local environment. This research demonstrates that giving users better awareness of the radiation distribution around them supports improving their activities (in terms of minimizing exposure to dose). However, in this research the radiation field visualization was done before the workers entered the environment and there was no real-time 3d display aspects using a handheld device.

In 2018, early work in the development of a solution to combine photographic data combined with radiological measurement data from a Compton camera to measure and visualize a radiation field in a virtual reality space was demonstrated in the Unity game engine and visualized with Google cardboard virtual reality technology[47]. In 2019 a conference presentation highlighted a number of recent activities in this area of virtually displaying radiation field data[48] showing approaches to combine static models of

environments in an NPP taken and processed from point clouds, combined with radiation measurement and interpolation techniques to approximate the radiation field distribution in an environment in an off-line setup. This technique is different than the real-time approaches pursued in this thesis research.

An approach in 2014 involved the visualization of a radiation field emitted by a robotised x-ray imaging device which was calculated using Monte Carlo methods in GEANT4[49][50][51] and displayed using a depth mapping camera as a clinician moved about the environment[52].

In 2017, researchers utilized results from a survey of high-dose locations in Fukushima prefecture region of Japan using a Compton camera was made which produced very comparable results to a traditional radiological survey[53]. Other approaches using a Compton camera mounted on a drone have also been shown in 2017[54]. These techniques have been utilized in areas outside in the Fukushima region in activities as recent as in 2020 with good success[55] and approaches have been shown using similar techniques elsewhere[56].

In 2019, an approach was demonstrated which the fusion of 3d data with radiation measurements. The intention of this was to facilitate and overcome one of the major public communication issues, which was the lack of a reliable method to visualize radiation to support broader public understanding[57].

Modern approaches in 2020 have leveraged real-time visualization capabilities of hardware such as the Microsoft HoloLens technology to provide real-time indication of radiation sampling locations with visualization of collected measurements in near real-time[58].

2.3. Source localization

Radiation measurement data represents an instance of a value associated with a location in 3d space. If taken with a handheld radiation detector, such as a small Geiger Muller tube or plastic scintillator, this data will also be associated with an integration time representing the period during which the 'measurement' data was captured to determine a unit of measurement per unit time. Taking measurement data and analyzing it can yield information related to where the location of radioactivity is originating thereby localizing the source of the radiation (within an approximate area).

In 2007, a paper compared three different approaches for localizing a single high activity (suitable for a dirty bomb or other radiological dispersal device scenario) radiation source[59]. The algorithms evaluated include the maximum likelihood estimator compared to Bayesian estimation[60] using an extended Kalman filter and an unscented Kalman filter, concluding that the maximum likelihood estimator approach performed the best[59].

A previously discussed, source localization technique from 2012 used reverse ray tracing approaches utilizing the properties of gels that reacted to radiation and collimated shielding to produce visible 'rays' that point towards the source of radioactivity[36] as its localization technique. While not exactly a solution useful for a first responder as each measurement requires the use of a single gel ball, the novelty of this solution for localization in high activity and other niche case environments is notable.

An approach was discussed in a paper in 2016 where a maximum likelihood localization approach for radiation sources with unknown source intensity was further developed[61]. In 2019, an approach was shown utilizing a Gaussian process approach for dose mapping radiation fields and was demonstrated for the purpose of mapping the doses in a complex radiation field in an experimental and a simulated environment[62].

In 2020, an automated heterogeneous robot system for radiation surveys was demonstrated which consisted of an unmanned aerial system taking measurements from the air which supported a terrestrial mapping solution with a ground based robot[63]. There is also research which discusses approaches for stationary detection networks to track the movement of a mobile radiation source[64] as they pass by sensors thereby localizing the radiation sources.

Low level radiation detection measurement requires careful consideration of integration time as there may be very few electronic events taking place in the detector to generate measurements which therefore require a longer counting time be used. This research focuses on the types of situations where first responders would be conducting monitoring where the source of radioactivity is clearly able to be distinguished above background levels and where measurements could represent a potential hazard to the surveyor or to the public if the source of radioactivity is not localized and controlled. In these types of situations, investigating new novel methods to localize a source of radiation and provide the potential for real-time advice would be useful for the first response community.

The use of handheld directional gamma ray detectors to support the rapid localization of radiological sources has been a tool available within the Canadian Department of

National Defence for a number of years (first shown in 2005) in several form factors[65][66]. These directional gamma ray detectors provide visual indications of the directionality of a radiation field to a user and display an approximate location for the source of the radiation as each measurement is taken in terms of providing a compass bearing. Advancing on this approach in 2014, there was a paper discussing approaches for the development of statistical methods to localize and source activity estimate with multiple directional gamma ray detector sensors [67] beyond simply providing a compass bearing. This research developed two models to analyze statistically fluctuating data taken from several directional sensors using a point of closest approach and maximum likelihood method evaluation processes to develop two generic algorithms to perform source localization given multiple-sensor detection system is deployed. Building on elements of this work a demonstration system was developed called the Point versus Area of Contamination Kit (PACK) system[68] which was designed to be able to infer if a radiation field is coming from a single isolated source (and where it is localized using the previous methods) or if it is coming from a larger distributed source (i.e. a field of contamination). The results showed that it was possible to gain information relevant to localize a source using this approach but that its performance was noted as poor given the experimental configuration with some suggestions recommended to improve for future research.

In some ways the literature described above addresses the basic scenario used in this research, making the assumption that a source is fixed in a location in the environment and that the radiation detector is mobile or a series of detectors are used providing an array of measurements from different positions. However, rather than seek an

analytical solution based on an algorithm derived from a point of closest approach or maximum likelihood method evaluation processes, this research sets out to achieve an alternative solution. Taking inspiration from the latest developments in the field of machine learning research, this research will instead leverage current methods to solve generic problems in a reinforcement learning environment and will create a game like scenario where an agent, driven by a neural network, will be tasked with solving the problem of estimating the location of the radiation source based on measurements taken from a process providing data similar to the augmented reality surveying process. The benefits of this novel approach are that such algorithms have been shown to be highly adaptable for solving a wide array of problems and therefore if demonstrated with a single generic radiation source localization problem, it would provide evidence that increasing the difficulty and scope of the scenario (i.e. more sources or more complex environments) would provide solutions to problems which are more challenging to solve with a derived analytical solution.

2.4. Virtual environments for training

The systematic approach to training is “*an approach that provides a logical progression from the identification of the competencies required to perform in a job to the development and implementation of training to achieve these competencies, and subsequent evaluation of this training*”[69]. Within the nuclear and first response industry, the systematic approach to training has been mostly standardized as the method for maintaining the stringent quality and required performance metrics for important roles, such as those related to safety. International safety guidance[70]

strongly states that “*A systematic approach to training should be used for the training of [nuclear power] plant personnel*”.

Training regimes vary across all parts of the nuclear industry. Very well documented and regimented training programs exist at nuclear power plants (NPPs) which are commensurate with the licensing requirements and the potential hazards of the work. Detailed records are kept to prove staff competencies to regulatory authorities. This can be contrasted with training given to those working in small laboratories or with mobile radiological devices (e.g., density gauges) which is less rigorous while still remaining commensurate with the potential radiological hazards of the work and any regulatory requirements.

A training program may include any number of different elements in its design. This could include a combination of classroom instruction, self-study material, simulator based exercises, full scale exercise scenarios or appropriate examination and performance review.

With the need for training comes a need for the verification of the students' successful comprehension of that material. Verification can take place through activities such as tests, examinations and/or monitoring performance during exercises. Tests and examinations allow for the verification of a student's acquisition and ability to apply knowledge. Exercises can provide an opportunity to evaluate the educated content (i.e., procedures, response arrangements, etc.) and test if there are gaps within this material.

Providing training and associated realistic exercises for extreme events such as a beyond design basis accident at a nuclear power plant or an event involving a Category

1 source[71] discovered outside of its shielding can be challenging. Classroom instruction often focuses on the theoretical, whereas interactive exercises can focus on the practical application of knowledge and procedures but only within safe environments. Exercises that involve scenarios with very high ionizing radiation environments can be challenging to realistically construct as there will be radioisotope license restrictions and safety concerns. This remains an area where technical innovation should be required to provide instructors a means to properly exercise students. Such challenges have resulted in all manner of innovative simulated radiation detectors which can generate real-time results to users showing high radiation measurements during training without exposing students to such fields. Modern technology continues to yield newer and better solutions for creating such artificial environments for training and exercising staff in all industries and as newer technologies are introduced, the nuclear industry should continue to evaluate their implications for potential impact on training.

CHAPTER 3: THEORETICAL BACKGROUND

3.1. Generic approach to radiation field visualization

In previous research from 2010, a generic methodology to visualize radiation field data with three dimensional context by constructing voxel models and using transparency effects to allow an observer to effectively 'see' through the field was developed and demonstrated as effective and robust[23][24]. The Figures which follow show the simplistic construction of one of these visualizations followed by an example of such a field viewed in a virtual room and superimposed over an image of a real environment[23][24].



Figure 7 - Elements types 1, 2, 3 and 4 (from left to right)

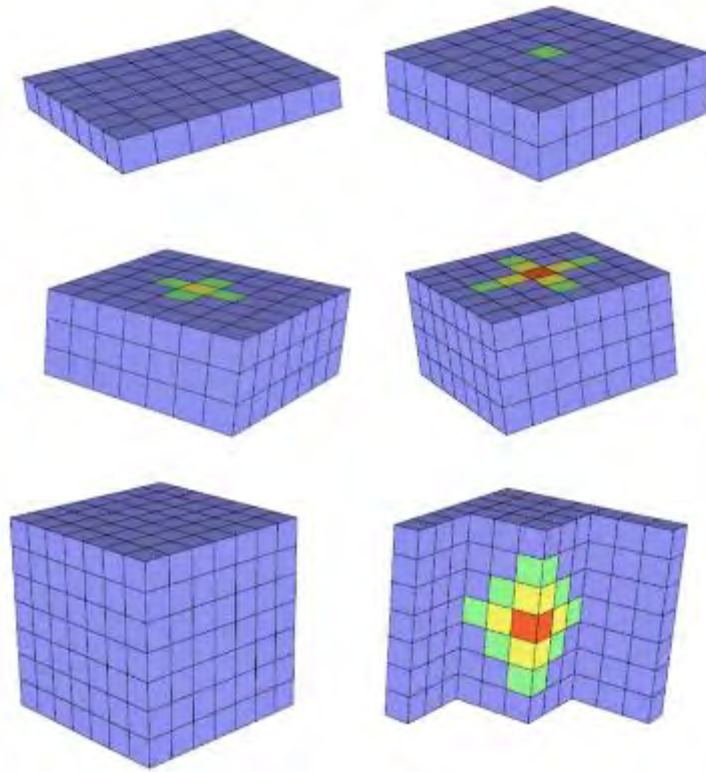


Figure 8 - Construction of the 7x7x7 field model sequence with radiation source at the center element

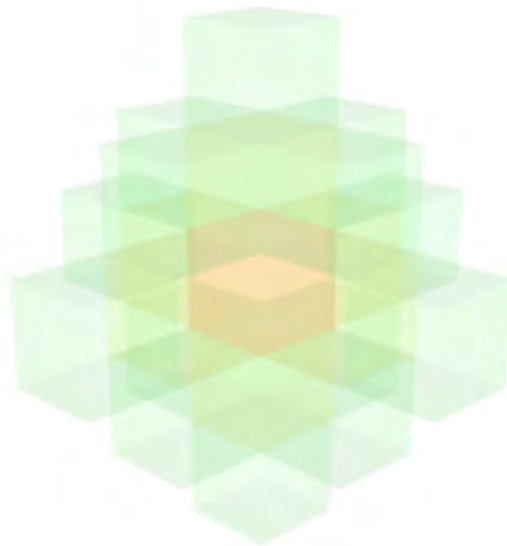


Figure 9 - Image from (previous figure) with one material made completely transparent and the others at 20% transparency producing the visualization effect

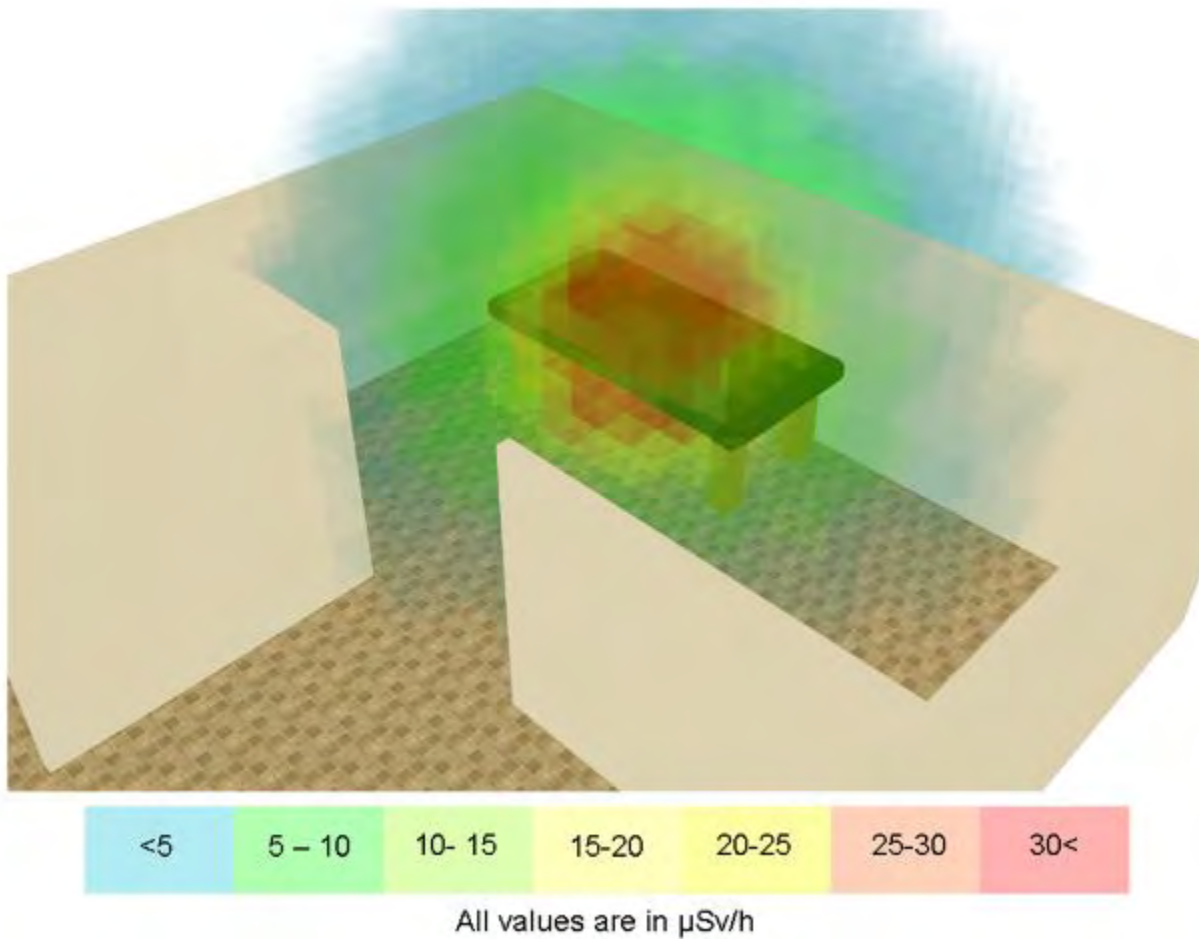


Figure 10 - Point source on a virtual table with scale (isometric perspective) and 3d visualized radiation field (higher fidelity model from a 25x25x25 grid with the source at the center)

This methodology is summarized as follows:

1. A data set containing (x_n, y_n, z_n, V_n) is taken (where x_n , y_n and z_n represent coordinates, and V_n represents a 'value' at those coordinates)
 - a. The process requires that x_n, y_n, z_n , values be at a fixed distance apart to establish a fundamental unit size for that model (e.g., if they are all values

at a 1m, 2m, 3m, etc in all directions, this process will establish that each (x_{n+1}) is equal to (x_n+1m) and the fundamental unit size is a cube of 1mx1mx1m)

2. A script (automated action instruction) is prepared to automate construction within a 3d modeling program. It reads those values and constructs a fundamental element at each location (a cube is used for convenience). This script includes a scale where the V value is assessed and each element is coloured based on its value.
 - a. The program is opened and the script is run to evaluate each (x_n, y_n, z_n, V_n) and a shape is built (centered at the coordinate or other reference point)
 - b. Based on the V_n value, that new object (volume) is assigned a colour, material, or whatever the term the program uses to define the external appearance of an element
3. This process then repeats until a shape has been built at all of the locations specified in the data file.
 - a. During the construction process, different thresholds can be assigned to values of V. For example, if V is: $5 > V > 3$, then color = light blue, which means any time a shape is built, and the V value is less than 5 but greater than 3, a color value of 'light blue' will be assigned.
 - b. A transparency effect can optionally be added to complete the visual effect.

This simple process (herein called the radiation field voxelization) can be used to display any appropriately characterized radiation field for communication purposes as well as used to display more limited data such as individual measurements. This process is similar to the methodology frequently employed to visualize MRI data except the radiation intensity of the field is substituted with the density of the volume as the basis for the visualization[37].

This generic methodology will be used as the basis of the radiation field measurement visualization due to its generic approach that should be applicable in most situations. In addition, as the goal of this research is to localize and provide visualizations of radiation sources using only a cell phone and linked radiation detector, this method leads itself to be well suited for representing sparse measurements as single voxels suspended in space which should not be computationally intensive.

The visualization approach used is similar to approaches that have been seen in previous research described in Chapter 2. This is not the only method that has been developed to understand and communicate the 3d transport of radiation, however it remains an openly available and fully documented solution which is easily transferable to different computing applications provided they meet a minimum set of requirements³. Furthermore the selection of this approach was partially inspired by a simulated picture of a radiation source on a table visualized by this generic methodology as seen in Figure 11[23]. An approach to produce similar results in real-time showing the

³ The minimum set of requirements for this technique to be used requires: The capability to automate construction actions within a 3d modeling program, the capability to read an external data source (containing x_n , y_n , z_n , V_n) and if desired, the capability to render transparency effects including through multiple layers of transparent materials.

measurements of such a source will be investigated and developed further into this research.



Figure 11 - Point source on a table with scale (isometric perspective) and 3d visualized radiation field

3.2. Requirements for a voxelization approach to be viable as a real-time display technique

In order for the voxelization approach to radiation field measurement visualization to be viable for the real-time augmented reality visual, certain information will be required to be known.

- Location of each radiation measurement with its context in relation to all others taken and the environment.
- The orientation on all axis of the viewing device and its field of view of into the environment.
- The position of the viewing device in the environment relative to the source of radiation.

With this information, it is possible using real-time 3d graphic techniques to overlay the 3d radiation measurement data over the live viewing data from a camera. Different techniques have been used over time to provide this data, which include using Global Positioning System data for 3d space reference, electronic compass data for angle of view space and technology such as accelerometers and gyroscopes to monitor the change of the position of a device as a user moves it.

3.3. Virtual environment for training and testing

3.3.1. Emergency response exercises

In general terms, the purpose of an exercise is to practice an activity so that those being trained are able to demonstrate their competency, verify that those who have taken training meet the expected performance level, evaluate plans and procedures to ensure they are comprehensive and effective and to explore alternative arrangements and procedures to see if improvements can be made[72]. The design of an exercise requires that specific objectives which are to be evaluated are developed and defined at the onset. The objectives can be as simple as successfully demonstrating performance capabilities via hearing verbal description of the response to a scenario provided to an

controller from a student, or more complex such as attempting the identification of deficiencies in performance of personal so areas where improvements to procedures to can be recognized or to evaluate the effectiveness of new procedures.

Different types of exercises can be part of a training program depending on programmatic needs and available resources. The type of exercise that may be needed within a program should be commiserate with the needs to verify the content that has been learned and/or the arrangements in place are sufficient. As the need for verification of the capabilities of a student(s) increase due to the hazard of the work, the potential complexity of the exercises needed to validate this also increases. A progressive approach to training is needed which corresponds to the validation need. Figure 12 summarizes the relationship between the scope of capabilities of the students needing to be validated and the complexity of the arrangements needed to conduct that validation.



Figure 12 - Progressive approach to training and exercises[73]

Seminars and workshops provide open discussion forums for the exchange of ideas and discussion of potential solutions. Students can be free to discuss and propose solutions which can be debated and discussed with other students and the instructors. These opportunities for discussions can allow students to discuss the procedures, issues and potential solutions without consequence for a ‘wrong’ solution (i.e., deviations from the procedures).

Tabletop exercises provide opportunities for group discussions of problems with the potential for all stakeholders in the solution being involved. The response to an issue can be discussed as procedural challenges are identified and the ultimate solution being

theoretical and based on the group discussions guided by a progressing narrative. Tabletop exercises provide positive opportunities for many stakeholders to discuss issues and actions in a comfortable environment. They can be particularly useful when discussing responses to complex events (such as an offsite release at a nuclear power plant) where many different organizations will have a role in the response arrangements. Tabletop exercises can allow the participation of representatives from all involved organizations, including remote participation, and can be cost effective.

Comprehensive training programs can include a combination of drills, partial and full-scale exercises[72]. Drills are small, focused scenarios where participants (single or small groups) can practice and demonstrate proficiency of specific skills under instructor/controller supervision. Partial and full-scale exercises are larger scope scenarios where entire organizational infrastructure including internal dependencies and outside interlinks with others can be included in the testing and evaluation. In the interest of realism, tools such as nuclear power plant control simulators and quantities of radiological material are frequently used so that participants can implement their full range of procedures, including the use of any needed tools and personal protective equipment to accomplish individual tasks to meet objectives.

Creating a realistic exercise environment remains a constant challenge for exercise designers in the nuclear and first response industries. Nuclear operator training has resulted in the nuclear industry creating full scale simulators which provide a complete duplicate work environment for operators to practice procedures in routine, upset and emergency conditions with a high degree of realism mostly limited by the capabilities of the physics simulations providing the response of the system. First response personnel

who may infrequently or rarely encounter radiological material may not be provided such comprehensive training environments in which to practice their procedures. Exercise designers must develop methods or purchase commercial equipment which can be used to simulate scenarios that may be encountered such as discovery of dangerous radiological source by a first responder. To simulate radiation measurements during exercises practical methods can be used which may include the use of small radiological sources which provides a detector response but at a lower level than would be encountered in reality. Specialized equipment is also available which can be used to simulate detector response to more active radiological material. This equipment can use different methods such as GPS telemetry to determine infield detector response, inaudible sound frequencies (which follow inverse an square response relationship with the distance to the audio source) with equipment which mimics the response of a radiation detector however is actually a specialized microphone, or even equipment which detects a chemical reaction which can be interpreted as localized contamination.

These different solutions are necessary because undue exposure to very high ionizing radiation fields during training and exercises is not practical as it does not meet radiation protection standards (e.g., adhering to the principals of ALARA) and it can potentially push employees towards radiological exposure limits which may then require them to be reassigned to other duties until their exposure period resets.

Since training is an essential element of maintaining nuclear safety culture, novel and innovative solutions are constantly being evaluated that can improve the type of training that can be delivered. The solution described herein provides a technique that maintains

a balance between the complexity and scope of a partial / full exercise while also having the cost benefit of being based on a computer simulation that can be repeated.

3.3.2. Virtual reality methodology

Virtual reality has been a developing field of computer science for decades and the use case for using this approach for training and education in other fields has been made in the past[74]. The technology has taken many forms, from head mounted displays to entire rooms with 360° projection display coverage to create the illusion of being in another location. Training in virtual reality has some advantages over existing methods for training. Virtual reality based scenarios allow a user an opportunity to use true to life movements to physically carry out many of the same motions they would if conducting operations in the field (such as moving their radiation detector upwards or downwards during the conduct of a survey).

Virtual reality technology has become far more commonplace in recent years as the technology has matured, the costs have become low enough for it to become a consumer product and the software development environment has become more accessible to novice designers. This research describes a methodology to construct virtual and augmented reality training environments that that can be navigated and a simulation of the response of an ionizing radiation detector is available to the user.

The hardware that was primarily used was the HTC Vive VR headset which was available for consumer purchase on 5 April 2016[75]. This hardware was chosen due to its commercial availability, the room scale tracking that was possible and the portability of the hardware. Figure 13 and Figure 14 show the included elements of the system.



Figure 13 - HTC Vive VR hardware (controllers, headset and headphones)



Figure 14 - HTC Vive base station mounted on camera stand (only one visible) and close-up

The HTC Vive system consists of a headset with a display that allows the user to see the virtual reality environment, two handheld controllers (often called wands) used to interact with the environment and two base stations which are used to help the headset and controllers determine their movements in the real-world environment. The system features SteamVR™ positional tracking which was developed by Valve Software[76] as its mechanism for tracking movement of the headset and controllers within the confines of a tracked volume (or play space)[77]. This setup supports a 3.5 m x 3.5 m space within which the user is able to physically move. Movement beyond this space requires an alternative locomotion method. Figure 15 shows a user with the headset on and

holding two controllers while in a virtual environment. Figure 16 shows a user demonstrating how they can reach and physically move to accomplish complex interactions in a virtual environment. The buttons on the controller for interacting in the virtual environment are labeled in Figure 17.



Figure 15 - A user in a VR environment



Figure 16 - A user in a VR environment demonstrating how interaction requires physical movement and engagement with the synthetic environment



Figure 17 - Labeled buttons on the Vive controller

The Unity Game Engine (herein denoted as Unity)[78] was used as the software platform to construct the virtual reality environment. Unity has a large and active community with many resources (both free and requiring payment)[79] and tutorials[80][81] available to learn how to use it. Alternatives exist which could have also been chosen which include options such as the Unreal Engine[82], CryEngine[83] or a custom constructed game engine.

In Unity, OpenVR[84]⁴ was used with the SteamVR[85] software development kit (SDK), which is available for free on the Unity asset store with its examples and scripts[86]⁵,⁶.

The hardware used for this example was a laptop with an i7-7820HK processor, 16GB of RAM, a 512 GB SSD as well as an NVIDIA 1080 GTX GPU⁷.

Prior to discussing how a virtual environment and simulated radiation field will be constructed in a later chapter, it is important to discuss the issue of movement and locomotion in a room scale virtual reality environment.

3.3.3. Moving in the virtual environment and user comfort

A room scale experience is one in which the player is able to move freely (as in walk, crouch or even jump) within a defined space in the real world. For the chosen VR technology, this space is minimally 2 m x 1.5 m and optimally 3.5 m x 3.5 m[87]. While within this volume the user is able to stand, turn, duck, crouch and otherwise freely navigate. When a play space is configured in SteamVR, the user manually defines the extents of this volume as part of the equipment setup activity. During use, when the user is approaching the limits, a chaperone system displays visible barriers indicating

⁴ In 2016 OpenVR was used for the training environments shown in this and the next chapter. As of 2020, Unity now provides a unified plug-in framework for VR called the XR Plug-in Framework (<https://docs.unity3d.com/Manual/XRPluginArchitecture.html>).

⁵ Although the HTC Vive is used throughout this thesis it is not the only virtual reality headset and controller technology. Using Unity XR with the SteamVR 2.0 interaction system provide more current solutions compatible with most existing VR hardware as well as potential compatibility for future hardware.

⁶ Although the research was conducted primarily with the SteamVR SDK, early interaction and testing took place also using the Virtual Reality Tool Kit asset that was available in 2016 and the numerous tutorials available on its creators Youtube channel at that time (<https://assetstore.unity.com/packages/tools/integration/vrtoolkit-vr-toolkit-64131>, <https://vrtoolkit.readme.io/docs> and <https://www.youtube.com/user/thestonefox/videos>).

⁷ The specific model was a 2017 Alienware m17 R2 laptop. Any laptop or desktop PC that is designated as VR capable should be sufficient for a similar setup.

they are approaching the edges of this space[88]. This is a warning to the user that they are approaching the limits of the defined safe space and should retreat. In order for a simulation to exceed the physical space limit a solution is needed which implements a form of locomotion.

Locomotion within a virtual reality environment is not a simple issue. The balance between keeping users comfortable and maintaining freedom to navigate is delicate. The most commonly employed solution is to use teleportation to move the player from one point to another. Teleportation typically leverages the 'controllers' as a pointer with a curved beam (curving upwards as it moves forward for a few meters then downwards) which intersects with the environment⁸. Where the beam intersects is the position the player will be moved to once they press the teleport button. A rapid fade visual will virtually move the player from its current location to the place this cursor points. This is by far, the most common method of traversing a virtual reality environment as experience has shown it is unlikely to contribute to user discomfort for the majority of people.

For radiological safety training, the use of teleportation in a virtual reality environment is not ideal. When planning and exercising possible exposure to ionizing radiation there is a need to traverse a scene in a realistic manner if a representative dose is to be monitored for the user based on their choice of navigation pathway in the environment. To calculate potential radiation exposure, the VR environment must be traversed at a

⁸ A straight beam is not used frequently due to its traversal limitations. A simple explanation is that if a straight beam is used the player will be unable to navigate to any surfaces greater in height than they can raise their hand and point. If an upwards pointing curved downwards beam is used, a player can angle the beam upwards on-top of higher objects to gain elevation not possible with a straight pointer

realistic speed with the user entering into all areas where real-time exposure would need to be monitored. In order to facilitate this, a common locomotion method which gently 'slides' the user around the environment was used (herein denoted as 'sliding mechanics').

The sliding mechanics work in the following manner which is visualized in Figure 18 and Figure 19:

- On the touchpad of the controller, the user places their thumb in the centre
- As the position of the thumb is moved forward, backward, left or right (or some combination of those directions) the user's position will be moved appropriately in that direction relative to the angular direction the controller is pointed (some users prefer relative to the headset direction).

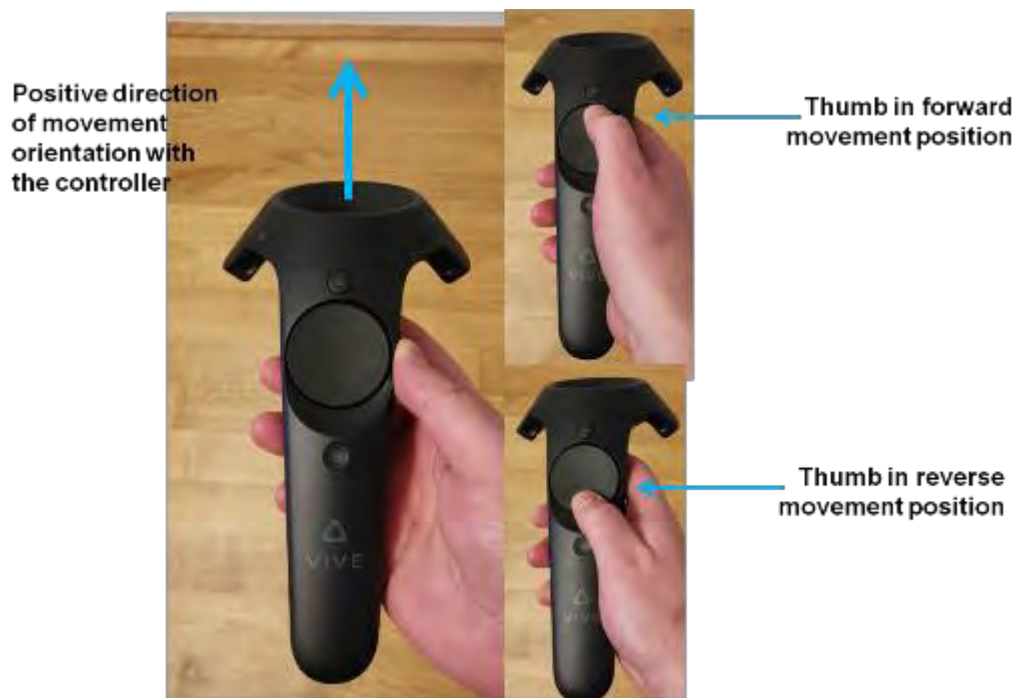


Figure 18 - Summary of forwards backwards slide locomotion

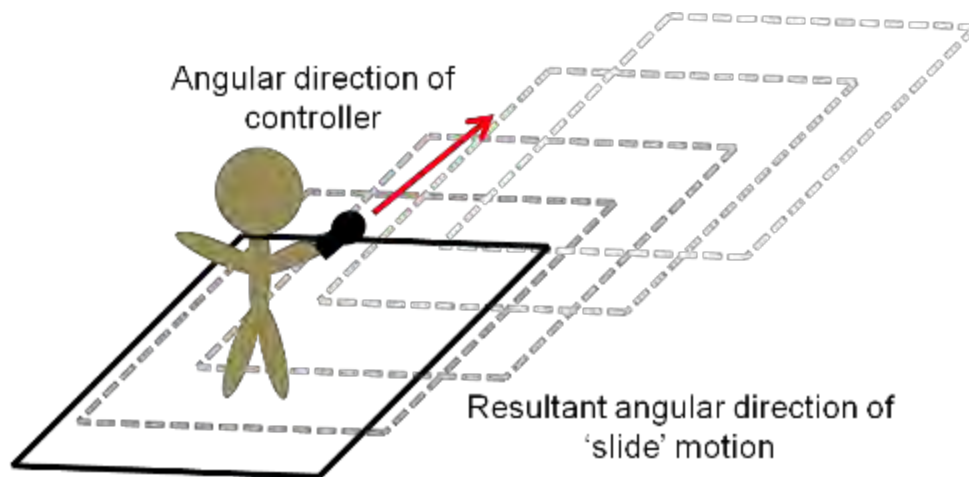


Figure 19 - Example of forwards slide locomotion

Several mechanics were experimented with before deciding which locomotion method to implement. While there are disadvantages and advantages to the different methods, this sliding mechanic successfully demonstrated itself to be robust and comfortable in the majority user experiences. However, issues that can occur in the uncommon instances where a user is uncomfortable when using the sliding mechanic include dizziness and nausea. These issues can be overcome by giving the users more time to familiarize themselves and become comfortable being in a virtual reality environment as well as taking measures to reduce the impact of the sliding motion. This can be accomplished by reducing the maximum speed the player is able to move, reducing the field of vision of the scene during movement with a visual effect (often called as tunnel vision) and reducing the change of acceleration of the movement to be more instantaneous which supports user comfort. Detailed discussions of the challenges

regarding locomotion and the different methods which have been implemented, are available in literature resources[89][90][91][92][93].

Locomotion is the most common cause of user discomfort when inside a virtual reality environment. However, the rate at which a visual frame is rendered and displayed to the user in the headset can also have an impact on comfort. The HTC Vive display has a resolution of 1080 x 1200 display pixels per eye (2160 x 1200 pixels combined) and operates at a refresh rate of 90 Hz. Maintaining a consistent frame rate locked at 90 frames per second (FPS) is important for ensuring user comfort. Many issues can impact the FPS delivered to the headset including the power of the hardware running the software and the complexity of the environment designed in the software that has to be rendered which could be manifested in the geometry of the environment, the lighting and graphical effects used in the scene or parallel non-graphic rendering related calculations that take place during the virtual reality experience and take processing capability away from the computer rendering the scene. Any scene designed for virtual reality is therefore needed to be tested and optimized as much as possible to maximize the FPS able to be delivered to the headset⁹.

⁹ Not maintaining a consistent FPS is a much studied problem in computer graphics for virtual reality. Solutions exist for situations where the target FPS is not able to be maintained. These solutions generally involve predicting when a frame will not be finished before the device refreshes the display (meaning the previously rendered frame would be displayed), and once that has been predicted they evaluate the previously drawn graphic frames and using motion prediction algorithms a synthetic frame is generated. Within SteamVR this is referred to as motion smoothing. (<https://store.steampowered.com/newshub/app/250820/view/2898585530113853534>). Other virtual reality technologies deploy similar custom solutions for this problem such as Asynchronous Spacewarp used by Oculus (<https://www.oculus.com/blog/introducing-asw-2-point-0-better-accuracy-lower-latency/>).

CHAPTER 4: RADIATION MEASUREMENT VISUALIZATION

4.1. Development of a flexible radiation transport model for virtual and augmented reality simulations

Throughout this research, an approach to model radiation transport was needed that could be implemented in different virtual and augmented reality environments. Several approaches were considered including implementing links with Monte Carlo models from software such as the Los Alamos National Laboratory Monte Carlo N-Particle (MCNP) Transport Code[94] and the development of a custom Monte Carlo approach which could be implemented via methods described in previous research[23][24]. As the focus of this research was on the modeling, detection and visualization of dangerous radiological sources, likely to be encountered by first responders, it was decided that the modeling approach did not require many of the advanced physics simulation possibilities from other codes and it would be sufficient if based on quicker to run calculation methods that can be performed in real-time in virtual and augmented reality.

4.1.1. Radiation modelling

In this section the nomenclature and discussion of how the gamma radiation measurement is simulated is provided. The way the formulas are expressed and the variables used are intended to as closely as possible match the description in the C# code that is provided as an annex so that it can be more easily understood.

A clear objective of the virtual reality simulation is to allow users the possibility to manipulate shielding material and sources in real-time in the virtual environment and to

quickly determine the radiation field reading in any location the user is able to access either virtually or physically (in the case of augmented reality with a simulated radiation source). In addition, for simulated radiation measurements in augmented reality, the calculation needs to run efficiently on the reduced capabilities of a cellular phone processor in real-time. As user comfort has to be maintained, it is very important that the radiation calculation process is efficient and that it not significantly impact the graphical rendering of the scene. For a virtual detector to provide a visual indication to a user of its reading, the time it takes for that calculation to run needs to be rapid as any extended calculation may impact performance of the system which can ultimately lead to user discomfort¹⁰. A real-time radiation calculation was built Unity which balances the need for optimization with the need for providing realistic detector response.

An approach was developed whereby the transport of gamma photons to the detector(s) in the scene would be modelled using the distance between the source and the detector combined with mass attenuation coefficients based on the energy of the photons to reduce the intensity of the radiation appropriately for any shielding material in between. A library of photon decay energies and associated probabilities of emissions was obtained[95] and put into a comma delineated text file that can be read by Unity. A library of mass attenuation coefficients for different materials¹¹ was constructed using the Radiological Toolbox (Rad Toolbox)[96][97] which has data available from a large library[98].

¹⁰ Careful planning and play testing is required to ensure that any potential sources for discomfort have been identified and to ensure that the scene has been appropriately optimized.

¹¹ The materials that were within this library were determine as needed for each environment and as new scenarios were constructed additional materials were added to the library. A default density was also stored for each material which could be manually overwritten as needed.

A virtual radiation detector was constructed in the Unity environment where the result of the radiation calculation would be displayed to the user. The calculation requires several key features:

- Ionizing radiation sources will need to be identified and associated with a radionuclide(s), an activity and an object in the environment. The radiation will be considered to be released from a single point at the centre of the radiation source.
- Objects in the environment which are to provide shielding will need to be identified and the necessary mass attenuation coefficients for each will need to be calculated based on the ionizing radiation source(s) and the energies of photons that are emitted.
- A virtual radiation detector will need to be constructed which will provide a measurement by computing the contribution of any radiological sources and accounting for any shielding present between the radiation source and the detection volume of the detector, which will be represented as a single point.

For an ionizing radiation source within an arbitrary environment (without shielding) the total energy fluence of photons $\phi_{unshielded}$ in $\text{cm}^{-2} \text{s}^{-1}$ incident at a distance d in m from the source can be determined taking into account each of the total photon energies n , that are being emitted E_i in keV, the associated probabilities P_i in % and the overall activity of the source A where:

$$\phi_{unshielded} = \sum_{i=1}^n \phi_i \quad (4.1)$$

Where the fluence decreases with distance following

$$\phi_i = \frac{E_i A}{4\pi \cdot d^2} \quad (4.2)$$

Therefore:

$$\phi_{unshielded} = \sum_{i=n}^n \frac{E_i P_i A}{4\pi \cdot d^2} \quad (4.3)$$

The radiation source and the radiation detectors will be considered as points in Cartesian coordinate space (i.e. (x, y, z)). Therefore, if the radiation detector is located at (x_{det}, y_{det}, z_{det}) and the radiation source is located at (x_s, y_s, z_s) then the distance between the two can be calculated as:

$$d = \sqrt{(x_{det} - x_s)^2 + (y_{det} - y_s)^2 + (z_{det} - z_s)^2} \quad (4.4)$$

d can be inserted into the previous formula:

$$\phi_{unshielded} = \sum_{i=n}^n \frac{E_i P_i A}{4\pi \cdot (\sqrt{(x_{det} - x_s)^2 + (y_{det} - y_s)^2 + (z_{det} - z_s)^2})^2} \quad (4.5)$$

This simplifies to:

$$\phi_{unshielded} = \sum_{i=n}^n \frac{E_i P_i A}{4\pi \cdot ((x_{det} - x_s)^2 + (y_{det} - y_s)^2 + (z_{det} - z_s)^2)} \quad (4.6)$$

To account for the impact of a shield between the source of radiation and the detector, the fluence of photons can be reduced using mass attenuation coefficients based on the material, its density and the length of travel on the portion of that vector between the point where it enters the object and exits. For a fluence of monoenergetic photons at a distance from a source that passes through a shield along the way, the formula is as follows:

$$I_i = I_0 e^{-(\mu_{den} * d)} \quad (4.7)$$

Where, I_0 is the initial (unshielded) particle fluence ($\text{cm}^2 \text{ s}^{-1}$) of monoenergetic photons at the detector

d is the penetration distance through the shielding material (cm)

μ_{den} is the density of the shielding material (g cm^{-3})

μ is the mass attenuation coefficient ($\text{cm}^2 \text{ g}^{-1}$) for the set energy of the particle fluence

And I_i is the shielded particle fluence ($\text{cm}^2 \text{ s}^{-1}$)

Considering the single source of monoenergetic photons from a source passing through an object prior to reaching a detection point, the previous formula could be modified to become:

$$\phi_{shielded} = \phi_{unshielded} e^{-(\mu_{den} * d)} \quad (4.8)$$

If the shielding object represents a closed watertight mesh based polygonal geometry in the same 3d Cartesian coordinate space as the source and detector positions, the entry

and exit points on the 3d polygonal surface could be represented as (x_{en}, y_{en}, z_{en}) and (x_{ex}, y_{ex}, z_{ex}) respectively. Therefore the previous formula can be modified to become:

$$\phi_{shielded} = \phi_{unshielded} e^{-(*d_{den}(\sqrt{(x_{en}-x_{ex})^2+(y_{en}-y_{ex})^2+(z_{en}-z_{ex})^2}))} \quad (4.9)$$

Substituting in the previous formula including the location of the source and detector the entire formula becomes¹²:

$$\phi_{shielded} = \frac{E_i P_i A}{4\pi \cdot ((x_{det}-x_s)^2 + (y_{det} - y_s)^2 + (z_{det} - z_s)^2)} e^{-(*den(\sqrt{(x_{en}-x_{ex})^2+(y_{en}-y_{ex})^2+(z_{en}-z_{ex})^2}))} \quad (4.10)$$

If multiple shields are in between the source of photons and the detector the exponential expands and becomes a summative of the contributions for each individual shield with separate d_{den-i} material density and μ_i mass attenuation coefficients with $(x_{en_i}, y_{en_i}, z_{en_i})$ and $(x_{ex_i}, y_{ex_i}, z_{ex_i})$ enter and exit coordinates along the surface of the water tight mesh that defines each shielding object.

As a final step, the particle fluence for each energy can be converted into a more useful form which is representative of the type of measurement that would be made by a handheld radiation detector. To approximate what measurement a handheld gamma sensitive meter would be measuring the final value will be converted into the ambient dose equivalent $H^*(10)$, which is the preferred unit for representing operational measurements[99] and most handheld gamma survey meters are calibrated with a response that is approximately close to this. Figure 20 provides a plot of the conversion from fluence into Gy and Figure 21 provides the plot from Gy into $H^*(10)$ [96].

¹² Removal of sigma element of formula as this applies to a single monoenergetic source of photons

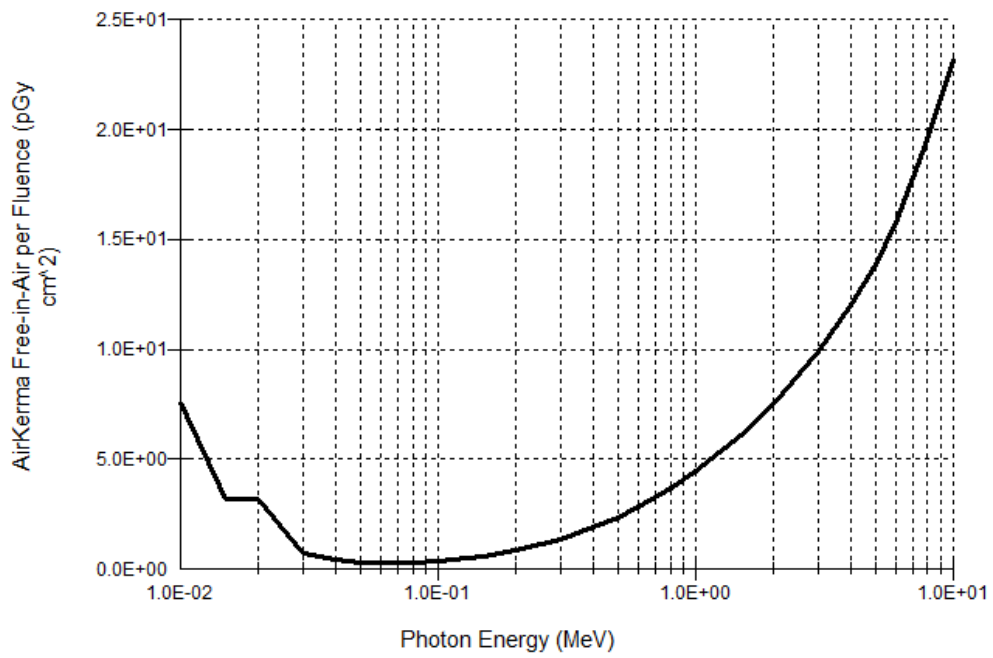


Figure 20 - Conversion for Air Kerma Free in Air per Fluence (pGy cm²)[96]

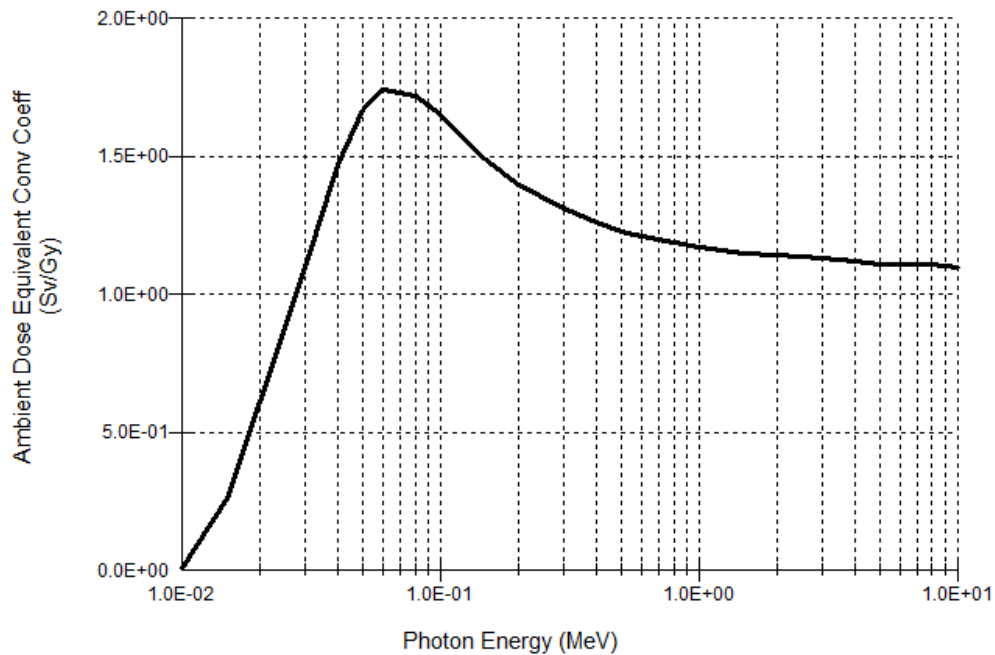


Figure 21 - Conversion for H*(10) from Gy[96]

The tabled values for the dose conversion factors used are provided in an Annex. As the dose conversion factors are provided as tabled values (available in Table 11), for each photon energy the dose conversion factor that is used is determined through a linear interpolation between the two closest values.

