Improving X-Ray Operator Performance Using Virtual Environments

by

Mark Breitinger

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

Master of Applied Science in Nuclear Engineering

Faculty of Energy Systems and Nuclear Science

University of Ontario Institute of Technology (Ontario Tech University)

Oshawa, Ontario, Canada

August 2021

THESIS EXAMINATION INFORMATION

Submitted by: **Mark Breitinger**

Master of Applied Science in **Nuclear Engineering**

Thesis title: Improving X-Ray Operator Performance Using Virtual Environments

An oral defense of this thesis took place on August 11, 2021 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. Glenn Harvel
Thesis Examiner	Dr. Sanjoy Mukhopadhyay, Remote Sensing Laboratory, United States of America

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

There are significant limitations in the training for x-ray operators under current regulatory and operational constraints. Advances in technology now allow for the creation of realistic virtual reality environments, which, when designed and implemented based on scientific principles and instructional system design concepts, can allow operators to test the full range of their tactics, techniques, and procedures.

This research thesis presents novel synthetic virtual reality training environments for explosive ordnance disposal and security screening. This includes the development of physics simulation package for generating and imaging x-rays which can be applied in interactive, realistic and variable training environments for operators.

Keywords: x-ray; virtual reality; nuclear security

AUTHOR'S DECLARATION

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.

MARK BREITINGER

STATEMENT OF CONTRIBUTIONS

The research described in Chapter 4.1, 4.3, and Appendix A was developed based on the Unity Point Kernel (UPK) model for radionuclide transport developed in cooperation with UOIT Ph.D. candidate Joseph Chaput. The UPK model concept was adapted and modified by the author into the UPK X-ray (UPK-X) model described in this publication.

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

ACKNOWLEDGEMENTS

The author would like to acknowledge the support of Joseph Chaput, a Ph.D. candidate in the Faculty of Energy Systems and Nuclear Science, whose support and guidance in virtual reality technologies and the Unity game engine were critical to the germination and initial success of the virtual reality x-ray concept, most notably in quickly and easily manipulating 3D objects in the Unity Inspector.

The author further acknowledges the support of Dr. Edward Waller, the research supervisor, whose guidance on the importance of applying established training and education principles to new technology guided the development of the two demonstrated virtual reality scenarios.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS AND SYMBOLS

CNSC	Canadian Nuclear Safety Commission
CSV	Comma Separated Values
DOTS	Data-Oriented Technology Stack (Unity)
EOD	Explosive Ordnance Disposal
GPU	Graphics Processing Unit
MCNP	Monte Carlo N-Particle
PNG	Portable Network Graphic
SAT	Systematic Approach to Training
UOIT	University of Ontario Institute of Technology
UPK-X	Unity Point Kernel – X-Ray
VR	Virtual Reality
XTK	X-Ray Toolkit

Chapter 1. Introduction

1.1 X-ray applications

X-rays have been increasingly used since their discovery in 1895 by Wilhelm Conrad Röentgen (Glasser, 1993) and the development of the ability to collimate their beams for targeted use (Michette, 1996).

X-rays are used in everyday life for a number of beneficial applications related to imaging and diagnostics. For regulatory purposes and licensing, these can be broadly categorized into medical and non-medical uses. Some of the most common medical uses include dental inspection, fracture detection, detecting infections such as pneumonia or bronchitis, detecting tumors, and other imaging applications such as arthritis, calcifications, and bone loss (Brown, 2013). Most of these applications involve fixed x-ray devices in a licensed facility.

Commercial and industrial uses include fixed facilities as well as many applications of mobile x-ray devices at work sites outside a fixed facility, which are also well regulated. These range from non-destructive testing and analysis of objects such as pipelines, welds, and other items to security applications such as airport screening and explosive ordnance disposal (EOD) bomb squad diagnostics (Brown, 2013). However, these are only some common applications of x-rays. They are also used in more niche industries including detecting counterfeit materials and art (Rizzutto, Kajiya, H.O.V. de Campos, & Almeida, 2013).

1.2 X-Ray Radiation

X-ray radiation is the term usually used to classify ionizing radiation whose origin is from the energetic excitation and rearrangement of shell electrons of a target atom via bombardment of electrons which have been accelerated through an electric field as well as the associated electromagnetic radiation emissions as the accelerated electron reacts to coulomb forces (Russo, 2017). This results in two effects where ionizing radiation is released:

1. The loss of the energy due to coulomb forces as a negatively charged and accelerated electron interacts with the nucleus of an atom can alter the direction of the accelerated electron resulting in a loss of energy, a change in the direction of travel and the emission of electromagnetic radiation (photon). This is bremsstrahlung (breaking) radiation. This radiation is emitted as a continuous spectrum of potential energies based on the angle of interaction, the target atom nucleus and the accelerated energy of the electron (Russo, 2017). This is illustrated in Figure 1 and Figure 2.
2. The loss of energy due to direct interaction of an accelerated electron with the electron shells (orbital electrons) of an atom can result in the impartment of energy and subsequent excitation of a shell electron into a higher level. For such an interaction to take place the accelerated electron must impart an amount of energy which exceeds the binding energy of the electron in its shell. The accelerated electron is subsequently scattered and the new vacant electron shell space is filled with an electron from a higher shell (or orbital). The difference in binding energy between the higher and lower electron shells

is emitted as a photon of a fixed energy (Russo, 2017). This is illustrated in Figure 3.