4.1.2. Creating the model in Unity – the Unity Point Kernel (UPK) approach

To implement this approach in Unity several scripts were created to handle different aspects of the calculation as well as storage of the relevant radiological and material property information needed. This approach, referred to as the Unity Point Kernel (UPK), is a novel approach to real-time radiation transport optimized for VR. The scripts

are written in the C# programming language¹³, which can manipulate and access properties of the various game objects managed by Unity. The following is a description of the scripts and their setup:

- Any 3d game object which represents a potential shielding object has a script attached to it which performs the following:
 - During the design phase the script allows the user to designate the material from the library of material properties[100] (which are stored locally in a materials property text file). A default density (g/cm^3) is stored in the library for each material which can be manually overwritten by the user. Any object designated as a shield is also configured to have a mesh collider which is required for the raycast physics.
 - When the scene is initially loaded a script will automatically search the game environment and determine what type of radiological material has been included. Based on the radionuclides that have been added to the scene, the script will determine the emitted photon energies, search the database of mass attenuation coefficients values versus photon energy and will linearly interpolate an attenuation coefficient for each photon energy required as well as the relevant dose conversion factors for each energy. This information remains stored on the game object and accessible for other scripts that will require it. Figure 22 provides an

¹³ Documentation of the available data in Unity which can be accessed through C# scripts is available here: <https://docs.unity3d.com/2017.4/Documentation/ScriptReference/>

example of the interface used to configure the script attached to a shielding object.

- Any 3d game object which represents a radiological source has a script attached to it which performs the following:
 - During the design phase the script allows the user to designate the source radionuclide from the library of radionuclides[95] (which includes the associated emitted photons and their related probabilities¹⁴) and the user provides a relevant activity in either Becquerel or Curies.
 - When the scene is initially loaded the script will automatically load into its properties the relevant photon energies and probabilities of emission from the library. This information remains accessible for other scripts which require it. Figure 23 provides an example of the interface used to configure the script attached to a radiation source.
- Any 3d game object which represents a radiation detector will have a script attached to it which performs the following:
 - When the scene is initially loaded this script will search the game environment for each game object with a radiation source script attached and will save that to a list. When a reading is needed to be displayed on the detector, the script will determine the position from the radiation

¹⁴ Photon probabilities <1% are ignored but remain available in the library.

detector to any radiological source(s) using the Unity raycast function¹⁵. If any shield(s) are determined to be in between the radiation source and the radiation detector the script will calculate the penetration distance in the object as it overlaps the raycast between the source and the radiation detector using the coordinates on the shielding object that are hit in the forward and backwards raycast directions (detector to source and source to detector). The script will then calculate for each emitted energy of photon from the radiological source its contribution to the radiation detectors measurement. Finally, the radiation detector will sum up all of the contributions to its measurement and will output the result on the detector screen, which is configured in Unity as a text gameObject. A summary of the raycast process is illustrated in Figure 24.

¹⁵ For comprehensive detail on the use of Unity, raycasting and colliders, it is recommended to read the Unity manual documentation.

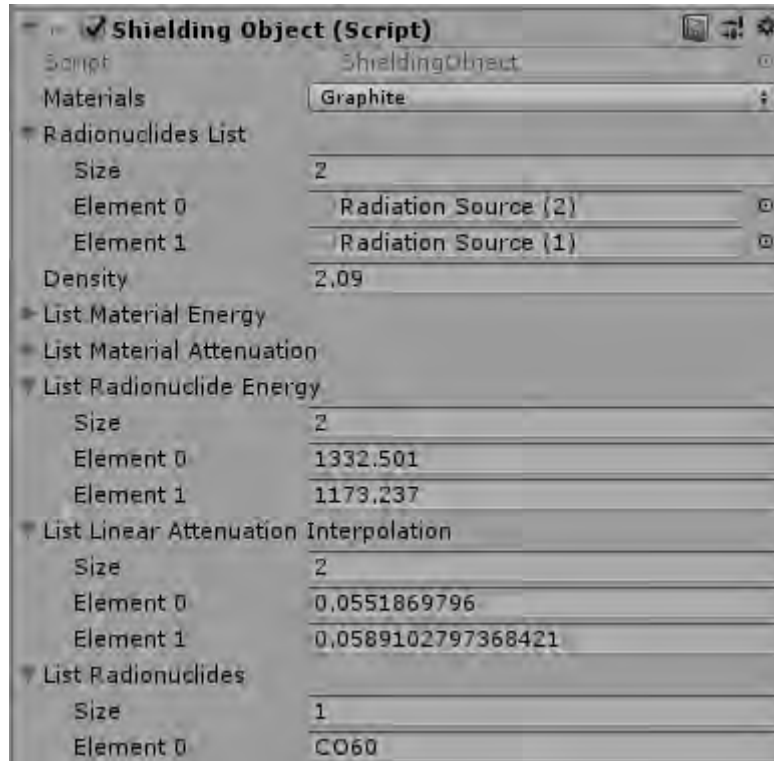


Figure 22 - Example of Shield source script configuration (note the visible attenuation values that have been linearly interpolated for Co-60)

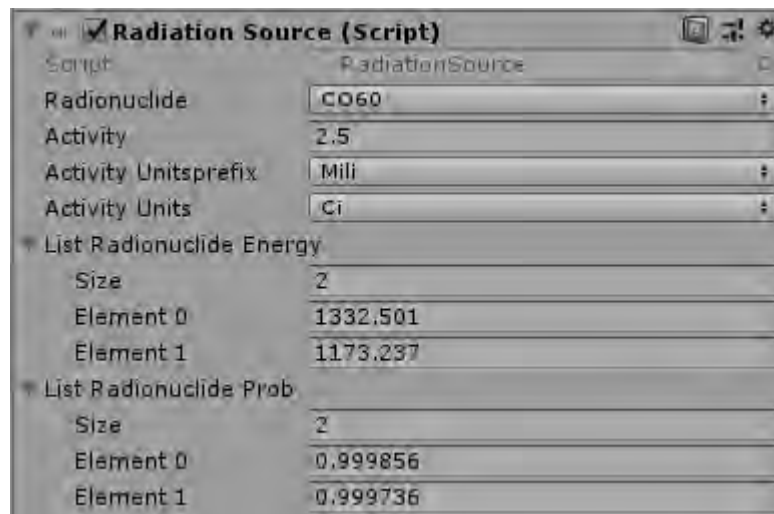


Figure 23 - Example of Radiation source script configuration for Co-60

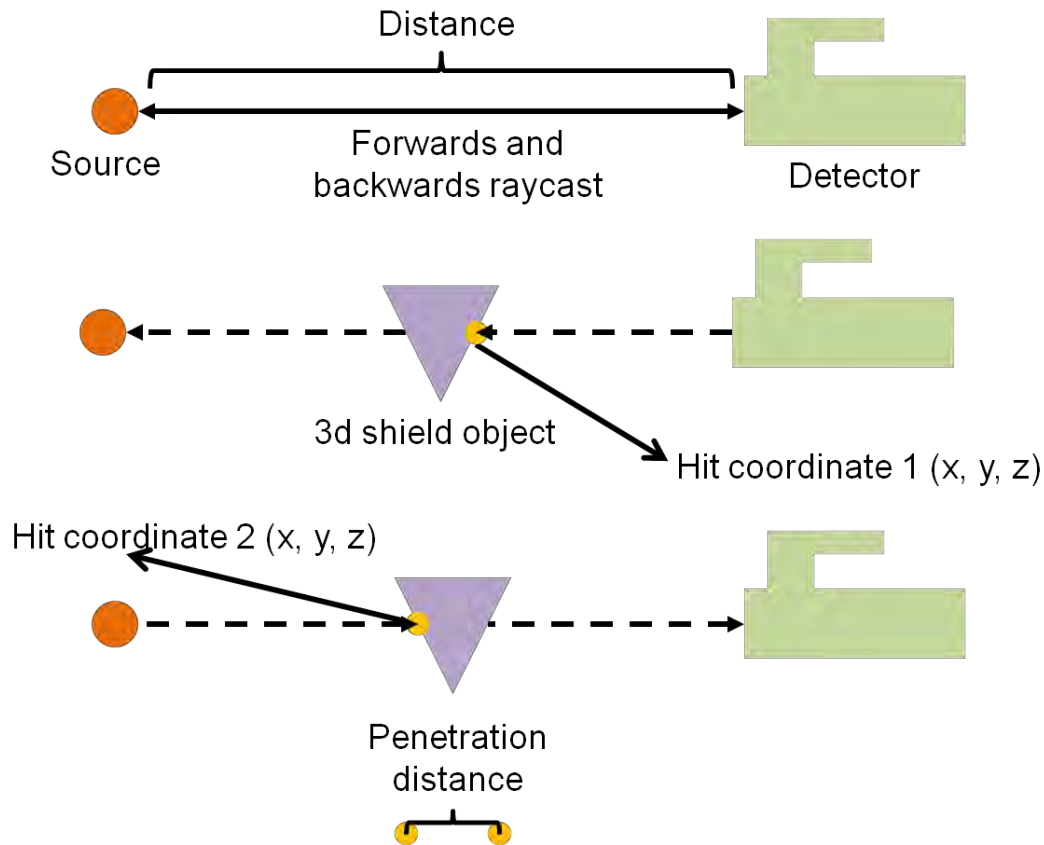


Figure 24 - Illustration of how the unity raycast is used to retrieve the hit points on the shield object needed to calculate the penetration distance needed for taking into account the shielding

4.1.3. Complex source distributions and other UPK applications

This research is focused on scenarios with isolated point sources and environmental shielding. However, as an application of the robustness of UPK, other applications of this approach should be discussed. Defining multiple sources can be used to simulate more complex radiological distributions by approaching the scene creation as a point kernel problem, where a complex source is broken down into many individual smaller sources where the total activity of the smaller sources equals the activity of the larger complex source. Large sources which are identified and measured from close can be

further broken down into smaller voxel kernel units to simulate more complex effects such as self-shielding of the source material, if needed. This is illustrated in Figure 25 and Figure 26. However, if such complex interactions are desired, another approach may be more useful in terms of accurately representing the desired physics.

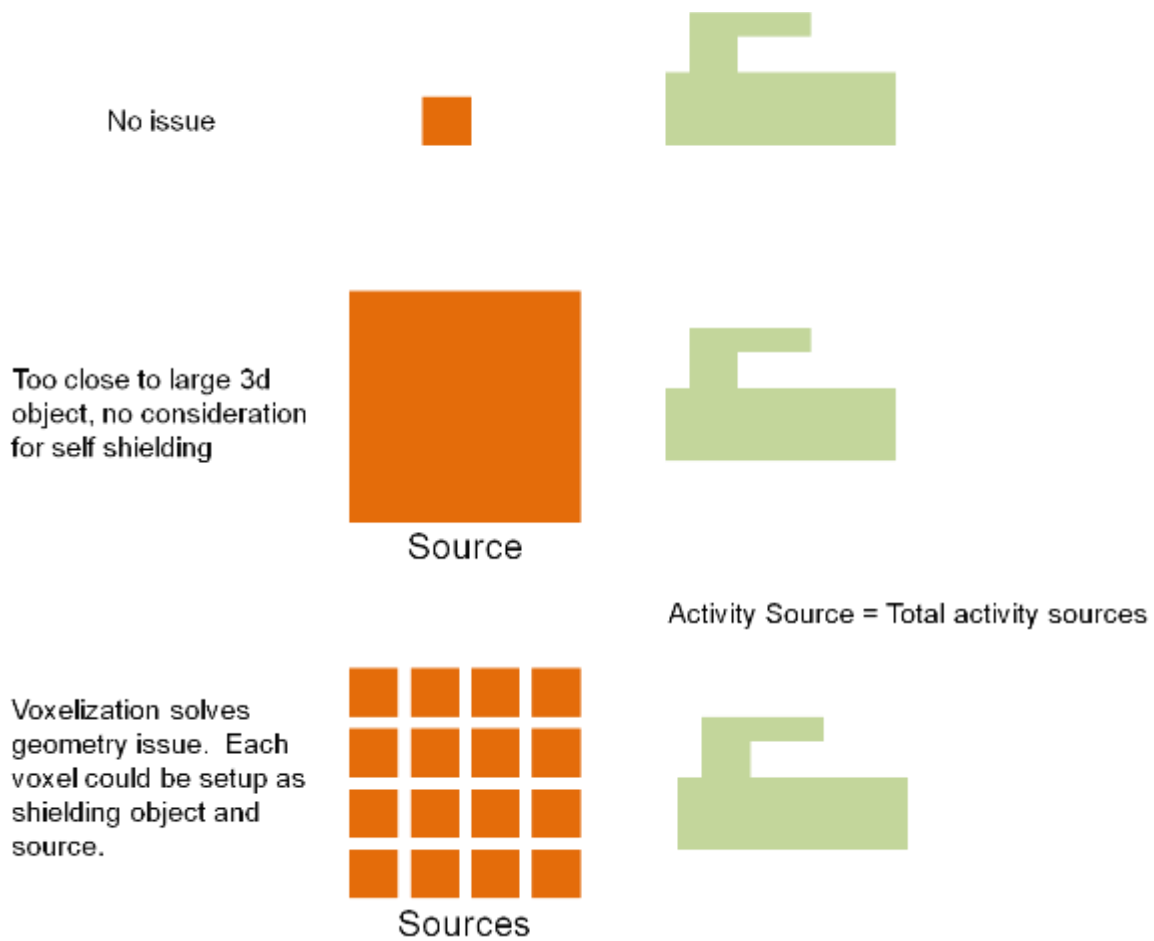


Figure 25 - Example of point kernel approach for measuring larger sources from close distances

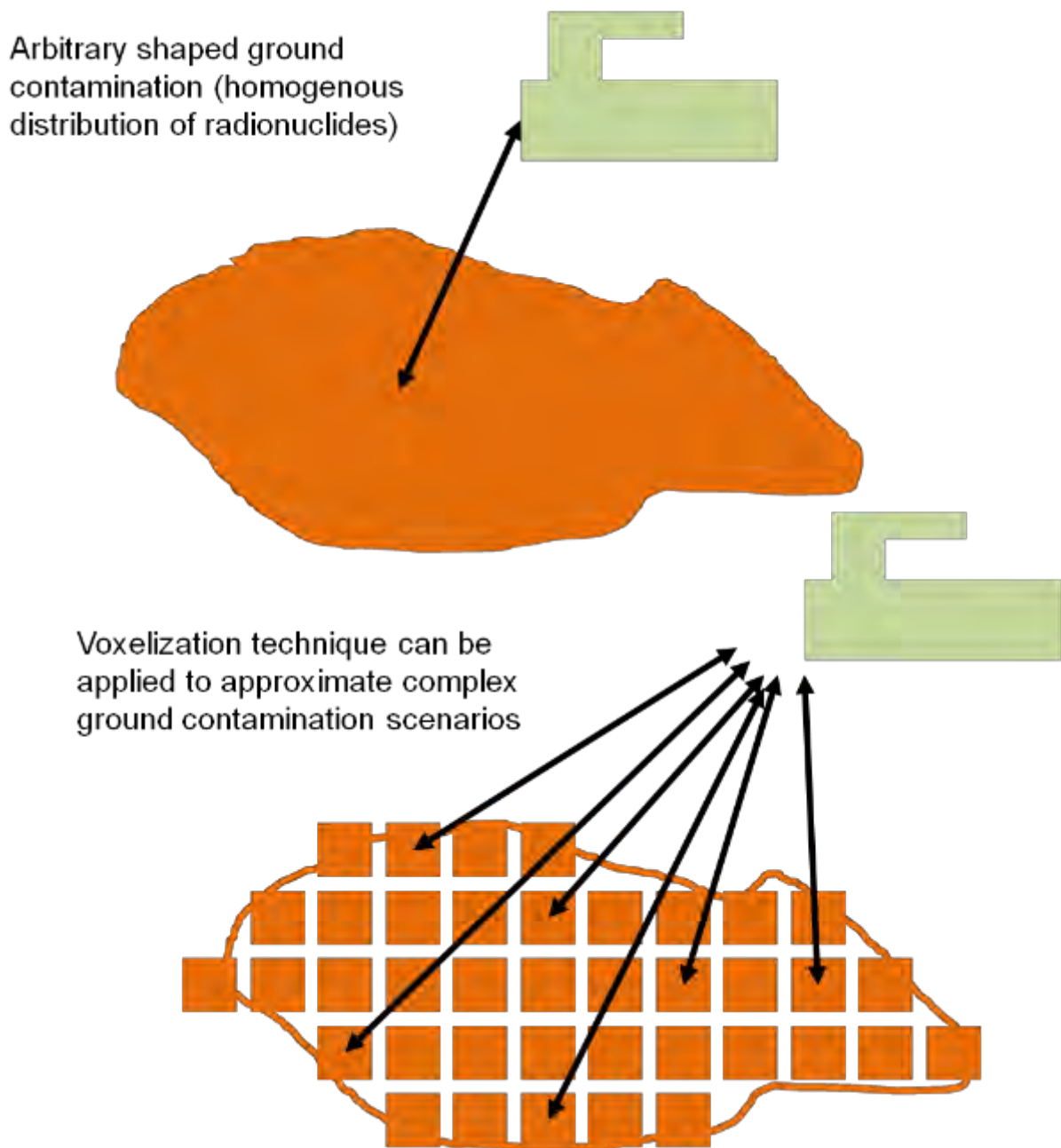


Figure 26 - Example of point kernel approach for an arbitrary ground contamination scenario

There are alternative approaches that are not based in Unity which leverage a similar approach in their specific virtual environments[101][102] but none directly applied to Unity and VR real-time visualizations that would not hinder the user experience.

4.1.4. Validation of the UPK model

A comparison with hand calculations was conducted to validate the UPK model. The validation scene is a detector measuring the dose rate from a radiological source at a fixed distance with and without shielding. In this example, a 1 Ci source of Co-60 source will be used. The source is positioned at location (0, 0, 0) and the detector at (0, 0, 100cm). Co-60 emits two photons of 1332 keV and 1173 keV with probabilities of 0.9998 and 0.9997, respectively. Looking at the data provided in Table 11 and linearly interpolating between the two closest dose conversion factors for fluence to Gy for 1332 and 1173 keV photons provides 5.567 and 5.04 and for Gy to Sv provides 1.156 and 1.163 respectively. Substituting the values into the previously described formula yields:

$$\begin{aligned}
 & \text{Dose (pSv/s)} \\
 & = \frac{1332\text{keV} * 0.9998\% * 5.567 (\text{pGy}\text{cm}^{-2}) * 1.156(\text{Sv/Gy}) * 3.7 * 10^{10}\text{Bq}}{4\pi \cdot ((0 - 0)^2 + (0 - 0)^2 + (100\text{cm} - 0)^2)} \\
 & + \frac{1173\text{keV} * 0.9997\% * 5.03 (\text{pGy}\text{cm}^{-2}) * 1.163(\text{Sv/Gy}) * 3.7 * 10^{10}\text{Bq}}{4\pi \cdot ((0 - 0)^2 + (0 - 0)^2 + (100\text{cm} - 0)^2)}
 \end{aligned} \tag{4.11}$$

$$\text{Dose (pSv/s)} = 3621679 \tag{4.12}$$

When converted into Sv/h, this is approximately a dose rate of 0.013038048 Sv/h or 13 mSv/h. Setting the same scenario in Unity provides the following results in Figure 27 which are in reasonable alignment with a comparison of 13 mSv/h for the hand calculation and 13 mSv/h for the calculation in Unity (note that the text mesh gameObject with the results in scientific notation in the same image is automatically populated by the script):

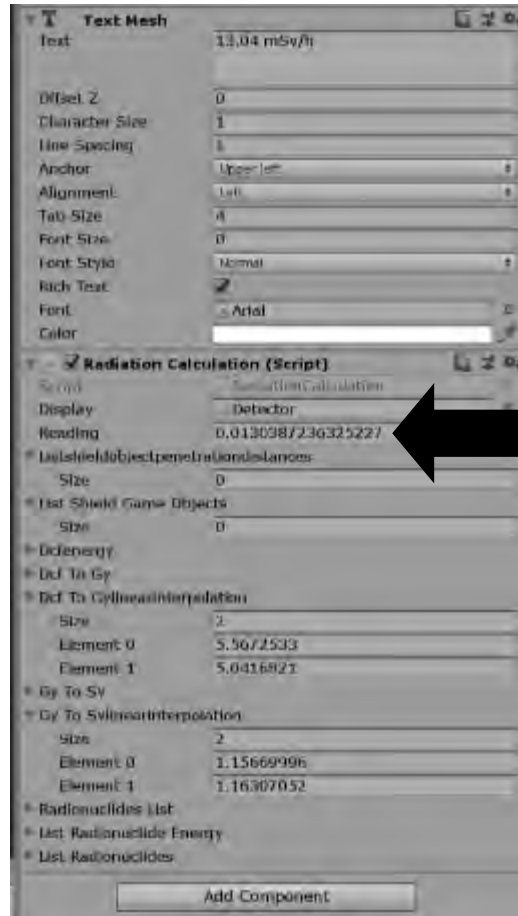


Figure 27 - Results from Unity for dose rate from 1 Ci activity Co-60 source at 1 meter

If a 1 cm lead shield is introduced into the scenario between the detector with coordinates (0, 0, 50 cm) and (0, 0, 51 cm) for entry and exit points along with the corresponding mass attenuation factors of 0.0568 and 0.060 and with a density of 11.34g/cm³, then the solution would then become:

$$\begin{aligned}
 \text{Dose (pSv/s)} = & \frac{1332\text{keV} * 0.9998\% * 5.567 \text{ (pGy} \cdot \text{cm}^{-2}) * 1.156 \text{ (Sv/Gy)} * 3.7 * 10^{10} \text{ Bq}}{4\pi * ((0-0)^2 + (0-0)^2 + (100\text{cm}-0)^2)} e^{-\left(0.0568\text{cm}^2/\text{g} * 11.34\text{g}/\text{cm}^3 \left(\sqrt{(0-0)^2 + (0-0)^2 + (50\text{cm}-51\text{cm})^2}\right)\right)} + \\
 & \frac{1173\text{keV} * 0.9997\% * 5.03 \text{ (pGy} \cdot \text{cm}^{-2}) * 1.163 \text{ (Sv/Gy)} * 3.7 * 10^{10} \text{ Bq}}{4\pi * ((0-0)^2 + (0-0)^2 + (100\text{cm}-0)^2)} e^{-\left(0.060\text{cm}^2/\text{g} * 11.34\text{g}/\text{cm}^3 \left(\sqrt{(0-0)^2 + (0-0)^2 + (50\text{cm}-51\text{cm})^2}\right)\right)} \quad (4.13)
 \end{aligned}$$

$$\text{Dose (pSv/s)} = 1869546 \quad (4.14)$$

When converted into Sv/h, this is approximately a dose rate of 0.0067303 Sv/h or 6.7 mSv/h. Setting the same scenario in Unity provides the following results in Figure 28 which are in reasonable alignment with a comparison of 6.7 mSv/h for the hand calculation and 6.8 mSv/h for the calculation in Unity:

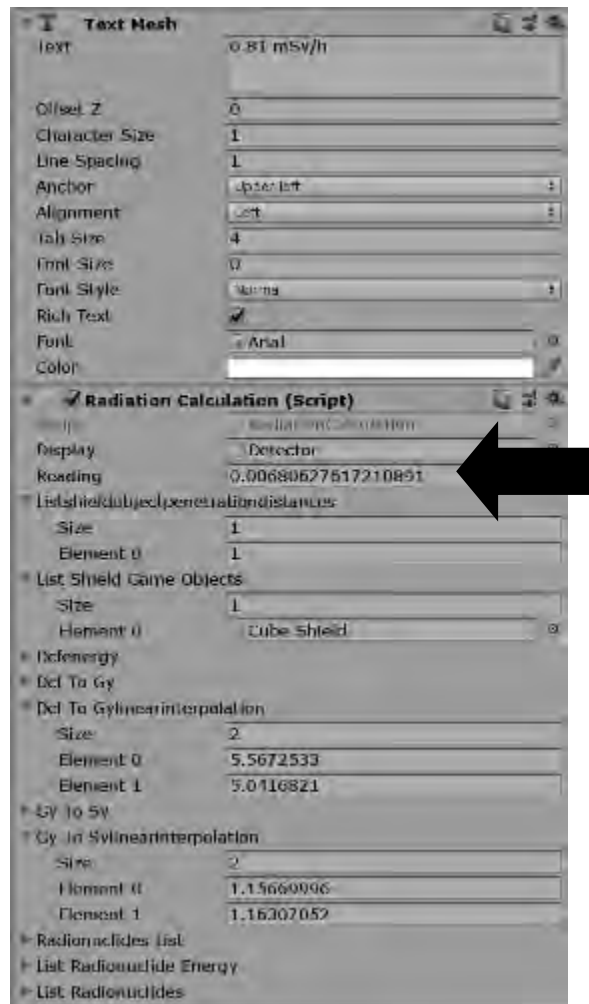


Figure 28 - Results from Unity for dose rate from 1 Ci activity Co-60 source at 1 meter with 1cm of lead shielding between source and detector

4.1.5. UPK Performance

When rendering a single frame for a virtual reality experience the frame rate is an important consideration, as was discussed in Section 3.3.3. The HTC Vive supports a display refresh rate of 90Hz, which means that there are 90 frames per second being delivered to the headset for an optimal experience and therefore the program rendering each frame as 1/90 seconds, or 11.1 milliseconds in between each frame to run all calculations and render the scene. For the UPK to be useable in real-time VR applications it is required that the execution time be less than that. In fact, as there are many other activities that occur when a frame is rendered, it is important for the execution time to be considerably less than that¹⁶. In evaluating the performance of UPK, the Unity Profiler can be used.

¹⁶ There are alternative approaches which could be used to spread the calculation out over several frames or more of display, however in the interest of simplicity, this discussion will be focused on examining the calculation in terms of what can be conducted in the time it takes to render a single frame for the HTC Vive at its native 90 Hz display refresh rate.

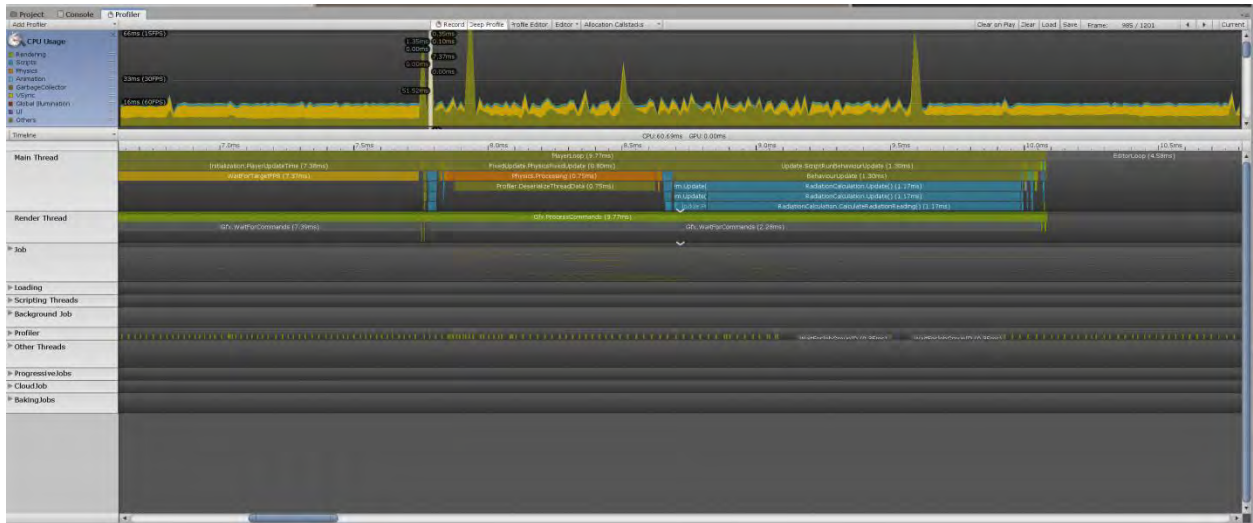


Figure 29 - Screenshot of the Unity profiler being used to investigate performance of UPK

The Unity profiler allows each frame that is made by Unity to be examined and the user to see how long individual processes in the rendering of that frame took to complete. When running Unity from the editor with the profiler enabled there is overhead introduced, so the execution time of a script in a built and compiled program will usually be quicker, but the profiler is useful for ascertaining a representative for how performant the code is given different situations. For this analysis an empty Unity scene is constructed which consists of a single detector and a single source. Additional sources, detectors and shields are adding into the scene so that their impact can be investigated. For the purpose of this investigation the source used in this scene will be a 1 Ci Co-60 source. When considering the performance of UPK, it needs to be realized that for each energy in a specific sources emission there is additional overhead. The results of this investigation are shown in Figure 30.

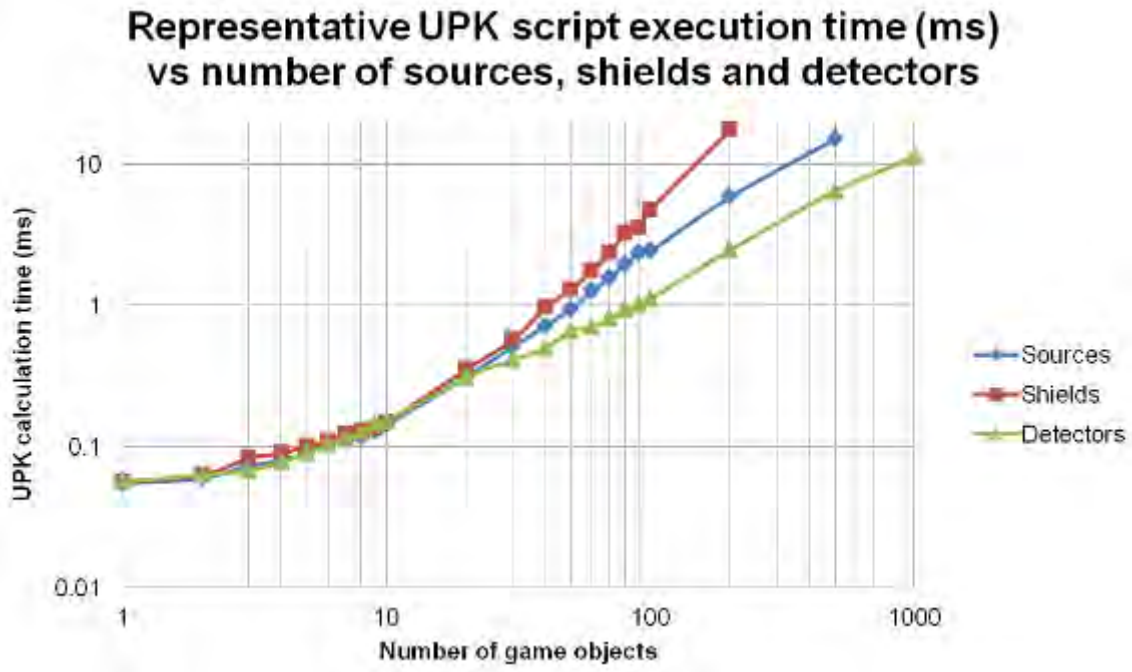


Figure 30 - UPK execution time (ms) vs. number of game objects (sources, shields and detectors)

UPK was designed to front load parts of the calculations and operations wherever possible so that they are performed on start-up rather than multiple times after the scene has loaded. For example, the linear interpolation of the mass attenuation coefficients and dose conversion factors for each radionuclide and shielding material occur only once on start up. Accessing the database of source terms, mass attenuation coefficients and material properties is also only conducted on start-up. The results show the execution time for dozens of sources, shields or detectors remains below 1 ms total time to compute which provides good headroom for the creation of complex source, detector and shielding setups during a virtual reality experience. UPK would need to be further optimized if more complex scenarios were required. .

4.1.6. Alternative approaches

There are alternative approaches to UPK however, UPK represents a balance between technical accuracy and operational efficiency (performance). The radiation field measurement could be determined in advance of runtime, using methods such as Monte Carlo simulations which can be simulated across the entire 3d space and subsequently tabulated for individual areas and displayed back to the end user as the detector moves through each area. An advantage of this approach is that detailed physics could be included such as realistically modelling scattering, secondary particle generation and other features. The disadvantage of such a pre-calculated solution is that objects which dynamically alter the radiation field and allow for interactive shielding, will not be able to be accounted for without a solution to account for this desired capability. This renders this option unrealistic for virtual reality.

Alternative approaches for handling large sets of voxelized data could be used which draw inspiration from how MRI data is often handled[37]. Tutorials exist outlining how these techniques can be implemented in Unity[103] which include the use of 3d textures or the development of a custom graphics shader where colour intensity could be made as a representative for the dose rate.

To reduce the number of calculations required to be performed, an alternative approach could have been to simply use a single dose conversion factor for each radionuclide, rather than photon energy specific factors which require considering all decay energies. It was desirable to maximize the flexibility of this approach so others could benefit other research. For example, during the conduct of this research, UPK was modified for the simulation of x-ray imaging operations for searching for explosive devices[104]. Figure

31 provides an example of UPK applied towards the imaging of a concealed explosive device and Figure 32 provides an example of UPK applied towards the imaging of a detailed mesh model of a human phantom used in detailed dosimetry Monte Carlo simulations[105].

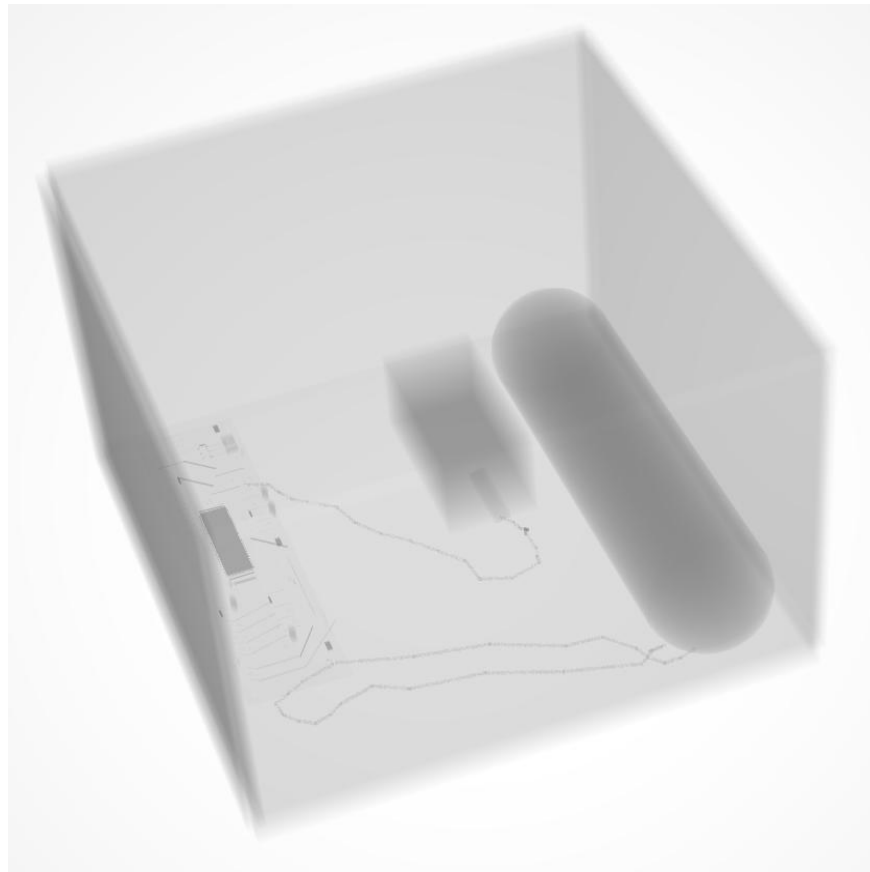


Figure 31 - Example of the use of UPK adapted to generate an approximation of an X-Ray visualization of an explosive device[104]



Figure 32 - Example of the use of UPK adapted to visualize a simulated x-ray of a human body

The purpose of UPK is to provide reasonably realistic radiation calculations and suitable performance in a framework that can be expandable for other uses. The use of buildup factors was not included as part of the base of the model as the scenarios that we

eventually constructed were ones with small amounts of shielding. However it is evident that incorporating buildup factors to improve the results, particularly for the situations where sources and thick shielding is involved is achievable as it would only need to modify the original formula as shown below to include a buildup factor and leverage an appropriate database of factors which are available in literature[106].

$$I_i = I_0 B e^{-(\mu_{den} * d)} \quad (4.15)$$

Where B represents the build up factor for a specific energy of photon.

4.2. Displaying a radiation field in augmented reality

4.2.1. Initial approach

Technology is always rapidly evolving and the methodology and solution developed later in this thesis was made with the best consumer level hardware and software circa 2016-2021. However, in 2010 a preliminary experimentation of an approach to display a radiation field in 3d was made with more primitive hardware and will be briefly described. The purpose of this experiment was to simulate the visualization of a point source in AR using commercially available mobile phone hardware. The technology of the Layer Reality Browser[107]¹⁷ was selected as it allowed a visual overlay to be made with a cell phone camera display and it supported visualizing 3d models that were fixed to a location in the real-world via GPS coordinates and hosted on a web service. The reference viewpoint of the cell phone for the 3d model is then determined using the cell phones internal GPS, compass and 3-axis accelerometer to determine the orientation of

¹⁷ The research and tests conducted and shown were using the Layer Platform 4.0 and 5.0 versions. Accessing the details of the platform as they exist at that time can be made through the Internet Archive Wayback Machine at <https://web.archive.org/web/20101110211124/http://site.layar.com/create/platform-overview/> (as of 22 December 2020).

the model relative to the cell phones camera field of view which can be overlaid in real-time[107].

The hardware used in this initial approach was an early model Android Operating System based phone made available by Google called the Nexus One[108] which had the required hardware and compatible software (Android version 2.1 initially) compatible with the version of the Layer Reality Browser available for developers in late 2010.

As the processing power available was limited, it was decided that the point source visualization would not be made using the geometrically dense voxelization approach but instead would use a 3d contour polygon mesh plot coloured and elevated based on the intensity of the radiation as distance from the source increased. The environment around the point source was considered to be an empty vacuum and the radiation intensity was considered emanating in all directions without scattering or other interactions. The point source decays at rate A, than the radiation quanta intensity I_{int} , in terms of quanta/cm², at any given distance d from the source was calculated as¹⁸:

$$I_{int} = \frac{A}{4\pi d^2}, \text{ where} \quad (4.16)$$

$R = \text{quanta emitted per unit time}$

$d = \text{distance from detector (in m)}$

Using a series of points in Microsoft excel set around a 10 x 10 m area, the radiation intensity for the contour plot was calculated in 10 cm intervals as can be seen in Figure 33.

¹⁸ This is not dissimilar to the previous described model except that this model is not converting quanta into dosimetry units, rather is simply used as a means to produce a 1/r² visualization without accounting for any shielding or dose conversion factors.

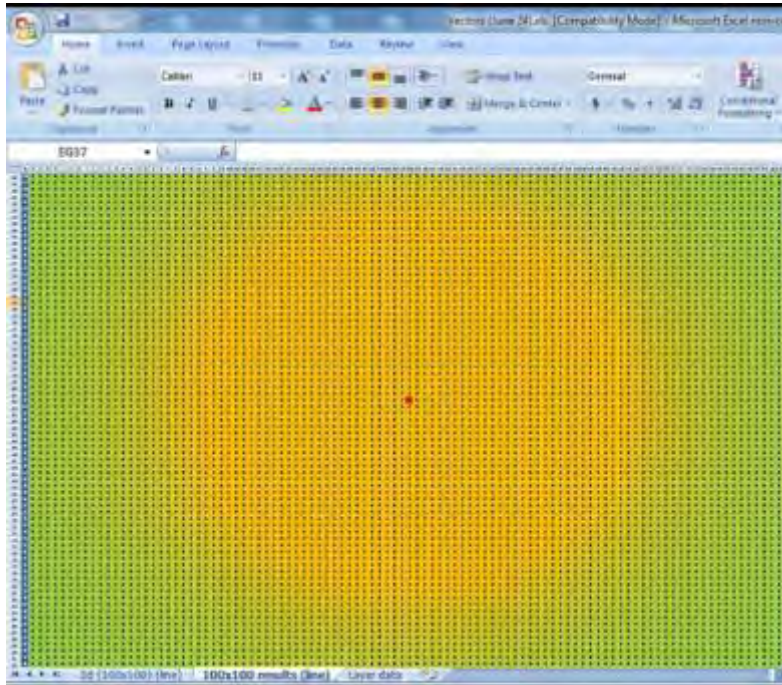


Figure 33 - Visualization of point source intensity data from Microsoft excel (using conditional formatting)

A custom script was prepared for use in Google SketchUp 7[109]¹⁹ written in the Ruby application programming interface (API) which imported a subset of this series of points in 3d with x, y representing the location and the log value of the intensity representing the z coordinate relative to the lowest value normalized versus 0 following methods adapted from what is described in previous referenced documents[23][24]. This can be seen in Figure 34.

¹⁹ Note that in 2010 SketchUp was a product owned by Google. It was subsequently acquired by Trimble Inc., who remain the current owners and distributors of the software.

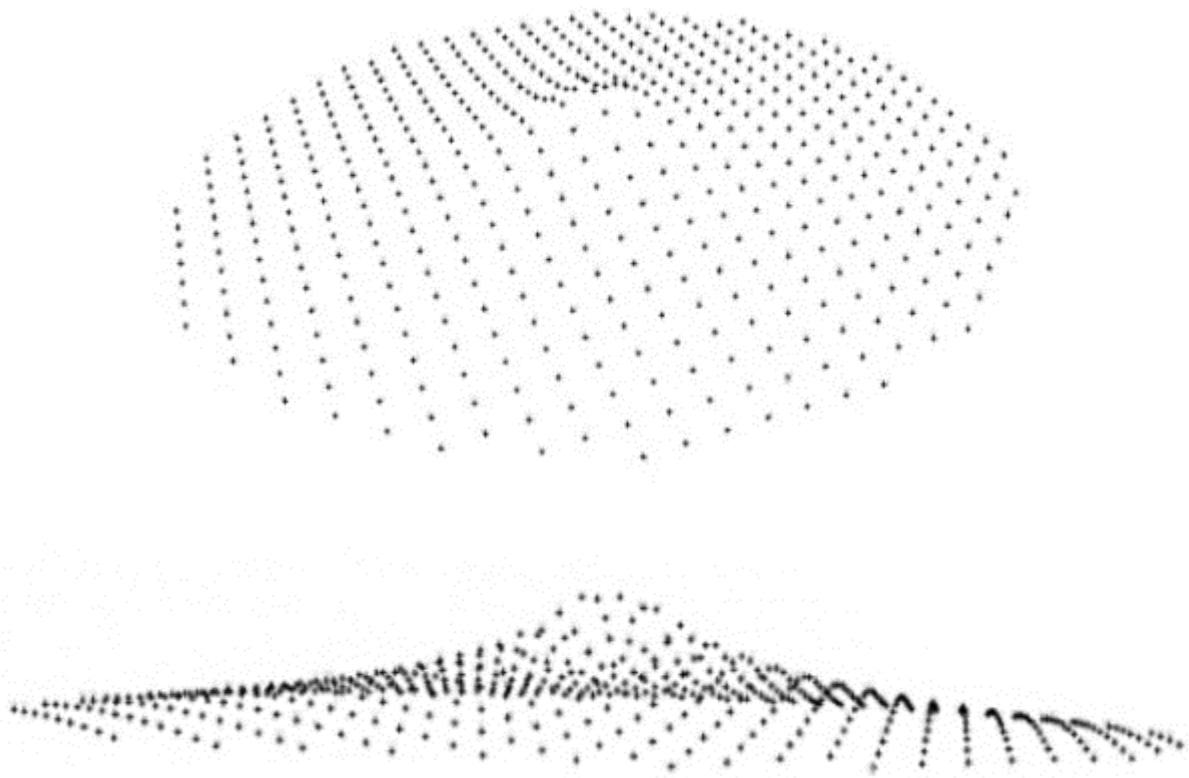


Figure 34 - Visualization of imported point source intensity data in Google SketchUp 7

Due to the low processing power available, the overall number of points imported was reduced so that a minimal polygonal surface could be constructed using these points. A simple colour system was applied to each polygon based on its height (which was relative to its value) and resulted in the model that can be seen in Figure 35.

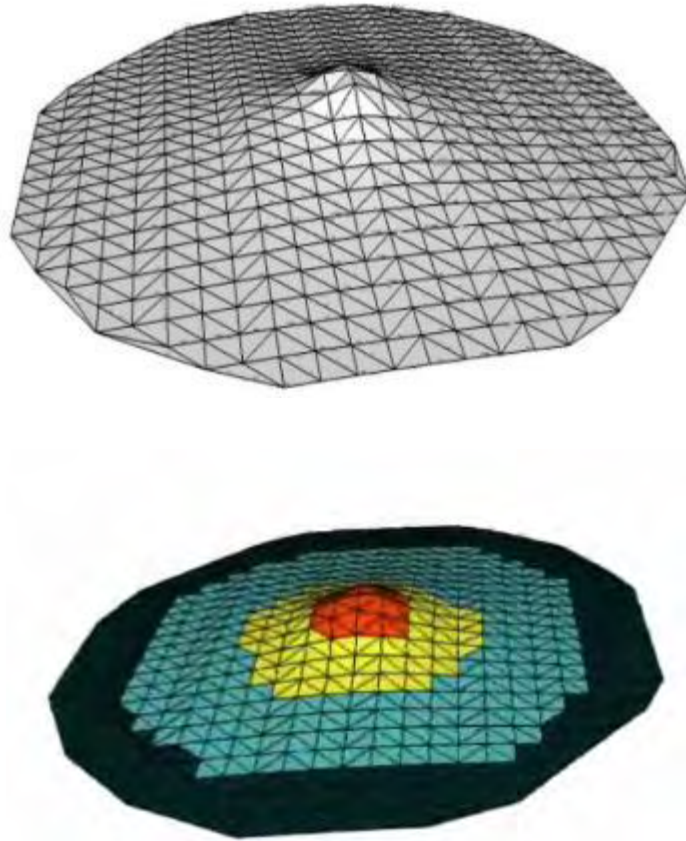


Figure 35 - Colored surface mesh of point source intensity data in Google SketchUp 7

The 3d model was made available on a webhost and a coordinates in the real-world were selected. Using the Layer Reality Brower the radiation field visualization was approached through the Nexus One cellular phone. The resultant visualization as seen from a distance of approximately 20 m and while approaching can be seen in Figure 36 and Figure 37.



Figure 36 - Real-time visualization of point source 3d mesh in AR interface via Layer Reality Browser (seen from distance)



Figure 37 - Real-time visualization of point source 3d mesh in AR interface via Layer Reality Browser (seen as approaching)

This approach was expanded upon to see how a more complicated model with a mixture of 5 radiation sources of unequal intensity would appear if they were randomly distributed within the same environment. The model was generated in the same method as before except the contribution from each source was summed into a total radiation field model that is visualized in Figure 38.

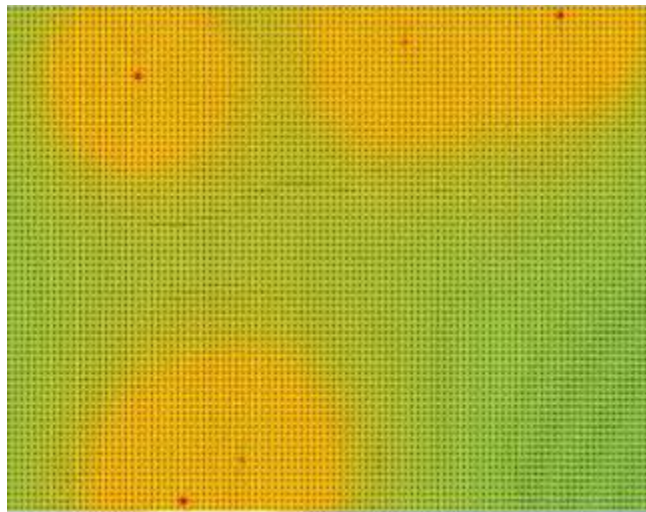


Figure 38 - Visualization of several point sources showing intensity data from Microsoft excel (using conditional formatting)

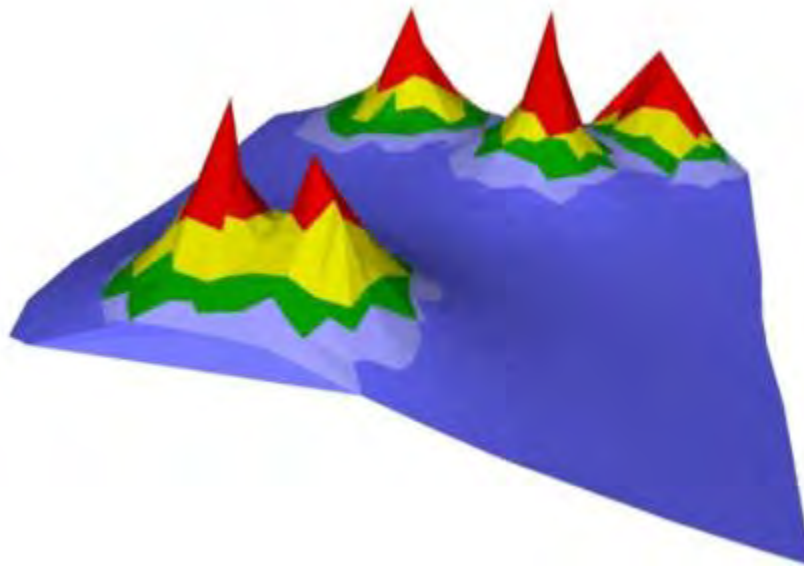


Figure 39 - Colored surface mesh of several point sources intensity data in SketchUp 7



Figure 40 - Real-time visualization of several point sources intensity 3d mesh in AR interface via Layer Reality Browser (seen as approaching)

This approach had a number of technical and performance limitations. The visualization technique relied on real-world GPS coordinates, which in turn relied on the accuracy of

the cell phones GPS receiver. This resulted in a visualization that was inconsistent in its 'placement' and appeared 'jumpy' when viewed through the cell phone. The limited graphical processing power of the cell phone required that the visualized model not have more advanced graphical effects, such as transparency applied, which further created a challenging to comprehend scenario when the radiation field was approach. Although this effort was considered novel and interesting at the time and served as inspiration for the methodology and approach described herein, due to the limitations in the hardware and software environments this specific AR approach was not pursued beyond what has been discussed.

4.2.2. Advanced approach for visualization of radiation measurement information (Google Project Tango)

While the previous approach was not pursued beyond the initial exploration and experimentation efforts, a new approach, in part inspired by examples of real-time 3d reconstruction/scanning[110][111] and the potential for data fusion between 3d reconstruction/scanning environmental information and radiological monitoring data was investigated for applicability. For the 3d reconstruction and scanning, the Google Project Tango (GPT) Tablet[112] was made available to the developer community in 2014²⁰ running the Android operating system. The system utilized information from an inertial measurement unit, fisheye camera and depth camera combined with onboard processing to track the 3d position of the tablet as it moves around an environment and is capable of creating a polygon mesh of the environment at the same time with

²⁰ The system was acquired in mid-2015

included code examples in its software development kit. Figure 41 shows a system diagram of how the software and hardware environment of the tablet is oriented and an in-depth discussion is available in literature[110].

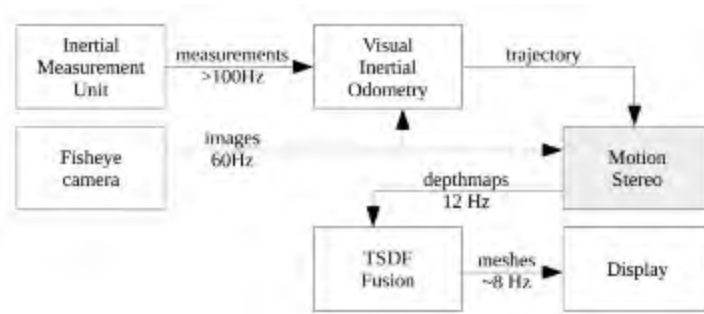


Figure 41 - System diagram for the Google Tango Tablet[110]

Included with the GPT Tablet was a developer tool called the ‘Constructor Developer Tool’ which provided an example of an Android software tool that would perform real-time meshing of the environment and could construct reasonably understandable maps of an environment. Figure 42, Figure 43, Figure 44 and Figure 45 depict some interior screen captures of an apartment building, including a stairwell showing the meshing process in progress (Figure 42 and Figure 43) and the quality of the resultant 3d model post scan (Figure 44 and Figure 45) visualized in the computer software MeshLab[113][114].

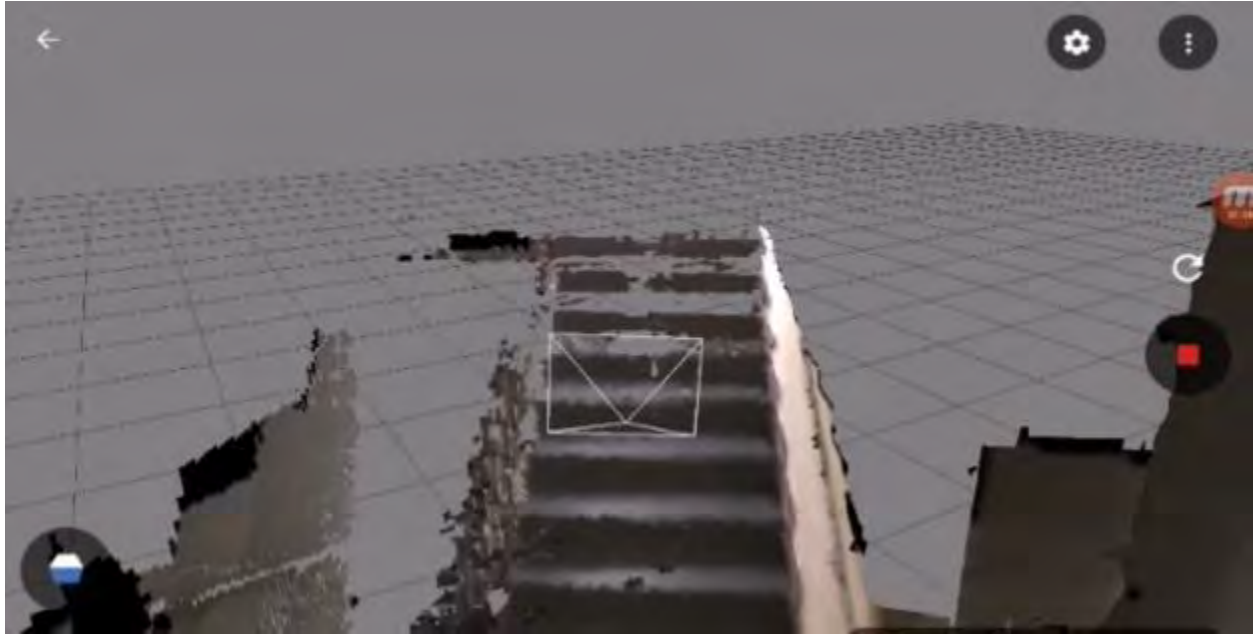


Figure 42 - Preliminary environmental scan taken with included Tango constructor application (approaching staircase)

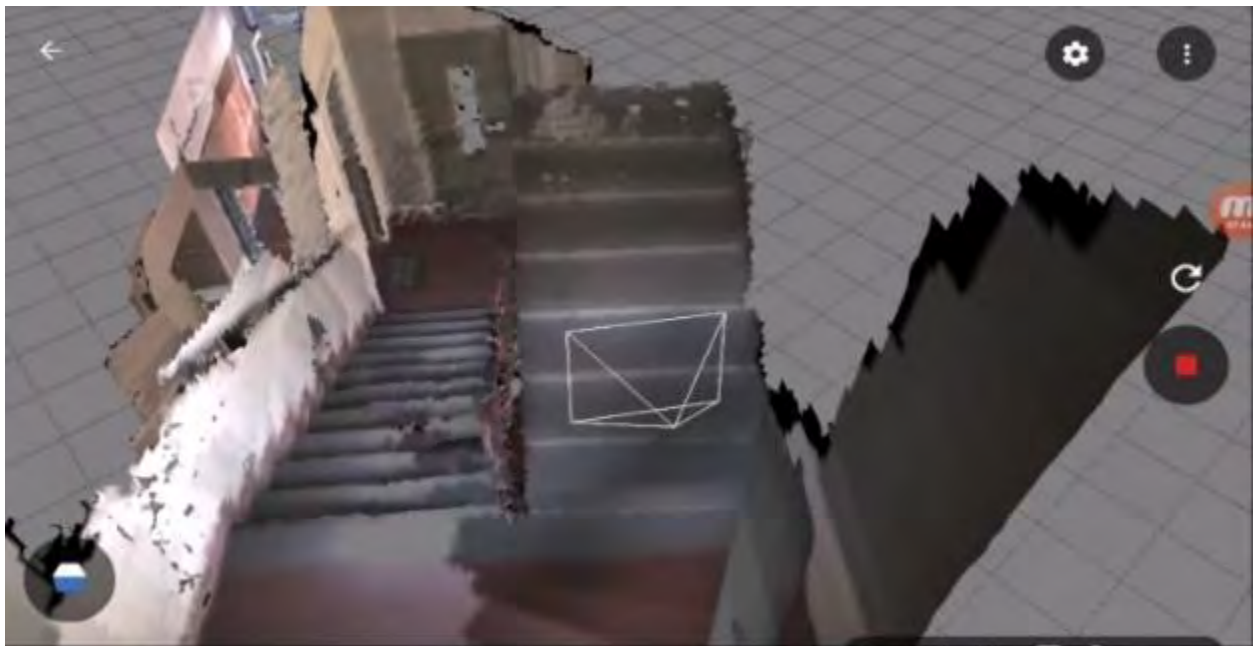


Figure 43 - Preliminary environmental scan taken with included Tango constructor application (after traversing staircase)

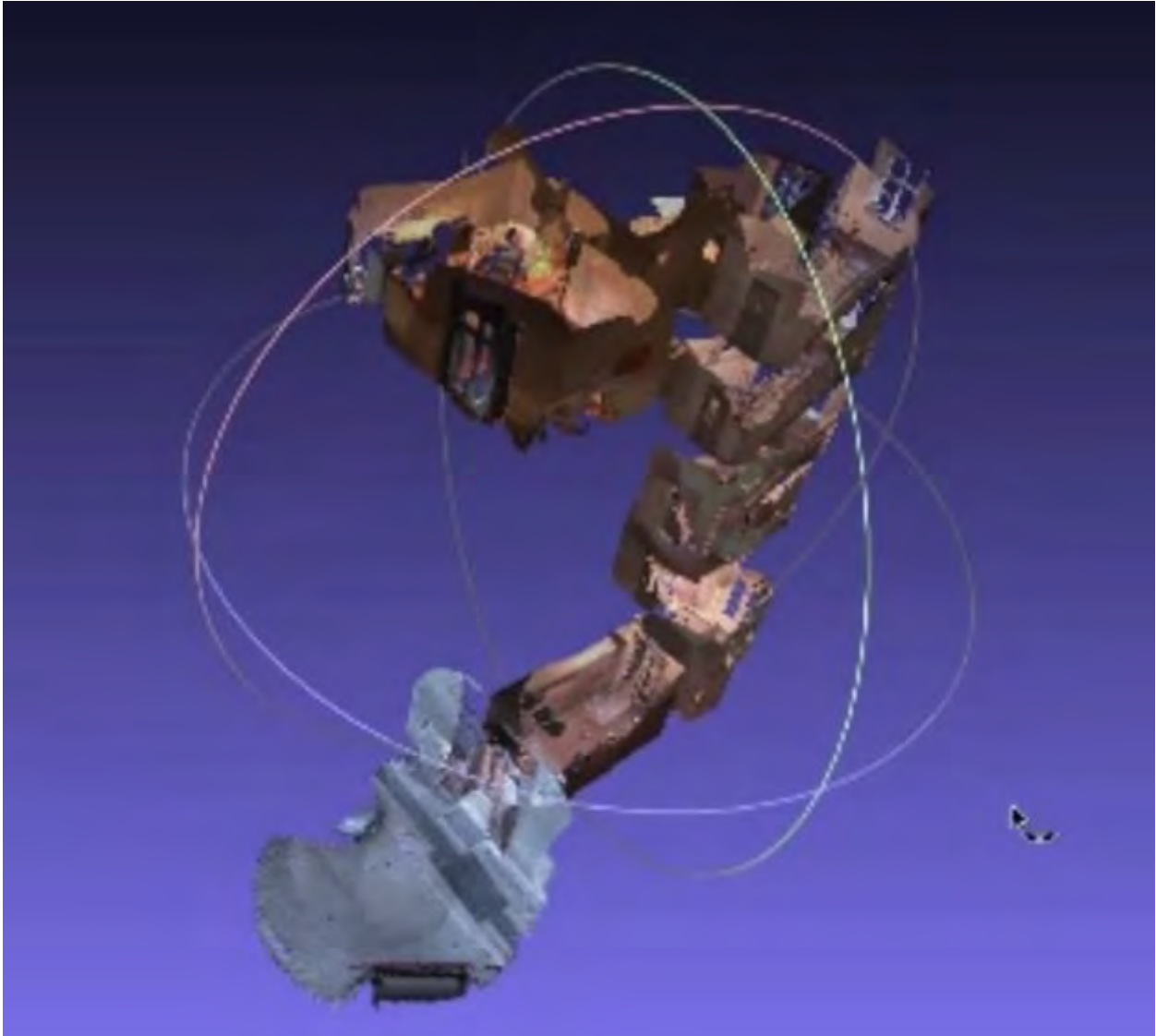


Figure 44 - Preliminary environmental scan taken with included Tango constructor application (showing several stories and the inside of an apartment)



Figure 45 - Preliminary environmental scan taken with included Tango constructor application (highlighting detail captured in close range)

The GPT Tablet was compatible with Unity development for Android applications and it included a software development kit with a set of software APIs for use by the development community which allowed access to the capabilities of the tablet including the augmented reality real-time tracking as well as the 3d environment meshing. Using the development environment and the UPK model, a software environment was constructed which featured the real-time meshing of the environment combined with the capability for a user to place radiation sources of a fixed activity into the 3d mesh (therefore giving them a fixed point in the environment). Once several sources were placed in the environment the user was then able to begin both mapping the environment as well as receiving a real-time radiation reading which was the summation of the contributions from all of the sources distributed into the environment. As a

measurement was made by the tablet, a voxel of 25cm x 25cm x 25 cm, fixed in 3d space in the environment, was positioned and colored based on the radiation reading in that voxel and a default scale. As additional readings were acquired the user was able to 'rescale' the existing measurements such that the highest measurements were red progressing down to orange, yellow, green and eventually blue. The results of this experiment in 2016 were extremely positive and can be seen in Figure 46, Figure 47 and Figure 48 with the combined environmental meshing giving the user very good perspective on the location of the source(s), utilizing the best available cameras, tablets, and meshing available at that time.

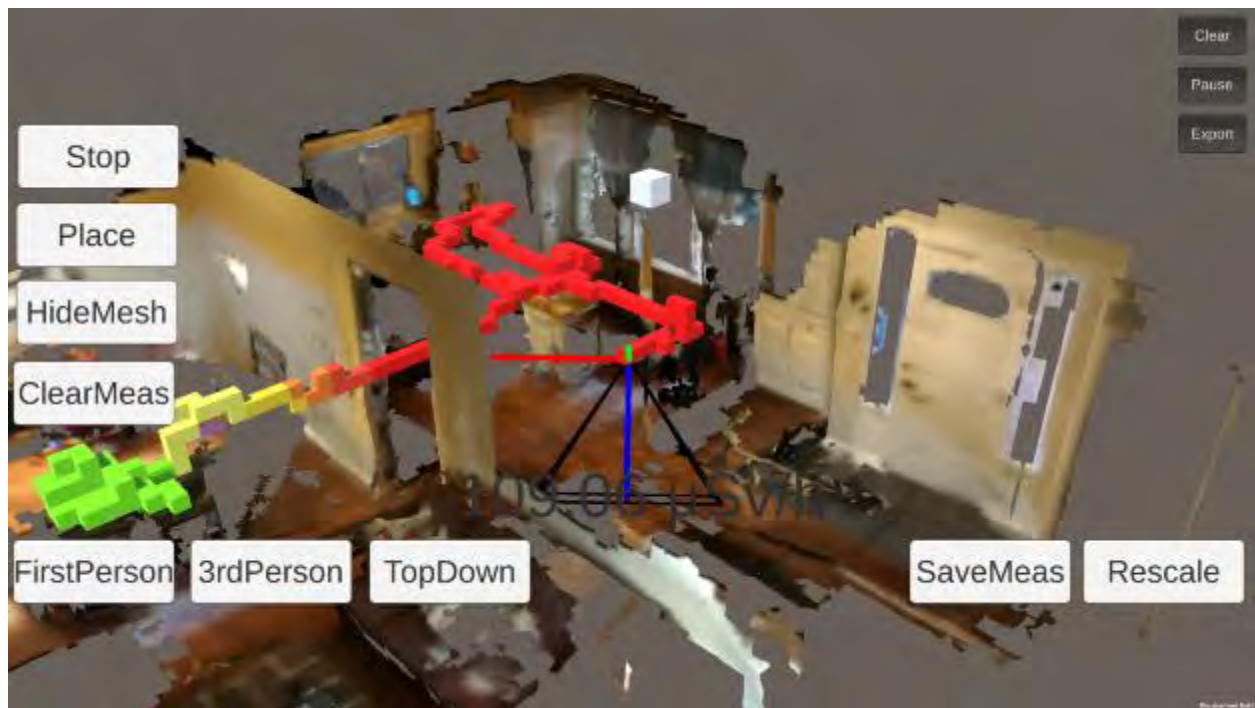


Figure 46 - Preliminary radiation visualization results (1)

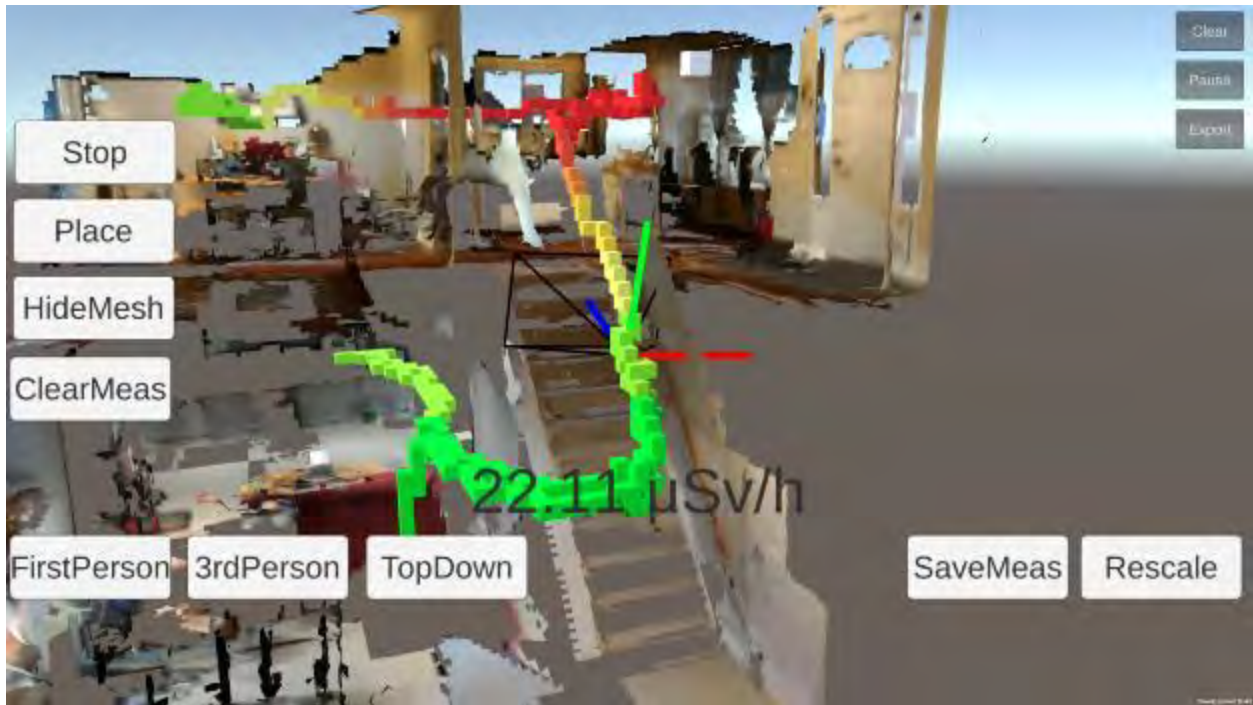


Figure 47 - Preliminary radiation visualization results (2)



Figure 48 - Preliminary radiation visualization results (3)

When evaluating the previous figures, it is important to consider that for the simulation being performed in the GPT Tablet, the radiation field contribution to the tablet is calculated each updated frame in the simulation. The density of the trail of voxelized measurements is therefore significantly greater than if the measurements were taken with a radiation detector with an integration time of several seconds that would result in much sparser trail with space between the measurements, unless the radiation survey was moving very slowly through the environment. Figure 49 provides an example of a series of measurements taken where the environmental meshing was hidden from the user. It can be seen that understanding the relative location of the measurements as they related to key features in the environment are more challenging to understand in this case and this provided further evidence regarding the important of environmental meshing in giving the user context for the location of radiation measurements.



Figure 49 - Preliminary radiation visualization results without environmental meshing

Alternative technologies have become available in recent years which also have the capabilities similar to the Google Project Tango Tablet, such as the Microsoft HoloLens. These alternatives were not widely available at the time this initial research was conducted. However, there are literature discussions of the differences between the systems[115] and the methodology used in later chapters with the Google Tango Tablet would be applicable with similar tracking technology.

4.2.3. Advanced approach for visualization of radiation measurement information (Unity AR Foundations and Google ARCore)

After the experiments with the previous approach to visualize synthetic radiation measurements utilizing the Google Tango Tablet (and two commercially available

hardware devices), two new technology alternatives became available: Unity AR Foundations[116] and Google AR Core[117]²¹. The benefit of transferring this approach to these new systems was that the number of devices this approach would be able to run on would be much higher[118] than just the Google Project Tango Tablet and the two commercially available cell phones that were released featuring that technology. AR experiences utilizing this technology are still able to be tracked in the same manner as the previous approach but with the absence of the depth sensor there is a bit of a loss in the fine detail related to a scene depth that can be determined (however it is entirely possible to develop a sense of depth from a scene without a dedicated depth sensor as has been shown with the ARCore Depth Lab samples[119]). For the demonstration of the real-time visualization of radiation fields that follows, the AR environment aspects have been built in AR Foundation based on the original approach used with the Google Tango Tablet. The code base remains mostly the same and the included axis and visualization of the view space (black bars from the camera in the third and top-down views) were ported over from the Google Project Tango development SDK to keep the visuals consistent between the two.

4.3. Development of a real-time capability to display radiation measurement information in augmented reality

To demonstrate that this approach would work with a field deployable radiation detector, it will be needed to demonstrate that the environment constructed in Unity for the simulated augmented radiation field visualization development would also work suitably

²¹ AR Core originally started as a stand along package not requiring AR Foundations[139]. This research and the code was originally ported to that package. When AR Foundation was made available with the AR Core this work was again ported to AR Foundations to remain relevant.

when provided live data from a radiation detector. A radiation detector which provides data in a format that can be inserted into Unity is needed to be selected. For this purpose, an Abacus Radiation Detector produced by S.E. International[120] was selected and is pictured in Figure 50, Figure 51, and Figure 52 and the specifications are in Table 1.



Figure 50 - Side view of the Abacus Radiation Detector



Figure 51 - Top view of the Abacus Radiation Detector



Figure 52 - Bottom view of the Abacus Radiation Detector

Table 1 - Specifications of the Abacus Radiation Detector[121]²²

Detector: Halogen-quenched Geiger-Mueller tube. Effective diameter 1.75" (45 mm).

Mica window density: 1.5 - 2.0 mg/cm².

Operating range: CPS - 0 to 12,500 cps.

Gamma Sensitivity: ~7.5 CPS/mR/hr referenced to Co60. Smallest detectable level for I-125 is .02 µCi at contact.

Averaging Periods: Display updates every second.

Power: 2 AA Batteries.

The radiation detector supports communication via serial communication protocol over Bluetooth connection. Using an asset purchased from the Unity Asset Store[122] which facilitates Bluetooth to serial connections, it becomes possible to connect to the radiation detector and receive data from it. Annex III provides excerpts from the C# script that is used in Unity to query and receive data²³ from the radiation detector via Bluetooth once connected²⁴.

Using this code provided a plain text value of counts per second which could then be parsed and used to display coloured voxel data in the same method used for the simulated data keeping its measurement position fixed in space while the camera freely navigated following the motion of the user.

²² Full technical specifications available in the manual

²³ Support for this activity was provided by Callan Brown, University of Ontario Institute of Technology

²⁴ The C# code that was purchased Unity asset has not been included, as it is proprietary code from a third party

4.4. Development of a virtual reality environment for the realistic training of radiation detection concepts

In order to demonstrate the potential for applying training for first responders (and others) in virtual reality, a scenario was constructed in Unity to demonstrate how UPK can be applied in a virtual reality environment with Unity game objects. The goal of this scenario was to create an environment with the following features:

- Ability for a user to measure a gamma radiation field consisting of contributions from 1 to n number of radiation sources with a handheld radiation detector.
- Ability for user to select any combination of predetermined radionuclides for each radiological source and provide them with an associated activity that corresponds with list of photon energies and probability of emission per decay.
- Ability for the user to position and move the simulated radiation detector to any position within the virtual environment.
- Ability of the simulated radiation measurement to account for shielding of the emitted photons taking into account the properties of different shields, its position within the environment and the energies of the photons emitted from the radiological source(s).

Recognizing that many of the participants who were going to be asked to try this new learning method would be entering virtual reality for the first time, a suitable scenario was designed to teach them how to move about and interact with objects including the virtual radiation meter. This scenario had several key features which included:

- Table at the start with blocks which they could interact and otherwise grab and throw to learn and understand the physics of the simulation.
- Series of larger and larger pyramids of blocks to teach users how objects can be scaled larger and larger for real-life perspectives.
- A training table with:
 - Three radiation sources
 - Radiation detector
 - Ten radiation shielding bricks

The tutorial scene was intended for an exercise controller to supervise. Their purpose was to both guide the player through how to operate and use the locomotion method in VR and also to judge how comfortable the participant was in the environment. In accordance with the Systematic Approach to Training described earlier, if the user was not comfortable in the environment, the exercise controller would then run future exercises that were to follow in a simplistic manner, guiding them to look at objects and explain how they would react, to maximize learning objectives. If the participants were comfortable in the VR environments, the exercise controller would (in the scenarios that followed), have the participants engage in more interactive and otherwise complex interactions with the environment. How future exercises would be conducted, was based on the judgement of the exercise controller after observing the students performance in the tutorial.

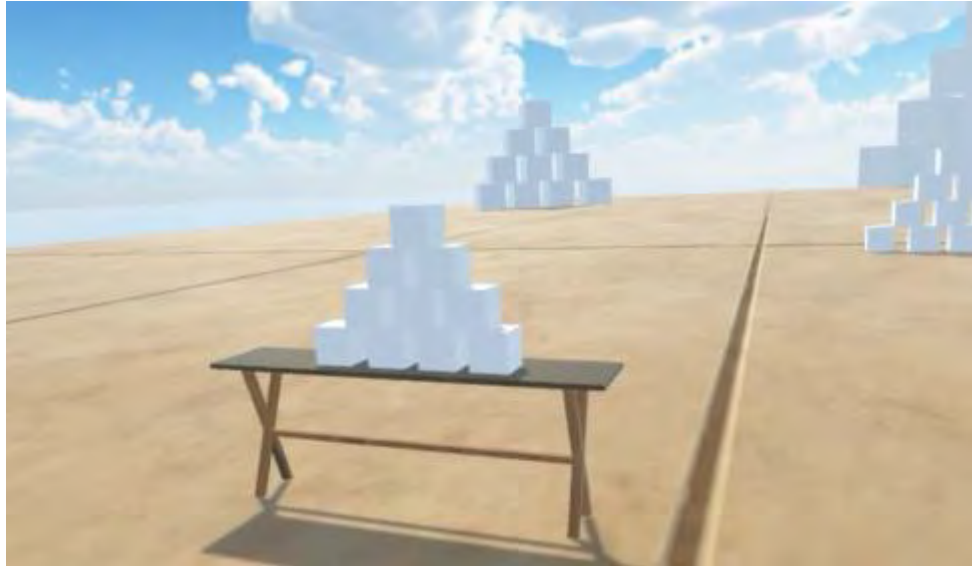


Figure 53 - Screenshot of tutorial scene starting point

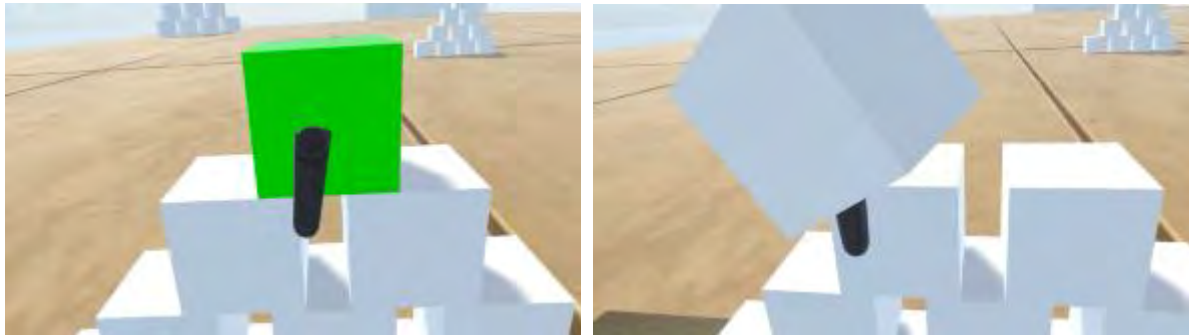


Figure 54 - Screenshot of user learning basic interactions at starting table in tutorial scene



Figure 55 - Screenshot of user approaching radiation demonstration table in tutorial scene



Figure 56 - Screenshot of user holding radiation detector and learning to perform a measurement

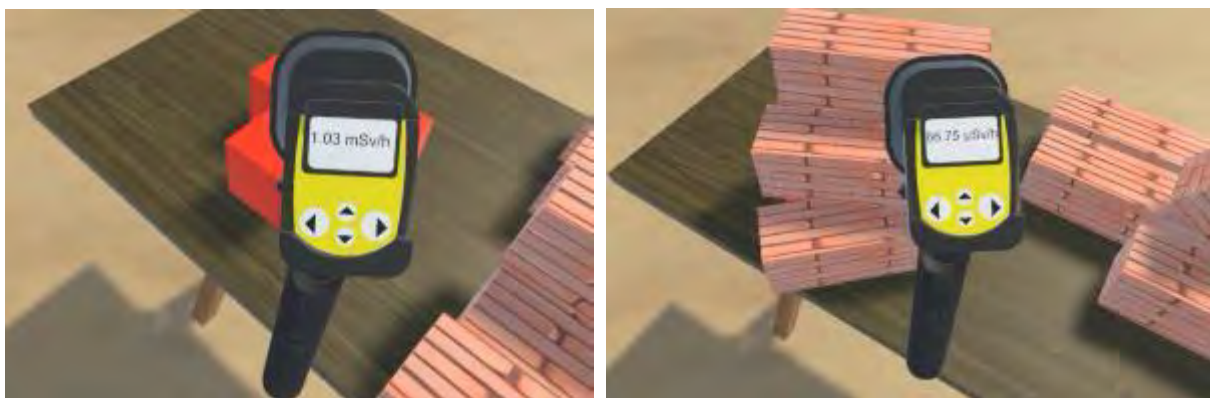


Figure 57 - Screenshot of user measuring the training sources and successfully building a small castle around the sources to see the shielding effect

The tutorial scene demonstrates an environment where a user can become familiar with navigation in VR as well as the basics for interacting with several sources of radiation,

several shields and a radiation detector. The simple elements of movement and interactive measurement will form the basis of scenarios described later in this research.

CHAPTER 5: REINFORCEMENT LEARNING APPROACH TO RADIATION SOURCE LOCALIZATION SUPPORT

In this chapter an approach leveraging the latest modern machine learning methods is developed and applied showing the potential for localizing a source of radiation leveraging the data that is obtained when conducting an augmented reality visualization of radiation field measurements.

5.1. Reinforcement learning

Artificial intelligence (AI) is a long-standing research branch in computer science which crosses into applications in many different fields from videogames to business intelligence to stock trading to general problem solving. Machine learning (ML) is a branch of AI research where patterns are deduced from sets of data without explicitly programming in the basis of the patterns[123]. Reinforcement learning is an approach within ML leveraging methods where desired results are encouraged and undesired results are discouraged over time.

As a comparison, infants learn to crawl learn in different ways. They can be shown a solution by observing their parents moving in a manner which they then try and imitate to the cheers and praise from their parents. They could be motivated by having an interesting toy, placed slightly out of reach that encourages them to try and learn how to manipulate their body to retrieve it and then they are happy when they finally figure out a way to reach their goal. In both situations the infant is making observations of the environment, determining a course of action for its next move and either receiving praise from its parents or from itself when it succeeds in achieving its goal or getting

frustrated when it fails to achieve its goal and trying a different approach leveraging what it learned in its previous failures. This is simplistic description of reinforcement learning capturing the high-level essence of the concept.

Reinforcement learning is useful for developing models that are able to solve complex problems. The research developed a methodology that takes input data and returns a decision based on that input data using reinforcement learning methods. This method was developed through training, where a large number of instances of input data are provided and the output decision is made, which is compared with a desired output and either rewarded and positively propagated or punished and negatively impacted.

5.2. Unity ML Agents

A Unity package called the Unity ML-Agents Toolkit is available[124] to support the creation of intelligent agents for videogames and other purposes. This package provides a general toolkit for creating, training and running ML agents in Unity[125] which leverages the open source library PyTorch[126]. The toolkit provides access to their state-of-the-art ML training algorithms through Unity which is built on top of PyTorch[125].

A few terms need to be described in the context of the application of reinforcement learning described in this chapter.

- Environment
 - This is the surroundings of an area which include floor, ground, walls and, of particular interest in this research, the calculated radiation field contribution from a radiation source that can be located anywhere (within

a user defined restricted volume) in that environment. In discrete steps of time (t) the environment exists in a given state (S_t).

- Agent
 - The agent can be thought of as the algorithm trying to solve a given problem after having been provided some observations from the environment. During training when an agent decides an action (A_t) to take it is determining how to modify the environment which will then require feedback to determine if its action was desirable. Based on the agent's chosen action the environment is subsequently modified (S_{t+1}) and observation and the decision cycle can continue.

- Observations
 - This is data provided to the agent regarding the current environment (often called sensors). Determining the observations that are needed to be provided to the agent is akin to determining what key information would be needed to solve a problem deterministically. However, it is not necessary to limit the observations in such a manner - providing additional observations, when it is easy to do so, can help a problem be solved more reliably.

- Policy
 - This is the 'decision' making element of the reinforcement learning process and it governs the behaviour of the agent. A policy is linked to a neural network of a user definable number of layers and neurons that is

eventually provided observations by the agent which are processed through the network and an output is the result. The output of the policy is understood by the agent to represent a decision and the exact type and number of outputs is dependent on the design of the policy, the input observations provided, and the type of problem being solved.

- Training

- A policy represented as a neural network needs to be constructed, evaluated, tested and improved in order for a desired solution given a set of observations to be made. Training is the process wherein observations are provided to a neural network with existing weights (on initiation they can be randomly established) and the output is then evaluated. If the output is positive towards achieving the goal of the policy, then that decision process may be encouraged through backwards evaluating the weights and adjusting so that the same decision would be more likely in the future given similar inputs, if it is negative towards achieving that goal then it may be discouraged in future decisions. The weights within the neural network are adjusted accordingly. During the training process, data is being provided to the neural network policy to develop relationships between observations and outcomes as they related to the goals of the policy and agent.

- Reward

- During the evaluation of a decision the weights of the neurons in the neural network need to be adjusted to encourage or discourage similar

decisions from being made. This is done through a reward (R) process wherein an output decision is evaluated against a known or suspected ideal decision which is then further compared with a reward function that determines the desirability of the reward. This results in strengthening or weakening the bias of the neural network for future decisions given similar observational input data. The policy attempts to maximize the reward given to the agent and avoid negative rewards.

- Inferencing
 - Once a policy has been developed and trained, applying that policy to new data is called inferencing. In this process the policy is no longer training to improve its response rather it has been trained and is now the agent is used to solve whatever types of problems it was designed to solve.

Reinforcement learning rewards can be *intrinsic* or *extrinsic*[127] where an intrinsic reward is related to an agent responding to self motivation (such as curiosity regarding the environment and a desire to explore) whereas an extrinsic reward is related to a specific task, such as accomplishing a goal in the environment (such as reaching a specific location).

The goal of the agent during training is to maximize the expected reward. ML-Agents Toolkit implements policy gradient methods which *rely upon optimizing parametrized policies with respect to the expected return (long-term cumulative reward) by gradient descent*[128] which ultimately means the motivations of the agent will be to maximize that total reward each 'game'. An advantage of using Unity ML-Agents Toolkit is that it provides a convenient interface between the virtual environment of the simulation in

Unity and the training environment handled in PyTorch. Furthermore, it includes the capability to take models after they have been trained and run them in Unity with simulated data. There are several algorithms including in the Unity ML-Agents Toolkit which include Proximal Policy Optimization (PPO)[129], Soft Actor Critic (SAC)[130], Generative Adversarial Imitation Learning (GAIL)[131] and Behavioral Cloning (BC). A generalize reinforcement learning cycle is shown in Figure 58.



Figure 58 - Reinforcement learning generalized cycle

A recent detailed discussion on the mathematics behind ML-Agents and its application as a solution for different problem is available[132].

5.3. Design of the reinforcement learning scenarios and agent training

Unity ML-Agents Toolkit was used for the implementation of reinforcement learning within the established framework developed for virtual environments [125]. A training environment was created where the following chain of events happens:

- Within a 10x10m room (with the centre of the room representing location 0,0,0), a radiation source will be randomly positioned in the environment at a height from 0 to 2 m
- An agent in that room and is provided 10 different measurements of that radiation source:
 - The agent is requested to 'guess' the location of the radiation source in the environment using the dose rate and positions of all 10 measurements²⁵. This represents an input set of 40 values.
 - During training, the guess is compared with the known radiation sources position and a reward function is applied that encourages guesses close to the source position and discourages guesses that are far away. The agent is able to continue to guess as its reward is accumulated and when the environment episode limit is reached (this varied from 10 to 1000), the episode is ended and the source is repositioned, the agents measurements are randomly selected again and the episode begins again with an accumulated reward of 0.

Configuring the agent for training requires determining the selection of a hyperparameter file that will choose the configuration of the training environment and network that will be used. In this example, a policy based on PPO with a 3-layer network of 1024 neurons each is established and an agent is trained. The agent can be

²⁵ Assuming an average walking speed[140] of 1.3m/s in a non-linear walking path during a survey, 10 was selected as a reasonable amount of information that could be accumulated by a surveyor while exploring a 10x10m space.

seen training²⁶ in Figure 59 and Figure 60. Figure 61 provides a generalized view of the training process under this given setup.

Table 2 - Example hyperparameter file for input to train a network (called RadiationSourceLocalizing)

```
behaviors:  
RadiationSourceLocalizing:  
  trainer_type: ppo  
  hyperparameters:  
    batch_size: 5120  
    buffer_size: 102400  
    learning_rate: 0.0003  
    beta: 0.002  
    epsilon: 0.2  
    lambda: 0.95  
    num_epoch: 5  
    learning_rate_schedule: constant  
  network_settings:  
    normalize: false  
    hidden_units: 1024  
    num_layers: 3  
    vis_encode_type: simple  
  reward_signals:  
    extrinsic:  
      gamma: 0.99  
      strength: 1.0  
  keep_checkpoints: 5  
  max_steps: 500000000  
  time_horizon: 1000  
  summary_freq: 50000
```

²⁶ During training it is quicker to conduct parallel training of multiple agents in order to more rapidly gather experience data.

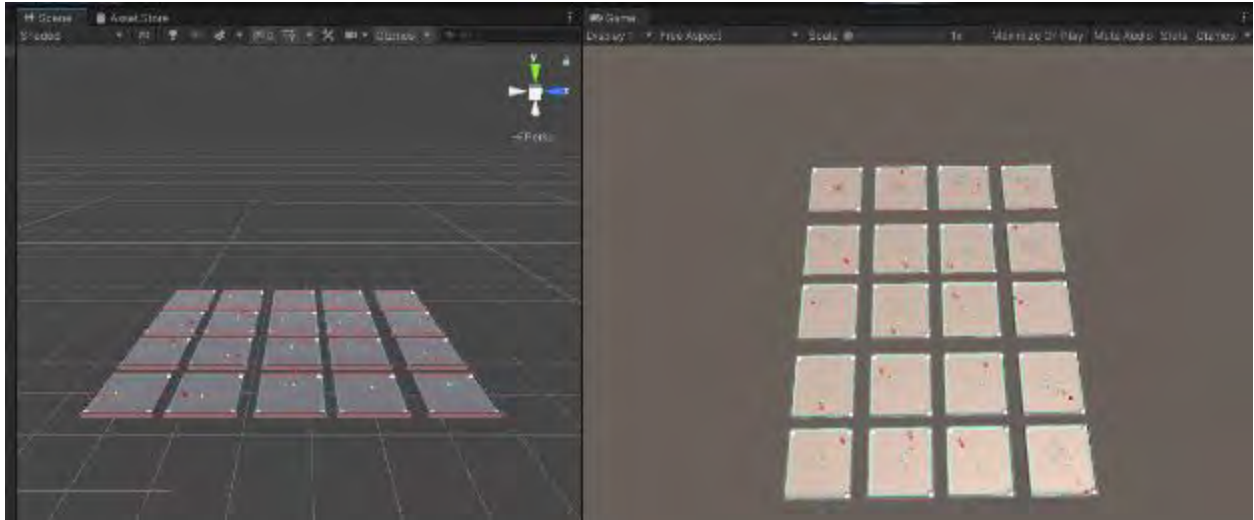


Figure 59 - Screenshot of the agents during training (1)

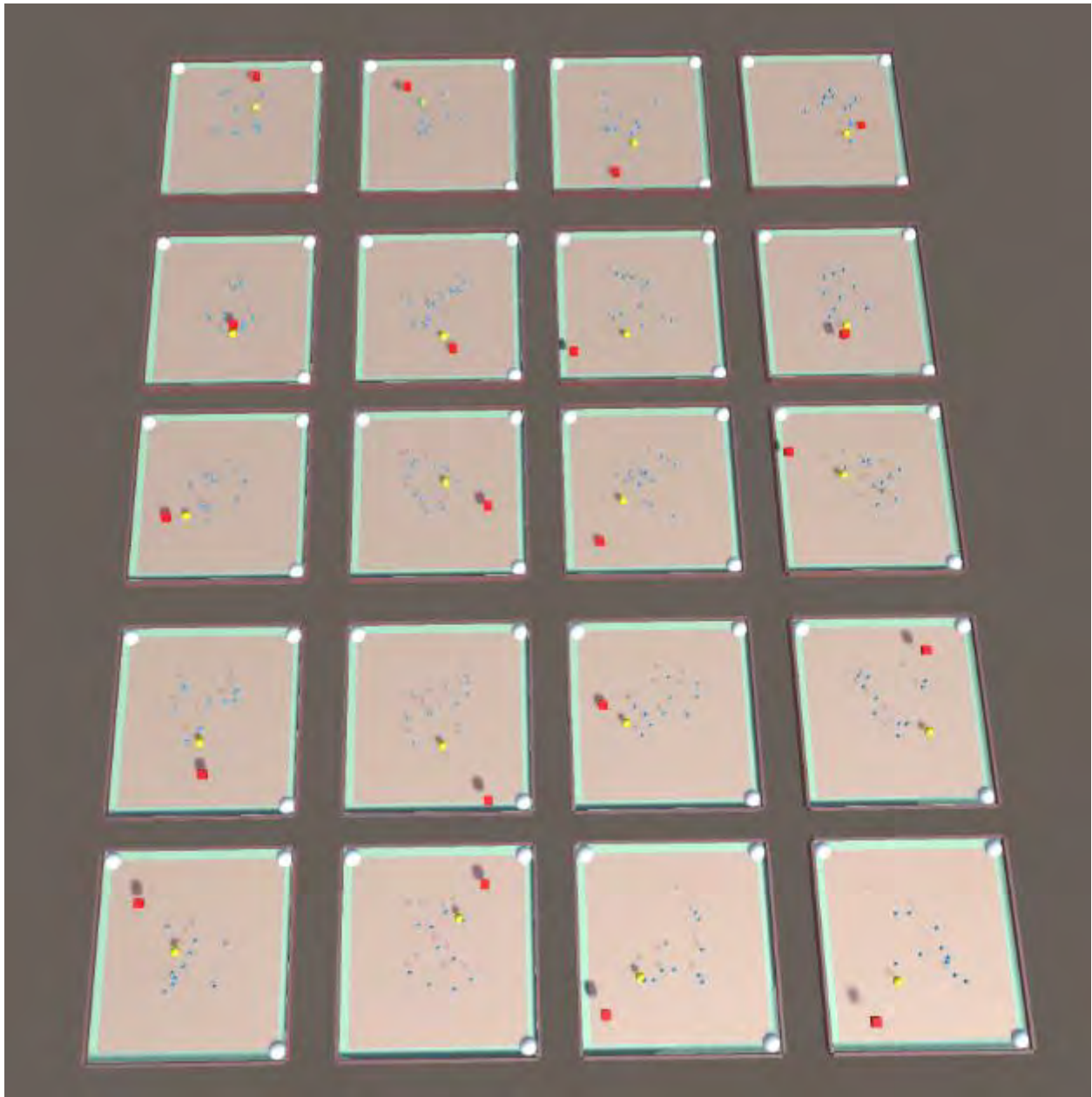


Figure 60 - Screenshot of the agents during training (2)

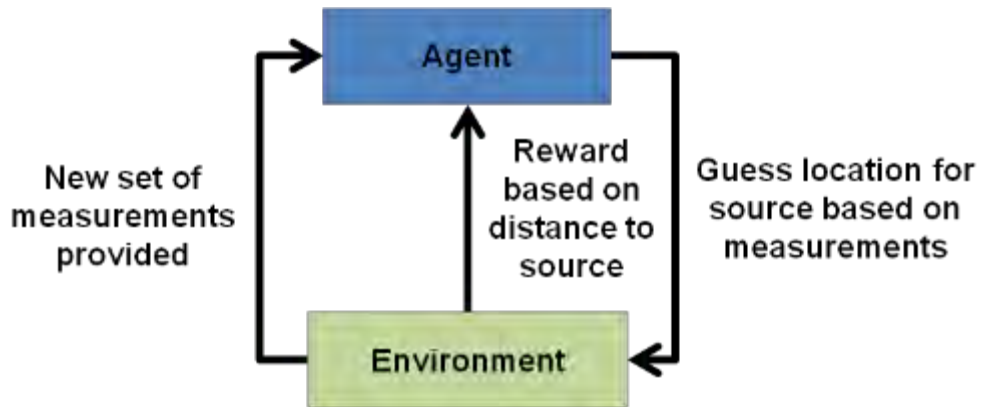


Figure 61 - Summary description of reinforcement learning for finding radiation source based on random set of measurements

5.3.1. Episode Selection

At the start of each episode the source would randomly be moved to anywhere within the environment room between -5 m and 5 m and 0 and 2 m in height. Measurements would be taken within the environment. Within -2 and 2 and 0 to 2 m in height, 10 measurement points would be randomly chosen.