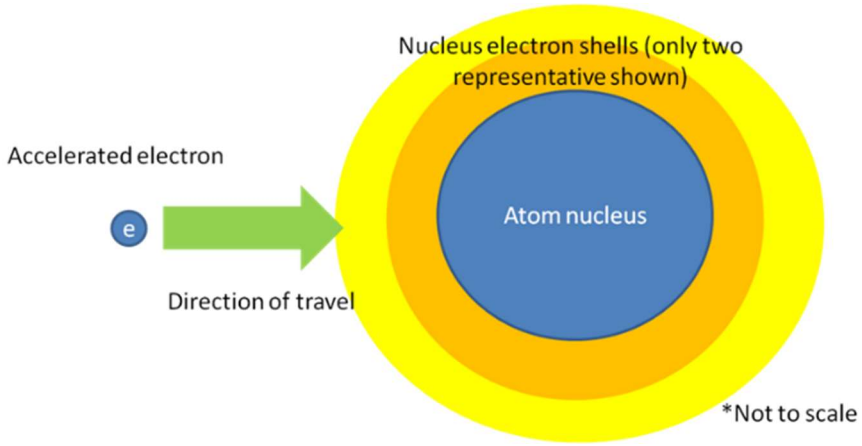

Figure 1. Accelerated electron approaching target atom

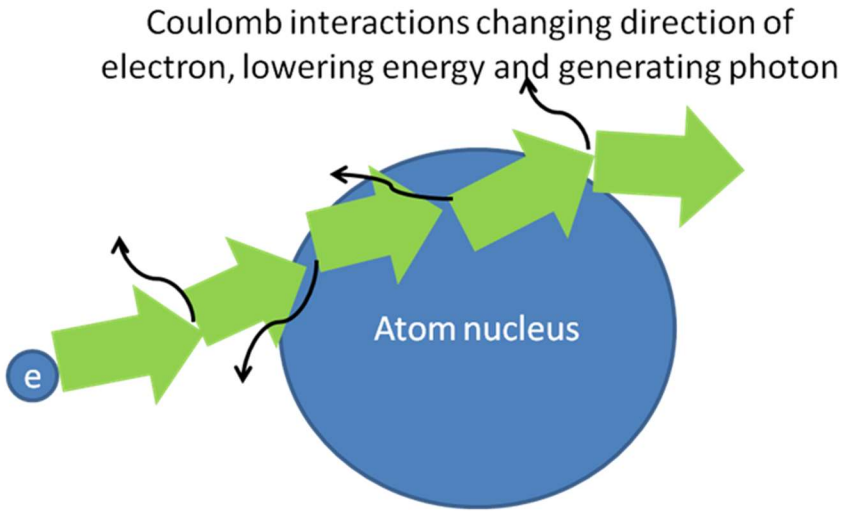

Figure 2. Coulomb force interaction between electron and atom resulting in bremsstrahlung radiation

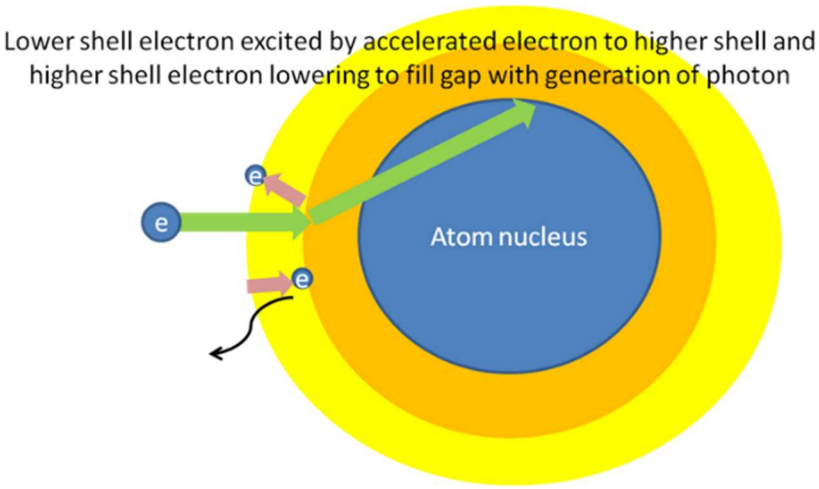

Figure 3. Lower electron shell excitation by accelerated electron causing lower shell orbital electron to excite into a higher orbit and a higher shell orbit electron to lower and fill newly opened gap

The resultant spectra of x-ray energies that are produced would look approximately as shown in Figure 4, adapted from (de Beer, 2018).