Figure 62 - Image of training environment as seen from top-down perspective

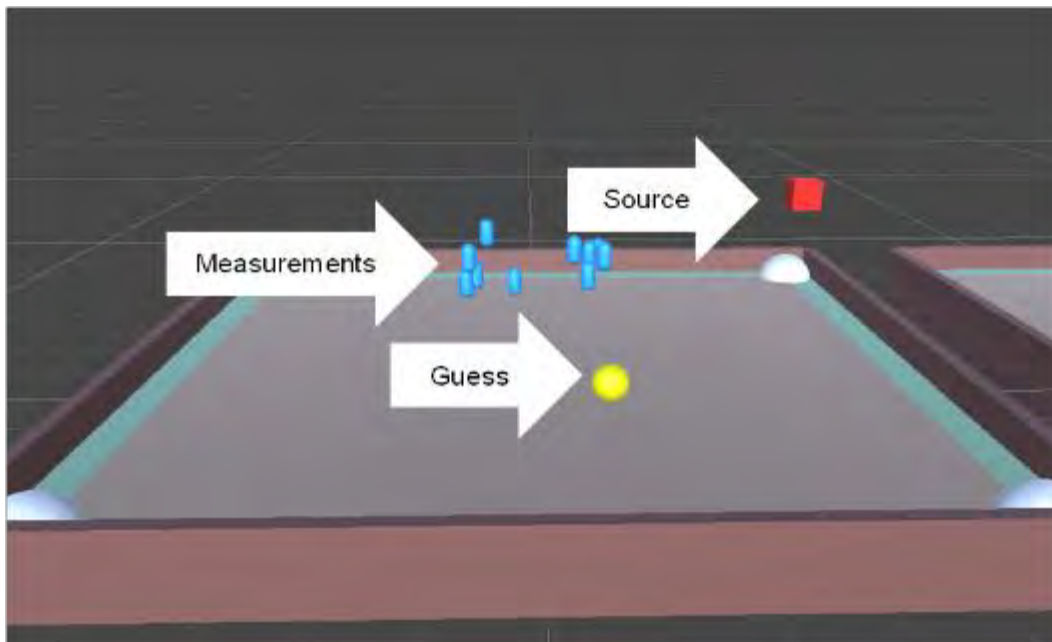


Figure 63 - Image of training environment as seen from side perspective

The instructions for the Unity ML Agents software package state that the network will train more efficiently if the inputs are normalized to fall within a range of -1 to 1, therefore each position value is normalized by 5 or 2. To normalize the measurements a simple linear method was used where the highest measurement in the collection was made equal to 1 and all other values were normalized based on that.

The following data was provided as input to the neural network for each measurement:

$(X_{\text{normalized-pos}}, Y_{\text{normalized-pos}}, Z_{\text{normalized-pos}}, \text{Doserate}_{\text{normalized}})$

Measurements			
Measurement Positions			
Size	10		
Element 0	X 0.7955036	Y 1.054604	Z -0.4222416
Element 1	X -0.8995731	Y 1.438246	Z -0.7712595
Element 2	X 0.4424016	Y 1.128855	Z -0.09768403
Element 3	X -0.6259921	Y 1.707463	Z 0.6737683
Element 4	X 0.5182235	Y 1.092564	Z 0.3868881
Element 5	X 0.7021244	Y 1.756997	Z 0.1936272
Element 6	X -0.7560477	Y 1.181077	Z -0.8077619
Element 7	X -0.111616	Y 1.210508	Z 0.6079614
Element 8	X -0.8391702	Y 1.204233	Z -0.3959409
Element 9	X 0.1297327	Y 1.473409	Z 0.3108117
Measurement Doubles			
Size	10		
Element 0	3.66960320339445e-06		
Element 1	5.20035793876606e-06		
Element 2	4.30835373244809e-06		
Element 3	6.87359918061327e-06		
Element 4	4.62384330759462e-06		
Element 5	4.01976884446349e-06		
Element 6	5.06208099499962e-06		
Element 7	5.94468938358841e-06		
Element 8	5.83473785087318e-06		
Element 9	5.03031207167715e-06		
Measurement Doubles Normalized			
Size	10		
Element 0	0.533869244768364		
Element 1	0.756569855489025		
Element 2	0.626797347247079		
Element 3	1		
Element 4	0.67269609211954		
Element 5	0.584812808957657		
Element 6	0.736452746514087		
Element 7	0.864858311836859		
Element 8	0.84866210230731		
Element 9	0.731830870479757		

Figure 64 - Example of data being tracked during the training process showing positions of the measurements in the room, the measurements and the normalized measurements

To improve efficiency in the training process a Generative Adversarial Imitation Learning (GAIL) was tested[131]. The use of GAIL in reinforcement learning allows the opportunity to provide in the training process idealized results which the Agent will study in order to better understand what actions are expected in a given situation. As the location of the radiation source is always known in each virtual environment (but always unknown to the Agent), the correct location can always be identified. Although this

method ultimately was not used in the final networks that were generated as they were efficiently being trained without this technique, it did prove promising for allowing more complex AI behaviours to be trained (such as adding shielding or multiple sources into the environment to be evaluated). Behavioural cloning is an alternative technique compared to the used of GAIL in this process but it was also not pursued as the results based on PPO were successful as a demonstration and hyperparameter optimization was not the focus of this research.

5.4. Selection of an agent training process

Different techniques were used to train agents. These included:

- Modifying the number of measurements given to an agent
- Allowing the agent multiple opportunities to make a guess from given data (from 10 to 1000) before being provided new measurement data
- Allowing the agent only a single opportunity to guess the location of a source from a given set of measurements
- Modifying the size of the space in which the measurements were provided to an agent
- Modifying the parameters of the neural network being trained such as:
 - Number of hidden layers
 - Number of neurons
 - Modifying parameters of the PPO training process
- Modifying the reward function with approaches such as:

- A reward proportional to the distance from the source with a goal distance being +1 reward and all other distances from that being $1/(\text{distance away from ideal})$
- A sparse reward function that was only positive when the agent exactly achieved its goal of finding the source
- A more frequently negative reward function providing frequent negative rewards for incorrect guesses outside of the desired range.

The process with the highest success rate leveraged an approach with the agent being provided 10 random measurements. The scene intentionally resets after each 'guess' by the agent. Therefore, the agent is provided only a single opportunity to 'guess' the location of a radiation source based on a single set of 10 measurements before the source is randomly moved and a new set of randomized data is provided. This was determined based on experience from training a varied set of models and to maximize the agent's exposure to the widest set of environments. A reward function was designed where the guess would be evaluated and a reward provided with the following parameters shown in Table 3 and plotted in Figure 65.

Table 3 - Tabled reward function (provided when value is less than or equal to)

Distance Guess from Source (cm)	Reward provided during training
1000	-1
900	-1
800	-1
700	-1
600	-0.75
500	-0.5
400	-0.1
300	0.001
200	0.02
150	0.1
125	0.5
100	1

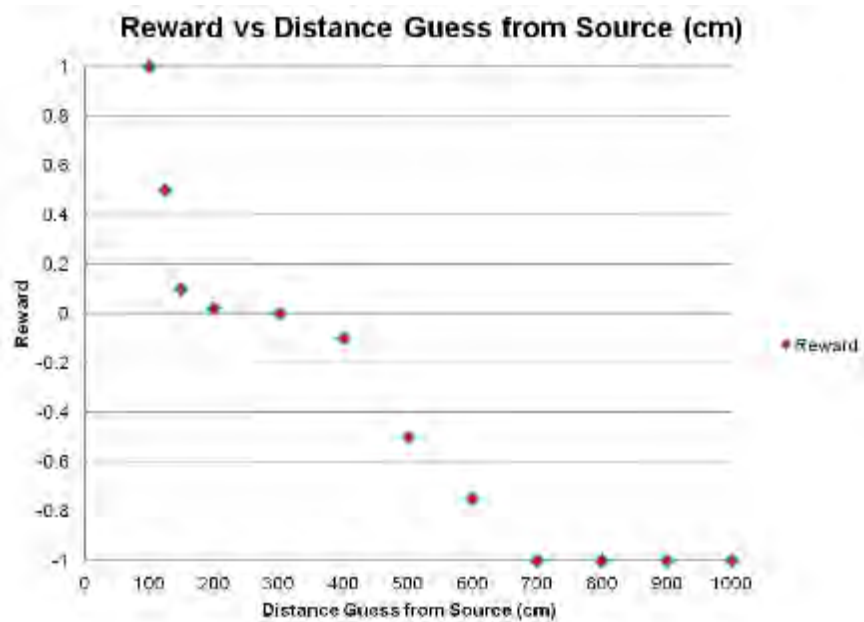


Figure 65 - Plotted reward function

During training the agents response to this reward function can be seen in Figure 66 and Figure 67.

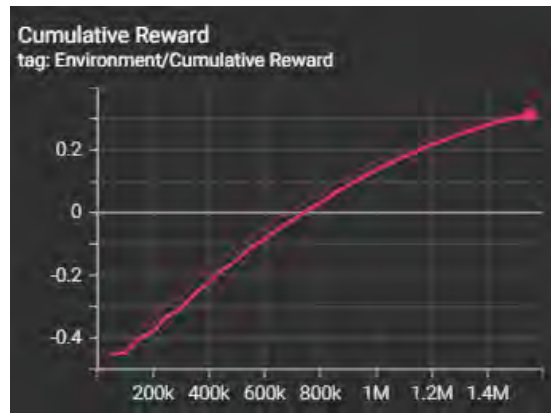


Figure 66 - Agents response to reward function during training (at step ~1.5 million)



Figure 67 - Visualization of agent response during training with the most accurate 'guesses' circles in red (at step ~1.6 million)

While it can be seen that the Agent is making some guesses that are getting closer and closer to the source as the accumulated reward per environment (or guess), it can also be seen that the Agent is still making guesses that can at times be inaccurate. Figure 68 is a histogram plot showing over time how the frequency of a given reward is selected. This shows that over time the agent is tending to receive a full -1 reward much less frequently while also receiving a +1 reward much more frequently.

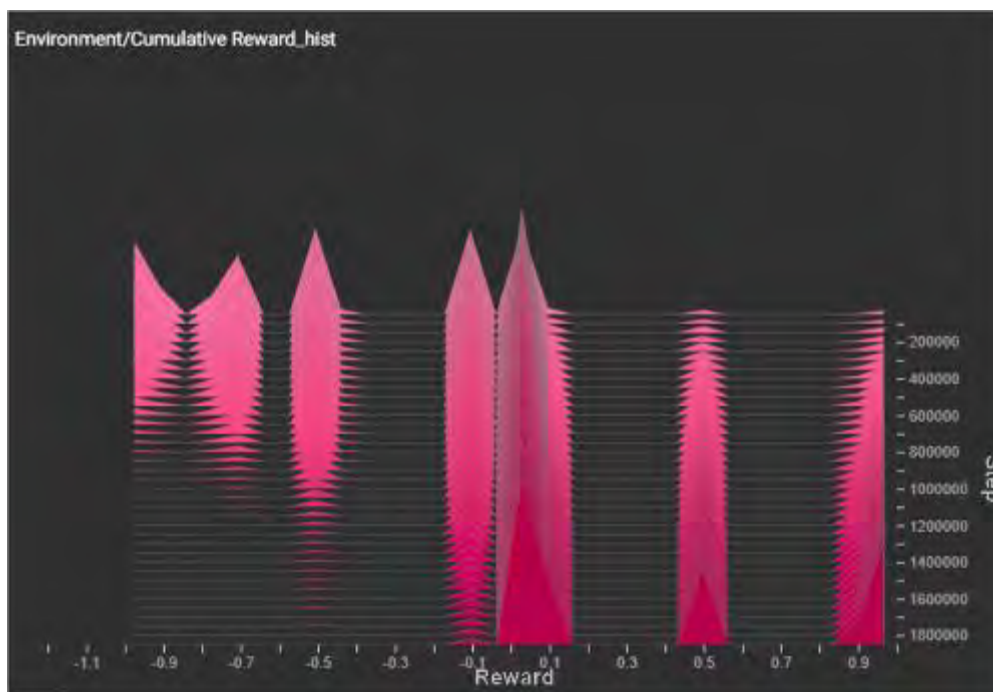


Figure 68 - Histogram plot of the frequency of reward vs. time during training (at step ~1.8 million)

As the training process with the agent continues it improves its ability to get closer and closer in its guess.

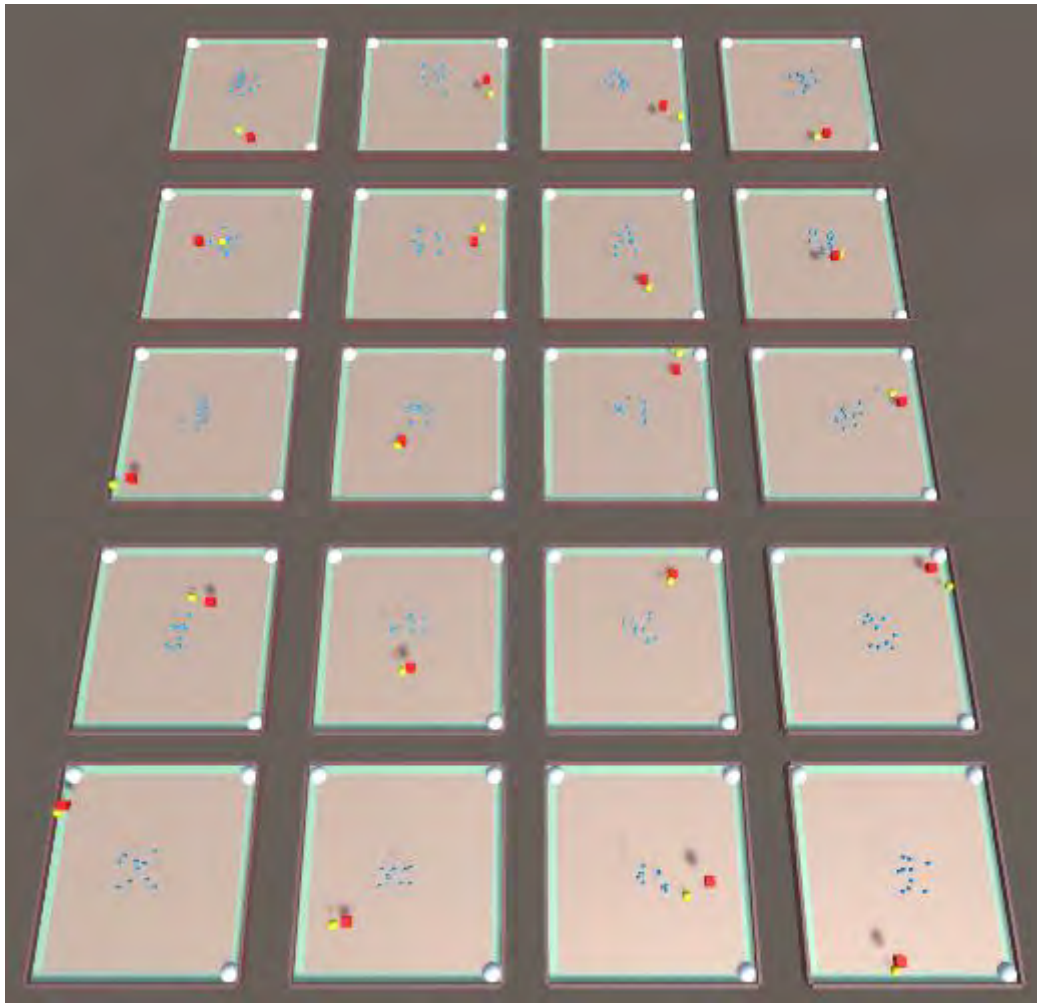


Figure 69 - Visualization of agent response during training (at step ~4 million)

Different techniques were used during this research in an attempt to find a solution that would reach a performance guess with an average near 1 meter based on any random input set of 10 measurements. This included trying many different settings within the hyperparameter file and experimenting with using PPO vs. SAC algorithms. Ultimately,

as this research was focused on demonstrating a generic operational solution, optimization of the hyperparameters for the agents was left for future research to investigate with more complex environments.

CHAPTER 6: SYNTHESIS OF RADIATION MEASUREMENT VISUALIZATION, SOURCE LOCALIZATION AND AUGMENTED REALITY TRAINING APPROACHES

6.1. Radiation measurement visualization

A scenario was made wherein radiation data could be displayed and visualized to a user using the previously developed connection between the radiation detector and the portable phone hardware device. Three sources of radiation that could be used to produce a response in the detector to show the real-time visualization of the measurements were acquired and consisted of two pieces of Fiesta ware containing uranium oxide (Figure 70 and Figure 71) and a gas lantern mantle which contains thorium (Figure 72). These sources were selected based on the ability to provide a response in the detector, while remaining under regulatory limits not requiring any licensing or authorization for their use allowed for a proof of concept demonstration evaluation.



Figure 70 - Fiestaware dish 1



Figure 71 - Fiestaware dish 2



Figure 72 - Thorium mantle

These three sources were then put in an indoor environment, separated from each other, as can be seen in Figure 73. The scenario required the user to survey and locate these three sources of radiation.



Figure 73 - Three sources of radiation placed in an indoor environment

During the conduct of the survey the phone was supported by a simple 'selfie stick' which could be held at the same time as the radiation detector. This was required as the location of the detection volume was programmatically expected to be at approximately the same location as the camera²⁷. The setup can be seen in Figure 74 and Figure 75.

²⁷ This can be adjusted when placing measurements in the environment. As long as the detector remains at a fixed position in space relative to the radiation detector the corresponding measurements can be placed in the correct position in the environment by using the cell phones current location and translating the measurement by the known fixed position of the measurement volume of the detector.



Figure 74 - Radiation detector and phone apparatus setup for surveying (front view)



Figure 75 - Radiation detector and phone apparatus setup for surveying (front view)

Figure 76 through Figure 80 show the results of the real-time visualization during the survey. The specifics of the visuals are discussed in detail in Chapter 7.

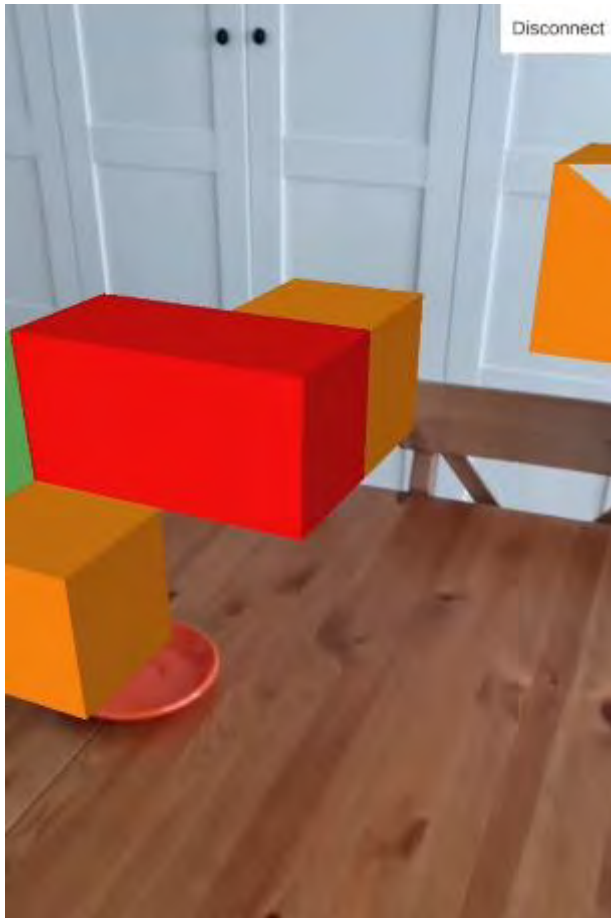


Figure 76 - Augmented reality radiation measurement real-time visualization screenshot (seen from first person perspective) (1)



Figure 77 - Augmented reality radiation measurement real-time visualization screenshot (seen from first person perspective) (2)

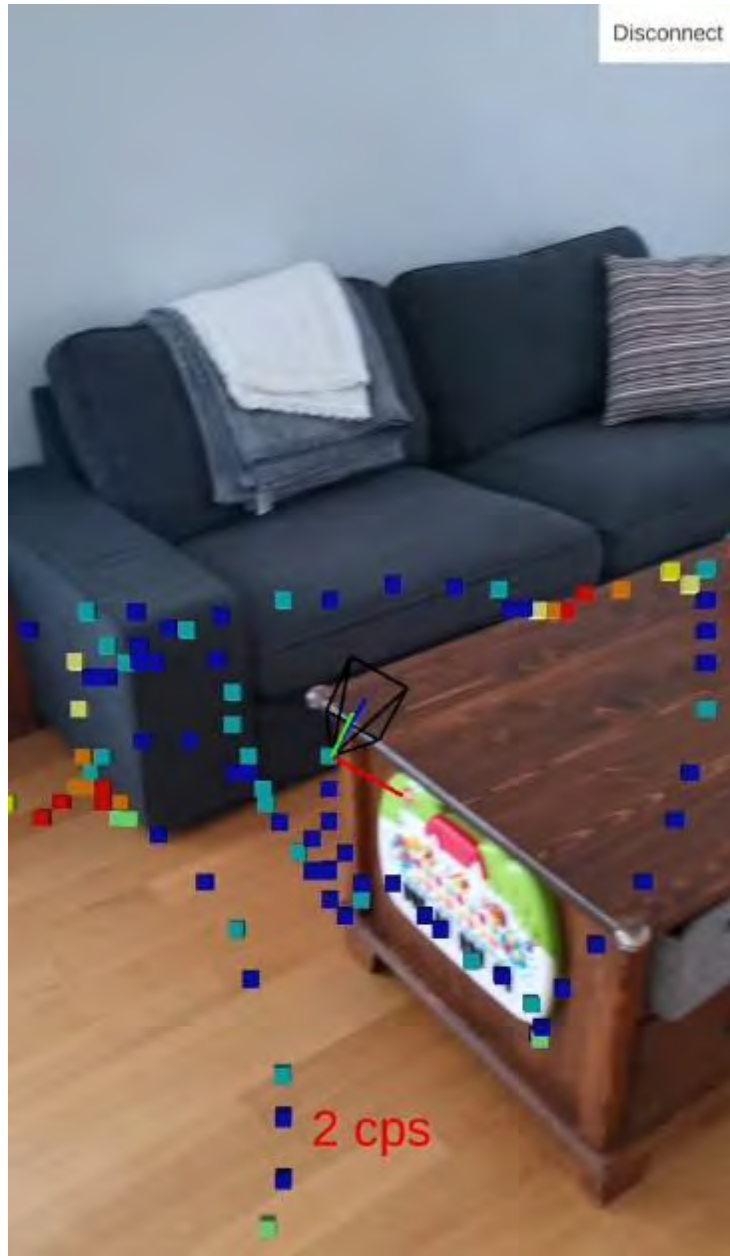


Figure 78 - Augmented reality radiation measurement real-time visualization screenshot (seen from top-down perspective) (3)

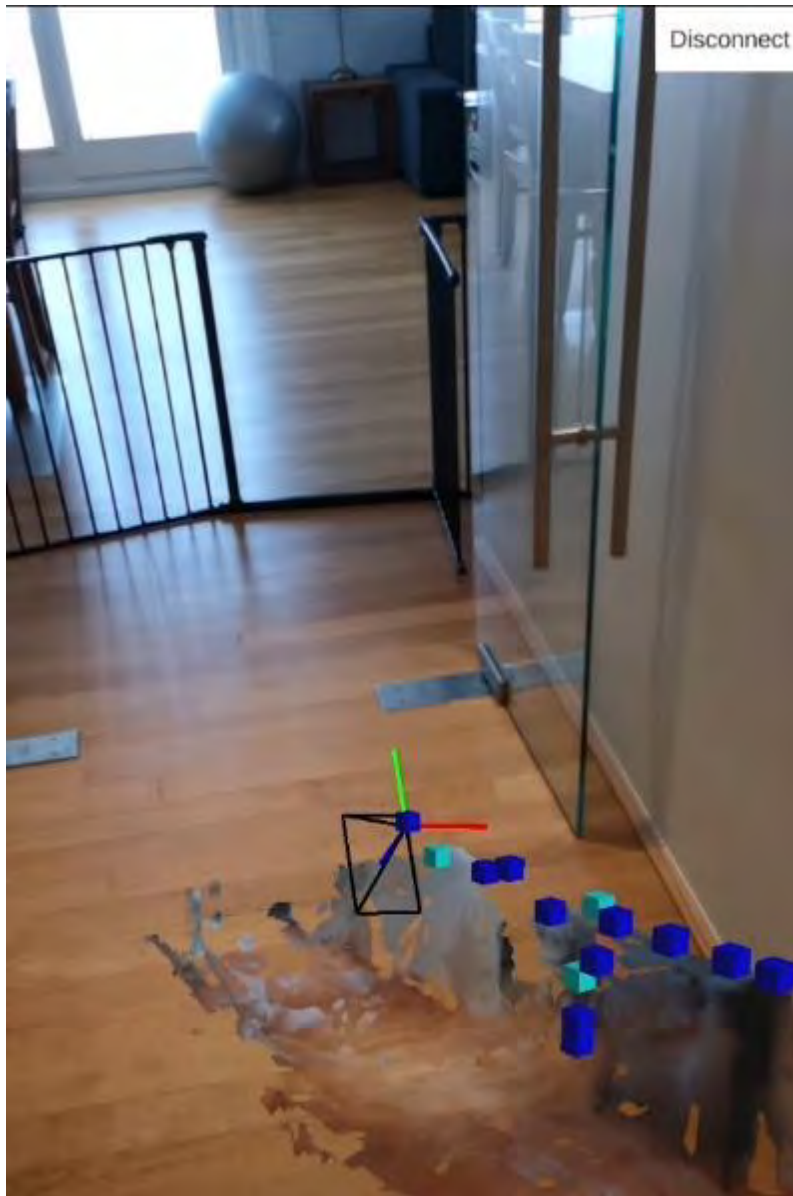


Figure 79 - Augmented reality radiation measurement real-time visualization screenshot (seen from third-person perspective with environmental meshing) (4)



Figure 80 - Augmented reality radiation measurement real-time visualization screenshot (seen from third-person perspective with environmental meshing) (5)

6.2. Radiation source localization support using ML

The augmented reality visualization and machine learning research was combined into a single phone application with the AR Foundations and AR Core technology. This application allowed the user to place a virtual radiation source in the environment (or have one randomly placed), take virtual measurements of that source and once a

minimum of 10 measurements were taken, the user could use those 10 measurements as input into a trained AI agent that would attempt to guess the location of the radiation source, providing the user with a generalized indication of where they should proceed. Meshing of the environment and multiple camera angles were implemented to also support the user conducting more detailed evaluation of the environment. Figure 81 provides a screenshot of the interface for the application as well showing the placement of a single source of radiation in the environment.

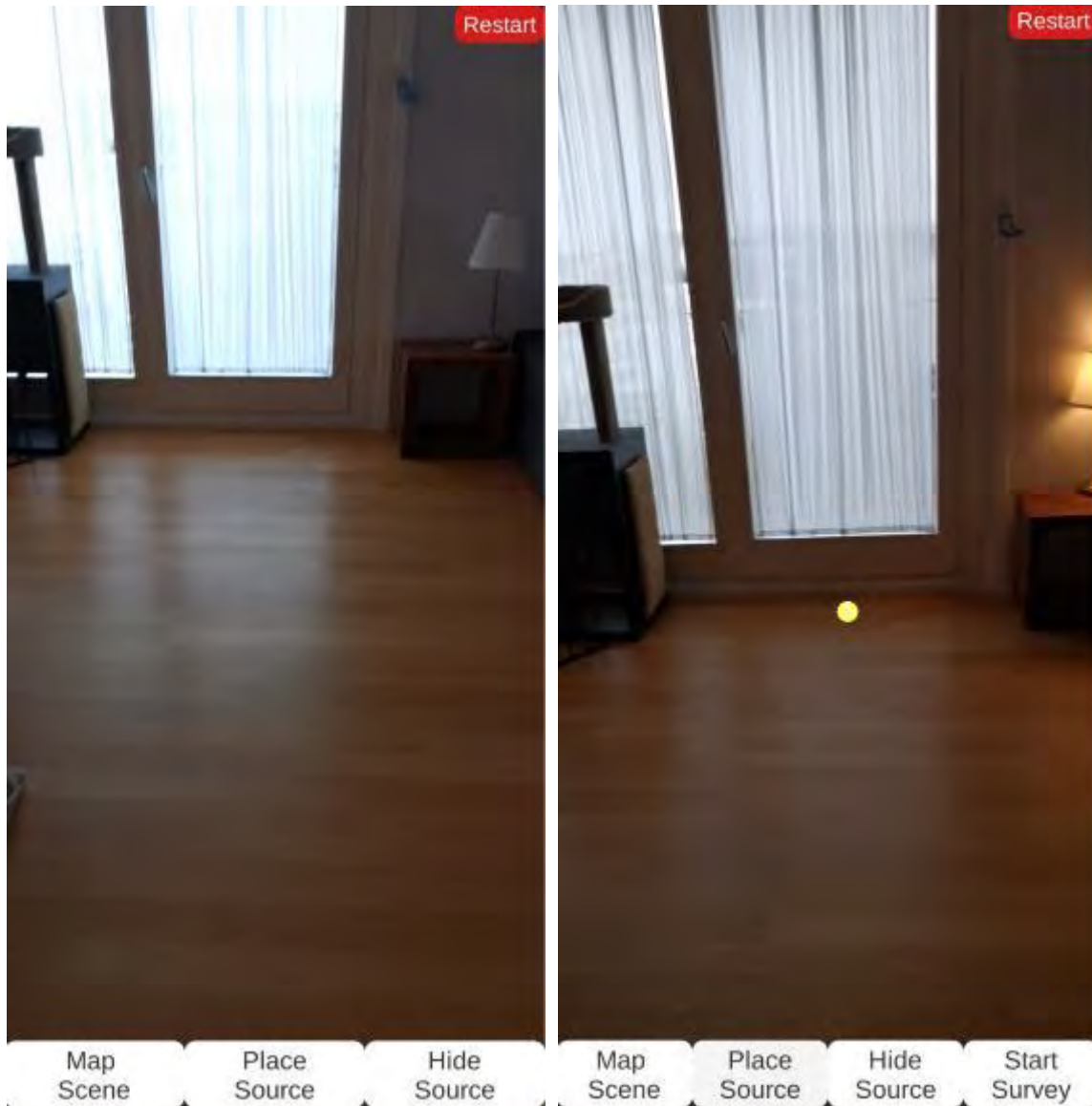


Figure 81 - Example primary interface for the application with and without the placement of a source of radiation

Figure 82 shows the example of the augmented reality visualization of the measurement of a single source of radiation using three different viewpoints (3rd person, top down and first-person perspectives) showing how the measurements taken by the user can be

evaluated and viewed from any desired angle to help them understand the radiological measurements better.

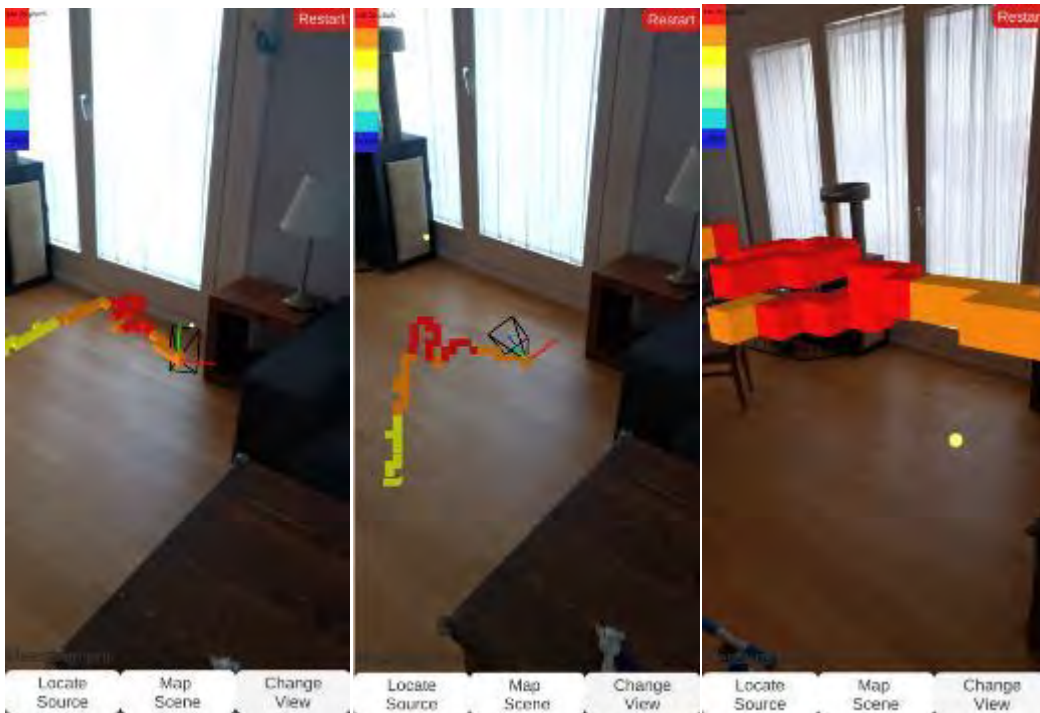


Figure 82 - Example of three view points (3rd person, top down and first person) in the example application interface measuring a single source

Figure 83 provides an example of the measurement of several sources of radiation in the augmented reality environment. Although the localization approach was developed for a single source of radiation, the capability to virtually measure the complex distribution of radiation from a mixed field of several sources was added into the application for experimental purposes which allowed for the augmented reality based visualization of more highly complex environments.

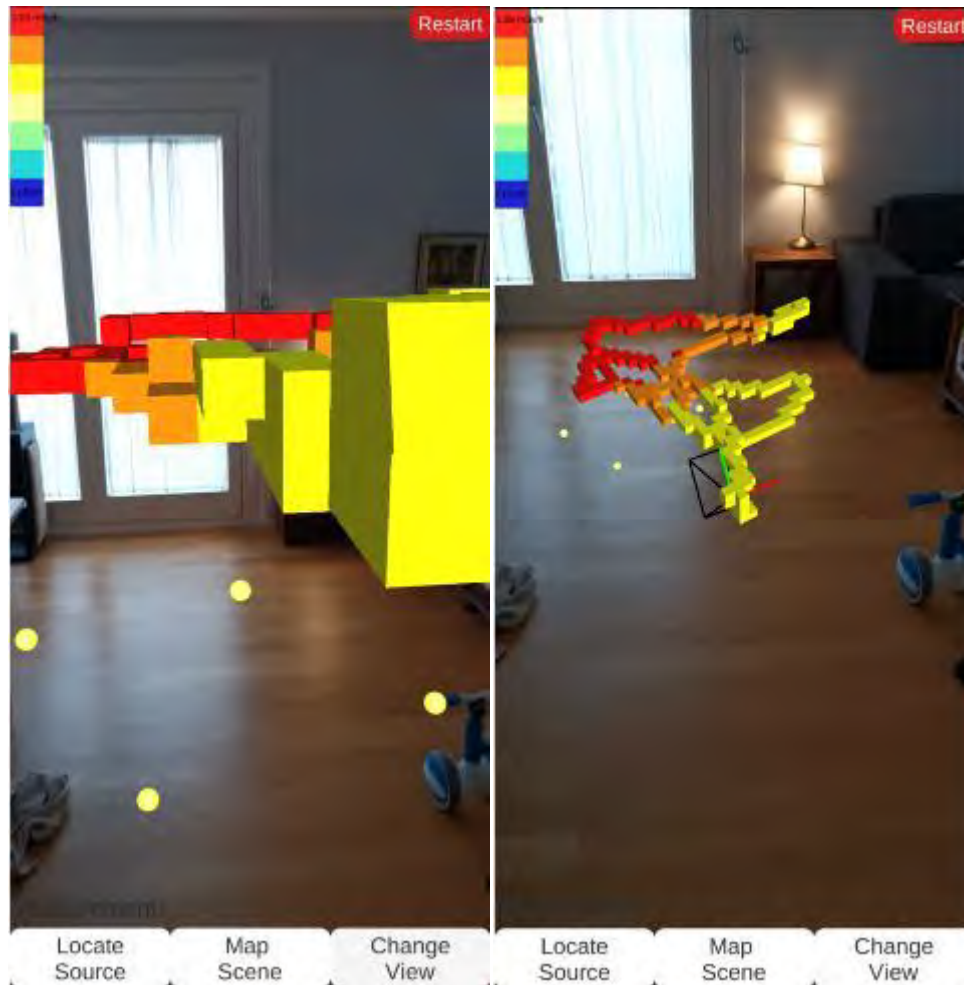


Figure 83 - Example of the measurement of 4 sources of radiation in the example interface from first and 3rd person viewpoints

Figure 84 provides an example of a single source being measured in the environment with the environmental meshing capabilities turned on, further demonstrating how this can enhance a user's ability to perform analysis of their measurements while deployed. Note that in this specific application a lower resolution environmental mesh is being used compared to what was previously used on the Google Project Tango Devices. This can be improved and more recent versions of AR Core have enhanced the output quality possible.

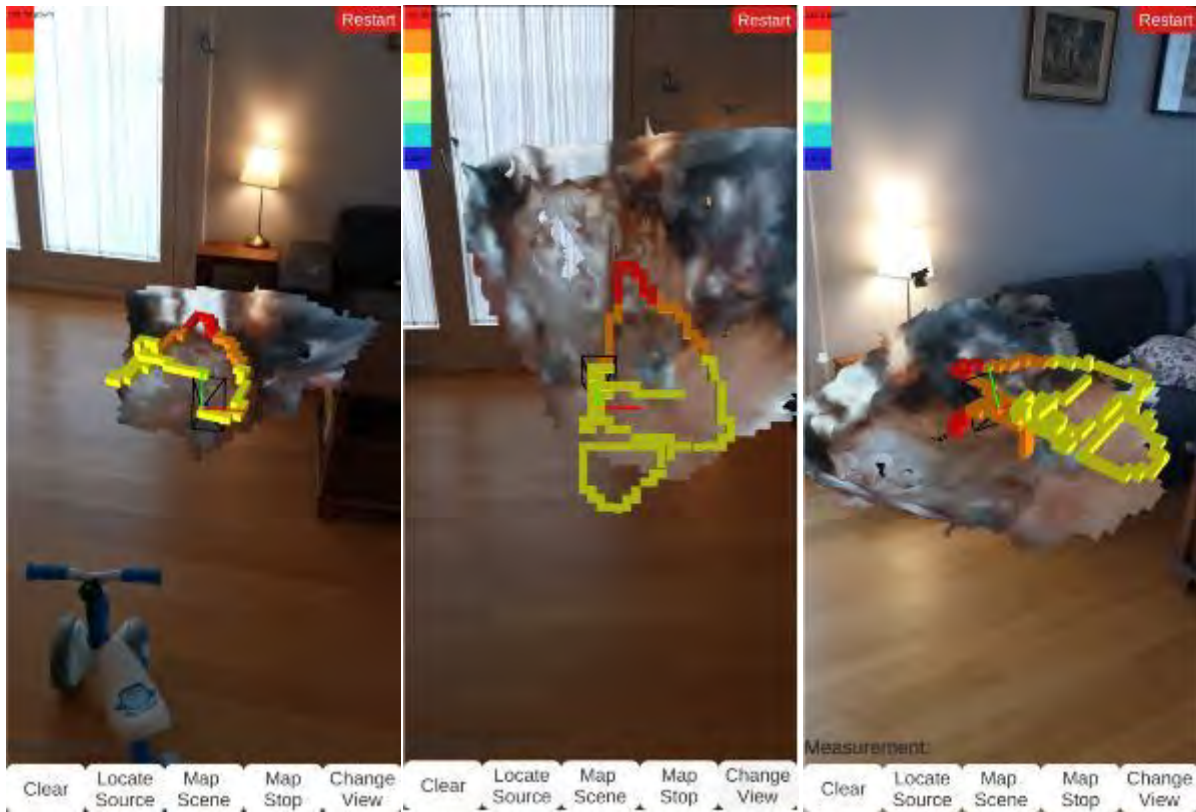


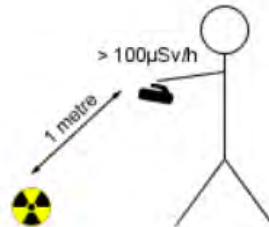
Figure 84 - Example of the measurement of a single source of radiation with the environmental meshing activated

Providing real-time guidance to first responders has been noted as an important area for development and international efforts continue in this area. Figure 85 provides an example of the type of guidance that should be available to responders recommended by the International Atomic Energy Agency[133].



Detection of elevated radiation levels

Elevated gamma radiation levels may be detected using a dose rate meter:



Any of the following readings are indications of elevated radiation levels:

- dose rate greater than 100 $\mu\text{Sv/h}$ at one metre from object or ground
- dose rate greater than 1000 times normal
- instrument goes off scale high

Figure 85 - Example of field guidance for first responders provided by IAEA Portable Digital Tool for Assisting First Responders to a Radiological Emergency²⁸

During a radiological survey where a single source is encountered, having the Agent identify an approximate location for the source provides additional useful analysis possibilities. Section 0 explores the results when the 'locate source' button is pressed and the measurements are given to the Agent to guess the location of the radiation source using a trained agent's neural network in this application and real-time advice is shown on the augmented reality screen.

²⁸<https://www.iaea.org/publications/8032/portable-digital-tool-for-assisting-first-responders-to-a-radiological-emergency>

6.3. Virtual reality training approach developed into scenarios

To apply the previously described virtual reality approach to training for first responders to a radiological or nuclear emergency, a number of scenarios were developed. These scenarios were delivered during a technical meeting in 2017 at the IAEA in Vienna, Austria²⁹. This meeting was attended by 62 international experts and its purpose was to review a draft IAEA Safety Guide on emergencies that may take place during the transport of radiological or nuclear material. As experience using VR during training was gained, these scenarios were modified and improved and new scenarios were created. The descriptions herein and the screenshots provided represent the scenarios as delivered during the 2017 technical meeting. The feedback from the participants is presented in Section 7.4. All four scenarios would be conducted in order, after the player successfully completes the training tutorial described in Section 4.4.

6.3.1. Transport accident with a radiological source in a vehicle (scenario 1)

This scenario focused on the challenges which may be encountered during the initial response to a mixed hazard emergency. The player, in this exercise, was provided a verbal briefing that they were of the carrier for a shipment of radioiodine. The narrative described to them by the instructor explains that they were transporting this material on a large roadway and were struck by a drunk driver in a yellow car. After the collision,

²⁹ <https://www.iaea.org/newscenter/news/iaea-closer-to-finalising-new-safety-guide-for-nuclear-transport-emergencies> and modified for use at other events such as <https://www.iaea.org/newscenter/news/iaea-hosts-workshop-on-nuclear-or-radiological-accident-assessment-and-prognosis> and <https://www.iaea.org/newscenter/news/virtual-emergency-response-experience-offered-at-the-iaea-general-conference>.

the drunk driver ran on foot away from the crash site into a corn field to flee from prosecution. The player is told that they are uninjured, but they were dazed from the accident, and have wandered several paces from the initial crash site. The exercise begins with the player 20 meters away.

The player is tasked with evaluating the area around the crash site for hazard. There are several visible hazards which they are expected to identify prior to moving closer to the scene:

- Smoke (implying possible fire) from the front of the yellow car
- Visible fluid leak from the yellow car
- Sparks from a damaged traffic line in front of the white van
- Visible signage indicating a radiological source is being transported
- Ambient traffic in the opposing traffic lane representing a hazard

The purpose of this exercise is to acclimate the players with the virtual reality environment. The scenario does not feature any radiological hazard nor does it require the use of any virtual tools (although a radiation detector is available which only monitors background).

Figure 86 shows the general arrangement of the scene. The hazards are deliberately placed in such a position that the player is required to walk completely around the outside of the crash prior to moving inwards.



Figure 86 - Screenshot from the transport accident with a radiological source in a vehicle showing the general environment of the scene

Once the player has identified all of the hazards, the instructor tells them they can now make a decision to move towards the scene. As they approach the white van with the source that was being transported, the player is able to see that the side door has been left ajar as shown in Figure 87.



Figure 87 - Screenshot from the transport accident with a radiological source in a vehicle showing the user approaching the accident

As they approach the open door the player is able to visually (and radiologically) inspect the package as can be seen in Figure 88. The package is able to be picked up and examined by the player from any angle. The scenario concludes once the player determines that the package has not been damaged.

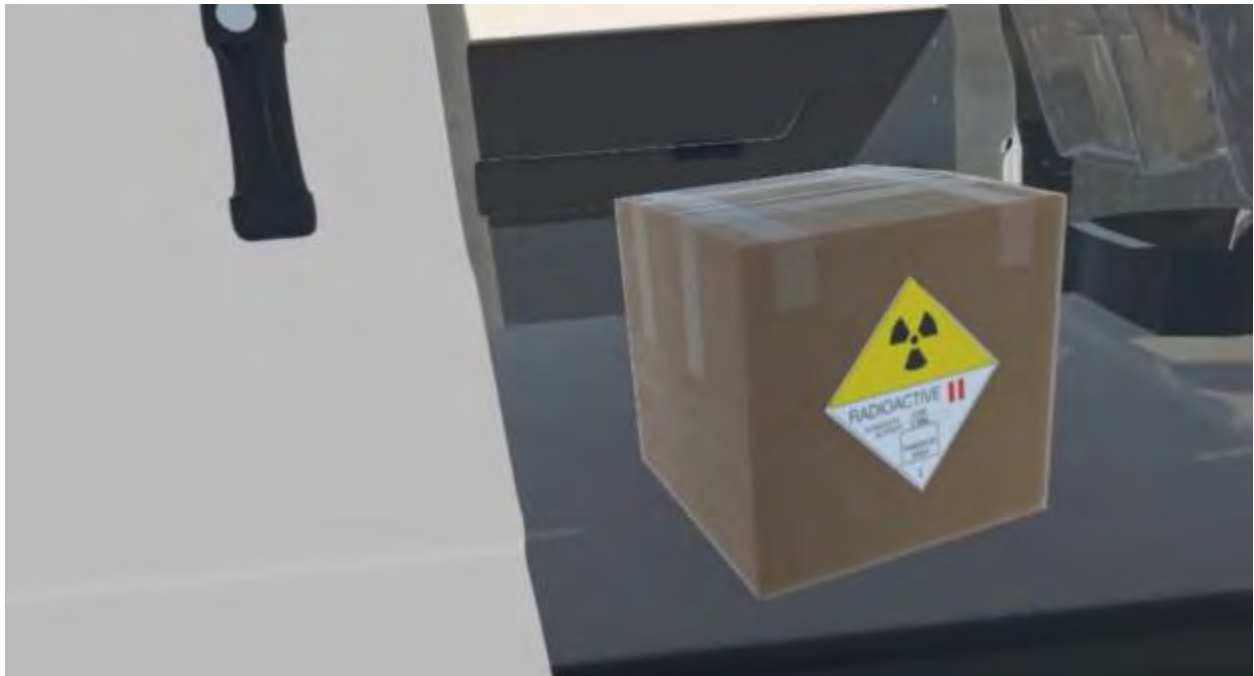


Figure 88 - Screenshot from the transport accident with a radiological source in a vehicle showing the user investigating the transport package

6.3.2. Transport accident with radiological material in severe weather (scenario 2)

This scenario focused on the challenges encountered while conducting radiation surveys in a complex environment. The player, in this exercise, is told that they are a qualified emergency response radiological assessor[134], who was called in to evaluate a transport emergency involving a truck shipping drums of natural uranium. From a distance behind an established cordon, the player is asked to describe the scene and their thought process to the instructor before they make entry into the scene from the vantage point seen in Figure 89.



Figure 89 - Screenshot from the transport accident with radiological material in severe weather with showing the general dark environment and severe rain (not visible is severe wind causing the trees to shake)

The participants were visually asked to identify the issues which made this scene complicated which include reduced visibility due to being set near sunset, heavy rain in the environment, and very high wind speeds which causes the trees to sway and move rapidly. Participants were challenged with questions regarding how the weather conditions could impact their response. Once they have had that discussion with the controller, they would then be expected to approach the scene carrying a radiation detector with them to inspect the drums which appeared to have been ejected from the open door on the truck storage area during the accident.