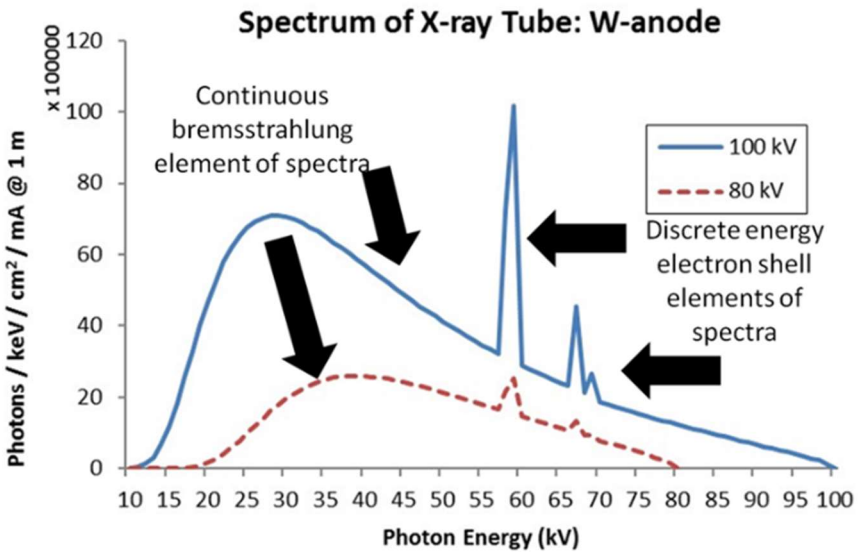

Figure 4. Annotated x-ray spectra

For x-rays to be generated a source of accelerated electrons is needed. For portable x-ray generators, such as the one that will be discussed later in this research, this usually requires an x-ray tube. An x-ray tube is comprised of a vacuum assembly, heated cathode, an anode and a voltage supply as shown in Figure 6 (International Atomic Energy Agency, 2014).

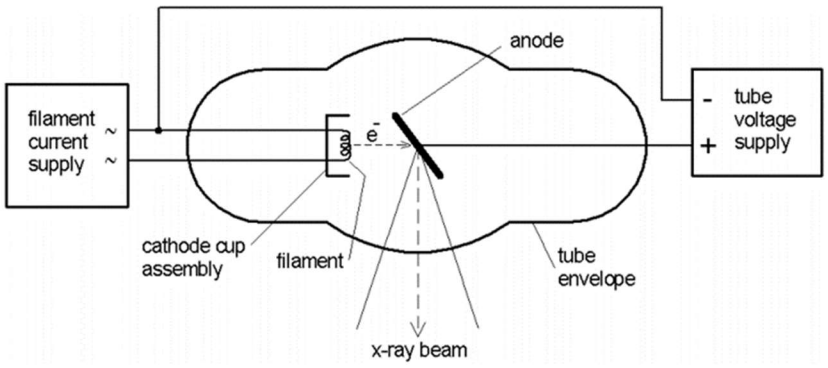

Figure 5. Principal components of an X-ray tube

The cathode filament is heated via the application of a current in a vacuum environment which in turn results in the thermionic emission of electrons. These electrons are released and accelerated towards the anode where they impact and produce x-rays. The voltage supplied between the anode and cathode can control the yield and energy of the x-rays that are generated, with a higher potential difference between the anode and cathode resulting in more energetic electron acceleration and therefore more x-ray generation as illustrated in Figure 7 (International Atomic Energy Agency, 2014). A thin opening in the vacuum environment is used to create a directional x-ray beam that is used for investigative purposes.

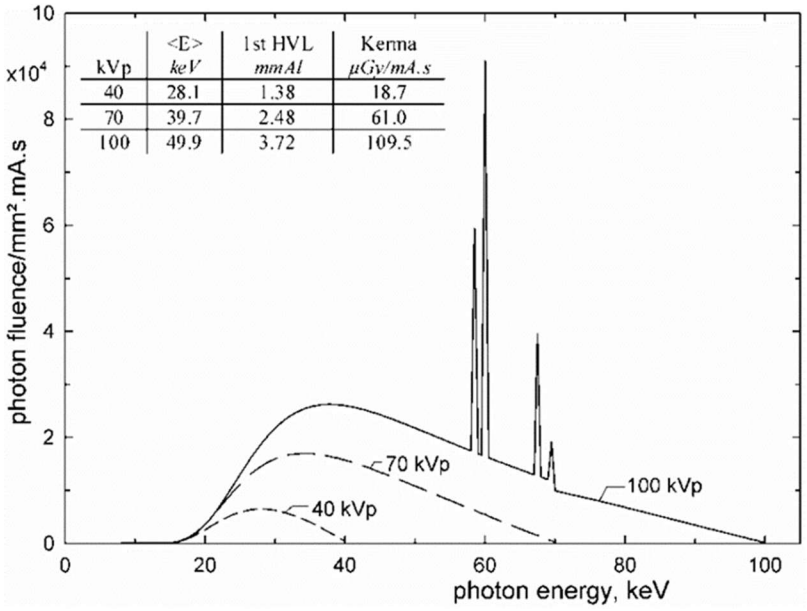

Figure 6. Impact of tube voltage on resultant x-ray spectra

Finally, to produce a visualization an image receptor is used. At the image receptor the resultant x-rays are measured and when the intensity at each point is compared, the medium which the x-rays were passing through can be investigated. Higher density material will scatter away more x-rays than lower density materials and therefore areas of low x-ray intensity can be inferred to have resulted due to the blockage by the high-density material. Prior to the creation of digital imaging capabilities, the image receptors were usually using methods with special type of film materials which can be combined with a phosphor to be made sensitive to x-rays. Currently electronic imagining techniques are used where large grids of mm sized electronics which are either directly sensitive or indirectly sensitive to x-rays, are used to measure the resultant intensity. An example of this can be seen in Figure 8 (International Atomic Energy Agency, 2014).