Figure 90 - Screenshot from the transport accident with radiological material in severe weather with showing the user investigating a drum, visually inspecting its signage, conditions and taking radiation measurements

The drums were all emitting elevated radiation measurements that were above background. They would visually inspect the drums and take surface measurements. The radiation detector was able to 'push' the drums and knock them over, forcing the players to be mindful of the position and angle of the measurements they were taking. A crushed drum was hidden in the middle of the intact drums knocked on its side. The players would then need to physically bend over to take a measurement of an object on the ground as can be seen in Figure 91. The requirement to be both mindful of moving the radiation detector around the drums and then needing to bend over to get measurements from the damaged drum contributed to the players having a better understanding of the VR environment and their movement in the virtual space.



Figure 91 - Screenshot from the transport accident with radiological material in severe weather with showing a user investigating a damaged drum requiring them to crouch down

6.3.3. Transport accident with a sea container carrying radiological material at a port (scenario 3)

This scenario focused on the challenges which can be encountered when surveying a high radiation environment where there are significant amounts of shielding complicating the radiation field distribution. The descriptive background given to players by the controller at the start of this scenario explained that they are an emergency response radiological assessor called to respond to an incident at a port. As packages from a large ship were offloaded, several stationary radiological alarms were triggered.

A crane carrying the packages off the transport ship failed and all of the contents from a sea container were dropped onto the surface of the port. The tasks for the participant were to examine the scene and determine which packages contained radiological material and which did not. The scene was deliberately designed with similar packages lacking any signage to force the players to think and move slowly during their survey. The initial configuration of the scene can be seen in Figure 92.



Figure 92 - Screenshot from the transport accident with a sea container carrying radiological material at a port showing the general environment

Three sources are hidden in different boxes amongst the damaged packages. The sources provided a surface contact dose rate on the packages of 2.5 mSv/h. The other boxes acted as shielding material. As the player searched the scene, they would be able to pick up and move any packages. When a player successfully identified a

package as having a source, which could be confirmed by a controller, the controller would remove it from the scene and the searching would continue. The players were not told how many sources were in the scene and after finding the third package, players would be allowed to continue searching until they were satisfied that the scene was only measuring background. Figure 93 and Figure 94 show a player navigating the environment and searching through the packages. The packages were stacked on top of each other so that players were required to bend over and move in all directions in the play space.



Figure 93 - Screenshot from the transport accident with a sea container carrying radiological material at a port showing a user investigating the scene and trying to identify which packages carry radiological material



Figure 94 - Screenshot from the transport accident with a sea container carrying radiological material at a port showing a user investigating the scene and trying to identify which packages carry radiological material

6.3.4. Transport accident with radiological material on a train in a rural area (scenario 4)

This scenario focused on the challenges which could be encountered during the evaluation of an accident with a transport package of spent nuclear fuel. The scenario focused on a train derailment in a rural area where two of the cars on the train were Type B(U) packages[135] with appropriate signage. The community was framed as an isolated city in northern Canada. In this exercise, the player is a first responder who arrives at the scene of the train derailment. The primary task was to determine if the environment was safe to navigate and if so, navigate and inspect the scene noting important observations to the controller.

The train was derailed in this scene. There was visible smoke coming from the locomotive. The first train car was a 'ruined' car which was externally visible and made to look wrecked. The majority of the remaining train cars were intentionally designed to approximate the visual appearance of oil transport containers[136] but upon closer inspection were labelled as containing milk. The front of the train is shown in Figure 95.



Figure 95 - Screenshot from the transport accident with radiological material on a train in a remote community showing the train from close perspective

The vantage point of the player was more than 100 m away as shown in Figure 96. Players would be asked to make observations and describe the scene to the controller. The players would then be told that there is a scope on a nearby table that they could use to help them determine if they would approach this scene as can be shown in Figure 97. Players would then be asked to continue making observations using the

scope and tell the controller if they believed it was safe to approach the scene. Ultimately the players would not find any radiological hazards as the safety functions of the Type B(U) packages were intact.



Figure 96 - Screenshot from the transport accident with radiological material on a train in a remote community showing the train from the viewpoint of the user



Figure 97 - Screenshot from the transport accident with radiological material on a train in a remote community showing the user using a scope to investigate the train and read signage and assess hazards from a distance

6.4. Mixed reality training approach

During the conduct of this research another innovative technique was explored that allows the VR scenarios to be used in a modified way, called Mixed Reality (MR) filming. Mixed reality filming is a technique where the positions of the VR headset and controllers relative to a filming camera are tracked in real-time. With this information, the footage from the camera mixed with the synchronized positional data from the controllers and headset can be used to blend in the 3d environment with the person in the virtual experience. Several of the VR scenarios were filmed with this mixed reality filming technique[137] and the results were shown to be useful as an introduction for the VR experiences and as teaching aids to explain how a scenario should progress.

Figure 98 shows the space that was used for the mixed reality filming using a green screen behind the user. Figure 99 through Figure 103 show some elements of a scenario filmed in mixed reality involving a radiological source being transported in a white van which is struck by another vehicle based on a modified version of Scenario 1.



Figure 98 - A space setup for mixed reality filming using a green screen



Figure 99 - User beginning a scenario picking up a radiation detector from a table and examining its response with a check source (filmed in mixed reality)



Figure 100 - User approaching the scene of a traffic accident involving the transport of a radiological source (filmed in mixed reality)



Figure 101 - User visually examining and measuring a package with a radiological source (filmed in mixed reality)



Figure 102 - User retrieving a package with a radiological source from a traffic accident (filmed in mixed reality)



Figure 103 - User putting a package with a radiological source into a concrete container for temporary storage (filmed in mixed reality)

CHAPTER 7: RESULTS

7.1. Results from the use of augmented reality to visualize a radiation field

7.1.1. Results of simulated radiation field visualization using Layer Augmented reality web browser

The initial experiments using the Layer Augmented Reality web browser were interesting but ultimately were far too limited in capability to be useful for a visualization approach that would be flexible and field deployable. One of the major issues with this method was that the limited 3d rendering computing capabilities of the hardware required that the visualization approach be significantly simplified. The GPS and compass tracking used to determine the position of the device and the direction the camera was facing resulted in an overall visual that appeared to be floating in space on top of the view from the cell phone camera. This effect can be seen in the sequence of three photos in Figure 104, which were taken from the same position but show the model has moved in all three images.

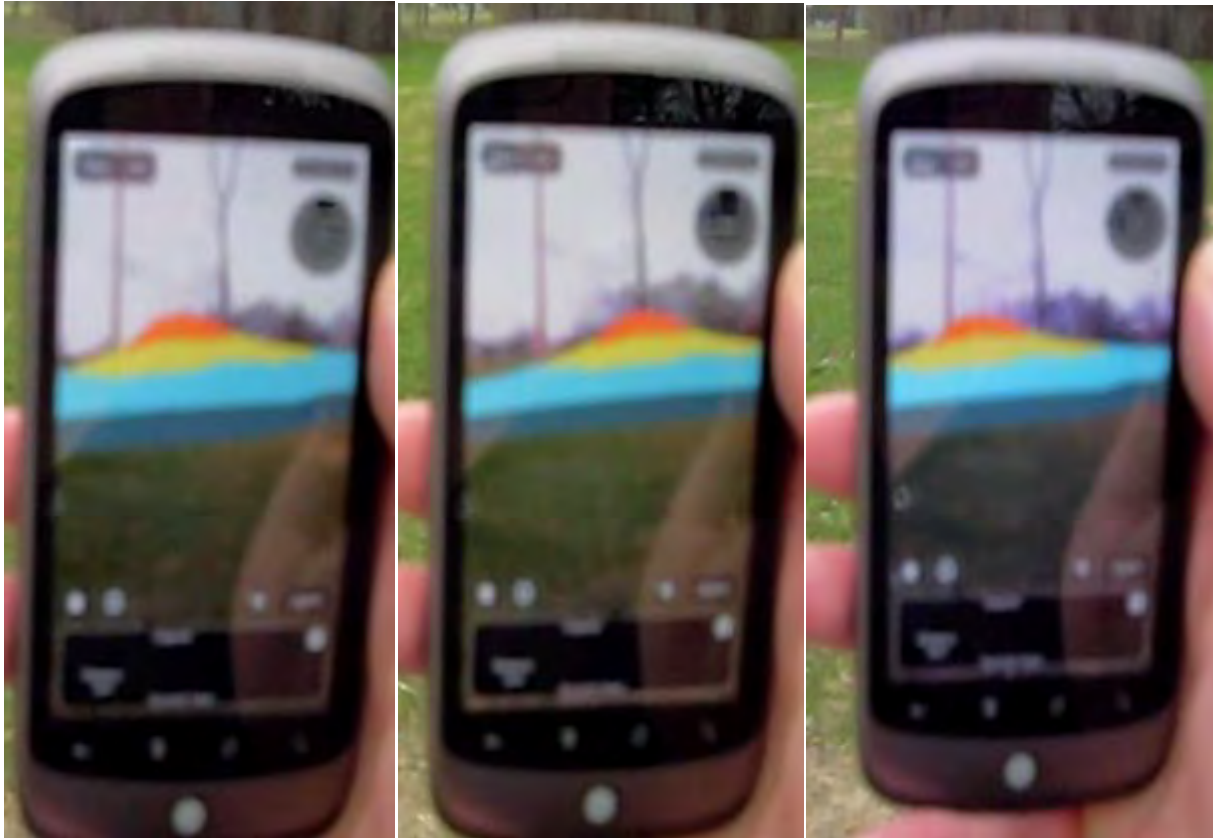


Figure 104 - Series of images from the same location showing the radiation visualization failing to 'lock in place'

Overall, the initial visualization approach was seen as a stepping stone to the more advanced approach developed with the Google Project Tango hardware and eventually AR Foundations with AR Core in Unity.

7.1.2. Results of simulated radiation measurement visualization using Google Project Tango and AR Foundations with AR Core

The experiments using the Google Project Tango and later AR Foundations and AR Core as the means to track the phone through the environment proved much more successful than the initial approach with the Layer Augmented Reality web browser.

The increased computational capabilities of the hardware allowed for the radiation field to be simulated in real-time as a user navigated the environment and each measurement could be placed as it was taken in the simulation. The environmental 'fixing' of a 3d measurement in space was very stable and allowed for simulated measurements of the radiation field to be viewed from any angle reliably and repeatedly.

The capabilities of the hardware to provide a real-time generated 3d mesh of the environment proved very useful in helping to understand where previous measurements were taken when viewing the results from different angles. The user was not required to always view the environment in a first-person perspective and was able to use any perspective they wanted with a free roaming camera and options for first-person, third-person or top-down default camera angles being provided. Figure 105 shows an example of some measurements that have been taken in AR using the advanced approach where environmental meshing is disabled as is the AR viewpoint (i.e. real world camera view). Each cube represents an individual measurement and they are maintained locked in positions as the camera representing the user (seen at the bottom center as lines representing the field of view of the user) moves freely about.



Figure 105 - Example of advanced AR approach where data remains fixed in the environment for the user to analyze

Figure 106 shows an example of adding the real-time meshing into the experience so that the user is able to see the surroundings and contextualize the measurements better when evaluating them. In this analysis the user has traversed 3 flights of stairs in a small suburban house. Although the meshing adds some visual noise, the capability of rotating and moving the camera in real time helps to contextualize and therefore understand the environment. The AR viewpoint remains disabled in this figure.



Figure 106 - Example of advanced AR approach showing the benefits of real-time meshing to help contextually understand where measurements are being taken in the environment

Figure 107 shows an example of a user passing down a flight of stairs, taking some measurements then returning to the same location to go back upstairs. The meshing of the environment and the previous measurements which were taken appear locked in place have not drifted in the frame of reference from the phone.



Figure 107 - Example of advanced AR approach showing that even when reentry into the same environment is done the original results remain fixed in the environment

Figure 108 shows a top-down perspective of a single level apartment where some measurements of a virtual source are taken. Having the measurements available in real-time allows the user to conduct analysis procedures such as adjusting the scale used to colour the measurements as can be seen in Figure 109 helping them to determine which areas are of the most interest for their survey.



Figure 108 - Example of advanced AR approach showing a top down perspective helping the user understand where the source may be located



Figure 109 - Example of advanced AR approach showing a top down perspective helping the user understand where the source may be located with a modified scale to more clearly indicate the highest locations

When the virtual measurements are combined in a fully augmented reality environment the results are more effective. Figure 110 shows two separate angles of a fully augmented reality environment where a virtual radiation source is being measured and viewed from two separate angles. The measurement visuals remain fixed in the environment and the user is able to move fully around in real-time and investigate the scene.



Figure 110 - Example of advanced AR approach showing virtual measurements taken in real-time locked in the environment from two different angles

Figure 111 shows the results of a survey down a hallway viewed from three different perspectives: Top down, 3rd person free camera and first person perspective in full augmented reality. The results show that the analytical freedom provided to the users allowed them the flexibility to analyze and critically view their data from any angle which would support their situational awareness in localizing a radiation source and understanding a complex radiation field distribution.

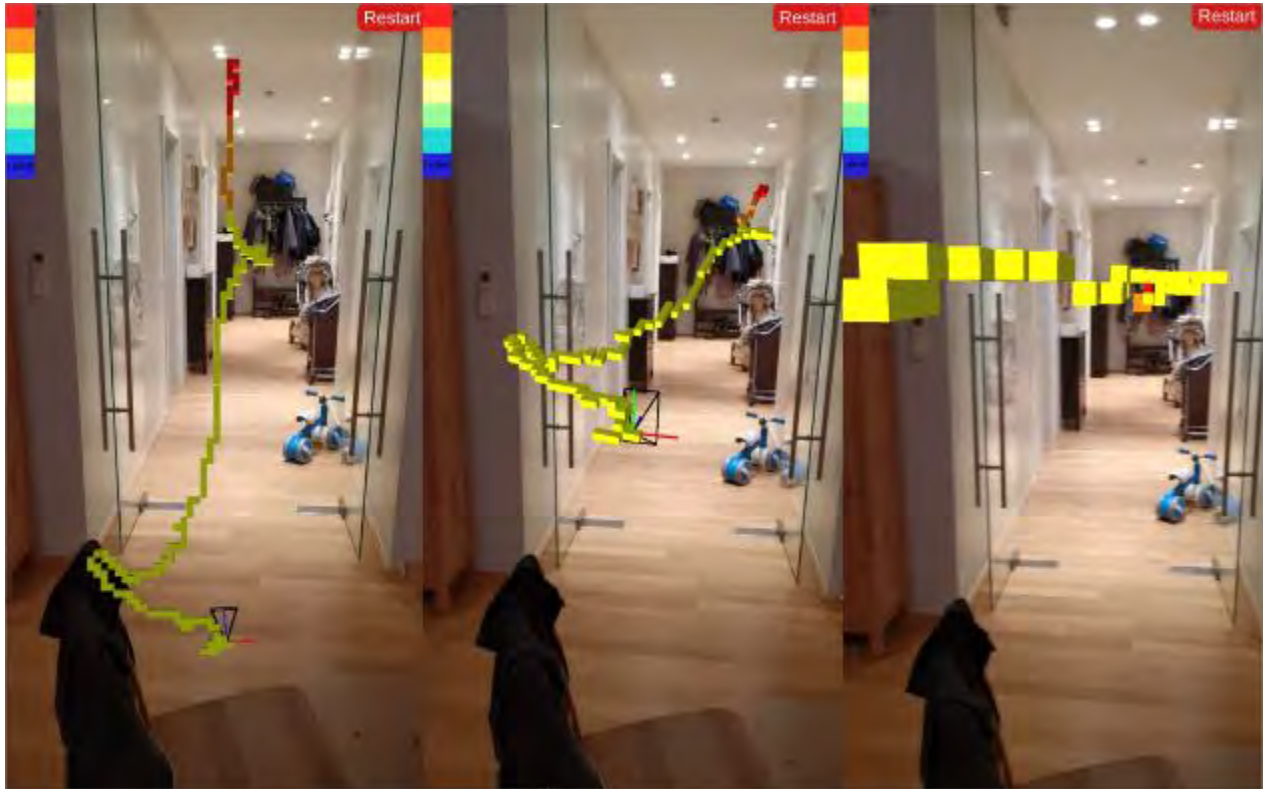


Figure 111 - Example of advanced AR approach showing virtual measurements taken in real-time locked in the environment and viewed from top down, 3rd person and first person perspectives

The results of the advanced augmented reality approach demonstrate how to leverage mobile handheld augmented reality technology to successfully visualize radiation measurements from a virtual source in real-time. This demonstration successful shows that it is possible to use augmented reality to visualized radiation measurements in a number of unique ways including from a first person perspective directly in augmented reality, from top down and third person perspectives, both with and without environmental meshing to support comprehension of where each measurement was taken. This approach was demonstrated on the Google Project Tango tablet hardware,

as well as the two commercially released cellular phones with this technology (Asus Zenfone AR and the Lenovo Phab 2 Pro). In addition, this approach was improved to incorporate Unity's AR foundations and AR core to demonstrate its applicability to a broader range of hardware³⁰ which does not require Google Project Tango specific hardware features.

7.2. Results from the use of augmented reality to visualize radiation measurements

Following the successful demonstration of the visualization of measurements taken during the survey of a simulated source of radiation, the same approach was used but applied during the survey of radiation sources where data was provided via a Bluetooth connected radiation detector. Figure 112 shows the results of a survey in augmented reality from the top-down angle with environmental meshing. As the user navigates the environment the measurement results continue to remain fixed in place with the environmental meshing continuing to provide some context to help the user understand where each measurement was taken.

³⁰ A full list of supported hardware is available here: <https://developers.google.com/ar/devices>. All hardware listed do not support all AR features, which primarily limits the availability of environmental meshing in this approach.



Figure 112 - Example of advanced AR approach showing measurements of three sources from top-down angle with environmental meshing

Figure 113 demonstrates the results of the survey of the three sources of radiation where the user conducts the entire survey in augmented reality. The user is able to view the results from any camera angle they choose, and the measurements remained fixed in place in the environment during the course of the survey.

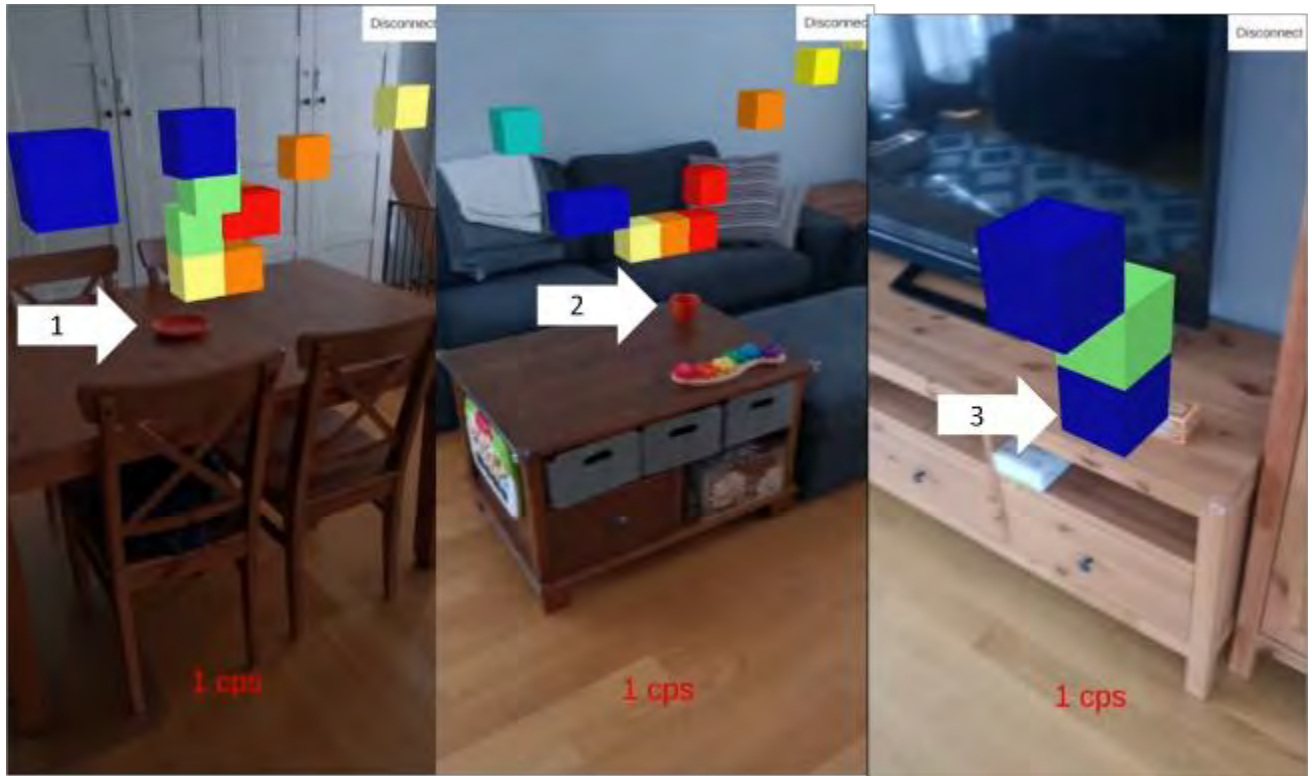


Figure 113 - Example of advanced AR approach showing measurements of three sources from first person perspective

Figure 115 demonstrates the results of the survey of the three radiation sources with environmental meshing turned on and the user viewing the results from a 3rd person perspective. As with the simulated measurements, the user is able to view the results from any angle and zoom in and rescale the coloring of the measurements as needed to support their analysis.

An important aspect of the visualization using this approach is that each measurement is plotted at the location of the detector when it is received by the phone hardware. This means that there is an integration period of a second before the data value is sent to the phone (or more if there are hardware delays in sending the serial data). The

repercussion of this is that the location where the measurement is placed could potentially not represent the full location where the measurement was taken. An exaggerated image depicting this is shown in Figure 114. In practice, this effect was overcome by ensuring the user was moving very slowly through the environment. In practical terms, since the location and movement of the hardware is known, it would also be possible to have an onscreen prompt that warns the user when they are moving too quickly.

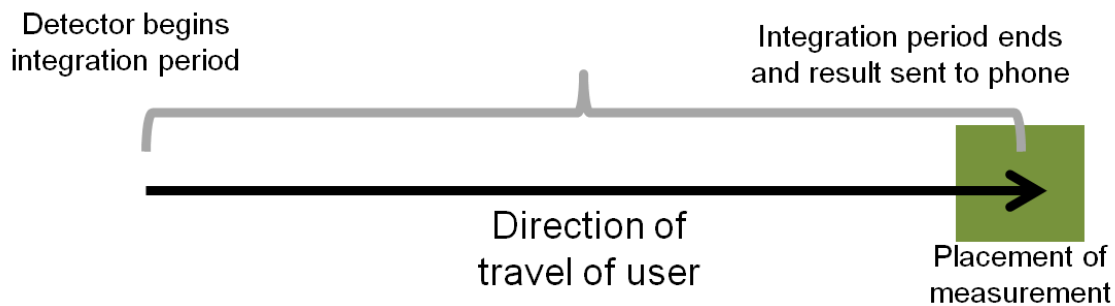


Figure 114 - Example of influence of integration period on measurement placement in augmented reality

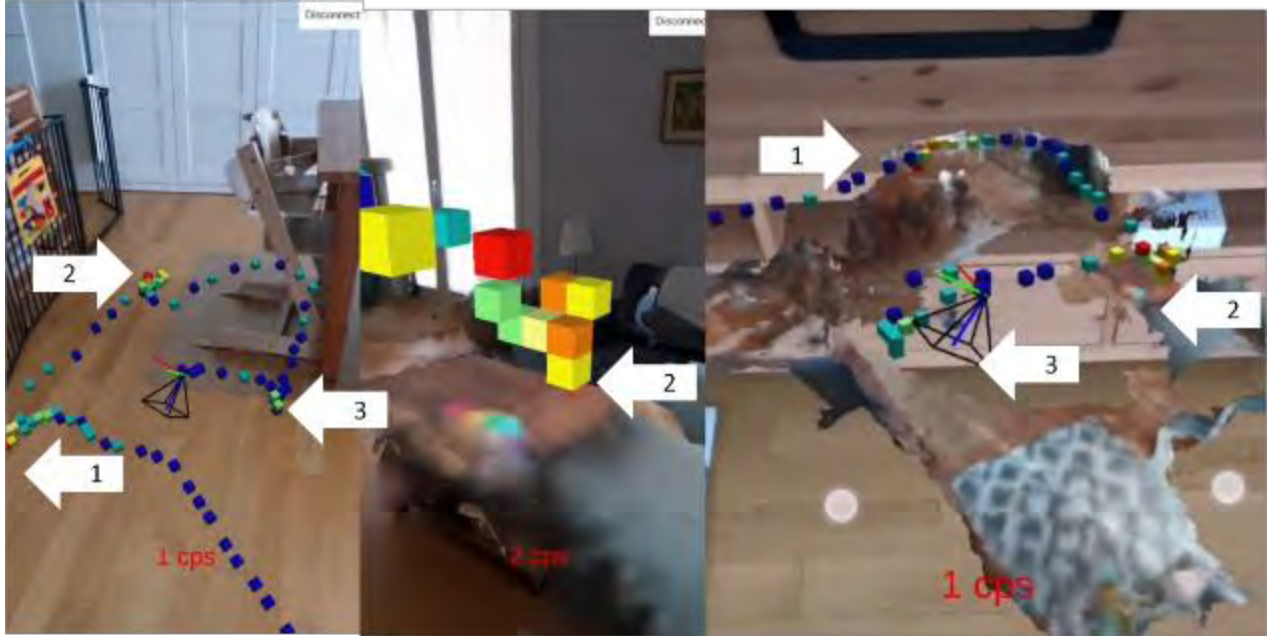


Figure 115 - Example of advanced AR approach showing measurements of three sources from 3rd person perspective with environmental meshing

The results from this approach have successfully demonstrated that this approach is viable for both simulated and live data. In this demonstration, it is shown that data received from a radiation detector can be successfully visualized in augmented reality. The use of different camera angles combined with environmental meshing to help a surveyor better understand where each measurement was taken in the environment is also shown.

Key to the results discussed in the next section is an understanding that using this approach provides the surveyor a real-time dataset of information for each measurement in the form of (x_n, y_n, z_n, V_n) as discussed in 3.1.

7.3. Results from the use of machine learning approach to localizing radiation sources

Following the successful demonstration of visualizing live measurement results in real-time in augmented reality, the use of the reinforcement agent to predict the location of a source based off in situ measurements in augmented reality was investigated. A virtual radiation source was placed in the environment and the user was tasked with taking 10 virtual measurements, simulated with UPK. The user could then at any point press a command on screen which would feed the 10 most recent measurements into the agent and produce an output guess. The guess would then be visually locked in the environment as a white sphere to guide the player in the proper direction. The sphere was visible in first-person, third-person and top-down views. As a check on the results, a line was drawn from the guess to the virtual source (which the agent was not provided any information regarding) to check the accuracy of the resulting guess. Figure 116 provides an example of a guess from the first- and third-person perspectives which was 201 cm away from the location of the source. Figure 117 provides a different example of a guess from the top down, first and third person perspectives, which was 156 cm away from the location of the source in that simulation. Figure 118 provides an example of a different guess seen from the third person perspective which was 220 cm away from the location of the source. Figure 119 and Figure 120 show the results of the same guess which was 144 cm away from the source seen from all three perspectives.

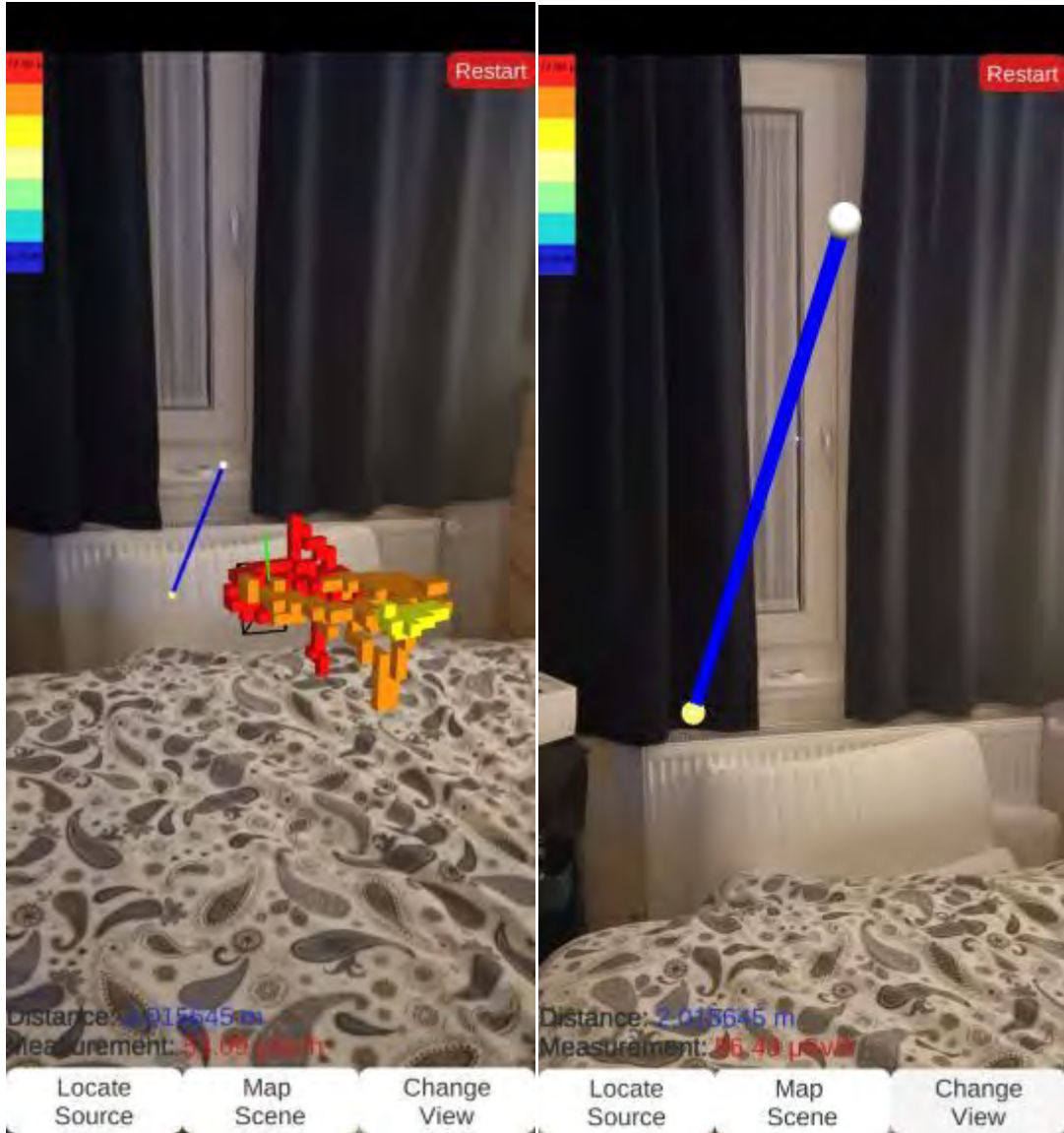


Figure 116 - Third person and first person perspective of measurements and a guess (white sphere) 2.01 m (blue line) from actual source (yellow sphere)

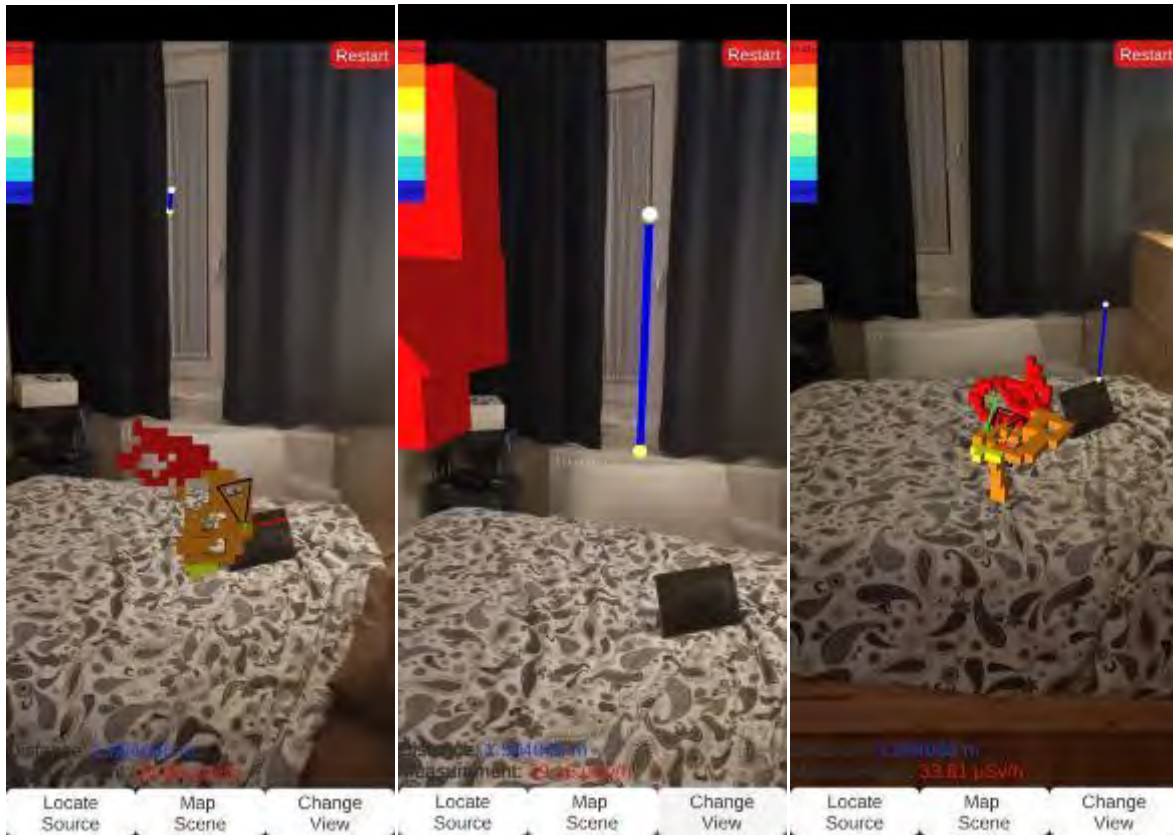


Figure 117 - Third person, Top down and first person perspective of measurements and a guess (white sphere) 1.56 m (blue line) from actual source (yellow sphere)



Figure 118 - Third person perspective of measurements and a guess (white sphere) 2.20 m (blue line) from actual source (yellow sphere)

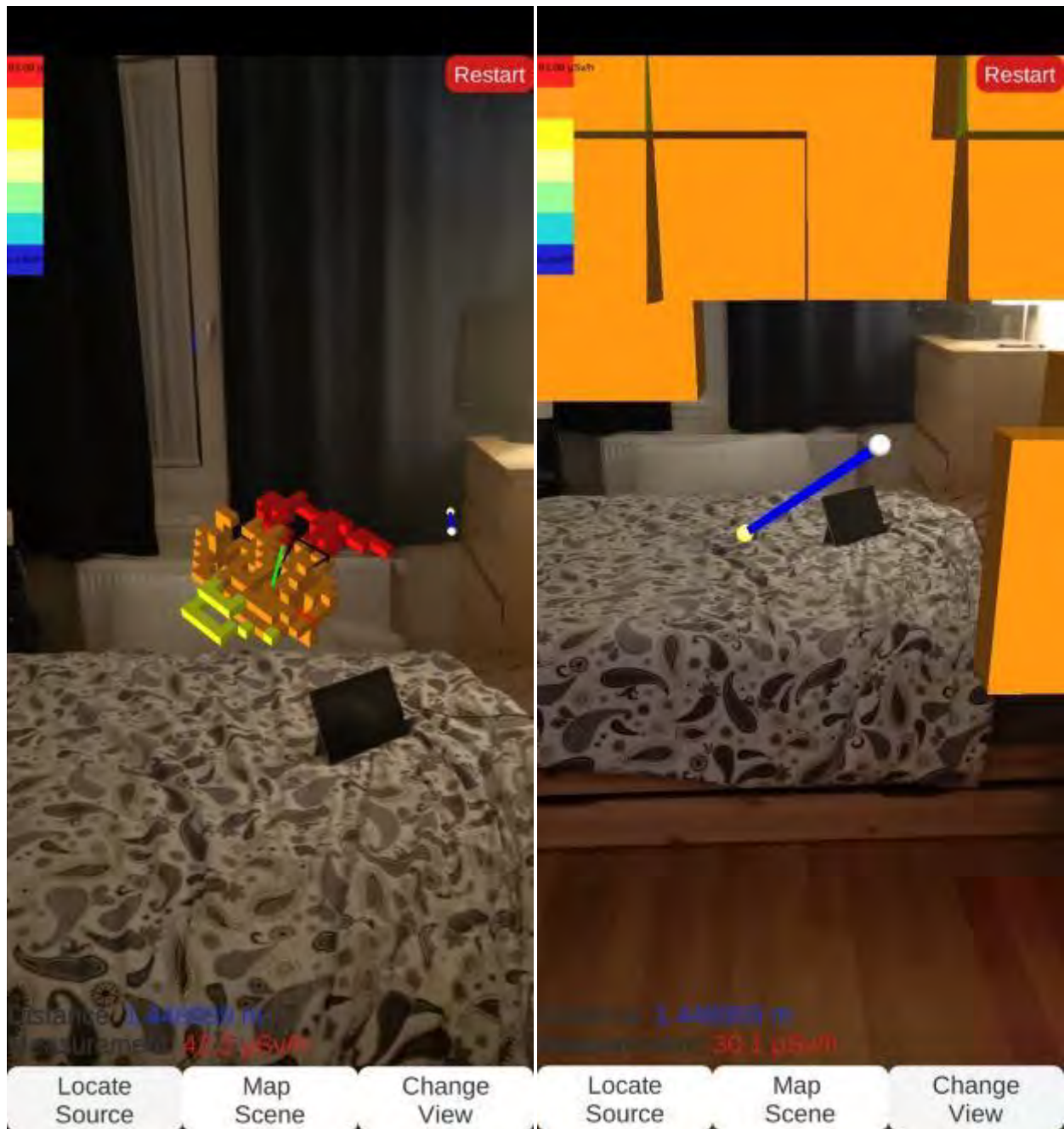


Figure 119 - Third person and first person perspective of measurements and a guess (white sphere) 1.44 m (blue line) from actual source (yellow sphere)

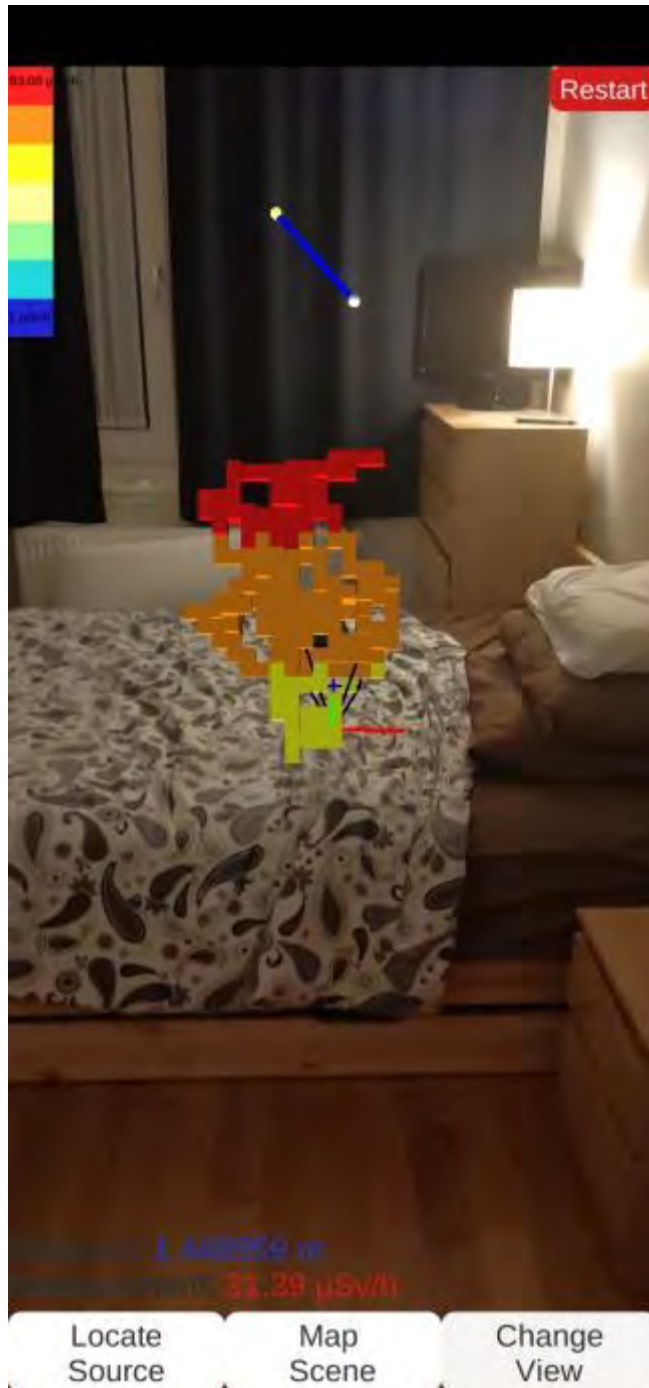


Figure 120 - Third person and first person perspective of measurements and a guess (white sphere) 1.44 m (blue line) from actual source (yellow sphere)

The results of the real-time ML based advice feature were successful. Although this process was capable of providing an estimate to where the source was located in a given space given 10 measurements, the estimate could be in excess of several meters. As the user should have no knowledge of the actual location of the source in a given environment, as was the original intent, the positional advice could have the potential to add further confusion, but this advice, when combined with the situational awareness and training of the surveyor, would be useful in guiding the decisions for additional measurements and areas of interest.

Importantly, the advice could be shown to be within the desired 100 cm range and consistently, with the exception of some outlier guesses which based on the input data could vary. In order to demonstrate the error expected with this, 25000 simulations were run virtually with the trained agent being provided 10 random measurements and its guess source location compared to the actual location recorded. The distance between the agents guess vs. the actual location of the source was counted in steps of 10cm (0-10 cm, 10-20, 20-30cm, etc). The results are shown in Figure 121 and Figure 122 which show an average distance of 98.85 cm for all guesses with a standard deviation of 41.98 cm.

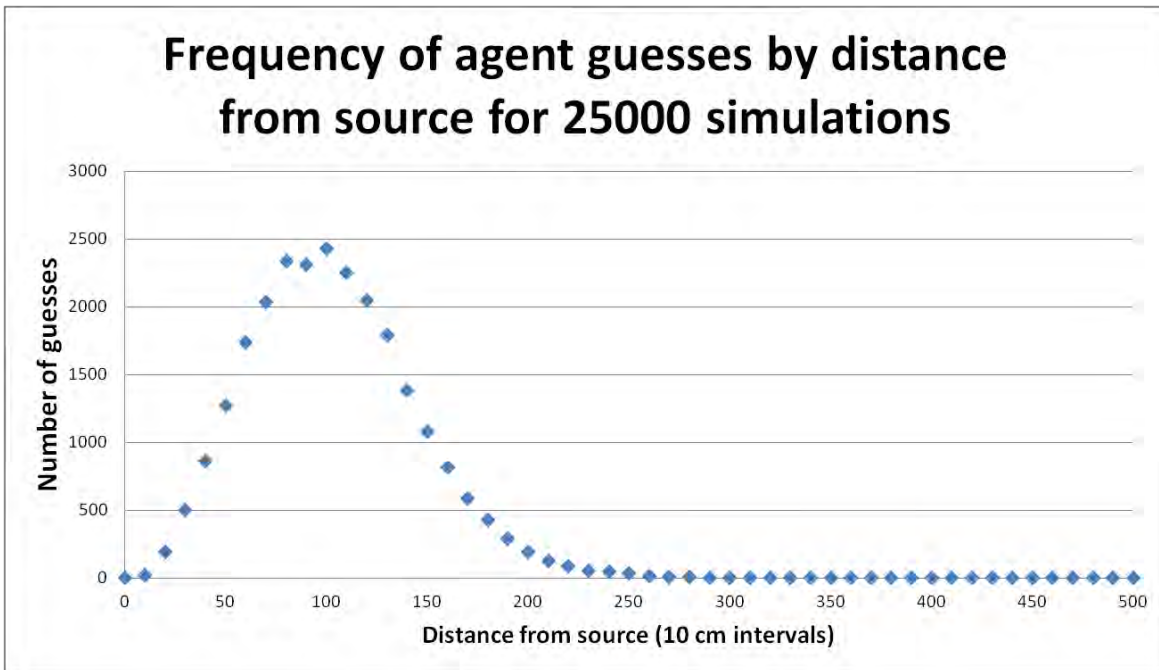


Figure 121 - Distance guessed location from source for an AI agent in 25000 simulations (linear scale)

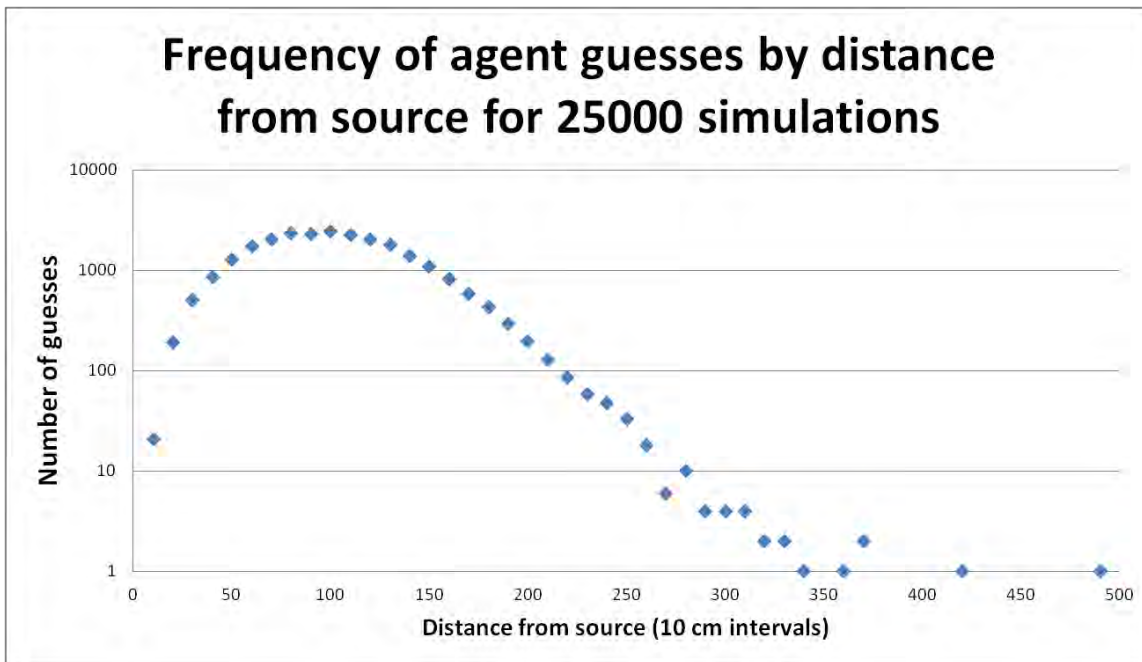


Figure 122 - Distance guessed location from source for an AI agent in 25000 simulations (log scale)

There are avenues to overcome this situation which include changing the training approach or increasing the training time of agent policies that are trending towards a successful outcome. In the experimentation during this research more than 100 separate training approaches were tried with different designs and reward functions. Figure 123 and Figure 124 show the results of 80 of the different training processes plotted by cumulative reward into the training vs. step and computational time.

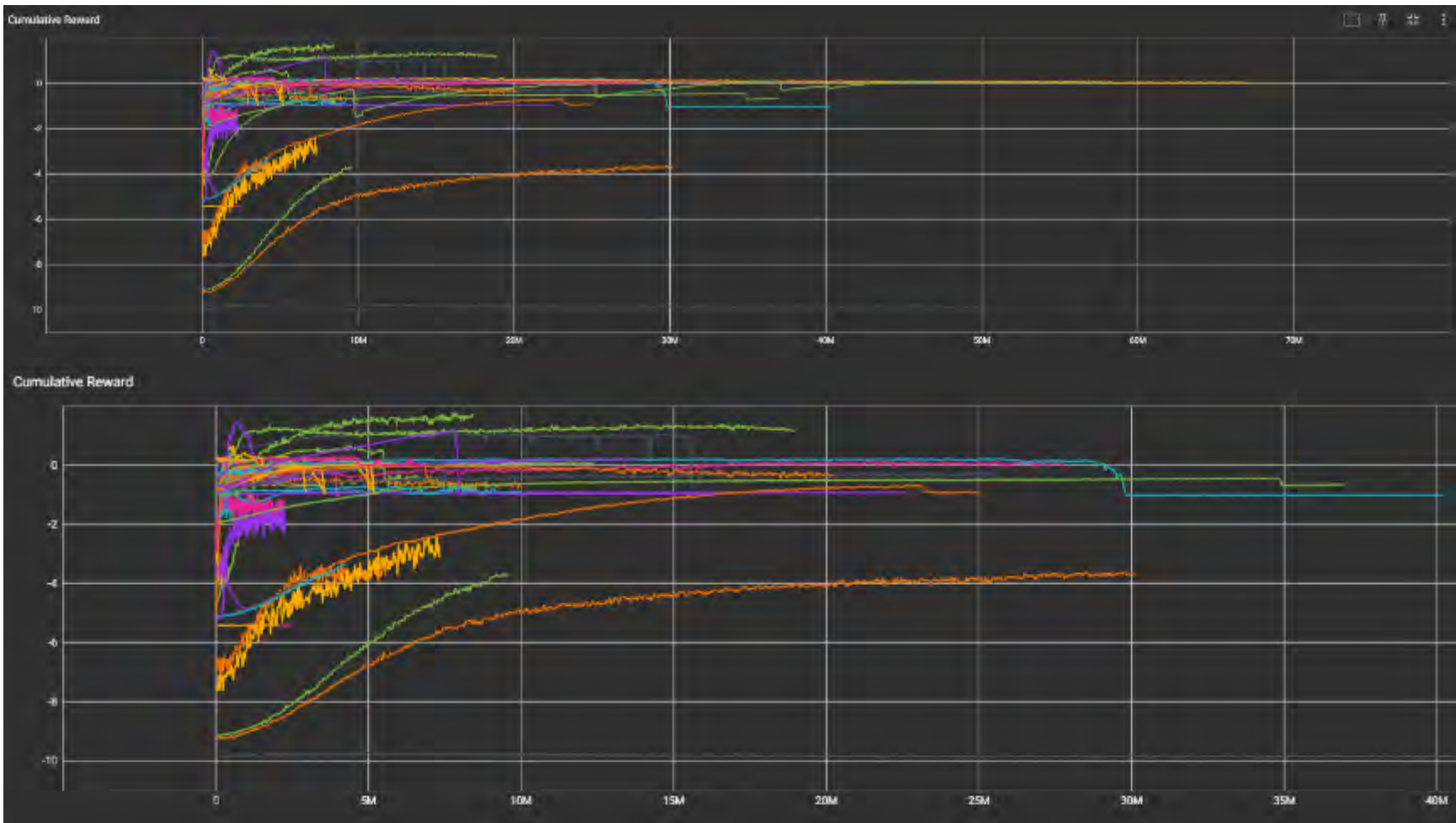


Figure 123 - Example of 80 different training sessions cumulative reward progress compared (note horizontal axis represents steps into the training process)

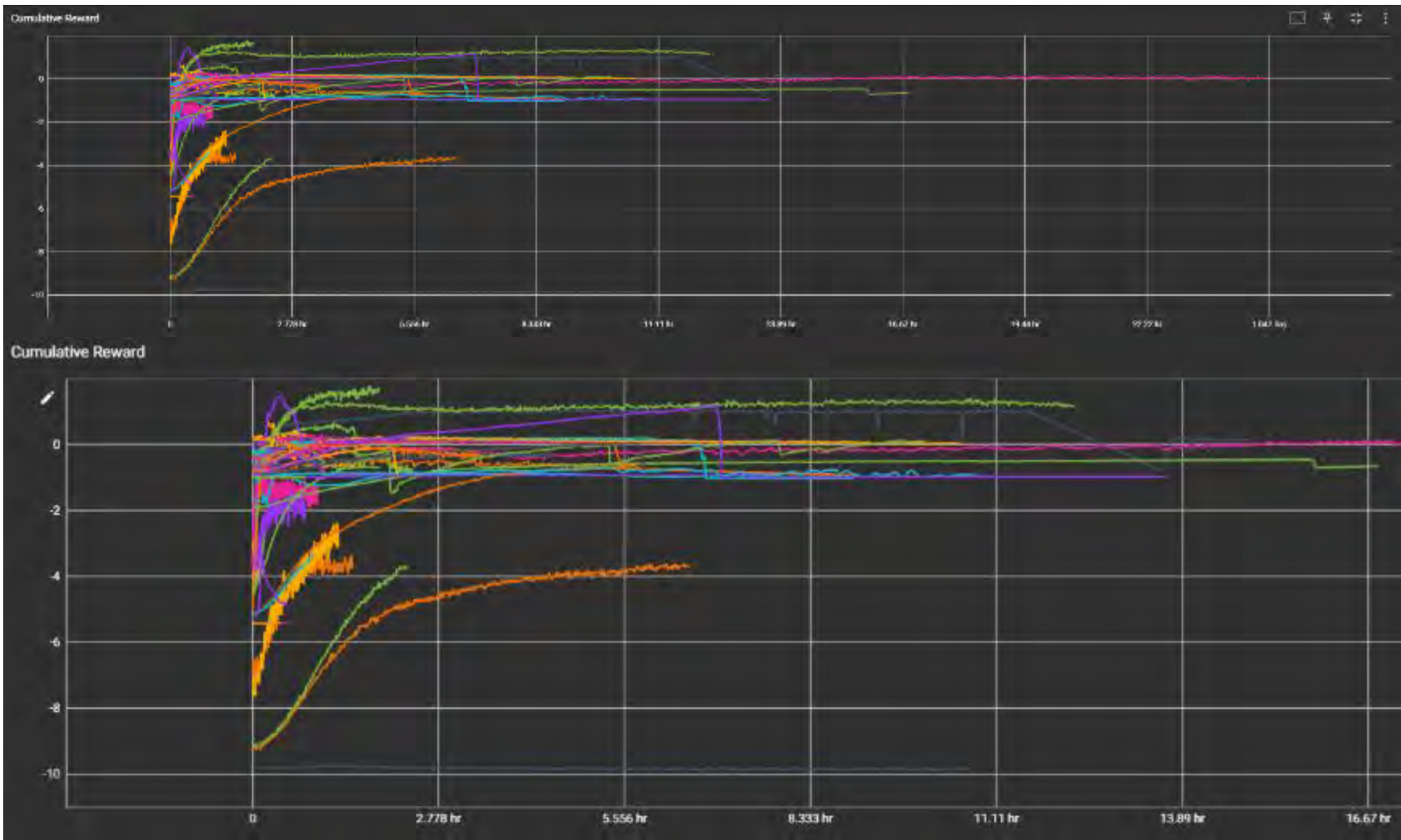


Figure 124 - Example of 80 different training sessions cumulative reward progress compared (note horizontal axis represents total time spent in the training process)

The results of these plots demonstrate that it is possible for the training process to improve over time in its ability to provide a recommendation for the location of the source based on the provided data. While a 'perfect' solution was not found in this experimentation the conceptual solution remains demonstrated as viable and this process could be improved upon and refined.

7.4. Results from the use of virtual reality technology for the conduct of emergency response training

7.4.1. Evolution

The use of the room scale VR scenarios for training was a unique experience when it was conducted. It was established³¹ that the far majority of the participants in this Technical Meeting were complete newcomers to Virtual Reality. The scenarios were very carefully constructed so that the player interactions would increase in complexity as each scenario progressed. The tutorial was designed so that the exercise evaluator could monitor the players comfort and skill operating within the virtual reality environment. Based on their evaluation, when the first scenario (car crash) was loaded, they would choose whether to question them on only what they could see visually from their location, other if they wanted to have them move to more complex viewing locations.

³¹ This was conducted by a verbal question during the opening of the meeting which no participants responded that they had attempted virtual reality previously

7.4.2. Feedback

The scenarios described in 6.3 were provided as a public demonstration during an IAEA meeting in October 2017 involving 62 emergency response and transport safety experts evaluating a new safety guide on transport related emergencies³². During the meeting two stations were setup where participants were provided the opportunity to go through as many of the scenarios as they were comfortable trying. In addition, interest was seen from IAEA staff members and other visiting professional that were given the opportunity to try the scenarios at the same time. Over the week long exhibition, more than 100 people attempted at least one of the scenarios. For those experts who were from the transport safety guide meeting, 46 were willing to answer some brief questionnaire³³ questions and provide general feedback that was recorded by the exercise controllers. Informal feedback was also captured from the IAEA staff members and other visiting professionals but their results were not included in the surveys. Questionnaire items were asked before and after the training scenarios were conducted. The following questions were asked to the participants with their response options being strongly agree, agree, neutral, disagree and strongly disagree (Likert scale).

- Questions pre-conduct of the VR training scenarios
 - I see value in VR for nuclear safety training
 - I have experience with virtual reality technology

³² A press article describing the event is available online: <https://www.iaea.org/newscenter/news/iaea-closer-to-finalising-new-safety-guide-for-nuclear-transport-emergencies>

³³ A copy of the survey is provided in Annex VI

- I have experience with virtual reality for nuclear applications
- I am a video game player
- I follow the latest technological developments
- Questions post-conduct of the VR training scenarios
 - I see value in VR for nuclear safety training
 - I could easily understand the scenario concepts
 - I found the VR scenarios realistic
 - I felt comfortable in the VR environment
 - I was not disoriented and/or did not feel dizzy while in VR
 - The length of time for the scenarios was sufficient
 - I would recommend VR as an option for my training organization

The results of the questions pre-conduct of the exercises are available in Figure 125, Figure 126, Figure 127, Figure 128 and Figure 129, and the results of the questions post-conduct of the exercises are available in figures Figure 130, Figure 131, Figure 132, Figure 133, Figure 134, Figure 135 and Figure 136.

I see value in VR for nuclear safety training



Figure 125 - Survey results question 1 (n=46) (pre-conduct)

I have experience with VR technology



Figure 126 - Survey results question 2 (n=46) (pre-conduct)

I have experience with VR for nuclear applications



Figure 127 - Survey results question 3 (n=46) (pre-conduct)

I am a video game player



Figure 128 - Survey results question 4 (n=46) (pre-conduct)

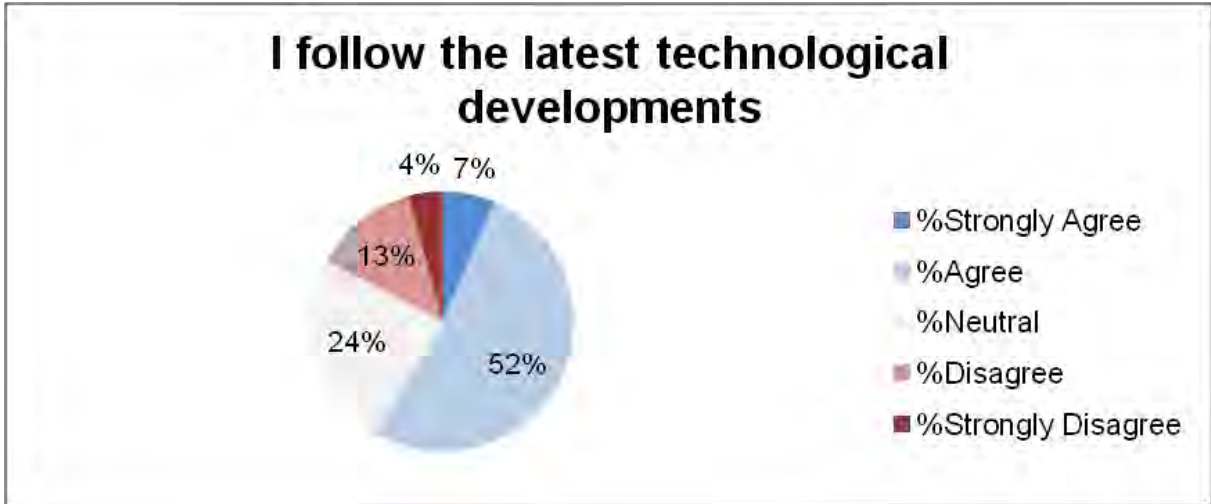


Figure 129 - Survey results question 5 (n=46) (pre-conduct)

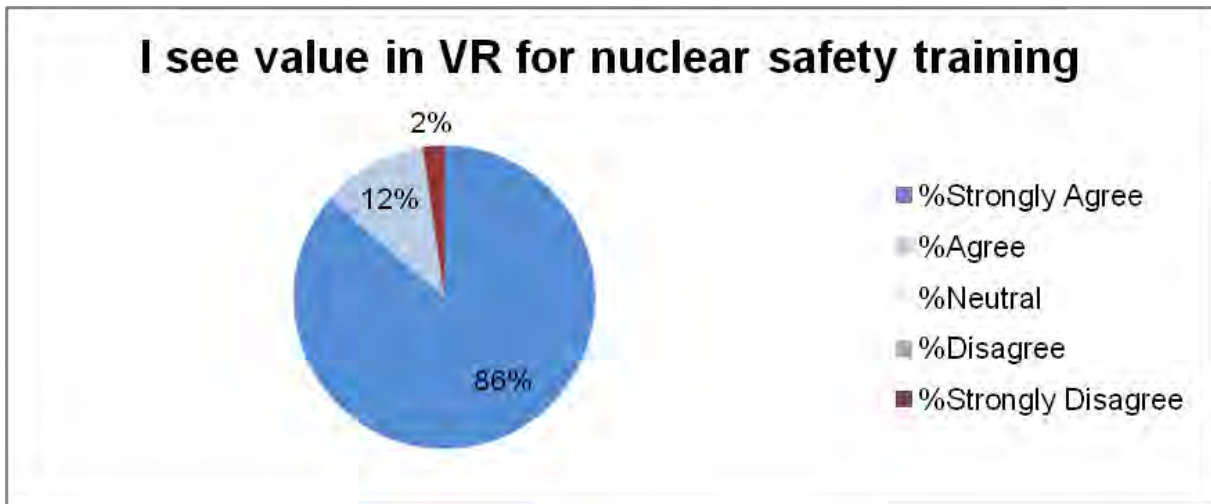


Figure 130 - Survey results question 6 (n=44) (post-conduct)

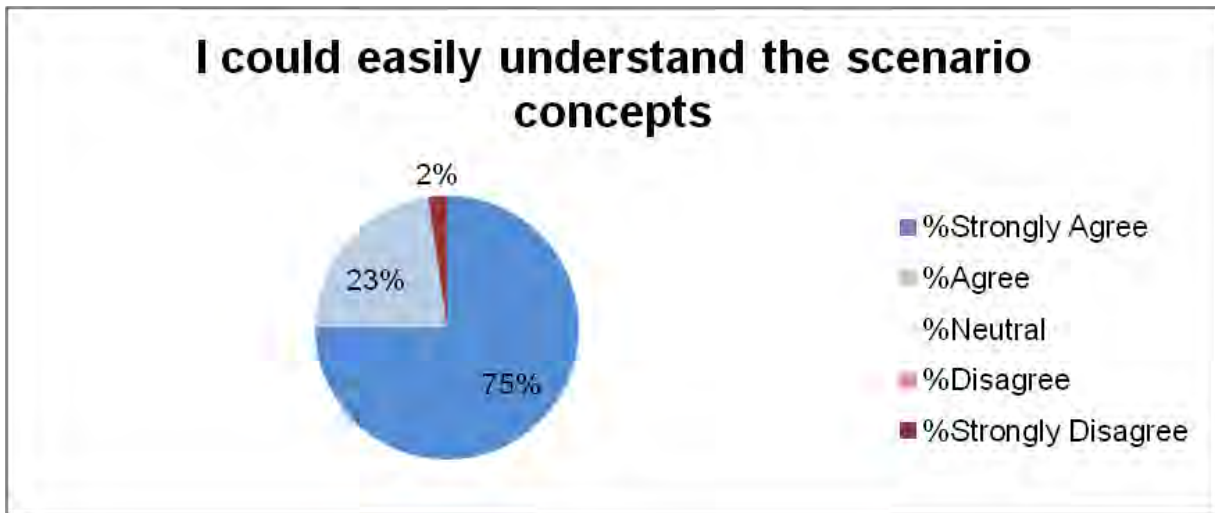


Figure 131 - Survey results question 7 (n=44) (post-conduct)

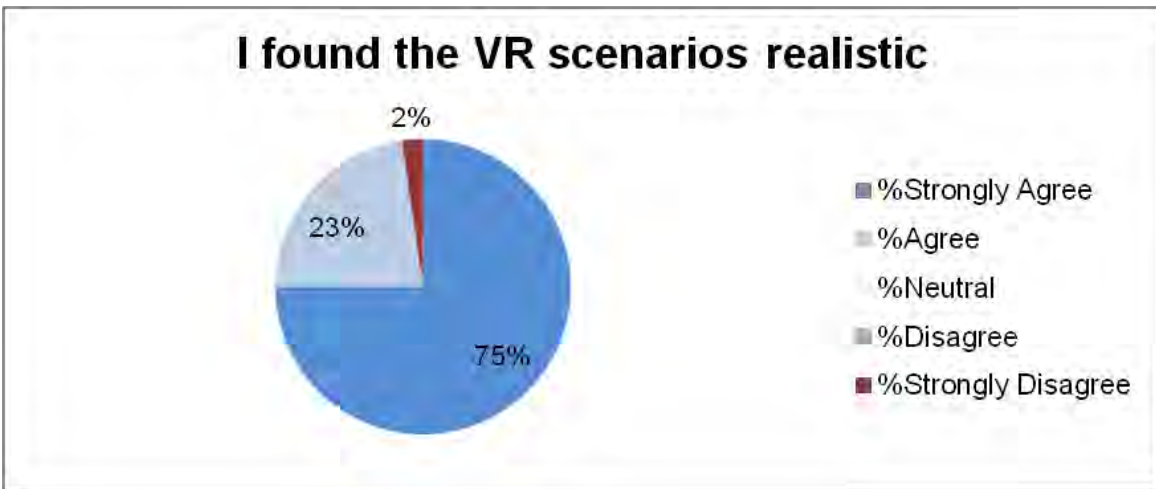


Figure 132 - Survey results question 8 (n=44) (post-conduct)

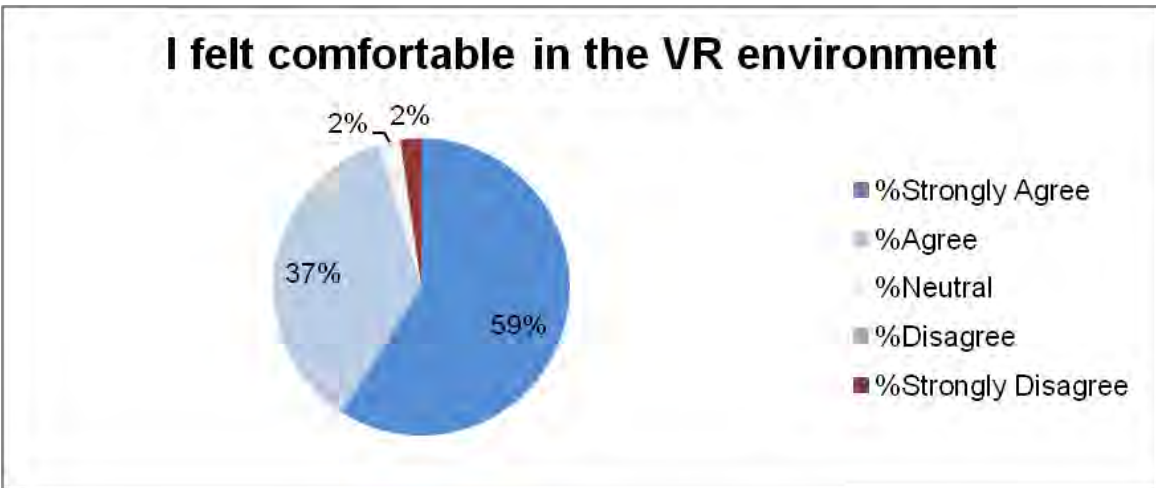


Figure 133 - Survey results question 9 (n=44) (post-conduct)

I was not disoriented and/or did not feel dizzy while in VR



Figure 134 - Survey results question 10 (n=44) (post-conduct)

The length of time for the scenarios was sufficient



Figure 135 - Survey results question 11 (n=44) (post-conduct)



Figure 136 - Survey results question 12 (n=44) (post-conduct)

In addition, of the people that tried all 4 scenarios AND reported the one they found most compelling ($25/46=54.3\%$), (3/25)12% chose scenario 1 (car crash), (7/25) 28% chose #2, (12/25) 48% chose #3, and (3/25) 12% chose #4 as can be seen in Figure 137.

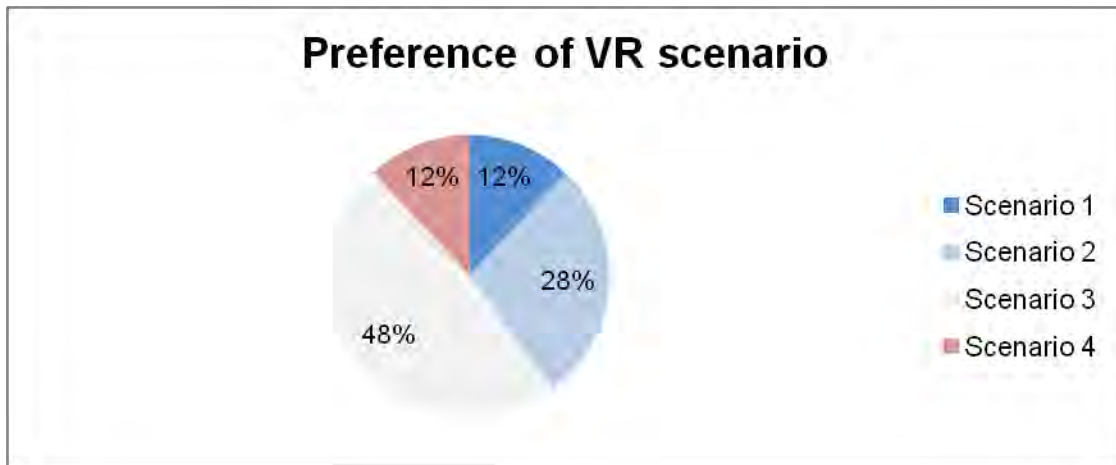


Figure 137 - Survey results preference of VR scenario (n=25)

Of the people who completed both the pre and post VR survey question "I see value in VR for nuclear safety training" (44/46=95.7%), following the demonstration: (15/44) 34.1% saw an improvement of their opinion of VR, (28/44) 63.6% didn't change their opinion, and (1/44) 2.3% had a reduced opinion of VR.

The results show good opinions of the VR training after the user's experience. It was seen Figure 126 that the majority of players had little experience with VR technology prior to the exercises and the vast majority did not have experience with VR for nuclear applications as seen in Figure 127. After the conduct of the exercises the majority felt strongly that there was value in the use of VR for training as seen in Figure 130 and the majority also strongly agreeing that the scenario concepts were easily understood and appropriately realistic as seen in Figure 131 and Figure 132. Overall it can be seen that the participants believed that this type of training should continue to be available and used in the future.

The only negative opinion of the scenarios was a single participant who was not comfortable throughout the exercise conduct. They verbally shared with the organizers that they were uneasy standing without seeing their body. They were provided a chair to allow them to sit (which 3 other attendees also used as an accommodation for their comfort), but even with the chair they were unable to feel completely comfortable with the experience. They were still interested in completing and trying each scenario and their verbal feedback related to their comfort was very useful in the creation of future exercise scenarios outside of this activity. A significant number of additional lessons were learned during the deployment of the VR scenarios for this event and subsequent activities between 2016 and 2020. These lessons learned are subjective in nature and

were taken from observations from controllers of the participants and their verbal feedback. They have been captured in an Annex.

7.5. Specific considerations for the use of VR technology under the circumstances of the Covid-19 pandemic

In late 2019 and very early 2020 the world slowly became aware of the threat posed from the Covid-19 virus. This has had a significant impact on society, economies and has resulted in a loss of life which as of the writing of this thesis has been estimated to exceed 6,200,000 lives³⁴. The use of VR hardware for deployment in a manner consistent with the methodology and for the applications described requires the use of a physical headset to display the environment to the user and handheld controllers for the user to interact with the environment. Unfortunately, with the Covid-19 virus in the current circumstances, these represent a very real contamination vector for the spread of the virus. As the world is currently coming to grips with how this virus continues to impact the lives of people during two years of this ongoing pandemic, its influence on the use of VR technology for education cannot be ignored. Given the risk for spreading the virus between participants sharing equipment, it is not recommended for this type of training to be run in the manner previously described where participants are sharing physical headset and controller hardware unless measures are taken to ensure all participants are vaccinated and test negative for a breakthrough case of the virus prior to the training.

³⁴ The actual number reference was taken from what was available from <https://www.worldometers.info/coronavirus/> as of 3 April 2022

However, there are considerations and practical solutions which can negate the potential for the spread of the Covid-19 virus. Providing each student, their own VR equipment which becomes part of their regular home/office equipment may be the most practical as the cost of newer VR headsets, such as the Oculus Quest 2 are less than \$500(CDN). This opens up the avenue for training in interactive and realistic environments while users are working from home in multiplayer scenarios, helping to increase the quality of their training if they are forced into lock down isolations to control the spread of the virus.

CHAPTER 8: CONCLUSIONS AND RECOMMENDATIONS

8.1. Conclusions

This research has demonstrated a methodology for the viable *in situ* visualization of radiation measurements in augmented reality. This approach can form the basis for new tools for radiation surveyors to increase their operational efficiency, hazard recognition and source localization skills, and overall increase safety. It was shown that the data gained when *in situ* measurements are taken can be leveraged as input into modern artificial intelligence based reinforcement learning methods to provide support for a user to localize a radiation source in the environment directly on the hardware they are using for visualizing the radiation measurements. Finally, this research demonstrated that the modelling approach developed has compatibility as a useful tool for building virtual reality training scenarios to enhance and improve the capabilities of first responders and other trainees as well as opening up the possibility of using the augmented reality platform as a tool for measuring virtual sources of radiation.

The augmented reality visualization research was demonstrated using two approaches. The first approach involved a primitive environmental placement technique leveraging the Layer Augmented Reality web browser. This approach demonstrated that it was possible to visualize this primitive visual of a radiation field, but it also was clearly limited and unable to keep the visuals 'fixed' into the environment very well. The second approach demonstrated using the Google Project Tango and eventually AR Foundations with AR Core demonstrated a much more capable and robust approach that is able to be deployed to a wide set of cellular phones. This method showed it was

possible to both simulate the measurement of virtual radiation sources in an augmented reality environment and also that when connected via blue tooth to a radiation detector it was possible to bring live measurement data into augmented reality as a tool for a surveyor to real-time visualize and analyze their survey results. The addition of multiple camera angles and 3d meshing of the environment further demonstrated added enhancements to increase the capabilities of users to interrogated the collected data in real-time.

The reinforcement learning based approach to source localization was shown using the training methods described to be able to localize a virtual radiation source within a ~1m average error region space. Importantly, this solution demonstrated that there is additional value not just in visualizing radiation measurements in augmented reality for situational awareness but also in providing a real-time advice capability for users based on trained AI agents. The reinforcement learning approach was based on a generic processes where the AI agent being trained and the algorithms being used were one specifically designed for radiation protection purposes, rather they were designed for solving generic problems. The radiation protection problem was turned into a 'game' which the agent could then be trained to solve. This type of solution has significant potential for applications to be built solving much more advanced radiation protection problems which could also be 'gamified' in a similar manner.

The approach to providing training in virtual reality was demonstrated as successful and very well received by the majority of users based on their feedback. This new approach for training radiation protection and emergency preparedness and response to students allows for the creation of complex environments and situations not always possible to

be provided due to cost, danger to students or unavailability of equipment. In addition, the ability for this virtual reality equipment to be picked up and transported to any classroom environment and its use for creating mixed reality footage provides versatility that is unique to the virtual reality teaching medium.

At the onset, this research established a need for the nuclear industry to always be looking for new methodologies and tools to increase a radiation surveyor's situational awareness during the conduct of a survey and increase their overall skill set. Adhering to the principals of ALARA requires providing the best training possible to potential radiation surveyors to develop their skills prior to any potential exposure and to provide tools to support situational awareness, where possible. The research conducted and the approaches developed directly addressed those issues from both directions and in time, have the potential to become standard elements in any radiation surveyors toolkit.

8.2. Recommendations for future research

The approach developed and demonstrated with UPK was simply one of many alternative approaches that could have been leveraged[138]. UPK itself could be further optimized by converting its approach into Unity's Data Oriented Technology Stack (DOTS) which would offer increased speed in its processing allowing for many more sources and shields in a given environment. The physics of UPK could be improved to take into account build up factors for different materials. Accounting for short range alpha and beta radiation in an environment would be another potential evolution of this approach as the detectors orientation is always known in the virtual environment and the distance to any contaminated surfaces or sources could be computed relatively easily.

The visualization approach for augmented reality could be further investigated. Issues such as how to better handle the detector integration with respect to the placement of the environmental representation of the measurement could improve the overall analysis of the survey data for the user. How to visualize and mix measurements where the detected radiation particle type (alpha, beta, gamma, neutron etc) is determined or elements such as how to visualize the photon energy in a way to help a user during a survey could be helpful in refining this approach for use in complex environments. An investigation into the benefits of incorporating the augmented reality display technology directly into the radiation detector, so that an external display is not needed, could prove useful for developing an all-in-one form factor.

The machine learning solution demonstrated for localizing radiation sources could be expanded upon in many different ways beyond what was shown in this research. The premise of this research was for scenarios with a single source in an optimal environment without any shielding material. The training environment could be dramatically increased in complexity, adding additional sources and semi-random shielding configurations to stress the agents understanding of the environment and allow it to provide more complex advice. Additional information could be given to the agent, such as the depth profile from the phones camera available in Unity from AR Core which would help the agent to understand the scene in the context of each reading captured and the direction of the viewpoint of the user. Optimizing the number of measurements given to the agent including allowing the agent 'to choose' which measurements should be considered (possibly filtering out background or outlier

information) would be another avenue for investigation to improve the capabilities of the agent.

The virtual reality training could be improved with the addition of more complex environments and more advanced scenes. Adding features such as multiplayer environments would facilitate cooperative problem-solving, allowing participants to interact with each other remotely. Increasing the available set of tools for the construction of a scene, such as adding more common first responder resources such as chemical sensors, forced entry tools, or communication tools, would enhance the realism and scenario options.

References

- [1] “International Atomic Energy Agency Safety Standards.”
<https://www.iaea.org/resources/safety-standards> (accessed Dec. 15, 2020).
- [2] “Canadian Nuclear Safety Commission - List of regulations.”
<http://nuclearsafety.gc.ca/eng/acts-and-regulations/regulations/> (accessed Dec. 15, 2020).
- [3] “Safety of nuclear power plants: Design IAEA Safety Series No. SSR-2/1 (Rev. 1),” Vienna, 2016. [Online]. Available: <https://www-pub.iaea.org/MTCD/publications/PDF/Pub1715web-46541668.pdf>.
- [4] “International Atomic Energy Agency Nuclear and radiological accident reports.”
<https://www.iaea.org/topics/accident-reports> (accessed Dec. 15, 2020).
- [5] “Lessons learned from the response to radiation emergencies (1945-2010), EPR-Lessons Learned 2012,” 2012. [Online]. Available: [https://www-pub.iaea.org/MTCD/Publications/PDF/EPR-Lessons learned 2012_web.pdf](https://www-pub.iaea.org/MTCD/Publications/PDF/EPR-Lessons%20learned%202012_web.pdf).
- [6] “United Nations Scientific Committee on the Effects of Atomic Radiation 2000 Report to the General Assembly, with scientific annexes Vol. II Annex J: Exposures and effects of the Chernobyl accident,” 2000. [Online]. Available: <https://www.unscear.org/docs/reports/annexj.pdf>.
- [7] “The Fukushima Daiichi Accident a report of the Director General,” 2015. [Online]. Available: <https://www.iaea.org/publications/10962/the-fukushima-daiichi-accident>.

- [8] "The radiological accident in Goiania," 1988. [Online]. Available: https://www-pub.iaea.org/MTCD/Publications/PDF/Pub815_web.pdf.
- [9] W. K. Gummer, F. R. Campbell, G. B. Knight, and J. L. Ricard, "Cosmos 954 The occurrence and nature of recovered debris, Atomic Energy Control Board," 1980. [Online]. Available: https://inis.iaea.org/collection/NCLCollectionStore/_Public/12/595/12595268.pdf.
- [10] "The Radiological Accident in the Reprocessing Plant at Tomsk," 1998. [Online]. Available: <https://www.iaea.org/publications/4740/the-radiological-accident-in-the-reprocessing-plant-at-tomsk>.
- [11] "The Radiological Accident in Lilo," 2000. [Online]. Available: <https://www.iaea.org/publications/5968/the-radiological-accident-in-lilo>.
- [12] "Report on the Preliminary Fact Finding Mission Following the Accident at the Nuclear Fuel Processing Facility in Tokaimura, Japan," 1999. [Online]. Available: <https://www.iaea.org/publications/5957/report-on-the-preliminary-fact-finding-mission-following-the-accident-at-the-nuclear-fuel-processing-facility-in-tokaimura-japan>.
- [13] "The Radiological Accident in Chilca," 2018. [Online]. Available: <https://www.iaea.org/publications/11095/the-radiological-accident-in-chilca>.
- [14] "The Radiological Accident at the Irradiation Facility in Nesvizh," 1996. [Online]. Available: <https://www.iaea.org/publications/4712/the-radiological-accident-at-the-irradiation-facility-in-nesvizh>.
- [15] "Nuclear Power Plant Personnel Training and its Evaluation a Guidebook,

- Technical Report Series No. 380,” 1996. [Online]. Available: https://www-pub.iaea.org/MTCD/Publications/PDF/trs380_web.pdf.
- [16] “Manual for First Responders to a Radiological Emergency, EPR-First Responders,” 2006. [Online]. Available: http://www-pub.iaea.org/MTCD/Publications/PDF/EPR_FirstResponder_web.pdf.
- [17] “IAEA Safety Standard Series No. GSR Part 7, Preparedness and Response for a Nuclear or Radiological Emergency, General Safety Requirements,” 2015. [Online]. Available: https://www-pub.iaea.org/MTCD/Publications/PDF/P_1708_web.pdf.
- [18] R. J. Ingram, “Emergency Response to Radiological Releases,” *Health Phys.*, vol. 114, no. 2, pp. 208–213, Feb. 2018, doi: 10.1097/HP.0000000000000757.
- [19] I. A. E. Agency, “Peer Appraisal of The Arrangements In Canada Regarding Preparedness And Response For A Nuclear Or Radiological Emergency,” 2019.
- [20] M. Royal, H. Colbert, and D. Gamble, “Canadian Next Generation First Responder Initiative: Preliminary Capability Assessment,” 2017. [Online]. Available: https://cradpdf.drdc-rddc.gc.ca/PDFS/unc274/p805347_A1b.pdf.
- [21] “Multi-agency radiation survey and site investigation manual (MARSIM). Final report,” Oak Ridge, TN, Dec. 2000. doi: 10.2172/569107.
- [22] “Actions to Protect the Public in an Emergency due to Severe Conditions at a Light Water Reactor, EPR-NPP-PPA 2013,” 2013. [Online]. Available: http://www-pub.iaea.org/MTCD/Publications/PDF/EPR-NPP_PPA_web.pdf.
- [23] J. Chaput, “A Generic Methodology for the 3-Dimensional Visualization of

- Radiation Fields,” University of Ontario Institute of Technology, 2010.
- [24] C. J. and E. Waller, “A Generic Methodology for the Three Dimensional Visualization of Radiation Fields,” 2010.
- [25] J. G. Paradis and Z. M. L., *The MIT guide to science and engineering communication 2nd edition*. MIT Press, 2002.
- [26] K. J. D. and S. K., *Handbook of Visual Communication Theory, Methods, and Media*. Routledge, 2020.
- [27] “Air Dose Rates in the Buildings, Tokyo Electric Power Company,” 2013. [Online]. Available: <http://www.tepco.co.jp/en/nu/fukushima-np/f1/images/f1-sv3-20130322-e.pdf>.
- [28] “Development of Remote Decontamination Technology in the Reactor Building” Robot Investigation Results of Unit 1 Reactor Building, Tokyo Electric Power Company,” 2012. [Online]. Available: https://www.tepco.co.jp/en/nu/fukushima-np/images/handouts_120525_06-e.pdf.
- [29] “Survey Map of the Entire Fukushima Daiichi Nuclear Power Station (May 7-10, 2013), Tokyo Electric Power Company.” [Online]. Available: <https://www.tepco.co.jp/en/nu/fukushima-np/f1/images/f1-sv2-20130521-e.pdf>.
- [30] “Extension Site of Distribution Map of Radiation Dose, etc./GSI Maps,” *Radioactivity Measurement and Investigation Project, The Ministry of Education, Culture, Sports, Science and Technology, Japan*, 2020. <https://ramap.jmc.or.jp/map/eng/about.html> (accessed Dec. 20, 2020).
- [31] K. A. Pradeep Kumar, G. A. Shanmugha Sundaram, B. K. Sharma, S. Venkatesh,

- and R. Thiruvengadathan, "Advances in gamma radiation detection systems for emergency radiation monitoring," *Nucl. Eng. Technol.*, vol. 52, no. 10, pp. 2151–2161, 2020, doi: <https://doi.org/10.1016/j.net.2020.03.014>.
- [32] "Guidelines for radioelement mapping using gamma ray spectrometry data, TecDoc-1363," 2003. [Online]. Available: https://www-pub.iaea.org/mtcd/publications/pdf/te_1363_web.pdf.
- [33] G. A. Drukier, E. P. Rubenstein, P. R. Solomon, M. A. Wojtowicz, and M. A. Serio, "Low cost, pervasive detection of radiation threats," in *2011 IEEE International Conference on Technologies for Homeland Security (HST)*, Nov. 2011, pp. 365–371, doi: 10.1109/THS.2011.6107897.
- [34] "Image Insight Inc. maker of GammapiX software." <https://www.imageinsightinc.com/> (accessed Dec. 15, 2020).
- [35] F. Sato *et al.*, "Visualization of high radiation field by radiophotoluminescence photography," *Radiat. Meas.*, vol. 68, pp. 23–30, Sep. 2014, doi: 10.1016/j.radmeas.2014.06.011.
- [36] E. B. Farfán *et al.*, "Locating Radiation Hazards and Sources within Contaminated Areas by Implementing a Reverse Ray Tracing Technique in the RadBall™ Technology," *Health Phys.*, vol. 102, no. 2, pp. 196–207, Feb. 2012, doi: 10.1097/HP.0b013e3182348c0a.
- [37] J. Zhou and K. Tönnies, "State of The Art for Volume Rendering," [Online]. Available: https://www.researchgate.net/publication/238687132_1_State_of_The_Art_for_Vo

lume_Rendering.

- [38] K. Vetter, R. Barnowski, A. Haefner, T. H. Y. Joshi, R. Pavlovsky, and B. J. Quiter, "Gamma-Ray imaging for nuclear security and safety: Towards 3-D gamma-ray vision," *Nucl. Instruments Methods Phys. Res. Sect. A Accel. Spectrometers, Detect. Assoc. Equip.*, vol. 878, pp. 159–168, Jan. 2018, doi: 10.1016/j.nima.2017.08.040.
- [39] O. Ivanov, A. Danilovich, V. Stepanov, S. Smirnov, and A. Volkovich, "Visualization of Radioactive Sources Without Gamma-Radiation With UV Imaging Systems," in *ASME 2009 12th International Conference on Environmental Remediation and Radioactive Waste Management, Volume 2*, Jan. 2009, pp. 321–325, doi: 10.1115/ICEM2009-16145.
- [40] F. Cosentino, N. W. John, and J. Vaarkamp, "An overview of augmented and virtual reality applications in radiotherapy and future developments enabled by modern tablet devices," *J. Radiother. Pract.*, vol. 13, no. 3, pp. 350–364, Sep. 2014, doi: 10.1017/S1460396913000277.
- [41] W. Li, A. Nee, and S. Ong, "A State-of-the-Art Review of Augmented Reality in Engineering Analysis and Simulation," *Multimodal Technol. Interact.*, vol. 1, no. 3, p. 17, Sep. 2017, doi: 10.3390/mti1030017.
- [42] A. Drøivoldsmo *et al.*, "Use of Radiation Maps in Augmented Reality," 2002.
- [43] J. Fritz *et al.*, "Augmented Reality Visualization with Use of Image Overlay Technology for MR Imaging–guided Interventions: Assessment of Performance in Cadaveric Shoulder and Hip Arthrography at 1.5 T," *Radiology*, vol. 265, no. 1,

- pp. 254–259, Oct. 2012, doi: 10.1148/radiol.12112640.
- [44] L. Chen, T. Day, W. Tang, and N. W. John, “Recent Developments and Future Challenges in Medical Mixed Reality,” Aug. 2017, [Online]. Available: <http://arxiv.org/abs/1708.01225>.
- [45] Z. Tang, P. Long, S. Huang, J. Zou, and Y. Qiu, “Real-time dose assessment and visualization of radiation field for EAST tokamak,” *Fusion Eng. Des.*, vol. 85, no. 7–9, pp. 1591–1594, Dec. 2010, doi: 10.1016/j.fusengdes.2010.04.054.
- [46] G. Yan and M. Li, “Path Planning and Visualizing for Maintenance Simulation of Nuclear Power Plant,” *Int. J. Online Eng.*, vol. 9, no. 6, p. 17, Nov. 2013, doi: 10.3991/ijoe.v9i6.3008.
- [47] Y. Sato, Y. Terasaka, S. Ozawa, Y. Tanifuji, and T. Torii, “A 3D radiation image display on a simple virtual reality system created using a game development platform,” *J. Instrum.*, vol. 13, no. 08, pp. T08011–T08011, Aug. 2018, doi: 10.1088/1748-0221/13/08/t08011.
- [48] Y. Zhao, “Visualization Technology and Preliminary Practice of Radiation Protection,” 2019, [Online]. Available: <http://www.isoe-network.net/docs/publications/proceedings/symposia/international-symposia/beijing-china-october-2019/slides-39/4374-zhao2019-ppt/file.html>.
- [49] S. Agostinelli *et al.*, “Geant4—a simulation toolkit,” *Nucl. Instruments Methods Phys. Res. Sect. A Accel. Spectrometers, Detect. Assoc. Equip.*, vol. 506, no. 3, pp. 250–303, Jul. 2003, doi: 10.1016/S0168-9002(03)01368-8.
- [50] J. Allison *et al.*, “Geant4 developments and applications,” *IEEE Trans. Nucl. Sci.*,

- vol. 53, no. 1, pp. 270–278, Feb. 2006, doi: 10.1109/TNS.2006.869826.
- [51] J. Allison *et al.*, “Recent developments in Geant4,” *Nucl. Instruments Methods Phys. Res. Sect. A Accel. Spectrometers, Detect. Assoc. Equip.*, vol. 835, pp. 186–225, Nov. 2016, doi: 10.1016/j.nima.2016.06.125.
- [52] N. Loy Rodas and N. Padoy, “3D Global Estimation and Augmented Reality Visualization of Intra-operative X-ray Dose,” 2014, pp. 415–422.
- [53] D. Tomono *et al.*, “First On-Site True Gamma-Ray Imaging-Spectroscopy of Contamination near Fukushima Plant,” *Sci. Rep.*, vol. 7, no. 1, p. 41972, Feb. 2017, doi: 10.1038/srep41972.
- [54] Y. Sato *et al.*, “Radiation Imaging System using a Compact Gamma-ray Imager mounted on a Remotely Operated Machine,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 1062–1066, 2017, doi: <https://doi.org/10.1016/j.ifacol.2017.08.218>.
- [55] H. Muraishi *et al.*, “Visualization of Low-Level Gamma Radiation Sources Using a Low-Cost, High-Sensitivity, Omnidirectional Compton Camera,” *J. Vis. Exp.*, no. 155, Jan. 2020, doi: 10.3791/60463.
- [56] F. Cong, Y. Tamura, K. Shimazoe, H. Takahashi, J. Ota, and S. Tong, “Radioactive source recognition with moving Compton camera imaging robot using Geant4,” *Nucl. Instruments Methods Phys. Res. Sect. A Accel. Spectrometers, Detect. Assoc. Equip.*, vol. 953, p. 163108, 2020, doi: <https://doi.org/10.1016/j.nima.2019.163108>.
- [57] K. Vetter *et al.*, “Advances in Nuclear Radiation Sensing: Enabling 3-D Gamma-Ray Vision,” *Sensors*, vol. 19, no. 11, 2019.

- [58] E. A. Medhurts, "Cloud Storage, Virtual Reality, And Augmented Reality To Guide Radiation Measurement Planning And Visualization Process," University of Illinois, 2020.
- [59] A. Gunatilaka, B. Ristic, and R. Gailis, "On Localisation of a Radiological Point Source," in *2007 Information, Decision and Control*, Feb. 2007, pp. 236–241, doi: 10.1109/IDC.2007.374556.
- [60] M. R. Morelande and B. Ristic, "Radiological Source Detection and Localisation Using Bayesian Techniques," *IEEE Trans. Signal Process.*, vol. 57, no. 11, pp. 4220–4231, Nov. 2009, doi: 10.1109/TSP.2009.2026618.
- [61] H. E. Baidoo-Williams, "Maximum Likelihood Localization of Radiation Sources with unknown Source Intensity," Jul. 2016, [Online]. Available: <http://arxiv.org/abs/1608.00427>.
- [62] B. A. Khuwaileh and W. A. Metwally, "Gaussian process approach for dose mapping in radiation fields," *Nucl. Eng. Technol.*, vol. 52, no. 8, pp. 1807–1816, 2020, doi: <https://doi.org/10.1016/j.net.2020.01.013>.
- [63] P. Gabrlik, T. Lazna, T. Jilek, P. Sladek, and L. Zalud, "Using an Automated Heterogeneous Robotic System for Radiation Surveys," Jun. 2020, [Online]. Available: <http://arxiv.org/abs/2006.16066>.
- [64] C. Liu, P.-L. Drouin, G. St-Jean, M. Deziel, and D. Waller, "Wireless Radiation Sensor Network with directional radiation detectors," in *2014 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC)*, Nov. 2014, pp. 1–6, doi: 10.1109/NSSMIC.2014.7431111.

- [65] C. L. Larsson and S. Djefal, "Development of a Directional Gamma Ray Probe," in *IEEE Nuclear Science Symposium Conference Record, 2005*, vol. 1, pp. 16–18, doi: 10.1109/NSSMIC.2005.1596198.
- [66] T. . Larsson, C.; Jones, "Testing of a new directional gamma survey meter," 2012.
- [67] C. Drouin, Pierre-Luc; Liu, "Development of statistical methods for radiation source localisation and activity measurement with multiple sensors," 2014.
- [68] F. Drouin, Pierre-Luc; Hassaine, "Radiation source localisation and type determination using the Point versus Area of Contamination Kit (PACK) system," 2019. [Online]. Available: https://cradpdf.drdc-rddc.gc.ca/PDFS/unc339/p810527_A1b.pdf.
- [69] "Nuclear Power Plant Personnel Training and its Evaluation: A Guidebook, Technical Report Series No. 380," 1996. [Online]. Available: https://www-pub.iaea.org/MTCD/Publications/PDF/trs380_web.pdf.
- [70] "Recruitment, Qualification and Training of Personnel for Nuclear Power Plants, Safety Guide, No. NS-G-2.8," 2002. [Online]. Available: https://www-pub.iaea.org/MTCD/Publications/PDF/Pub1140_scr.pdf.
- [71] "Categorization of Radioactive Sources, Safety Guide No. RS-G-1.9," 2005. [Online]. Available: https://www-pub.iaea.org/MTCD/Publications/PDF/Pub1227_web.pdf.
- [72] "Preparation, Conduct and Evaluation of Exercises to Test Preparedness for a Nuclear or Radiological Emergency, EPR-Exercise 2005," 2005. [Online]. Available: <https://www->

pub.iaea.org/MTCD/Publications/PDF/Exercise2005_web.pdf.