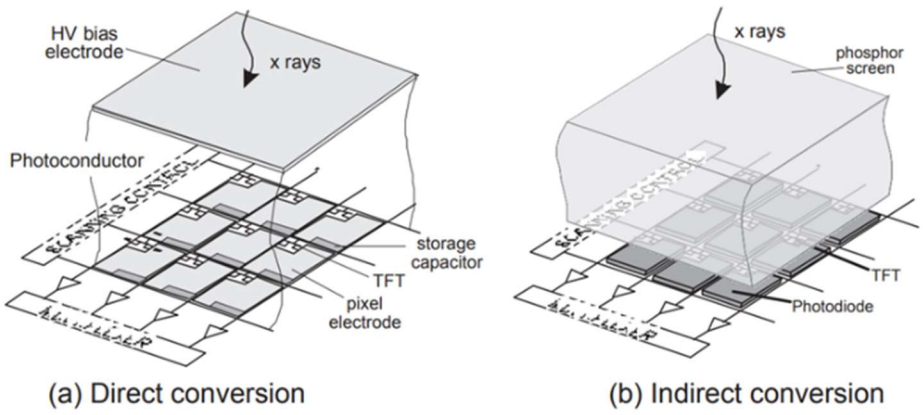

Figure 7. Example of electronic grids for measuring x-ray intensity through direct or indirect methods.

This research focuses on the novel application of training for the field deployment and teaching of x-ray equipment at the operational level. In depth literature discusses the more detailed aspects of the physics of detection, the considerations of optical effects on the imager device and the resultant image as well as computational techniques to improve imaging for cases such as tumor detection in medical fields (International Atomic Energy Agency, 2014).

1.3 X-Ray Terminology

To understand the operation of x-ray devices, in particular portable x-ray devices, requires an understanding of some of the terminology and operational language associated with their operation. Although there are variations in the definitions of some of the terms below, the definitions were selected based on their applicability to both fixed and portable x-ray devices. All definitions are from the U.S. Department of Justice, Office of Justice Programs and were created specifically for explosive ordnance disposal (EOD) applications (U.S. Department of Justics, 2007).

- Anode: The positive electrode/terminal of an electrical device. In x-ray tubes, the anode is typically made of tungsten and produces x-rays when bombarded by high energy electrons.
- Aperture: The surface of the x-ray generator module from which the useful x-ray beam exits and which is in a plane normal to beam direction.
- Control and Display Unit: The component of the portable x-ray system that enables the operator to activate and control x-ray generation and display the images obtained with the image capture unit.

- Exposure: For radiation protection purposes, exposure to ionizing radiation is measured as dose equivalent.
- Field of View: The horizontal and vertical extent of the target plane that is to be imaged.
- Image Acquisition Period: The interval required to capture (or expose) the image and then to display (or process) the image.
- Image Capture Unit: The component of the portable x-ray system that is used to acquire the x-ray image.
- Target: An object that is imaged by the portable x-ray system.
- X-ray output power: The power output of the x-ray generator modules, expressed as either the number of pulses for a pulsed x-ray generator or the combined values of kV and mA for a continuous-wave x-ray generator.

1.4 Governmental, Legal, and Regulatory Framework

Based on their widespread use and inherent hazards, the necessity to establish a governmental, legal, and regulatory framework (International Atomic Energy Agency, 2016) that ensures radiation safety (International Atomic Energy Agency, 2020) and security (International Atomic Energy Agency, 2019) of x-ray devices has been applied in most countries, including the necessity for making emergency arrangements (International Atomic Energy Agency, 2015). In Canada, X-ray

devices are regulated by Federal and provincial laws and associated regulatory guidance, which can vary by province[1].

At the Federal level, Health Canada and the Canadian Nuclear Safety Commission (CNSC) have responsibilities for regulating x-ray devices and uses based on authorities granted in the Radiation Emitting Devices Act (Government of Canada, 2016) and the Radiation Emitting Devices Regulations (Government of Canada, 2021). This Act and Regulations give Health Canada and the CNSC the ability to authorize, inspect, administer, and enforce rules regarding the use of devices, including x-ray devices.

Health Canada has developed a series of Safety Codes which provide further guidance and recommendations for the use of such devices. Safety Codes 29-35 are all applicable to x-ray devices but for the purposes of this research, Safety Code 29: Requirements for the Safe Use of Baggage X-Ray Inspection Systems (Health Canada, 1993) and Safety Code 34: Radiation Protection and Safety for Industrial X-Ray Equipment (Health Canada, 2003) are considered primary sources as they address specific applications and use cases discussed in this research and provide requirements for operator training.