- [73] "Preparation, Conduct and Evaluation of Exercises to Test Security Contingency Plans at Nuclear Facilities," Vienna, Austria, 2018. [Online]. Available: <https://www.iaea.org/publications/12369/preparation-conduct-and-evaluation-of-exercises-to-test-security-contingency-plans-at-nuclear-facilities>.
- [74] D. Kamińska *et al.*, "Virtual Reality and Its Applications in Education: Survey," *Information*, vol. 10, no. 10, 2019, doi: 10.3390/info10100318.
- [75] "Vive(tm) Hardware," *HTC Corporation*, 2020. <https://www.vive.com/eu/product/vive/> (accessed Dec. 21, 2020).
- [76] "Welcome to Steamworks, SteamVR(tm) Tracking," *Valve Optical Technologies, Valve Software*. <https://partner.steamgames.com/vrlicensing> (accessed Dec. 21, 2020).
- [77] A. Yates and J. Selan, "Positional tracking systems and methods," US 2016/013 1761 A1, 2016.
- [78] "Unity Real-Time Development Platform |3D, 2D VR & AR Engine," *Unity Technologies*. <https://unity.com/> (accessed Dec. 21, 2020).
- [79] "Unity Asset Store," *Unity Technologies*. <https://assetstore.unity.com/> (accessed Dec. 22, 2020).
- [80] "Unity Learn - Online and in-person courses & training in 2D, 3D, AR & VR." <https://unity.com/learn> (accessed Dec. 22, 2020).
- [81] "Unity Developer Public Forum," *Unity Technologies*. <https://forum.unity.com/> (accessed Dec. 22, 2020).

- [82] “Unreal Engine: The most powerful real-time 3D creation platform,” *Epic Games*.
<https://www.unrealengine.com/en-US/> (accessed Dec. 22, 2020).
- [83] “CRYENGINE | The complete solution for next generation game development,”
Crytek GmbH. <https://www.cryengine.com/> (accessed Dec. 22, 2020).
- [84] “Unity Manual Virtual Reality,” *Unity Technologies*.
<https://docs.unity3d.com/2017.1/Documentation/Manual/VirtualReality.html>
(accessed Dec. 21, 2020).
- [85] “SteamVR on Steam,” *Valve Corporation*.
<https://store.steampowered.com/app/250820/SteamVR/> (accessed Dec. 21,
2020).
- [86] “SteamVR Plugin | Integration | Unity Asset Store,” *Unity Technologies*.
<https://assetstore.unity.com/packages/tools/integration/steamvr-plugin-32647>
(accessed Dec. 21, 2020).
- [87] “VIVE Specs & User Guide - Developer Resources,” *HTC Corporation*.
[https://developer.vive.com/resources/vive-sense/hardware-guide/vive-specs-user-
guide/](https://developer.vive.com/resources/vive-sense/hardware-guide/vive-specs-user-guide/) (accessed Dec. 21, 2020).
- [88] “Chaperone - SteamVR - Knowledge Base - Steam Support,” *Valve Corporation*.
https://support.steampowered.com/kb_article.php?ref=6281-TOKV-4722
(accessed Dec. 21, 2020).
- [89] “VR Best Practices - Unity Learn,” *Unity Technologies*.
<https://learn.unity.com/tutorial/vr-best-practice> (accessed Dec. 21, 2020).
- [90] “Locomotion - Oculus for Developers,” *Facebook Technologies*.

- <https://developer.oculus.com/design/latest/concepts/bp-locomotion/> (accessed Dec. 21, 2020).
- [91] “Virtual Reality Best Practices | Unreal Engine Documentation,” *Epic Games*. <https://docs.unrealengine.com/en-US/SharingAndReleasing/XRDevelopment/VR/DevelopVR/ContentSetup/index.html> (accessed Dec. 21, 2020).
- [92] G. Loup and E. Loup-Escande, “Effects of Travel Modes on Performances and User Comfort: A Comparison between ArmSwinger and Teleporting,” *Int. J. Human–Computer Interact.*, vol. 35, no. 14, pp. 1270–1278, 2019, doi: 10.1080/10447318.2018.1519164.
- [93] E. Chang, H. T. Kim, and B. Yoo, “Virtual Reality Sickness: A Review of Causes and Measurements,” *Int. J. Human–Computer Interact.*, vol. 36, no. 17, pp. 1658–1682, 2020, doi: 10.1080/10447318.2020.1778351.
- [94] “MCNP - A General N-Particle Transport Code, Version 5” Volume I: Overview and Theory,” 2005. [Online]. Available: https://laws.lanl.gov/vhosts/mcnp.lanl.gov/pdf_files/la-ur-03-1987.pdf.
- [95] M. G. Stabin and L. C. Q. P. da Luz, “DECAY DATA FOR INTERNAL AND EXTERNAL DOSE ASSESSMENT,” *Health Phys.*, vol. 83, no. 4, pp. 471–475, Oct. 2002, doi: 10.1097/00004032-200210000-00004.
- [96] K. F. Eckerman and A. L. Sjoreen, “Radiological Toolbox User’s Guide (NUREG/CR-7166, ORNL/TM-2013/16),” 2013. [Online]. Available: <https://www.nrc.gov/docs/ML1314/ML13149A163.pdf>.

- [97] “Software | ORNL,” *Oak Ridge National Laboratory*, 2014.
<https://www.ornl.gov/crpk/software> (accessed Dec. 22, 2020).
- [98] J. H. Hubbell and S. M. Seltzer, “Tables of X-Ray Mass Attenuation Coefficients and Mass Energy-Absorption Coefficients (version 1.03),” Gaithersburg, MD, 2004. doi: <https://dx.doi.org/10.18434/T4D01F>.
- [99] ICRP, “The 2007 Recommendations of the International Commission on Radiological Protection (ICRP 103),” *ICRP Publ. 103. Ann. ICRP 37 (2-4)*., 2007, [Online]. Available: [https://www.icrp.org/publication.asp?id=ICRP Publication 103](https://www.icrp.org/publication.asp?id=ICRP%20Publication%20103).
- [100] R. J. J. McConn, C. J. Gesh, R. T. Pagh, R. A. Ruker, and R. G. I. Williams, “Compendium of Material Composition Data for Radiation Transport Modeling, PIET-43741-TM-963 PNNL-15870 Rev. 1,” Richland, USA, 2011. [Online]. Available:
https://www.pnnl.gov/main/publications/external/technical_reports/PNNL-15870Rev1.pdf.
- [101] M. Li *et al.*, “MIGSHIELD: A new model-based interactive point kernel gamma ray shielding package for virtual environment,” *Nucl. Eng. Technol.*, vol. 52, no. 7, pp. 1557–1564, Jul. 2020, doi: 10.1016/j.net.2019.12.026.
- [102] I. Prokhorets, s. I. Prokhorets, M. Khazhmuradov, Y. Rudychev, and D. Fedorchenko, “Point-Kernel method for radiation fields simulation,” *Vopr. At. Nauk. i Tech. = Pyt. At. Nauk. i Tech. = Probl. At. Sci. Technol.*, vol. 48, 2007.
- [103] M. Lavik, “Volume rendering in Unity,” 2020.
<https://matiaslavik.wordpress.com/2020/01/19/volume-rendering-in-unity/>

(accessed Jun. 01, 2020).

- [104] M. (UOIT) Breitinger, "Improving X-Ray Operator Performance Using Virtual Environments," University of Ontario Institute of Technology, 2021.
- [105] B. S. C.H. Kim, Y.S. Yeom, N. Petoussi-Henss, M. Zankl, W.E. Bolch, C. Lee, C. Choi, T.T. Nguyen, K. Eckerman, H.S. Kim, M.C. Han, R. Qiu, B.S. Chung, H. Han, "Adult mesh-type reference computational phantoms," *ICRP Publ. 145*, 2020, [Online]. Available: [https://icrp.org/publication.asp?id=ICRP Publication 145](https://icrp.org/publication.asp?id=ICRP%20Publication%20145).
- [106] American National Standards Institute, "American National Standard for Neutron and Gamma-Ray Fluence-to-dose factors," 1991. [Online]. Available: <https://webstore.ansi.org/standards/ansi/ansians1991>.
- [107] "An overview of the Layar platform," *Layer*. <http://site.layar.com/create/platform-overview/> (accessed Nov. 10, 2010).
- [108] "Nexus One press release," *Google Inc.* <https://sites.google.com/a/pressatgoogle.com/nexusone/press-release> (accessed Dec. 22, 2020).
- [109] "SketchUp: 3D Design Software | 3D Modeling on the Web." <https://www.sketchup.com/> (accessed Dec. 21, 2020).
- [110] T. Schops, T. Sattler, C. Hane, and M. Pollefeys, "3D Modeling on the Go: Interactive 3D Reconstruction of Large-Scale Scenes on Mobile Devices," in *2015 International Conference on 3D Vision*, Oct. 2015, pp. 291–299, doi: 10.1109/3DV.2015.40.

- [111] T. Schöps, T. Sattler, C. Häne, and M. Pollefeys, "Large-scale outdoor 3D reconstruction on a mobile device," *Comput. Vis. Image Underst.*, vol. 157, pp. 151–166, Apr. 2017, doi: 10.1016/j.cviu.2016.09.007.
- [112] "Google's Project Tango," 2016. <https://www.google.com/atap/project-tango/> (accessed Jun. 20, 2016).
- [113] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, and G. Ranzuglia, "MeshLab: an Open-Source Mesh Processing Tool," in *Eurographics Italian Chapter Conference*, 2008, doi: 10.2312/LocalChapterEvents/ItalChap/ItalianChapConf2008/129-136.
- [114] "MeshLab software." <https://www.meshlab.net/>.
- [115] U. Riedlinger, L. Oppermann, and W. Prinz, "Tango vs. HoloLens: A Comparison of Collaborative Indoor AR Visualisations Using Hand-Held and Hands-Free Devices," *Multimodal Technol. Interact.*, vol. 3, no. 2, p. 23, Apr. 2019, doi: 10.3390/mti3020023.
- [116] "Unity AR Foundations," *Unity Technologies*, 2021. <https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@4.2/manual/index.html> (accessed Jun. 01, 2021).
- [117] "ARCore Extensions for AR Foundation," *Google Inc*, 2021. <https://developers.google.com/ar/reference/unity-arf> (accessed Jun. 01, 2021).
- [118] "ARCore supported devices," *Google Inc*, 2021. <https://developers.google.com/ar/devices> (accessed Jun. 01, 2021).
- [119] R. Du *et al.*, "DepthLab: Real-time 3D Interaction with Depth Maps for Mobile

- Augmented Reality,” in *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, Oct. 2020, pp. 829–843, doi: 10.1145/3379337.3415881.
- [120] “S.E. International Inc. Radiation Alert Detectors and Monitors,” 2020. <https://seintl.com/products/abacus.html> (accessed Jan. 01, 2020).
- [121] “Abacus Operational Manual,” Summer Town, TN., S.E. International, Inc.
- [122] “Android & Microcontroller / Bluetooth.” <https://assetstore.unity.com/packages/tools/input-management/android-microcontrollers-bluetooth-16467> (accessed Jan. 18, 2021).
- [123] S. B. Kotsiantis, I. D. Zaharakis, and P. E. Pintelas, “Machine learning: a review of classification and combining techniques,” *Artif. Intell. Rev.*, 2006.
- [124] Unity, “Unity ML Agents Toolkit Github.” <https://github.com/Unity-Technologies/ml-agents>.
- [125] A. Juliani *et al.*, “Unity: A General Platform for Intelligent Agents,” Sep. 2018, [Online]. Available: <http://arxiv.org/abs/1809.02627>.
- [126] “PyTorch.” <https://pytorch.org/> (accessed Jun. 01, 2020).
- [127] A. Barto, S. Satinder, and S. Nuttpong, ““Intrinsically motivated learning of hierarchical collections of skills.” 2004.
- [128] J. Peters, “Policy gradient methods,” *Scholarpedia*, vol. 5, no. 11, p. 3698, 2010, doi: 10.4249/scholarpedia.3698.
- [129] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal Policy

- Optimization Algorithms,” Jul. 2017, [Online]. Available: <http://arxiv.org/abs/1707.06347>.
- [130] T. Haarnoja *et al.*, “Soft Actor-Critic Algorithms and Applications,” Dec. 2018, [Online]. Available: <http://arxiv.org/abs/1812.05905>.
- [131] J. Ho and S. Ermon, “Generative Adversarial Imitation Learning,” Jun. 2016, [Online]. Available: <http://arxiv.org/abs/1606.03476>.
- [132] T. M. Moreira, J. G. de Faria, P. O. S. Vaz-de-Melo, L. Chaimowicz, and G. Medeiros-Ribeiro, “Prediction-free, real-time flexible control of tidal lagoons through Proximal Policy Optimisation: A case study for the Swansea Lagoon,” *Ocean Eng.*, vol. 247, p. 110657, Mar. 2022, doi: 10.1016/j.oceaneng.2022.110657.
- [133] “Portable Digital Assistant for First Responders to a Radiological Emergency, IAEA-EPR-First Responders PDA,” 2006. [Online]. Available: https://www-pub.iaea.org/MTCD/Publications/PDF/EPR-First_Res-PDA/html/ti1-elevated-rad-levels.htm.
- [134] “Generic procedures for assessment and response during a radiological emergency, IAEA-TECDOC-1162,” 2000. [Online]. Available: https://www-pub.iaea.org/mtcd/publications/pdf/te_1162_prn.pdf.
- [135] “Regulations for the Safe Transport of Radioactive Material 2018 edition, Safety Specific Requirements, No. SSR-6 (Rev 1),” 2018. [Online]. Available: https://www-pub.iaea.org/MTCD/Publications/PDF/PUB1798_web.pdf.
- [136] “Containers for Transport of Dangerous Goods by Rail, a Transport Canada

Standard, TP 14877E,” 2018. [Online]. Available:
https://tc.canada.ca/sites/default/files/migrated/tp14877_en.pdf.

[137] Liv.tv, “LIV Developers software SDK webpage.” <https://www.liv.tv/developers>
(accessed Jan. 05, 2020).

[138] J. K. Shultis and R. E. Faw, “RADIATION SHIELDING TECHNOLOGY,” *Health Phys.*, vol. 88, no. 4, pp. 297–322, Apr. 2005, doi:
10.1097/01.HP.0000148615.73825.b1.

[139] “API reference - ARCore SDK for Unity,” *Google Inc*, 2021.
<https://developers.google.com/ar/reference/unity> (accessed Jun. 01, 2021).

[140] M. Schimpl *et al.*, “Association between Walking Speed and Age in Healthy, Free-Living Individuals Using Mobile Accelerometry—A Cross-Sectional Study,” *PLoS One*, vol. 6, no. 8, p. e23299, Aug. 2011, doi: 10.1371/journal.pone.0023299.

Annex I: UPK calculation code

This annex contains the C# code used for the calculation of radiation field intensity from n number of sources in UPK. This is comprised of the following C# scripts which are used in Unity:

- RadiationSource.cs is provided in Table 4
 - This C# script is used as an attribute on any object within a scene that it to function as source of radiation. When setting up the scene and placing this script on the game object radiation source, the user will be required to select the radionuclides of the source and its activity in Ci or Bq with associated Si prefix. When a scene is loaded this script will automatically import all required gamma photon emissions (>0.00001%) from a library file and will normalize the source photons from all radionuclides based on the relevant probability and activity to a list of photons of energy x per second which is stored in a list for future use.
- RadiationShield.cs is provided in Table 5
 - This C# script is used as an attribute on any object within a scene that is to function as a radiation shield when directly between a source of radiation and a detector. When setting up the scene and placing the script on the game object to act as a shield, the user will be required to select the material of the game object from a library of predetermined materials with default densities. When a scene is loaded this script will automatically import the mass attenuation coefficients related to that material from a library file. It will then search the entire scene and

determine every radionuclide that has been defined within the scene and it will look up the associated photon decay energies for all of the radionuclides. Then it will search through the library file of imported material mass attenuation coefficients and for each photon energy of the radionuclides in the scene it will linearly interpolate between the two closest mass attenuation coefficients that will get saved in a list for future use.

- RadiationCalculation is provided in Table 6
 - This C# script is used as an attribute on any object that will need to calculate a radiation field measurement. When a scene is loaded it will search the entire scene and determine every radionuclide that has been defined within the scene and it will look up the associated photon decay energies for all of the radionuclides. It will store a list of each defined radiation source in the scene. It will then lookup a table of conversion factors from photon energy intensity to Gy and for Gy to Sv(H*10) operational units and it will linearly interpolate between the two closest conversion factors for each photon energy and store that value in a list. During game play, when the function CalculateRadiationReading(), a raycast is sent from each radiation source to the detectors in both forwards (source to detector) and backwards (detector to source) directions returning a list of game objects that were hit by the rays as well as other associated data (such as the hit location). For each source the distance to the detector is calculated. If a gameObject that has been

marked as a shield (through tagging) is found the entry and exit points are taken to determine the penetration path length through that game object. The mass attenuation impact on each radionuclide is then calculated for that source and the total is incorporated into the final dose per decay at the detector. This process loops through all sources and the final results are summed and converted from Sv(H*10)/s to Sv(H*10)/h and when needed, Si prefixes are used to emulate a detector converting to the highest applicable prefix.

Table 4 - RadiationSource.cs C# code

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using System.IO;

public class RadiationSource : MonoBehaviour {

    'RadionuclideSource'

    public enum Radionuclides {

        //(Large list of radionuclides removed for brevity)
        //This list would be in the format of (shown for Cs-137 and Co-60)
        // CS137,
        // CS60,

    }

    public enum ActivityUnitsPrefix {
        pico,
        nano,
        micro,
        mili,
        none,
        kilo,
        mega,
        giga,
        tera,
```

```

    peta,
}

public enum ActivityUnits {
    Bq,
    Ci,
}

public Radionuclides radionuclide;
public double activity;
public ActivityUnitsPrefix activityUnitsprefix;
public ActivityUnits activityUnits;

public List<string> listRadionuclideEnergy = new List<string>();
public List<string> listRadionuclideProb = new List<string>();
public double adjustedactivityBq;
private double unitmultifactor;

    void Start () {
        //File path in the next line refers to a location where data is stored on the local
        machine in a comma-separated values (.csv). This approach is used in other scripts to
        access data.
        //It could also be stored in a local resources folder in the project using
        Application.dataPath + + "../(folder name)/(filename.extension)"
        //The file contains decay energy and probability data stored in a format like this
        (example from Cobalt 60)
        // CO60,1332.501,    0.999856,
        // CO60,1173.237,    0.999736,

        using(var radionuclideenergyreader = new StreamReader(@"[filepath]")) {

            while (!radionuclideenergyreader.EndOfStream) {
                var line = radionuclideenergyreader.ReadLine();
                var values = line.Split(',');

                if (values[0] == radionuclide.ToString()) {
                    listRadionuclideEnergy.Add(values[1]);
                    listRadionuclideProb.Add(values[2]);
                }
            }
        }

        if ("Bq" == activityUnits.ToString()){

            if (activityUnitsprefix.ToString() == "pico"){unitmultifactor = 0.000000000001;}
            if (activityUnitsprefix.ToString() == "nano"){unitmultifactor = 0.000000001;}
            if (activityUnitsprefix.ToString() == "micro"){unitmultifactor = 0.000001;}

```

```

if (activityUnitsprefix.ToString() == "mili"){unitmultifactor = 0.001;}
if (activityUnitsprefix.ToString() == "none"){unitmultifactor = 1;}
if (activityUnitsprefix.ToString() == "kilo"){unitmultifactor = 1000;}
if (activityUnitsprefix.ToString() == "mega"){unitmultifactor = 1000000;}
if (activityUnitsprefix.ToString() == "giga"){unitmultifactor = 1000000000;}
if (activityUnitsprefix.ToString() == "tera"){unitmultifactor = 1000000000000;}
if (activityUnitsprefix.ToString() == "peta"){unitmultifactor = 1000000000000000;}

adjustedactivityBq = activity * unitmultifactor;
}
if ("Ci" == activityUnits.ToString()){

if (activityUnitsprefix.ToString() == "pico"){unitmultifactor = 0.000000000001;}
if (activityUnitsprefix.ToString() == "nano"){unitmultifactor = 0.000000001;}
if (activityUnitsprefix.ToString() == "micro"){unitmultifactor = 0.000001;}
if (activityUnitsprefix.ToString() == "mili"){unitmultifactor = 0.001;}
if (activityUnitsprefix.ToString() == "none"){unitmultifactor = 1;}
if (activityUnitsprefix.ToString() == "kilo"){unitmultifactor = 1000;}
if (activityUnitsprefix.ToString() == "mega"){unitmultifactor = 1000000;}
if (activityUnitsprefix.ToString() == "giga"){unitmultifactor = 1000000000;}
if (activityUnitsprefix.ToString() == "tera"){unitmultifactor = 1000000000000;}
if (activityUnitsprefix.ToString() == "peta"){unitmultifactor = 1000000000000000;}

adjustedactivityBq = activity * unitmultifactor * (37000000000);
}
}
}

```

Table 5 - RadiationShield.cs C# code

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using System.IO;

public class ShieldingObject : MonoBehaviour {

    public enum Materials {

        (List of materials removed)
    }
    public Materials materials;
    public List<GameObject> RadionuclidesList = new List<GameObject>();
    public string density = "Default";
    public List<string> listMaterialEnergy = new List<string>();
}

```



```

public List<string> listMaterialAttenuation = new List<string>();
public List<string> listRadionuclideEnergy = new List<string>();
public List<double> listLinearAttenuationInterpolation = new List<double>();
public List<string> listRadionuclides = new List<string>();

void Start () {

    RadionuclidesList.AddRange (GameObject.FindGameObjectsWithTag
("RadionuclideSource"));
    //File path in the next line refers to a location where data is stored on the local
machine in a comma-separated values (.csv).
    //This approach is used in other scripts to access data.
    //It could also be stored in a local resources folder in the project using
Application.dataPath + + "../(folder name)/(filename.extension)"
    //The file contains material name, a range of photon energies and the attenuation
value. This is an example for the material 'skin' from 100-1000 keV.
    //Skin,100,0.1636
    //Skin,150,0.1466
    //Skin,200,0.1342
    //Skin,300,0.1168
    //Skin,400,0.1047
    //Skin,500,0.09565
    //Skin,600,0.08848
    //Skin,800,0.07776
    //Skin,1000,0.06992

        using(var materialsreader = new
StreamReader(@"[filepath]")) {

            while (!materialsreader.EndOfStream) {
                var line = materialsreader.ReadLine();
                var values = line.Split(',');

                if (values[0] == materials.ToString()) {
                    listMaterialEnergy.Add(values[1]);
                    listMaterialAttenuation.Add(values[2]);
                }
            }
        }

    foreach (GameObject Source in RadionuclidesList) {
        string temp =
Source.GetComponent<RadiationSource>().radionuclide.ToString();

        if(listRadionuclides.Contains(temp)){
        }
    }
}

```

```

        else{
            //File path in the next line refers to a location where data is stored on the local
            machine in a comma-separated values (.csv).
            //The file contains decay energy and probability data stored in a format like this
            (example from Cobalt 60)
            // CO60,1332.501,    0.999856,
            // CO60,1173.237,    0.999736,

            using(var radionuclideenergyreader = new
            StreamReader(@"[filepath]")) {

                while (!radionuclideenergyreader.EndOfStream) {
                    var line =
radionuclideenergyreader.ReadLine();
                    var values = line.Split(',');

                    if (values[0] == temp) {
                        listRadionuclideEnergy.Add(values[1]);
                    }
                }
            }
        }

        if (density == "Default"){
            //File path in the next line refers to a location where data is stored on the local
            machine in a comma-separated values (.csv).
            //This is only used if default is left as the value when setting up a sheild, otherwise
            a unique density for the material can be manually assigned.
            //The file contains default density values for materials. The format is as follows for
            skin and lead.
            //Skin,0.803
            //Lead,9.53

            using(var materialdensityreader = new
            StreamReader(@"[filepath]")) {

                while (!materialdensityreader.EndOfStream) {
                    var line = materialdensityreader.ReadLine();
                    var values = line.Split(',');

                    if (values[0] == materials.ToString()) {
                        density = values[1];
                    }
                }
            }
        }
    }

```

```

        int stopgoingthrough = 0;

        foreach(string radionuclideenergy in listRadionuclideEnergy) {

            int countmaterialenergy = 0;

            while(countmaterialenergy <
listMaterialEnergy.Count) {
                if (stopgoingthrough == 0) {

                    double radionuclideenergydouble =
System.Convert.ToDouble(radionuclideenergy);
                    double materialenergydouble =
System.Convert.ToDouble(listMaterialEnergy[countmaterialenergy]);

                    if (materialenergydouble > radionuclideenergydouble)
{
                        stopgoingthrough++;

                        double stepsize =
(System.Convert.ToDouble(listMaterialAttenuation[countmaterialenergy]) -
System.Convert.ToDouble(listMaterialAttenuation[countmaterialenergy-
1]))/(System.Convert.ToDouble(listMaterialEnergy[countmaterialenergy]) -
System.Convert.ToDouble(listMaterialEnergy[countmaterialenergy-1]));

                        double interpolatedvalue =
System.Convert.ToDouble(listMaterialAttenuation[countmaterialenergy-1]) + stepsize *
(radionuclideenergydouble -
System.Convert.ToDouble(listMaterialEnergy[countmaterialenergy-1]));

                        listLinearAttenuationInterpolation.Add(interpolatedvalue);

                        stopgoingthrough++;
                    }
                }
                countmaterialenergy++;
            }

            stopgoingthrough = 0;
        }
    }
}

```

Table 6 - RadiationCalculation.cs C# code

```
using System;
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using TMPro;
using System.IO;

public class RadiationCalculation : MonoBehaviour {

    public GameObject Display;
    public GameObject TextDisplay;

    public double Reading;
    private double DosePerSourceEnergy;
    private float temp3;
    private float distance;

    private double reductionfactor;
    private double totalreductionfactor;
    private Vector3 EntryPoint;
    private Vector3 ExitPoint;

    private Ray ray;
    private Ray raybk;

    private List<RaycastHit> shieldhit = new List<RaycastHit> ();
    private List<RaycastHit> shieldhit2 = new List<RaycastHit> ();
    public List<double> listshieldobjectpenetrationdistances = new List<double> ();
    public List<GameObject> listShieldGameObjects = new List<GameObject> ();
    public List<string> dcfenergy = new List<string> ();
    public List<string> dcfToGy = new List<string> ();
    public List<double> dcfToGylinearinterpolation = new List<double>();
    public List<string> GyToSv = new List<string> ();
    public List<double> GyToSvlinearinterpolation = new List<double>();
    public List<GameObject> RadionuclidesList = new List<GameObject>();
    public List<string> listRadionuclideEnergy = new List<string>();
    public List<string> listRadionuclides = new List<string>();
    void Start () {

        //File path in the next line refers to a location where data is stored on the local
        machine in a comma-separated values (.csv).
        //The file contains the values needed for converting Air Kerma Free in Air per
        Fluence and for converting to H*(10).
        //See figure 20 and figure 21 for details
```

```

using(var dcfreader = new StreamReader(@"[filepath]")) {

    while (!dcfreader.EndOfStream) {
        var line = dcfreader.ReadLine();
        var values = line.Split(',');

        dcfenergy.Add(values[0]);
        dcfToGy.Add(values[1]);
        GyToSv.Add(values[2]);
    }
}

RadionuclidesList.AddRange (GameObject.FindGameObjectsWithTag
("RadionuclideSource"));

foreach (GameObject Source in RadionuclidesList) {
    string temp4 =
Source.GetComponent<RadiationSource>().radionuclide.ToString();

    listRadionuclides.Add(temp4);
    //File path in the next line refers to a location where data is stored on the
local machine in a comma-separated values (.csv).
    //The file contains decay energy and probability data stored in a format
like this (example from Cobalt 60)
    // CO60,    1332.501,    0.999856,
    // CO60,    1173.237,    0.999736,

    using(var radionuclideenergyreader = new
StreamReader(@"[filepath]")) {

        while (!radionuclideenergyreader.EndOfStream) {
            var line =
radionuclideenergyreader.ReadLine();
            var values = line.Split(',');

            if (values[0] == temp4) {
                listRadionuclideEnergy.Add(values[1]);
            }
        }
    }
}

int stopgoingthrough = 0;

foreach(string radionuclideenergy in listRadionuclideEnergy) {

```

```

count needs to begin at zero when we enter the foreach loop
    int countmaterialenergy = 0;

    while(countmaterialenergy < dcfenergy.Count)
    {
        if (stopgoingthrough == 0) {

                double radionuclideenergydouble =
System.Convert.ToDouble(radionuclideenergy);
                double materialenergydouble =
System.Convert.ToDouble(dcfenergy[countmaterialenergy]);

                if (materialenergydouble > radionuclideenergydouble)
{
                    stopgoingthrough++;

                    double stepsize =
(System.Convert.ToDouble(dcfToGy[countmaterialenergy]) -
System.Convert.ToDouble(dcfToGy[countmaterialenergy-
1]))/(System.Convert.ToDouble(dcfenergy[countmaterialenergy]) -
System.Convert.ToDouble(dcfenergy[countmaterialenergy-1]));

                    double interpolatedvalue =
System.Convert.ToDouble(dcfToGy[countmaterialenergy-1]) + stepsize *
(radionuclideenergydouble - System.Convert.ToDouble(dcfenergy[countmaterialenergy-
1]));

                    dcfToGylinearinterpolation.Add(interpolatedvalue);

                    stopgoingthrough++;
                }
            }
            countmaterialenergy++;
        }
        stopgoingthrough = 0;
    }

    stopgoingthrough = 0;

    foreach(string radionuclideenergy in listRadionuclideEnergy) {

```

```

        int countmaterialenergy = 0;

        while(countmaterialenergy < dcfenergy.Count)
        {
            if (stopgoingthrough == 0) {

                double
radionuclideenergydouble = System.Convert.ToDouble(radionuclideenergy);
                double materialenergydouble =
System.Convert.ToDouble(dcfenergy[countmaterialenergy]);

                if (materialenergydouble > radionuclideenergydouble)
{
                    stopgoingthrough++;

                    double stepsize =
(System.Convert.ToDouble(GyToSv[countmaterialenergy]) -
System.Convert.ToDouble(GyToSv[countmaterialenergy-
1]))/(System.Convert.ToDouble(dcfenergy[countmaterialenergy]) -
System.Convert.ToDouble(dcfenergy[countmaterialenergy-1]));

                    double interpolatedvalue =
System.Convert.ToDouble(GyToSv[countmaterialenergy-1]) + stepsize *
(radionuclideenergydouble - System.Convert.ToDouble(dcfenergy[countmaterialenergy-
1]));

                    GyToSvlinearinterpolation.Add(interpolatedvalue);

                    stopgoingthrough++;
                }
            }
            countmaterialenergy++;
        }

        stopgoingthrough = 0;
    }

    void Update () {
        //Assigned on an update function runs this calculation each update. This is very
        costly.
        //Integration time of a detector can be approximated by calling this function every
        few seconds.
        //Integration can be further simulated by calling this function several times during

```

the integration period and averaging the values.

```
//Other options can be used to represent the behaviour of a specific detector.
```

```
    CalculateRadiationReading();
```

```
}
```

```
public void CalculateRadiationReading () {
```

```
    Reading = 0.0d;
```

```
    foreach (GameObject source in RadionuclidesList) {
```

```
        distance = (Vector3.Distance (source.transform.position,
transform.position))*100;
```

```
        double unnormalizedparticlefluence = 1 / (4 * 3.14159d *
(distance*distance));
```

```
        ray = new Ray (source.transform.position, transform.position -
source.transform.position);
```

```
        raybk = new Ray (transform.position, source.transform.position -
Display.transform.position);
```

```
        shieldhit.Clear ();
        shieldhit.AddRange (Physics.RaycastAll (ray.origin, ray.direction,
distance/100));
```

```
        shieldhit2.Clear ();
        shieldhit2.AddRange (Physics.RaycastAll (raybk.origin,
raybk.direction, distance/100));
```

```
        listShieldGameObjects.Clear ();
        listshieldobjectpenetrationdistances.Clear ();
```

```
        foreach (RaycastHit hit in shieldhit) {
```

```
            if (hit.collider.tag == "ShieldObject") {
```

```
                EntryPoint = hit.point;
```

```
                foreach (RaycastHit hit2 in shieldhit2) {
```

```
                    if (hit2.collider.GetInstanceID () ==
```



```

hit.collider.GetInstanceID ()) {

                                ExitPoint = hit2.point;

                                temp3 = Vector3.Distance (EntryPoint,
ExitPoint) * 100;

                                listshieldobjectpenetrationdistances.Add(temp3);

                                listShieldGameObjects.Add(hit.transform.gameObject);
                                }
                                }
                                }

                                double normalizedActivity;
                                normalizedActivity = source.GetComponent<RadiationSource>
().adjustedactivityBq;

                                double linearattenuation = 0;
                                double density = 0;

                                int i = 0;

                                foreach(string energy in source.GetComponent<RadiationSource>
().listRadionuclideEnergy) {
                                    totalreductionfactor = 0;

                                    int j = 0;
                                    foreach(GameObject shield in listShieldGameObjects) {

                                        density =
System.Convert.ToDouble(shield.GetComponent<ShieldingObject> ().density);

                                        int k = 0;
                                        foreach(string energy2 in
shield.GetComponent<ShieldingObject> ().listRadionuclideEnergy) {

                                            if (energy == energy2) {

                                                linearattenuation =
shield.GetComponent<ShieldingObject> ().listLinearAttenuationInterpolation[k];

```

```

        }
        k++;
    }

    reductionfactor = Math.Pow (2.71828d, (-
linearattenuation*density*listshieldobjectpenetrationdistances[j]));
    if (totalreductionfactor ==0){
        totalreductionfactor = reductionfactor;
    }
    else{
        totalreductionfactor =
totalreductionfactor * reductionfactor;
    }

    j++;
}
int l = 0;

foreach(string energy3 in listRadionuclideEnergy) {
    if (energy == energy3){

        if(totalreductionfactor == 0){
            totalreductionfactor=1;
        }
        DosePerSourceEnergy =
totalreductionfactor*normalizedActivity*unnormalizedparticlefluence *
System.Convert.ToDouble(source.GetComponent<RadiationSource>
().listRadionuclideProb[i]) * dcfToGylinearinterpolation[l] * GyToSvlinearinterpolation[l];

    }
    l++;
}

    Reading = Reading +
(DosePerSourceEnergy*60*60)*0.000000000001;
    i++;
}
}
//print(Reading) to see the reading in the console.
//As needed, convert to string and assign the reading to a text box or other display
to show it to the user.
//formatUnit and formatNumber can be used when converting the Reading to a
string to display the appropriate SI unit.
}

```

```

string formatUnit (float n) {
    string output;
    output = "Test";
    if (n < 0.001) output = "Low";
    if (n >= 0.001) output = "nSv/h";
    if (n >= 1) output = "µSv/h";
    if (n >= 1000) output = "mSv/h";
    if (n >= 1000000) output = " Sv/h";
    if (n >= 100000000) output = "kSv/h";
    if (n >= 100000000000)
        output = "Off/S";

    return output;
}

float formatNumber (float n, out float output){
    output = 1.0f;

    if (n < 0.001) {
        output = 0;
    }
    if (n >= 0.001) {
        output = 0.001f;
    }
    if (n >= 1) {
        output = 1;
    }
    if (n >= 1000) {
        output = 1000;
    }
    if (n >= 1000000) {
        output = 1000000;
    }
    if (n >= 100000000) {
        output = 100000000;
    }
    if (n >= 100000000000) {
        output = 100000000000;
    }
    return output;
}
}

```

UPK uses a number of associated library reference files. These files are the following:

- RadiationDecayEnergiesCSV.csv
 - This is reference file with a comma-separated value (CSV) list of data providing a radionuclide name and its associated gamma photon energies (in KeV) and emission probabilities (in %) per decay.
- MaterialsAttenuationCSV.csv
 - This is a reference file with a CSV list of data providing a material name and its associated mass attenuation coefficient ($\text{cm}^2/\text{g}^{-3}$) for energy provided in steps³⁵.
- MaterialDensitiesCSV.csv
 - This is a reference file with a CSV list of data providing a material name and a default density (g/cm^{-3}) taken from nuclear industry literature[100].
- AmbientDoseEquivalentCSV.csv
 - This is a reference file with a CSV list of data providing photon energy (keV) and conversion factors for converting to Gy and for converting Gy to Sv in H*10 operational units.

³⁵ Step width varies between material and generally has a range from keV to MeV in range

Annex II: Augmented reality visualization code

The following C# code was used with a third party package to connect to the Abacus detector and sent and receive data.

Table 7 - C# code in Unity used to query and receive data from the Abacus Radiation Detector

```
void Awake ()
{
    BluetoothAdapter.askEnableBluetooth ();
    BluetoothAdapter.OnDeviceOFF += HandleOnDeviceOff;
    BluetoothAdapter.OnDevicePicked += HandleOnDevicePicked;
    BluetoothDevice [] btList = BluetoothAdapter.getPairedDevices();

    for (int i = 0; i<btList.Length; i++)
    {
        //Specific to the detector used in this research
        if(btList[i].Name=="Abacus 001008")
        {
            connectMac = btList[i].MacAddress;
            deviceNumber = i;
        }
    }

    HandleOnDevicePicked(btList[deviceNumber]);
}

void HandleOnDeviceOff (BluetoothDevice dev)
{
    if (!string.IsNullOrEmpty (dev.Name))
        status.text = "Couldn't connect to " + dev.Name + ", device is OFF";
    else if (!string.IsNullOrEmpty (dev.Name)) {
        status.text = "Couldn't connect to " + dev.MacAddress + ", device is
OFF";
    }
}

void HandleOnDevicePicked (BluetoothDevice device)
{
```

```

        device.ReadingCoroutine = ManageConnection;

        devicNameText.text = "Remote Device : " + device.Name;
connect();
    }

    public void send ()
    {
        if (device != null && !string.IsNullOrEmpty (dataToSend.text)) {
            device.send (dataToSend.text);
            (System.Text.Encoding.ASCII.GetBytes
(dataToSend.text));
        }
    }

    IEnumerator ManageConnection (BluetoothDevice device)
    {

        InfoCanvas.SetActive (false);
        DataCanvas.SetActive (true);
        TangoMultiCameraJ.SetActive (true);

        byte[] msg = { };
        int loopCount = 0;

        if (firstRead == 0)
        {
            byte[] msgConnect = { 65 };
            device.send(msgConnect);
            readDataText.add(device.Name, "Connected");
            readDataText.add(device.Name, connectMac);
            abacusCPS.Clear();
            abacusDateTime.Clear();

            firstRead = 1;
        }

        while (device.IsReading) {
            if (device.IsDataAvailable) {
                msg = device.read();

                if (!(msg.Length == 5)){

```

```

        for (int i = 0; i < msg.Length; i++)
        {
            if (msg[i] < 32 || msg[i] > 122)
            {
                msg[i] = 32;
            }
        }
        string content = System.Text.ASCIIEncoding.ASCII.GetString(msg);
        readDataText.add(device.Name, content);
        msg = null;
    }

    if (msg != null && msg.Length > 0) {

        abacusCPS.Add(msg[0] * 10 + msg[1] + msg[2] * 10 + msg[3]);

        abacusDateTime.Add(System.DateTime.Now.ToString());

        readDataText.add (device.Name, "Index: " + loopCount + " Time: " +
abacusDateTime[loopCount] + " CPS: " + abacusCPS[loopCount]);

        Display.text = (abacusCPS [loopCount]).ToString();
        loopCount++;
    }

    }

    yield return null;
}
DataCanvas.SetActive (false);
InfoCanvas.SetActive (true);
}

```

Annex III: Reinforcement learning code

This annex contains the C# code used for the training of an Agent using Unity ML-Agent toolkit and that agent is capable of estimating the location of a radiation source from 10 measurements taken in the environment. For training, the following C# classes are required in Unity:

- GuessingAgent
 - This is an Agent class which when run in training configuration will randomly position a radiation source in a scene, then take 10 measurements from random positions which are provided to the reinforcement learning algorithm as normalized values between 1 and -1. The algorithm then 'guesses' the X, Y and Z location of the source between 1 and -1, which is then normalized to the size of the play space which was 5 and -5 in width and length and 0 and 2 in height. The known position of the source is then compared with the guess and depending on how far the guess is from the actual location a reward or a penalty is given to the Agent.
 - The agent could be provided a reward and asked to guess again until it gets within a fixed distance from the source and the episode ends.
 - The agent could be provided a reward and then 10 new measurements are given and it is asked to guess again until it gets within a fixed distance of the source and the episode ends.

- The agent could be provided a reward and the episode ends regardless of the distance the guess from the source is triggering a episode with a new random source location and new set of 10 randomly taken measurements.
- RadiationArea3d
 - This code controls the training environment which includes moving the source to a random location within the environment at the start of each episode.
- RadiationCalculationforML
 - This class controls all aspects of the virtual radiation measurements when in augmented reality. This code monitors the location of the phone as it is moved as a virtual radiation detector and proceeds to keep a running tally of the last 10 measurements taken and their associated positions. When the user presses an onscreen button, a function is called which takes those values, normalizes them in the same way as the GuessingAgent class and then feeds them into the previously trained neural network which produces an output that is normalized for the play space size it was trained in. The class then places a sphere in the environment to mark where the guess is to the user is able to see and it optionally can draw a line from the guess to the placed source in the environment giving its length, which represents the distance between the guess and the actual location in cm.