Additional regulations, typically under the relevant occupational health and safety acts, exist at the provincial level. In Ontario, this is contained in the Occupational

[1] A list of Federal and provincial regulations is maintained by the Radiation Safety Institute of Canada on their website: https://radiationsafety.ca/resources/regulatory-documents/

Health and Safety Act, R.R.O. 1990, REGULATION 861: X-Ray Safety. It specifies roles and responsibilities for various people involved in x-ray operations and sets requirements for training and operation (Province of Ontario, 2021).

1.5 Operator training

Although the details vary by location and intended use, in all cases, operators of x-ray devices have to be trained and certified according to set standards. These include a combination of classroom/theoretical knowledge and hands-on training using the equipment under normal conditions and postulated unexpected conditions (International Atomic Energy Agency, 1998) such as device malfunction.

For medical applications, individuals are certified as x-ray technicians or medical radiation technologist. The requirements for this certification are comparatively limited due to the low energy involved, limited variation of usage and therefore the overall potential hazard to the safety of the operator and the patient. In other applications, the requirements are more stringent.

Natural Resources Canada, in cooperation with Health Canada, maintains the X-Ray Fluorescence Analyzer Operator Certification that qualifies individuals to perform non-destructive testing using radiographic testing and imaging (Natural Resources Canada, 2021). The certification is available in two levels, both of which require hands-on training and a written examination. The training requirements are established in hours, not based on critical tasks, making a direct comparison to the VR approach described in this thesis difficult. The requirements for Level 1 certification include 5 hours on the demonstration and practice in using the portable analyzer to make accurate and safe measurements as well as the demonstration and

practice in the safe set-up, handling, operation, general maintenance and storage of the analyzer (Natural Resources Canada, 2019).

Health Canada, through Safety Code 34, requires the operators of all industrial x-ray devices to be qualified as a Certified Industrial Radiographer (Health Canada, 2003).

For law enforcement and other non-baggage screening security applications, there is no set qualification or certification specifically for the x-ray operation. Instead, the use of x-ray devices is built-in to the overall training of diagnostics and disablement of explosive and dispersal devices, which is currently a 24-day course by the Canadian Police College (Canadian Police College, 2021).

The Canadian Institute for Non-Destructive Testing offers Certified Industrial Radiographer Training for combined Level 1 and 2 certifications, which include 48 lab hours of practical instruction (Canadian Institute for Non-destructive Evaluation, 2021).

Because the training requirements, established specifically for x-rays, are set in hours instead of task-based objectives, a direct time comparison of efficiency of training is not possible. However, the inherent advantage of virtual training components is described in more detail later in the research, focusing on the minimal setup and take down time as well as the ability to quickly alter and iterate scenarios for the student learning objectives.

1.6 Research Overview

This thesis presents a novel virtual training system and x-ray transport code, named Unity Point Kernel – X-Ray (UPK-X) that can increase the efficiency and

effectiveness of x-ray operator training and be integrated into existing training and certification processes. The thesis demonstrates the application of UPK-X in two virtual training environments, one mobile x-ray environment for explosive ordnance disposal and one fixed x-ray environment for airport security screening. The benefits of this training approach include:

- Allowing instructors to rapidly iterate training aid configurations and hazards/threats
- Reduce operational and technical overhead limitations of in-person training
- Allow x-ray operators to quickly generate x-ray images, either in-game or exported from UPK-X, for analysis and assessment

Chapter 2. Problem analysis

There are a number of limitations to the current approach to conducting training for new x-ray operators and recurrent training for existing operators. These limitations range from regulatory frameworks to operational and practical challenges. While overall the existing training programs are largely effective, they can be improved upon to maximize the effectiveness of training and limit the negative impacts of these limitations. The three primary limitations of existing training are the overhead required by regulatory frameworks, an inability to rapidly cycle training scenarios or variables, and an inability to test tactics, techniques, or procedures which may result in unsafe conditions.

2.1 Regulatory limitations

As radiation generating devices, mobile x-ray machines are licensed by the relevant regulatory bodies and there are limitations on their use to ensure safety and compliance with regulatory radiation exposure limits. Similarly, in many countries, x-ray operators are classified as occupationally exposed workers, regulating the dose they are allowed to receive to ensure safety and requiring dosimetry and personal monitoring. Obtaining these authorizations and qualifications incurs expenses before training can begin and limits the range of scenarios which can be tested in training sessions.

2.2 Operational limitations

X-ray operators benefit from the opportunity to rapidly cycle training scenarios and variables within the scenarios, which presents limitations in field-based training. For

example, operators may be curious of the effect of changing the angle of objects, the order of shielding objects between the device and the target, or any other number of physical variables. In field-based training, the sessions must pause while the objects are moved to respond to the inquiry. Oftentimes, these inquiries are left unanswered due to short windows for training sessions, the physical challenge of moving objects, or the unavailability of the materials in question.

The authorization to operate these devices includes significant limitations to conduct training in public settings, in order to avoid doses to the public. For the public safety bomb squad community, in particular, this limits the facilities and areas in which they can train. Similarly, privately owned facilities which may present unique operational challenges, require coordination and approval of the venue owner. There are also limitations on the ability to conduct training in operating facilities, for example pipelines which are in use. In these cases, training is conducted in classroom sessions, using alternative training targets, or during scheduled maintenance / down time periods.

Some x-ray operators conduct mission critical applications, for example bomb squads, often times as one of a small number of qualified personnel using a small number of critical pieces of equipment. In these situations, there are limitations on the operators travelling for training or moving equipment out of the immediate operational area to which they may be called.

Finally, x-ray devices have limited life components that require maintenance and eventual replacement. With a unit cost of approximately USD$10,000, damage

incurred in training due to novel scenarios or testing new procedures may increase the operational cost of the equipment.

2.3 Literature Review

X-ray technology and associated training have been available for a number of years. Training is primarily conducted in classroom and in-person practical sessions. There are a number of studies that raise questions about the efficacy of existing training programs. One study in 2004 found that only 45% of radiologists could correctly identify the appropriate doses and concepts for medical radiation exposure (Jacob, Vivian, & Steel, 2004).

Computer-based and virtual training have been developed for a number of topics, including x-ray image interpretation. One study found that airport security screeners demonstrated increased effectiveness in detecting threat devices, in particular improvised explosive devices, when using properly designed computer-based training (S. Michel, 2007). There is a commercial company providing medical x-ray training in a virtual environment, however, their application uses stock x-ray images and focuses on virtual image assessment, not image generation (Virtual Medical Coaching, 2021). Neither of these applications use custom generated x-ray images inside the virtual environment.

Virtual training for other applications in nuclear safety, security, and emergency response has been piloted in other settings by international organizations (International Atomic Energy Agency, 2017). There is a limited commercial market for virtual emergency response training, however, this focuses on operational aspects

at an incident level and does not include direct operation of individual devices or pieces of equipment (XVRSim, 2021).

Whenever computer or virtual training methods are applied, it is important to align their development with established instructional systems design principles. There are many common pitfalls of poorly developed computer based training, including failing to define training objectives, designing training modules based on the technology and not learning objectives, using technology that requires extensive instruction and demonstration, assuming learners are homogenous, and failing to anticipate hardware and software environments over the training lifecycle (Akhter, Javed, Shah, & javaid, 2021) (Călin, 2018).

There is a growing field of study to apply machine learning and artificial intelligence to x-ray image interpretation, in particular for the medical field to supplement conventional radiologist interpretation and diagnosis. One recent study demonstrated greater than 70% effectiveness of a trained model in diagnosing chest x-rays, benchmarked against radiologist diagnosed data sets (Azemin, 2020). Another study showed that a model trained on a combination of real and custom generated synthesized data sets can, outperform models trained only on real data sets (H. Salehinejad, 2019).

The research discovered no training or certification programs involving virtually generated x-ray images or training programs similar to UPK-X which can alternate between virtual and conventional training programs and tools.

Chapter 3. Virtual reality approach

3.1 Advances in virtual reality technology

Virtual reality (VR) has been a developing field of computer science for decades. The technology has taken many forms, from head mounted displays to entire rooms with 360° coverage in all directions to create the illusion of being in another location (Kamińska, et al., 2019). Initial technology focused on immersive environments including room-scale projectors and cave-like environments.

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 for novice designers.

3.2 Benefits of a virtual reality approach

Training in virtual reality has some advantages over existing methods for training. VR-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 their survey), provided a realistic and robust radiation physics code has been developed to provide information to a virtual radiation detector which can monitor high activity radiation fields beyond which would be allowed in normal training situations.

3.3 Selected VR system

The Unity game engine was selected to construct an environment and scenarios. Unity was selected because it is freely available for research purposes and has a large active community with many resources and tutorials making it relatively easy to learn (Unity Technologies, 2021). Alternatives considered included the Unreal Engine, CryEngine or a custom constructed game engine. The primary limitations of the alternatives were fewer tutorials and increased time to learn or design the system.

In Unity, OpenVR (GitHub, 2021) was used for designing the virtual reality environment and the SteamVR software development kit (Valve Corporation, 2021) was used for implementing commonly available assets and object models.

Testing was done on the HTC Vive VR headset and controllers (Vive, 2021). At the time of development, the HTV Vive single player, wired VR headsets were state-of-the-art as one of the first mass consumer VR headsets to gain commercial success. Since initial development and testing, multiplayer, wireless VR headsets include those from Oculus and Vive have gained market share and would add additional capabilities for VR training. However, the HTC Vive was suitable for the research conducted.

The Unity game engine was run on standard laptop computers for testing and development during the research. For testing in VR, a specialized gaming laptop from MSI with advanced graphics capabilities was used. With future code optimization, non-PC based VR headsets or standard laptops would be suitable for running the current scenarios.

Figure 8. HTC Vive controllers and headset (transceivers not pictured)

Chapter 4. X-ray transport approaches and calculations

4.1 X-ray transport approach

To achieve the training objectives in a VR environment, it is necessary to design a realistic x-ray transport code that can simulate real-world behavior of x-ray devices. The following design requirements were considered as essential for modelling the calculation and producing the final x-ray image.