Table 8 - GuessingAgent.cs C# code

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using Unity.MLAgents;
using Unity.MLAgents.Actuators;
using Unity.MLAgents.Sensors;
using System.IO;
using System;

public class GuessingAgent : Agent
{
    //Note - several versions of ML Agents package were used. ML-Agents Release 13
    was the version used in this.
    //It is strongly recommended to use the latest version and to specifically check for
    documentation updates.
    public GameObject guessicon;
    public GameObject arena;
    public GameObject SingleSourceML;
    public GameObject[] Measurements;
    private float CountingSteps;
    public List<Vector3> MeasurementPositions = new List<Vector3>();
    public List<double> MeasurementDoubles = new List<double>();
    public List<double> MeasurmentDoublesNormalized = new List<double>();
    public GameObject Display;
    public double Reading;
    private double DosePerSourceEnergy;
    private float temp3;
    private float distance;
    private double reductionfactor;
    private double totalreductionfactor;
    private Vector3 EntryPoint;
    private Vector3 ExitPoint;
    private Ray ray;
    private Ray raybk;
    private List<RaycastHit> shieldhit2 = new List<RaycastHit>();
    public List<double> listshieldobjectpenetrationdistances = new List<double>();
    public List<GameObject> listShieldGameObjects = new List<GameObject>();
    public List<string> dcfenergy = new List<string>();
    public List<string> dcfToGy = new List<string>();
    public List<double> dcfToGylinearinterpolation = new List<double>();
    public List<string> GyToSv = new List<string>();
    public List<double> GyToSvlinearinterpolation = new List<double>();
    public List<GameObject> RadionuclidesList = new List<GameObject>();
    public List<string> listRadionuclideEnergy = new List<string>();
    public List<string> listRadionuclides = new List<string>();
```

```

public override void Initialize()
{

    using (var dcfreader = new StreamReader(Application.dataPath + "[filepath]"))
    {

        while (!dcfreader.EndOfStream)
        {
            var line = dcfreader.ReadLine();
            var values = line.Split(',');

            dcfenergy.Add(values[0]);
            dcfToGy.Add(values[1]);
            GyToSv.Add(values[2]);
        }
    }

    RadionuclidesList.Add(SingleSourceML);

    foreach (GameObject Source in RadionuclidesList)
    {
        string temp4 = Source.GetComponent<RadiationSource>().radionuclide.ToString();

        else
        {
            listRadionuclides.Add(temp4);
            using (var radionuclideenergyreader = new
            StreamReader(Application.dataPath + "[filepath]"))
            {

                while (!radionuclideenergyreader.EndOfStream)
                {
                    var line = radionuclideenergyreader.ReadLine();
                    var values = line.Split(',');

                    if (values[0] == temp4)
                    {
                        listRadionuclideEnergy.Add(values[1]);
                    }
                }
            }
        }
    }
}

```

```

    }
    }
}

int stopgoingthrough = 0;

foreach (string radionuclideenergy in listRadionuclideEnergy)
{
    int countmaterialenergy = 0;

    while (countmaterialenergy < dcfenergy.Count)
    {
        if (stopgoingthrough == 0)
        {
            double                radionuclideenergydouble                =
System.Convert.ToDouble(radionuclideenergy);
            double                materialenergydouble                =
System.Convert.ToDouble(dcfenergy[countmaterialenergy]);

            if (materialenergydouble > radionuclideenergydouble)
            {
                stopgoingthrough++;

                double                stepsize                =
(System.Convert.ToDouble(dcfToGy[countmaterialenergy])
System.Convert.ToDouble(dcfToGy[countmaterialenergy - 1])) /
(System.Convert.ToDouble(dcfenergy[countmaterialenergy])
System.Convert.ToDouble(dcfenergy[countmaterialenergy - 1]));

                double                interpolatedvalue                =
System.Convert.ToDouble(dcfToGy[countmaterialenergy - 1]) + stepsize *
(radionuclideenergydouble - System.Convert.ToDouble(dcfenergy[countmaterialenergy
- 1]));

                dcfToGylinearinterpolation.Add(interpolatedvalue);

                stopgoingthrough++;
            }
        }

        countmaterialenergy++;
    }
}

```

```

    stopgoingthrough = 0;
}

stopgoingthrough = 0;

foreach (string radionuclideenergy in listRadionuclideEnergy)
{

    int countmaterialenergy = 0;

    while (countmaterialenergy < dcfenergy.Count)
    {
        if (stopgoingthrough == 0)
        {

            double                radionuclideenergydouble                =
System.Convert.ToDouble(radionuclideenergy);
            double                materialenergydouble                =
System.Convert.ToDouble(dcfenergy[countmaterialenergy]);

            if (materialenergydouble > radionuclideenergydouble)
            {
                stopgoingthrough++;

                double                stepsize                =
(System.Convert.ToDouble(GyToSv[countmaterialenergy])
System.Convert.ToDouble(GyToSv[countmaterialenergy - 1])) /
(System.Convert.ToDouble(dcfenergy[countmaterialenergy])
System.Convert.ToDouble(dcfenergy[countmaterialenergy - 1]));

                double                interpolatedvalue                =
System.Convert.ToDouble(GyToSv[countmaterialenergy - 1]) + stepsize *
(radionuclideenergydouble - System.Convert.ToDouble(dcfenergy[countmaterialenergy
- 1]));

                GyToSvlinearinterpolation.Add(interpolatedvalue);

                stopgoingthrough++;
            }
        }
        countmaterialenergy++;
    }

    stopgoingthrough = 0;
}

```

```

    }
}

public override void OnEpisodeBegin()
{
    RadiationArea3d radiationarea3d =
(RadiationArea3d)arena.GetComponent(typeof(RadiationArea3d));
    radiationarea3d.ResetSource();
    MeasurementDoubles.Clear();
    MeasurementPositions.Clear();
    MeasurementDoublesNormalized.Clear();
    double HighestMeasurement = 0d;

    for (int i = 0; i < 10; i++)
    {
        //Randomly taking measurements - the space can be constrained or enlarged
        //depending on the training scenario. This is just an example.
        //In this example measurements are taken from within a 2m x 1m x 2m space
        //(assuming the detector originates at 0,0,0).
        float xposition = UnityEngine.Random.Range(-1f, 1f);
        float yposition = UnityEngine.Random.Range(1, 2f);
        float zposition = UnityEngine.Random.Range(-1f, 1f);
        Display.transform.localPosition = new Vector3(xposition, yposition, zposition);
        MeasurementPositions.Add(new Vector3(xposition, yposition, zposition));
        Measurements[i].transform.localPosition = new Vector3(xposition, yposition,
zposition);
        double reading = CalculateRadiationReading();
        MeasurementDoubles.Add(reading);
        if (reading > HighestMeasurement) { HighestMeasurement = reading; }
    }
    foreach (double measurement in MeasurementDoubles)
    {
        MeasurementDoublesNormalized.Add(measurement / HighestMeasurement);
    }
    CountingSteps = 1;
}

public override void CollectObservations(VectorSensor sensor)
{
    foreach (Vector3 vector in MeasurementPositions)
    {
        //Normalization so that values trained to the network are within -1 to 1 range if

```

```

required
    //sensor.AddObservation(vector.x / [factor]f);
    sensor.AddObservation(vector.y - 1f);
    //sensor.AddObservation(vector.z / [factor]f);
}

foreach (double measurement in MeasurmentDoublesNormalized)
{
    sensor.AddObservation(System.Convert.ToSingle(measurement));
}

}

public override void OnActionReceived(ActionBuffers actionBuffers)
{
    Vector3 guess = new Vector3(0, 0, 0);

    //Normalizing for training environment space of 2m x 10m x 10m - guess will return
    -1 to 1 as actions and the source is randomly located within that environment.
    guess = new Vector3(actionBuffers.ContinuousActions[0] * 5f,
    actionBuffers.ContinuousActions[1] * 2f, actionBuffers.ContinuousActions[2] * 5f);

    guessicon.transform.localPosition = guess;

    Vector3 correctanswer = new Vector3(0, 0, 0);
    correctanswer = SingleSourceML.transform.localPosition;

    //Checking how far the 'guess' is from the correct answer in terms of distance.
    float howwrong = (guess - correctanswer).magnitude;

    float reward = 0f;

    //This is an example of a simplistic reward response. Values can be modified as
    needed.
    //In this example the goal is to allow guesses to take place until either the guess is
    within 0.5 world units or 100 guesses have been made.
    if (howwrong > 6) reward = -1f;
    if (howwrong < 6) reward = -0.75f;
}

```

```

if (howwrong < 5) reward = -0.5f;
if (howwrong < 4) reward = -0.1f;
if (howwrong < 3) reward = 0.001f;
if (howwrong < 2) reward = 0.02f;
if (howwrong < 1.5) reward = 0.1f;
if (howwrong < 1.25) reward = 0.5f;
if (howwrong < 1) reward = 0.6f;
if (howwrong < 0.75) reward = 0.7f;
if (howwrong < 0.5)
{
    reward = 1f;
    SetReward(reward);
    EndEpisode();
}

    SetReward(reward);

if (CountingSteps == 100)
{
    EndEpisode();
}

    CountingSteps = CountingSteps + 1f;
}

void Start()
{

}

void Update()
{

}

```

(Previously provided radiation class code goes here - removed for brevity - it is the same as the VR code with only small modifications to return the measurement internally)

Table 9 - RadiationArea3d.cs C# code

```

using System.Collections;
using UnityEngine;

```



```

public class RadiationArea3d : MonoBehaviour
{
    public GameObject radiationsource;

    public void ResetSource()
    {
        //On episode creation the radiation source is positioned randomly within the play
        space of 10m x 2m x 10m
        float xposition = UnityEngine.Random.Range(-5f, 5f);
        float yposition = UnityEngine.Random.Range(0, 2f);
        float zposition = UnityEngine.Random.Range(-5f, 5f);
        radiationsource.transform.localPosition = new Vector3(xposition, yposition,
zposition);
    }
}

```

Table 10 - RadiationCalculationForML.cs C# code

```

using System;
using System.Collections;
using System.Collections.Generic;
using UnityEngine.UI;
using UnityEngine;
using System.IO;
using Unity.Barracuda;

public class RadiationCalculationForML : MonoBehaviour {

    public GameObject Measurement;
    public GameObject newMeasurement;
    public List<GameObject> Measurements = new List<GameObject>();
    public double MaxMeasurement;
    public GameObject CameraTransform;
    public GameObject CameraSwitch;
    private Vector3 RoundedPos;
    private List<Vector3> RoundedPoses = new List<Vector3>();
    private bool matchcheck;
    private double DoubletoSingleTest;
    public GameObject ScaleMaxText;
    public GameObject ScaleMinText;
    public GameObject DisplayMaxText;
    private int ArraySize;
    public GameObject[] Measurements10;
    public GameObject RadData;

```

```

public GameObject Display;
public double Reading;
private double DosePerSourceEnergy;
private float temp3;
private float distance;
private double reductionfactor;
private double totalreductionfactor;
private Vector3 EntryPoint;
private Vector3 ExitPoint;
private Ray ray;
private Ray raybk;
private List<RaycastHit> shieldhit = new List<RaycastHit> ();
private List<RaycastHit> shieldhit2 = new List<RaycastHit> ();
public List<double> listshieldobjectpenetrationdistances = new List<double> ();
public List<GameObject> listShieldGameObjects = new List<GameObject> ();
public List<string> dcfenergy = new List<string> ();
public List<string> dcfToGy = new List<string> ();
public List<double> dcfToGylinearinterpolation = new List<double>();
public List<string> GyToSv = new List<string> ();
public List<double> GyToSvlinearinterpolation = new List<double>();
public List<GameObject> RadionuclidesList = new List<GameObject>();
public List<string> listRadionuclideEnergy = new List<string>();
public List<string> listRadionuclides = new List<string>();
public NNModel GuessingModel;
public Model model;
private IWorker worker;
private int viewcount;
public GameObject guesslocation;
public GameObject DistanceScreenDisplay;
public LineRenderer LineRenderrer;

//Initalizing input with data
public float[] input =
{
    1f, 1f, 1f,
    1f, 1f, 1f,
    1f, 1f, 1f,
    1f, 1f, 1f,
    1f, 1f, 1f,
    1f, 1f, 1f,
    1f, 1f, 1f,
    1f, 1f, 1f,
    1f, 1f, 1f,
    1f, 1f, 1f,
    1f,
    1f,

```

```

        1f,
        1f,
        1f,
        1f,
        1f,
        1f,
        1f,
        1f
    };

    public void MakeAGuess()
    {
        int ArraySizeGuess = 0;

        while (ArraySizeGuess < 10)
        {
            input[ArraySizeGuess*3]           =
Measurements10[ArraySizeGuess].GetComponent<NNInputData>().x/3;
            input[(ArraySizeGuess*3)+1]       =
Measurements10[ArraySizeGuess].GetComponent<NNInputData>().y/2;
            input[(ArraySizeGuess*3)+2]       =
Measurements10[ArraySizeGuess].GetComponent<NNInputData>().z/3;
            ArraySizeGuess = ArraySizeGuess + 1;
            print("Adding coordinates");
        }

        ArraySizeGuess = 0;

        float MaxCurrentMeasurement = 0;

        while (ArraySizeGuess < 10)
        {
            if (MaxCurrentMeasurement <
Measurements10[ArraySizeGuess].GetComponent<NNInputData>().reading)
            {
                MaxCurrentMeasurement =
Measurements10[ArraySizeGuess].GetComponent<NNInputData>().reading;
            }
            ArraySizeGuess = ArraySizeGuess + 1;
        }

        ArraySizeGuess = 0;

        while (ArraySizeGuess < 10)

```

```

    {
        input[ArraySizeGuess + 30] =
Measurements10[ArraySizeGuess].GetComponent<NNInputData>().reading /
MaxCurrentMeasurement;
        ArraySizeGuess = ArraySizeGuess + 1;
        print("Adding Measurements");
    }

    var inputtensor = new Tensor(1, 40, input);
    worker.Execute(inputtensor);

    var outputTensor = worker.PeekOutput();

    guesslocation.transform.position = new Vector3(outputTensor[0] * 5,
outputTensor[1] * 2, outputTensor[2] * 5);

    DistanceScreenDisplay.GetComponent<UnityEngine.UI.Text>().text =
(Vector3.Distance(guesslocation.transform.position,
RadionuclidesList[0].transform.position) + " m");

    LineRenderer.startColor = Color.blue;
    LineRenderer.endColor = Color.blue;
    LineRenderer.startWidth = 0.05f;
    LineRenderer.endWidth = 0.05f;
    LineRenderer.SetPosition(0, guesslocation.transform.position);
    LineRenderer.SetPosition(1, RadionuclidesList[0].transform.position);
}

void Start () {
    (GuessingModel);
    worker = WorkerFactory.CreateWorker(WorkerFactory.Type.CSharpBurst,
model);
    ArraySize = 10;
    Measurements10 = new GameObject[ArraySize];

    MaxMeasurement = 0;

    TextAsset AmbDose =
(TextAsset)Resources.Load("AmbientDoseEquivalentCSV");
}

```

```

string[] linesFromfile = AmbDose.text.Split("\n"[0]);
int countingn = 0;
while (countingn < linesFromfile.Length-1)
{
    var line = linesFromfile[countingn];
    print(linesFromfile.Length);
    var values = line.Split(',');

    dcfenergy.Add(values[0]);
    dcfToGy.Add(values[1]);
    GyToSv.Add(values[2]);

    countingn++;
}

TextAsset RadDecay =
(TextAsset)Resources.Load("RadiationDecayEnergiesCSV");

RadionuclidesList.AddRange (GameObject.FindGameObjectsWithTag
("RadionuclideSource"));

foreach (GameObject Source in RadionuclidesList) {
string temp4 =
Source.GetComponent<RadiationSource>().radionuclide.ToString();

int stopgoingthrough = 0;

foreach(string radionuclideenergy in listRadionuclideEnergy) {

int countmaterialenergy = 0;

while(countmaterialenergy < dcfenergy.Count)
{
if (stopgoingthrough == 0) {

double radionuclideenergydouble =
System.Convert.ToDouble(radionuclideenergy)
double materialenergydouble =
System.Convert.ToDouble(dcfenergy[countmaterialenergy]

if (materialenergydouble > radionuclideenergydouble)
{

```

```

        stopgoingthrough

        double          stepsize          =
(System.Convert.ToDouble(dcfToGy[countmaterialenergy])
System.Convert.ToDouble(dcfToGy[countmaterialenergy-
1]))/(System.Convert.ToDouble(dcfenergy[countmaterialenergy]
System.Convert.ToDouble(dcfenergy[countmaterialenergy-1]));

        double          interpolatedvalue  =
System.Convert.ToDouble(dcfToGy[countmaterialenergy-1]) + stepsize *
(radionuclideenergydouble - System.Convert.ToDouble(dcfenergy[countmaterialenergy-
1]));

        dcfToGylinearinterpolation.Add(interpolatedvalue);

        stopgoingthrough++;
    }
        }
        countmaterialenergy++;
    }
    stopgoingthrough = 0;
}
stopgoingthrough = 0;

foreach(string radionuclideenergy in listRadionuclideEnergy) {

    int countmaterialenergy = 0;

    while(countmaterialenergy < dcfenergy.Count)
    {
        if (stopgoingthrough == 0) {

            double          radionuclideenergydouble  =
System.Convert.ToDouble(radionuclideenergy
            double          materialenergydouble      =
System.Convert.ToDouble(dcfenergy[countmaterialenergy]);

            if (materialenergydouble > radionuclideenergydouble)
{

```

```

        stopgoingthrough++;

        double          stepsize          =
(System.Convert.ToDouble(GyToSv[countmaterialenergy])
System.Convert.ToDouble(GyToSv[countmaterialenergy-
1]))/(System.Convert.ToDouble(dcfenergy[countmaterialenergy])
System.Convert.ToDouble(dcfenergy[countmaterialenergy-1]));

        double          interpolatedvalue  =
System.Convert.ToDouble(GyToSv[countmaterialenergy-1]) + stepsize *
(radionuclideenergydouble - System.Convert.ToDouble(dcfenergy[countmaterialenergy-
1]));

        interpolation

        GyToSvlinearinterpolation.Add(interpolatedvalue);

        stopgoingthrough++;

radionuclides energy
    }
    }
    countmaterialenergy++;
}
    stopgoingthrough = 0;
}
}

void Update () {

    double temp;
    //Displaying reading to screen
    Display.GetComponent<UnityEngine.UI.Text>().text =
(Mathf.Round((float)((Reading * 100) / formatNumber(Reading, out temp))) / 100) + " " +
formatUnit(Reading);

    //For placing visuals in environment and having them only be in an grid and not
overlapping. If a measurement already exists do not overwrite existing data.
    //If this is run with the radaition calculation in real time in an update loop then the
calculation is done each update. Integration time for simulating
    //the response of a detector can be added into this by only calling for a
measurement (and subsequent visual) at an interval of every few seconds.

```

```
//This is only called when a new measurement is received when receiving data from the
detector via Bluetooth connection
```

```
        RoundedPos = new
Vector3((Mathf.Round(CameraTransform.transform.position.x * 10)) / 10,
(Mathf.Round(CameraTransform.transform.position.y * 10)) / 10,
(Mathf.Round(CameraTransform.transform.position.z * 10)) / 10);
```

```
        foreach (Vector3 v3 in RoundedPoses)
```

```
        {
```

```
            if (v3 == RoundedPos)
```

```
            {
```

```
                matchcheck = true;
```

```
            }
```

```
        }
```

```
        if (matchcheck == false)
```

```
        {
```

```
            RoundedPoses.Add(RoundedPos);
```

```
            newMeasurement = Instantiate(Measurement,
```

```
RoundedPos, Quaternion.identity);
```

```
            newMeasurement.GetComponent<MeasurementValue>().Measurementd =
Reading;
```

```
            newMeasurement.GetComponent<NNInputData>().x =
CameraTransform.transform.position.x;
```

```
            newMeasurement.GetComponent<NNInputData>().y =
CameraTransform.transform.position.y;
```

```
            newMeasurement.GetComponent<NNInputData>().z =
CameraTransform.transform.position.z;
```

```
            newMeasurement.GetComponent<NNInputData>().reading = (float)Reading;
```

```
            Measurements.Add(newMeasurement);
```

```
            //When a new measurement is visualized all measurements are
evaluated to determine the highest measurement and then all measurements are
rescaled.
```

```
            //This is for visual effect. Color scaling can be manually tracked
alternatively or done through other means.
```

```
            ColorMeasurements();
```

```
            Measurements10[ArraySize] =
```



```

newMeasurement;
                                ArraySize = ArraySize + 1;
                                if (ArraySize > 9) { ArraySize = 0; }
                                }
                                matchcheck = false;
                                ScaleMaxText.GetComponent<TextMesh>().text =
(Mathf.Round((float)((MaxMeasurement * 100) / formatNumber(MaxMeasurement, out
temp))) / 100) + " " +formatUnit(MaxMeasurement);

}

string formatUnit (double n) {
    string output;
    output = "Test";
    if (n < 0.00000000000001) output = "Low";
    if (n >= 0.000000001) output = "nSv/h";
    if (n >= 0.000001) output = "µSv/h";
    if (n >= 0.001) output = "mSv/h";
    if (n >= 1) output = " Sv/h";
    if (n >= 1000) output = "kSv/h";
    if (n >= 100000)
        output = "Off/S";
    return output;
}

double formatNumber (double n, out double output) {
    output = 1.0d;

    if (n < 0.00000000000000000001) {
        output = 0;
    }
    if (n >= 0.0000000000001)
    {
        output = 0.000000000001d;
    }
    if (n >= 0.000000001)
    {
        output = 0.000000001d;
    }
    if (n >= 0.000001)
    {
        output = 0.000001d;
    }
}

```

```

    }
    if (n >= 0.001) {
        output = 0.001d;
    }
    if (n >= 1) {
        output = 1;
    }
    if (n >= 1000) {
        output = 1000;
    }
    if (n >= 1000000) {
        output = 1000000;
    }
    if (n >= 100000000) {
        output = 100000000;
    }
    if (n >= 10000000000) {
        output = 10000000000;
    }
    return output;
}

public void outputReadingToFile() {
    float counter;
    counter = 0;
        do {
            CalculateRadiationReading();
            string path = Application.dataPath + "../X-Ray
Outputs/Readingoutput.txt";
            StreamWriter writer = new StreamWriter(path,true);
            writer.WriteLine(Reading);
            writer.Close();
            transform.Translate(0,0.001f,0);
            counter++;
        } while (counter < 100);
}

void ColorMeasurements(){

        if (Reading > MaxMeasurement)
        {
            //Color scaling can be define by the user
            as needed.
            double temp;

```

```

    DisplayMaxText.GetComponent<UnityEngine.UI.Text>().text =
(Mathf.Round((float)((MaxMeasurement * 100) / formatNumber(MaxMeasurement, out
temp))) / 100) + " " + formatUnit(MaxMeasurement);
        MaxMeasurement = Reading;
        if (MaxMeasurement > 0.01){
            MaxMeasurement = 0.01;
        }
    foreach (GameObject oldmeasurement
in Measurements)
    {
        DoubletoSingleTest =
oldmeasurement.GetComponent<MeasurementValue>().Measurementd/MaxMeasurem
ent;
        if (DoubletoSingleTest > 1){
            DoubletoSingleTest = 1;
        }

        oldmeasurement.GetComponent<Renderer>().material.color = new
Color(1f, 0, 0);

        if (DoubletoSingleTest < 0.75){
            oldmeasurement.GetComponent<Renderer>().material.color = new Color(1f,
0.5f, 0);
        }
        if (DoubletoSingleTest < 0.5){
            oldmeasurement.GetComponent<Renderer>().material.color = new Color(1f, 1f,
0);
        }
        if (DoubletoSingleTest < 0.1){
            oldmeasurement.GetComponent<Renderer>().material.color = new Color(1f, 1f,
0.5f);
        }
        if (DoubletoSingleTest < 0.075){
            oldmeasurement.GetComponent<Renderer>().material.color = new Color(0.5f,
1f, 0.5f);
        }
        if (DoubletoSingleTest < 0.05){
            oldmeasurement.GetComponent<Renderer>().material.color = new Color(0,
0.75f, 0.75f);
        }
        if (DoubletoSingleTest < 0.01){

```

```

        oldmeasurement.GetComponent<Renderer>().material.color = new Color(0, 0,
0.75f);
    }
} else{
    DoubletoSingleTest =
Reading/MaxMeasurement;
    if (DoubletoSingleTest > 1){
        DoubletoSingleTest = 1;
    }
    newMeasurement.GetComponent<Renderer>().material.color = new Color(1f, 0,
0);
    if (DoubletoSingleTest < 0.75){
        newMeasurement.GetComponent<Renderer>().material.color = new Color(1f,
0.5f, 0);
    }
    if (DoubletoSingleTest < 0.5){
        newMeasurement.GetComponent<Renderer>().material.color = new Color(1f, 1f,
0);
    }
    if (DoubletoSingleTest < 0.1){
        newMeasurement.GetComponent<Renderer>().material.color = new Color(1f, 1f,
0.5f);
    }
    if (DoubletoSingleTest < 0.075){
        newMeasurement.GetComponent<Renderer>().material.color = new Color(0.5f,
1f, 0.5f);
    }
    if (DoubletoSingleTest < 0.05){
        newMeasurement.GetComponent<Renderer>().material.color = new Color(0,
0.75f, 0.75f);
    }
    if (DoubletoSingleTest < 0.01){
        newMeasurement.GetComponent<Renderer>().material.color = new Color(0, 0,

```

```
0.75f);  
    }  
}
```

Annex IV: Dose conversion factors

The following table of dose conversion factors was used to convert fluence into H*(10) [96].

Table 11 - Dose conversion factors for photon fluence to Gy to H*(10)

Photon energy (keV)	Fluence to pGy	Gy to H*(10) (Sv)
10	7.6	0.0608
15	3.21	0.8346
20	1.73	1.0553
25	1.083302	0.914116
30	0.739	0.8129
40	0.438	0.64386
50	0.328	0.54776
60	0.292	0.50808
70	0.300467	0.519584
80	0.308	0.52976
100	0.372	0.6138
150	0.6	0.894
200	0.856	1.1984
300	1.38	1.8078
400	1.89	2.3814
500	2.38	2.9274
600	2.84	3.4364
800	3.69	4.3911
1000	4.47	5.2299
1500	6.12	7.038
2000	7.51	8.5614
3000	9.89	11.1757
4000	12	13.44
5000	13.9	15.429
6000	15.8	17.538
8000	19.5	21.645
10000	23.2	25.52

Annex V: Subjective lessons learned during the development and deployment of VR training scenarios

This Annex is a collection of the subjective lessons learned from the deployment of VR training during a number of different activities between 2016 and 2021. The views expressed herein are based on the experience of the author and represent personal experience conducting these activities making on observations of the participants and based on their verbal feedback received. The views are personal and do not represent the views of any organization.

More than 500 people have been involved in exercises under the supervision of the author^{36,37}. The forums where this experience was gained include the activities described in Section 7.4, as well as the following activities:

- 2017-11: IAEA Regional Workshop on Assessment and Prognosis during a Nuclear or Radiological Emergency³⁸
 - International workshop with interactive virtual reality emergency response exercises integrated into a weeklong classroom content
- 2017-11 IAEA International Conference on Physical Protection of Nuclear Material and Nuclear Facilities

³⁶ This figure is a best estimate, as during many of these events it is challenging to keep an exact count of the number of participants.

³⁷ This figure does not include the number of students who activity participant in these events during classroom training, where a single user is inside the VR environment but a classroom of 20-30 people are involved in the training activity.

³⁸ <https://www.iaea.org/newscenter/news/iaea-hosts-workshop-on-nuclear-or-radiological-accident-assessment-and-prognosis>

- Presentation on: Training and exercising the nuclear safety and nuclear security interface incident response through synthetic environment, augmented reality and virtual reality simulations (E. Waller & J. Chaput)³⁹
- 2018-05: Canadian Radiation Protection Association 2018 conference
 - Lessons learned from the use of virtual reality as a training tool for emergency response applications⁴⁰
- 2018-06: Ninth Meeting of Representatives of the Competent Authorities identified under the Early Notification Convention and the Assistance Convention
 - The use of virtual reality as an instruction and exercise tool for emergency response training
- 2018-09: IAEA General Conference
 - An Interactive Demonstration of Virtual Reality in Emergency Preparedness and Response: Using Technology to Enhance the IAEA's EPR Training Programme⁴¹
- 2018-10: IAEA School of Radiation Emergency Management
 - Conduct of interactive virtual reality emergency response exercises in the three-week course content.
- 2018-12: International Conference on the Security of Radioactive Material: The Way Forward for Prevention and Detection

³⁹ <https://www-pub.iaea.org/MTCD/Meetings/PDFplus/2017/cn254/cn254BookOfSynopses.pdf>

⁴⁰ <http://crpa-acrp.org/home/wp-content/uploads/2019/05/CRPA-2019-Presentation-abstracts.pdf>

⁴¹ <https://www.iaea.org/newscenter/news/virtual-emergency-response-experience-offered-at-the-iaea-general-conference>

- The Application of Virtual Reality to Support Training for Nuclear Security. Lessons learned from the use of virtual reality technology in emergency preparedness and response training and its applications for nuclear security training (M. Breitingner & J. Chaput)
- 2019-04: Technical Meeting on Advances In Emergency Preparedness And Response Technology and Arrangements
 - Lessons learned on the use of virtual reality in the IAEA School of Radiation Emergency Management
- 2019-05: Canadian Radiation Protection Association 2019 conference
 - Lessons learned from the use of virtual reality as a training tool for emergency response applications⁴²

These lessons learned are subjective. They are provided so that others who choose to develop similar training experiences can benefit from the experience gained during this research.

Development lessons learned

The following lessons were learned related to the development of VR scenarios:

- The resources required to develop a VR scenario can be significant. The skill set for VR development requires knowledge of programming, 3d modeling, texturing, 3d lighting, game design and human interaction/hardware interface design. This can be daunting for someone considering entering this field without previous experience. There are many helpful tutorials available online that cover all these

⁴² <http://crpa-acrp.com/home/wp-content/uploads/2019/04/CRPA-2019-Scientific-Program-12-Apr-2019.pdf>

aspects and teach the necessary skills. However, when relying on such tutorials it is important to make sure that the version of the development software you are using is not too new compared to the tutorial, as things quickly become dated and if you are using the latest version of the software there may be interface and terminology changes.

- Development of any training scenario requires consideration towards what the learning objectives should be and serious consideration for why it should be conducted in VR instead of some other training format. For example, development of a VR scenario where a user can conduct an activity that can be done easily in a classroom (such as navigating a menu on a detector to explore functions) would not be an optimal experience in a VR environment. Activities that are impractical, such as exploring high dose rate environment or structures which are not possible to explore otherwise, are better suited for such training.
- The design for running a scenario will directly impact the development time required. Designing scenarios which are supervised by an instructor who walks the player through the expected actions, following a script, taking notes and asking questions, was found to be the least resource intensive way to design and deploy a training scenario. In terms of person power requirements, this requires a fulltime controller present for each headset station. For classroom settings, learning objectives can be achieved without the need for each student to directly play out a scenario. Having a volunteer in front of a classroom sharing their visual view on a screen has proven it can work as a way to both engage others in the class who may be unwilling to try the hardware (for various personal reasons)

and also were useful in breaking up the tedium of a dry technical set of lectures with an interactive challenge.

- Development tools specific for aiding in creating VR content have evolved considerably since the research began in 2016. The tools that were originally used within Unity required a considerable learning curve to be able to integrate everything into the environment with VR interactions that were comfortable and felt natural to the user. However, a new user looking to jump into this path for development should look closely at available tutorials and toolkits (free and those that are available to purchase). These newer toolkits can save a large amount of time in the development process as they address frequently needed topics (like premade locomotion solutions or built in solutions for quickly making an object interactive) and they often feature tutorial environments which can help instruct a novice user on how they may want to design their own environments.
- Initially, the development of virtual environments focused significant resources on visual fidelity. The goal was to present the user with a ‘photorealistic’ environment that was difficult to distinguish from the real world. Tools such as photogrammetry software was used to produce some ‘photorealistic’ approximations of radiation detectors and interior and exterior environments⁴³ without the need for manually 3d modelling an environment. The Google Project Tango devices used in Section 4.2.3 were able to use an application called the

⁴³ Photogrammetry is a process which leverages photographs of an environment or object, taken from a multitude of angles in order to extract the positions of where the camera was when each photo was taken and generate a point cloud of the environment. This in turn can be made into a polygon mesh that can be textured using the photographs generally providing a very realistic visual. This technique is used frequently in the videogame, film and movie industries.

Constructor Developer Tool which could manually scan an environment and in real-time construct a 3d mesh of the environment with vertex colours and capturing images for texturing during post processing afterwards⁴⁴. An example of the results can be seen in Figure 138 and Figure 139. These results while serviceable may have areas that would feature incomplete elements of the scan as seen in Figure 140. Manual repairing of the mesh using the tools available in MeshLab was possible as can be seen in Figure 141, but ultimately the need for this was abandoned as more experience was gained.



Figure 138 - Example environment for scanning into 3d

⁴⁴ Constructor Developer Tool by Google <https://play.google.com/store/apps/details?id=com.projecttango.constructor>. Note: the meshing capability was also available in the development kit in Unity as well.



Figure 139 - Results from Constructor Developer Tool



Figure 140 - Results of mesh capturing highlighting imperfections in the mesh

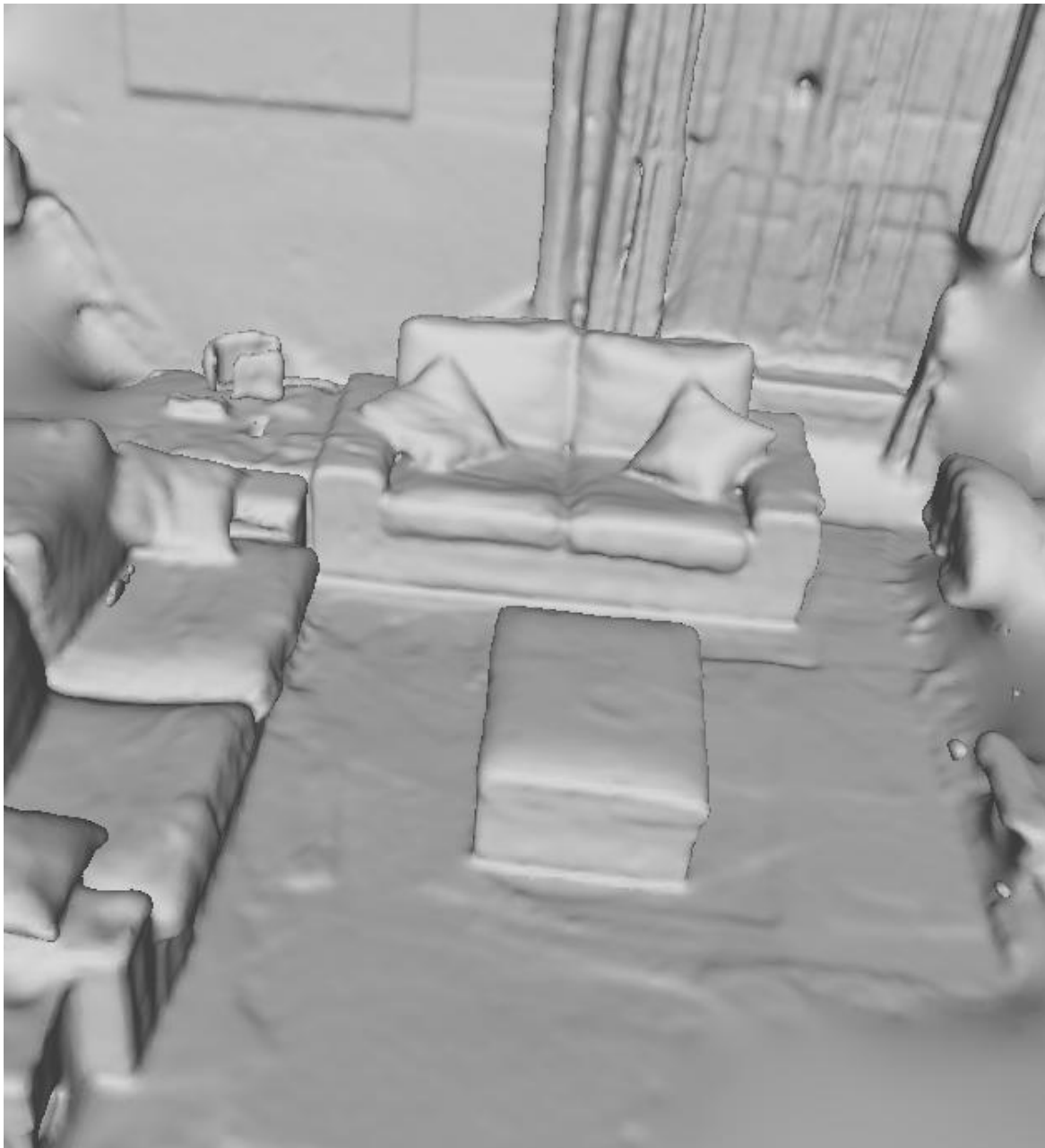


Figure 141 - Results of mesh repairing in MeshLab afterwards

- More simplistic 3d graphics which, while still remaining visually impressive, were not at the level of photo realism were determined to be the most suitable for the training environments. There were two reasons for this:

- The efforts required to create a photorealistic textured model of a detector were not significant (less than 1 day), but to construct a large environment took a considerable amount of time, both to acquire the data needed and to process the data into a complete model. Once initial models were constructed and combined into a VR environment, it was seen that although users did recognize the photorealistic nature of the environments, they would frequently become distracted by any visual imperfections. The time investment required to remove these visual imperfections was not seen as worthwhile for rapidly iterating during the research in this thesis. However, larger scale teams for commercial projects would likely see some benefit.
- When developing a VR environment, consistency between the visuals of all the elements in the scene was recognized as an important requirement. Mixing low detailed objects with high detailed objects with different visual styles (example - cartoonish 3d models and photorealistic radiation detectors), creates a muddled and distracting experience.

Integration into training program lessons learned

The following lessons were learned related to the integration of VR into a training program using the Systematic Approach to Training:

- At the onset, detailed consideration is required to determine why VR being integrated into a training program. As VR can provide supplemental exercises to enhance classroom learning activities, it is a useful tool. However, when examining how and where to integrate VR into a training course, a close look into

why an exercise is being developed for VR as opposed to just using a 3d program on a monitor, needs to be considered to avoid the entire VR aspect being labelled as a gimmick with minimal or no added value.

- The hardware and software life cycle of any developed training needs to also be considered. VR technology is rapidly moving forward. Within this thesis research the VR technology which began with the HTC Vive progressed and was used on the Oculus Quest, the Valve Index and finally the Oculus Quest 2. The rapid progression of the technology brings greater capabilities but also more expectations from users as they become familiar with VR and showing an older design for an experience as a component in a training course may make the overall material look dated quickly. In practice, it was seen that ensuring a scenario is not designed to relying on a hardware exclusive feature such as the touch pad on the HTC Vive controller or the finger tracking in the Valve Index controllers made porting and verifying the material worked on other hardware easier.
- The operating environment during training has an impact on the scenario conduct (running within the headset locally or remotely on a PC). When running a scenario, the 3d processing can be done on a remote computer system and provided to the user via a video cord (such as with the HTC Vive), run locally on the headset (such as with the Oculus Quest 1 and 2) or a mixture with the user wearing the computer hardware as a backpack or leveraging wireless transmission (such as Oculus Air Link) to send the video and audio to the headset. Initially the training deployed using a laptop with a cable running to the

HTC Vive during the conduct of scenarios. This proved extremely reliable and for most cases and the cable was not cumbersome to the user. However, during training occasionally the controller was needed to micromanage the tethered cable as the users were not used to managing its position while in VR. Running a scenario wirelessly was experimented on using the Oculus Quest with processing being done on the same laptop but the video signal sent via a dedicated wireless network that would be setup each time training was conducted. This also proved successful in allowing for cable free training but due to the nature of wireless technology, the local environment could never be guaranteed compatible (i.e., there could be too many other wifi signals causing interference). How the training environment will be delivered needs consideration as the benefits of a wireless setup may not be worth the issues encountered when trying to setup in an unknown environment where reliability is not guaranteed. Having a scenario running on the standalone headset locally is an alternative way to manage this however if the scenario is to be streamed to an external display for a classroom activity, wireless technology is still being used. Standalone headsets also come with other constraints such as reduced local processing capabilities of the Oculus Quest and Quest 2 requiring more optimized design of the environments and lowering of the visual fidelity.

Human factors lessons learned

The following lessons were learned related to human factors in the design of VR scenarios:

- Users in virtual reality environments tend to feel very much exposed and on display. When the headset is on, it removes their awareness of the people and objects around them. It was seen that this can make some people feel uncomfortable while they participate in a scenario and often would contribute to some people feeling reluctant to volunteer to try the technology. This can be overcome by ensuring there is the option of privacy during the conduct of a scenario. For example, portable walls can be used to surround the play space of the user so that they feel less exposed and more protected with just them and the controller of the scenario meanwhile their visual information can still be routed to a larger screen to support a classroom of people following the activity.
- Users, most frequently those who have never done room scale VR, often lose track of their physical location in the room as they engage in a scenario. While this is desirable in terms of making them feel present in the virtual environment, it can cause them to step out of boundaries (often overlooking the built in warnings from the chaperon or guardian system) and is a potential hazard for damaging equipment or tripping the user over objects outside of the designated play space. This can be accounted for with the introduction of something physical, like a raised carpet which is only inside the play space. This gives a user a tactile feeling when they leave the 'safe' environment that they can feel, even with shoes on, and helps to keep them from exceeding the designated play space.
- Not all users are comfortable interacting in VR while standing. Having a chair with wheels that can rotate 360 degrees available so that any participant can use

this to help them experience the content is important. Equally important is testing and ensuring that the entire experience can be conducted while seated in a chair.

- In public settings, such as a classroom, frequently users can overheat while wearing a VR headset due to the pressure of being on display and possibly the physical actions required in the scenario. Having a large fan available and turned on pointed at the play space can help to address this issue.
- Although a developed scenario is best run as a built project for the hardware being used (i.e., a windows executable or android APK file), there were several situations when having the Unity project available and running the scenarios in the editor was useful. For example, when engaging with experts who might have valuable feedback regarding placement of objects in a scene that would enhance realism, being able to make those changes in the editor and apply them immediately in a rebuilt scene (or running directly from the editor) was useful.
- Even before Covid-19, hygiene was an important issue. While using a headset a user may be sweating or possibly having some face make-up or cream rub off onto the headset foam or silicon surface (depending on the head-set type). Disposable paper covers that the user wears can provide a barrier between direct skin contact help to alleviate this issue to some degree and frequent cleaning of the elements of the headset that may contact skin is recommended.
- As mentioned previously, ensuring scenarios are accessible for a user who remains seated (for comfort or accessibility issues) is an important deployment and design consideration. Other forms of accessibility need to be considered include:

- Ensuring a scenario can be completed with only a single controller
- Verifying that the height of the user, if standing, does not make certain objects in the environment that they need to interact with inaccessible as they are too low or too high.
- Ensuring that if a user is unable to interact with the environment, but still able to wear the headset, that the controller of the scenario can manually move them to the points of interest so they can make observations.
- If the use of a player avatar is desired its use needs to be carefully considered. It is possible in VR to allow a player to see a representation of themselves, usually called an avatar. The avatar could be a cartoonish representation of a person or using an inverse kinematics system could be a human representation which provides to the player a visual of their feet and arms where the local position of each is estimated using the system. This creates a scenario whereas a designer, player comfort with the physical presence of their avatar needs to be considered. Aspects of this include the size, potential gender, skin colour and other features that may distract or potentially make a user uncomfortable. This was briefly experimented with during the design of the training scenarios but was ultimately abandoned as without an avatar representation players were successful in completing the training objects and therefore this was designated as unneeded for the use cases described. However, if multiplayer experiences are being designed then this issue will need to be solved so that players can visually 'see' each other (such as by letting the user choose their avatar when

they begin allowing for the selection of desired physical features, keeping the avatars cartoonish and generic in their depiction or other creative solutions).

- At some point, during the conduct of a training activity with a student unfamiliar in a VR environment, a controller may need to place their hands on the student to gently guide them where they need to go (for example helping them to turn the correct direction, helping them find a controller button or even preventing them from accidental injury such as tripping on a cable). The potential for any physical contact during the conduct of training needs to be clearly discussed, understood and agreed with the player before entering in VR. Unless there is an immediate safety issue (such as preventing a player from stumbling), it should be announced verbally and agreed to by the player prior to any physical contact to help correct their actions. Adhering to this principal while remaining professional at all times and avoiding any unnecessary contact with the player was shown to keep the player comfortable during the training and also encourage others who may be hesitant to participate due to concerns over such this issue, to volunteer to participate.

Tutorial and 4 scenarios lessons learned

The following lessons were learned related to the tutorial and the 4 scenarios discussed in the main body of the thesis.

- Tutorial
 - Whether in a private situation or in a public demonstration it should be anticipated that each user who may try a scenario has never used a VR headset before. Having a simple tutorial that encourages use of the basic

mechanics needed to progress in each scenario is essential. In the case of this research those basic mechanics were:

- Understanding the locomotion solution that will be used. This mechanic was taught during the tutorial by requiring the player to navigate from the starting point in the tutorial to the end point
- Understanding how to grab and move objects. This mechanic was taught during the tutorial at the beginning when users have a large quantity of blocks to grab and throw, as well as in the final section of the tutorial where users have some radiological sources and shielding blocks to manipulate
- Understanding that the environments are scaled to be real world in size and therefore the perspective for solutions needs to be based on that scaling. This mechanic was taught during the movement from the start to the end of the tutorial where a set of blocks was scaled progressively larger and larger until the player had to struggle to look up to see the top of the house sized large pile of blocks, giving them the awareness that in VR objects can be extremely large, like buildings in the real-world.
- Understanding how to take a radiation measurement. This mechanic was taught at the end of the tutorial where students were provided a table with a radiation detector, several sources and several bricks for shielding material to experiment with.

- The time spent in the tutorial should not be fixed. Different players learned the mechanics of the tutorial at different rates. As each player was always under the supervision of the controller, it was a subjective decision of the controller how long the players would stay in the tutorial section. Some students completed the tutorial in 60 seconds. Some took their time and experimented and remained in the tutorial for 5 or more minutes.
- Transport accident with a radiological source in a vehicle (scenario 1)
 - The first scenario in a sequence of scenarios should not be overly challenging to a user. This scenario was deliberately 'easy', in that the student could see all hazards from the starting point and would be able to conduct all activities except for examining the source package without moving from where they started. This is important as depending on the comfort of the user, if there is a sequence of scenarios, they may 'give up' if the first one is overly challenging (such as the port based scenario) or requires the use of mechanics they are still actively learning to advance.
 - Students will overlook what may seem obvious to the designer. The ambient traffic in the opposing lane in this scenario was frequently overlooked by students. They would be fixated on the accident scene and while approaching to look for hazards they would stumble into the other lane of traffic and get 'hit' by a car. Collisions were not enabled on vehicles so the player would harmlessly pass through the vehicle. However the shock and realization that they overlooked such an obvious

hazard was very common after a moment of disorientation as their view point clipped through the vehicle geometry. Although in the design phase it seemed like an obvious, easy to spot hazard that students would see, it was not the case when students were in the scenario.

- Once a scene is constructed, reuse of a scene with small changes can completely change the response. The car crash scene was modified and reused at many different opportunities over the course of this research. It was modified to be set at night with the students provided a flashlight to help visibility but otherwise are required conducting the same activities. It was modified to have a large concrete shield where students would have to retrieve the now damaged package and place it into the concrete container, verifying it was successful. It was modified to be featured as part of a virtual press conference at a symposium for public information specialists who would get to view the scene and then would be challenged to provide answers to the public regarding what were the hazards at the scene in front of a virtual audience. The ease of reuse and modification of a single scene can be leverage in many different ways once it has been developed.
- Transport accident with radiological material in severe weather (scenario 2)
 - Environmental graphical effects such as wind and weather, even if not up to the standards of a AAA video game development company, are sufficient for communicating the situation to users during training. The weather effects used in this scene were rather primitive in their design.

However, the impact and the understanding of how this complicates the response to a scene were effectively communicated to the students. The desire for visual fidelity in all aspects of the creation and design of a scenario by the developing team should be managed with realistic views.

- Allowing the physics of the detector or other handheld objects to interact with the environment can make a big difference in how students approach a problem. In this scenario there were several drums standing on the roadway. The students were asked to physically examine them and take some radiation measurements around them (leading to the eventual discovery of the damaged broken drum). As soon as any student noticed that their radiation detector could interact and 'bump' or even tip over the drums in the scene their play style immediately changed. What started as large clumsy motions became more deliberate, including carefully angling the detector to scan the tops and sides of the drums. This slowed their progress forcing them to move in a more realistic manner and enhanced their approach in subsequent scenarios.
- Transport accident with a sea container carrying radiological material at a port (scenario 3)
 - A scenario that seems rather simple in the design phase can be very challenging, even for professional radiological monitoring personnel. This scenario featured 3 sources hidden in 3 boxes in a pile of ~50 other boxes, all looking similar. Although it only took a few seconds to find them, if you knew where they were positioned, almost every student

struggled initially in this scene. This was a positive experience as they were learning how the boxes were all shielding each other and in different amounts relative to how they were positioned between the sources and the detector. Allowing the students to move and position the boxes freely provided a very immersive and enjoyable experience as they developed their own sorting strategy to move and navigate through the pile.

- An innocuous detail of interactivity can provoke a significant positive reaction. In this scene there were two details that were not explained to the user during their narrative briefing. The first detail was that the crane in the environment was still hanging overtop the collecting of boxes as seen in Figure 142. The majority of students did not consider the hazard of the crane when they chose to approach the scene as they were fixated on the radiological puzzle they were about to solve. A detail such as this greatly supported the immersion of players who did notice the crane and told controllers they would not enter that area if it was unsafe. The second detail was the cones which marked out the cordon distance around the containers. The cones were arranged around the scene, and several were specifically put in areas where a measurement of $>100 \mu\text{Sv/h}$ (a frequently used international standard for establishing an inner cordon) would be measured if a user did conduct a survey to verify the cordon was properly established. For those users that chose to exercise this part of their emergency response procedures (this was not requested by the controllers as it was not a learning objective), they were surprised and

extremely satisfied to learn that they could reposition the cones to better mark that area.



Figure 142 - Overhead crane still representing a hazard in the scene

- Transport accident with radiological material on a train in a remote community (scenario 4)
 - The use of a scope to observe a scene from a distance was very well received by the users. The fact that they would have to close one eye in order to see clearly through the scope (due to the stereoscopic rendering in the headset) really helped to add to the immersion in this scene.
 - The scene conceptually was very much enjoyed by those with a background in transport accidents involving trains. The potential to run emergency response exercises for different scenarios with virtual trains as opposed to needing to take physical trains out of service was seen as an

incredibly useful potential application of this approach by a specific attendee.

Annex VI: Virtual Reality exercise feedback survey

Virtual Reality Simulations for
Radiological Transport Scenarios

Virtual Reality Simulations for Radiological Transport Scenarios

Your feedback is important. It provides us with information we need to improve and helps us to produce workshops and meetings that best fit your needs. We appreciate you taking the time to do this.

Instructions: Please place an (X) box that best represents your response to the statement presented (from strongly agree – strongly disagree). Please add any written comments about the course at end of this form (use additional sheets as necessary). Provide any other feedback as you see fit.

About you:



Name (optional):

Age:

What type of organization do you work for? (Gov't, TSO, consignor, carrier, etc)

In an emergency, would you fill the role of radiological assessor? YES NO

SURVEY

Statement	Rating				
	Strongly agree	Agree	Neutral	Disagree	Strongly disagree
General Pre-VR Questions (complete BEFORE the interactive session)					
I see value in VR for nuclear safety training					
I have experience with virtual reality technology					
I have experience with virtual reality for nuclear applications					
I am a video game player					
I follow the latest technological developments					

Figure 143 - VR transport survey feedback survey page 1

Virtual Reality Simulations for
Radiological Transport Scenarios

General Scenario Specific Questions					
Which scenarios did you try? (circle all that apply)	1	2	3	4	None
Which scenario did you find the most compelling? (circle one)	1	2	3	4	None
Why did you find it compelling?					
What types of facilities and activities would VR be best suited?					

Scenario Specific Questions: SCENARIO 1 (Van/car accident and medical sources)	
Who would you notify?	
What do you report?	
What hazards did you see?	
What immediate actions do you take?	

Scenario Specific Questions: SCENARIO 2 (Truck accident with drums)	
What hazards do you note before making entry to the cordoned-off area?	
What PPE should you be wearing?	
How many packages have been damaged?	
Describe the extent of contamination in the environment.	
Recommend clean-up (decon) methods and procedures for the decision makers	

Figure 144 - VR transport survey feedback survey page 2

Virtual Reality Simulations for
Radiological Transport Scenarios

Scenario Specific Questions: SCENARIO 3 (Pier-side accident loading high activity sources)	
How many sources did you locate?	
What other hazards may be at the scene?	
What authorities required notification?	
Recommend next steps for the recovery of the sources	

Scenario Specific Questions: SCENARIO 4 (Rail accident of Type B packages of fissile material)	
What cordon area should be established?	
What are the main hazards present at the scene?	
What response actions would you recommend?	
What PPE should responders wear inside cordon area?	



Statement	Rating				
	Strongly agree	Agree	Neutral	Disagree	Strongly disagree
General Post-VR Questions (complete AFTER in the interactive session)					
I see value in VR for nuclear safety training					
I could easily understand the scenario concepts					
I found the VR scenarios realistic					
I felt comfortable in the VR environment					
I was not disoriented and/or did not feel dizzy while in VR					
The length of time for the scenarios was sufficient					
I would recommend VR as an option for my training organization					

Figure 145 - VR transport survey feedback survey page 3

Virtual Reality Simulations for
Radiological Transport Scenarios



Statement	Rating				
	Strongly agree	Agree	Neutral	Disagree	Strongly disagree
OVERALL IMPRESSIONS					
The workshop increased my knowledge and skills about VR in EPR					
I understand more about the role of VR in EPR					
The interactive VR scenarios met my expectations					
I will be able to apply what I learned in this workshop					
I would recommend this workshop to colleagues					
OPEN DISCUSSION					
<p>Please provide any other comments you have about this workshop:</p>					

Figure 146 - VR transport survey feedback survey page 4