- The radiation transport calculation was required to be 'fast'. In practical terms this limitation was related to operator feedback and usability. To meet the training objectives, the time to 'produce' the image for the operator needed to be quick enough that they could rapidly review the output, alter the geometry of the scenario and produce a new image, without interruption in the experience.
- The image to be output needed to be in a lossless image format which could be analyzed with the same types of tools used by operators in the field environments. This research selected the X-Ray Toolkit (XTK) software, an industry standard explosive ordnance disposal software developed by Sandia National Laboratories and widely used in the United States and Canada.
- The radiation transport calculation needed to be appropriately realistic. Photon interaction and relative fluence reduction at the detector/film due to interaction between the x-rays generated at the source and their transit through dense material was a requirement.

- The final image needed to be exportable outside of the virtual environment in order to maximize the training value by using standard digital tools and software used in x-ray operations.

Based on these requirements, a mass attenuation coefficient transport model, was selected as the ideal combination of technical accuracy and operational efficiency. For an ionizing radiation source within an environment (without shielding) the fluence of photons incident at a distance from the source can be determined taking into account all of the different photon emissions and the associated probabilities with:

$$\phi_i = \frac{P_i E_i A}{4\pi \cdot d^2}$$

Where, P_i is the probability of emission of a photon of energy E_i

d is the distance from point source to receptor point (cm)

A is the activity of the source in decays per second (Becquerel, Bq) (applied as the intensity of the x-rays emitted in photons per second)

And ϕ_i is the particle fluence at that distance d (cm^{-2} s^{-1})

Practically, ϕ_i can be converted into a more useful detector response in terms of an operational quantity such as H*(10). H*(10) is used to approximate the x-ray panel detector and imaging. The H*(10) conversion can be replaced with specific x-ray panel electronics and imaging algorithms for specific uses.

To account for the impact of a potential shield using the mass attenuation coefficients the following formula can be used to reduce the photon fluence accordingly for the photons of each energy that are emitted by the radionuclide:

$$I_i = I_0 e^{-(\mu d_{den}\, d)}$$

Where, I_0 is the initial (unshielded) particle fluence ($cm^{-2}\ s^{-1}$) of photons of a set energy

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

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

μ is the mass attenuation coefficient ($cm^2\ g^{-3}$) for the set energy of the particle fluence

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

The final code in C#[2] is referred to as the Unity Point Kernel – X-ray (UPK-X). It contains the following constructs, outlined in more detail in Appendix B:

[2] C# is an object-oriented, general purpose programming language developed by Microsoft to allow programming in a virtual machine and automatically handle memory management (unlike C++)

1. Calculate the fluence from the x-ray generating device to each point on the x-ray panel: Although most x-ray generating device information is proprietary and not readily available, the research applied a representative spectrum that is suitable for a common handheld x-ray device. The following energy distributions and probabilities were built into UPK-X in a CSV file which can be easily modified for alternate devices should proprietary information be available to the user:

Energy (keV)	Probability
10	0.011235955
20	0.04494382
30	0.112359551
40	0.179775281
50	0.213483146
60	0.179775281
70	0.112359551
80	0.06741573
90	0.033707865
100	0.02247191
110	0.011235955
120	0.005617978
130	0.005617978

Table 1. X-ray energy and probability used in UPK-X

2. Calculate the fluence of each x-ray energy contribution for the distance from the x-ray imaging device to the current location on the detector panel: A basic inverse square law calculation from the x-ray source to the specific pixel on the x-ray detector/film.

3. Determine whether any objects are in the path between the x-ray imaging device and the detector.
4. Determine the distance penetrated through each object in the path. Use the mass attenuation coefficients from a database (Oak Ridge National Laboratory, 2014) based on material and photon energy to determine the amount of overall particle fluence for each energy level of x-ray that will be reduced.
5. Convert the final photon fluence at the x-ray panel to Grey based on the dose conversion factors (Oak Ridge National Laboratory, 2014) for the objects.
6. Assign the value to the current location on the x-ray panel and determine if it is a minimum or maximum value.
7. Normalize all values based on the minimum and maximum values and produce a gradient array which can be output as a lossless PNG image file.

4.2 Alternative approaches considered

It is possible to use alternative methods to produce the radiation transport calculation and ultimately output the x-ray. However, these options were not selected based on the design requirements above.

- A discrete mathematical solution was seen as too challenging (virtually impossible) to be produced given the desired variability of objects, detector and imaging plate positions. As each discrete solution would require a unique equation to be derived representing the complex geometry of the scenario, it would not have been desirable to limit the research work with this burden. However, as part of a Q/A check on the developed solution, a discrete mathematical solution was compared with the UPK approach.

- A Monte Carlo based simulation was considered as another possible solution. Monte Carlo simulations have the benefit of providing a convenient method to relatively accurately model the radiation transport process from the x-ray source to the imaging device. This method is in fact, frequently used in commercial and industrial applications to design and construct new devices. However, there are several limitations that were seen as undesirable for a Monte Carlo based approach:
    - The Monte Carlo approach, by the nature of how it works, requires extensive processing time, as particles transports are individually modelled and statistics on the imaging plate are developed. Coupling a Monte Carlo approach with a Virtual Reality simulation was seen as impractical due to simulation time requirements to develop the needed statistics to produce the output image for operator interpretation.
    - The virtual reality approach was to be developed in the game engine Unity. Unity is an advanced tool developed for handling and visualizing 3d geometry within interactive environments. Unity is capable of handling many different geometry format types. Monte Carlo simulations (such as MCNP, GEANT, etc.) generally are not easily compatible or able to handle such a wide variety of geometry inputs. This would have required a real time interpreter be developed to convert the real-time Unity environment – at run time – into a Monte Carlo compatible input format. Although not impossible, this was determined to be impractical and given the previously

mentioned simulation time requirements for Monte Carlo statistical acquisition, this was ultimately determined to be unneeded.

Overall, the UPK-X model was determined to represent a reasonable compromise in terms of simulating the physics of the radiation transport calculation while also maintaining speed of processing. Although the Monte Carlo option was recognized as being very common in the field of medical imaging, it was decided that the processing time would be too long for mobile x-ray operator environments and on common laptops / computers running the Unity game engine in parallel.

4.3 Applying the model in Unity

To implement UPK-X in Unity, several scripts were created which will handle different aspects of the calculation as well as storage of the relevant x-ray and material property information needed. These scripts are described in detail in Appendix B. Scripts are written in the C# programming language which can manipulate and access properties of the various game objects within Unity. The following setup was used to create the scenario:

- 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 (which are stored locally in a materials property text file) which includes designating the material type and providing its density. A default density is stored

in the library for each material which can be manually overwritten by the user.

    - When the scene is loaded the 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 automatically determine the emitted photon energies, will search the database of mass attenuation coefficients values versus photon energy and will linearly interpret an attenuation coefficient for each photon energy required. This information remains accessible for other scripts that will require it.

- Any 3d game object which represents a radiological source will have a script attached to it which performs the following:
    - During the design phase the script allows the user to designate the x-ray source from the library.
    - When the scene is 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.
- A 2d canvas game object which represents the x-ray panel is created and has a script attached to it which performs the following:
    - When the x-ray device is activated, this script searches the game environment for the x-ray source. When a reading is needed to be displayed on the detector, the script will determine the position from

the radiation detector to the x-ray source using the Unity raycast function. If any shield(s) are determined to be in between the radiation source and the radiation detector the script will calculate the penetration distance of the object as it overlaps the raycast between the source and the radiation detector. The script will then calculate for each emitted energy of photon from the radiological source its contribution to the radiation detectors measurement. This process will repeat for each pixel in the x-ray panel. All images displayed in this research were set to a standard panel resolution of 1000x1000 pixels, which took approximately 3 minutes to run each x-ray on a standard consumer laptop. The resolution can be increased or decreased to optimize resolution and time for the specific training objectives. The section addressing future research includes areas in which the code could be optimized to increase efficiency.

- The minimum and maximum values for the x-ray are identified and all intermediate values are assigned an interpolated value in a range within the byte size of a PNG image and stored in an array. The array is then exported to a folder on the user's computer with a custom file name – including the date and time – which can be easily accessed for seamless integration into non-VR training tools and software.

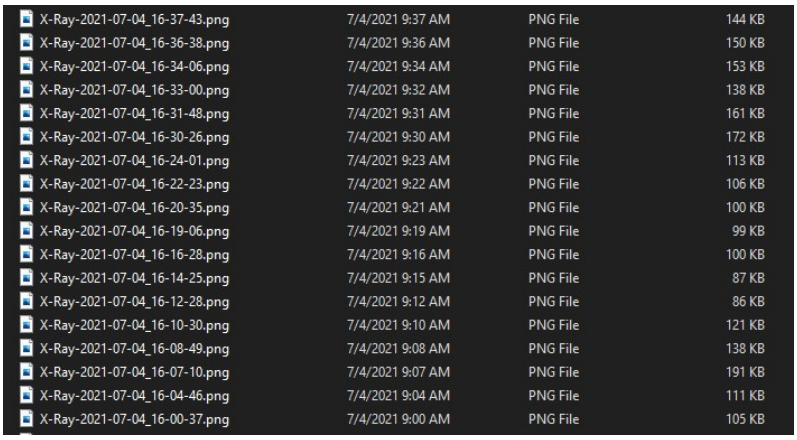

Figure 9. Windows Explorer folder of UPK-X generated images

4.4 UPK-X transport code benchmark

Although a discrete mathematical approach was not feasible for the Unity virtual reality solution for the reasons described previously, it was feasible to conduct a one-time validation of the transport model code against a mathematical solution using an identical scenario setup. This benchmark will compare the fluence at points on an x-ray panel based on photons of a single energy which pass through a shielding object to confirm that the resultant fluence calculated in Unity matches the manual calculation. Looping through all energies of the representative x-ray spectrum used later in this research is unnecessary to confirm the shielded and unshielded fluence calculation is correct.

To complete the benchmark, a methodology was developed to setup identical ‘scenes’ in Unity and Microsoft Excel to x-ray a cube with a direct orthogonal path from the x-ray source through the center of a cube and to an x-ray screen. The scene

was modeled in excel in 2d whereas in Unity it was modeled in 3d but only the results for a single 2d set of calculations were used. The scene is depicted in Figure 10.

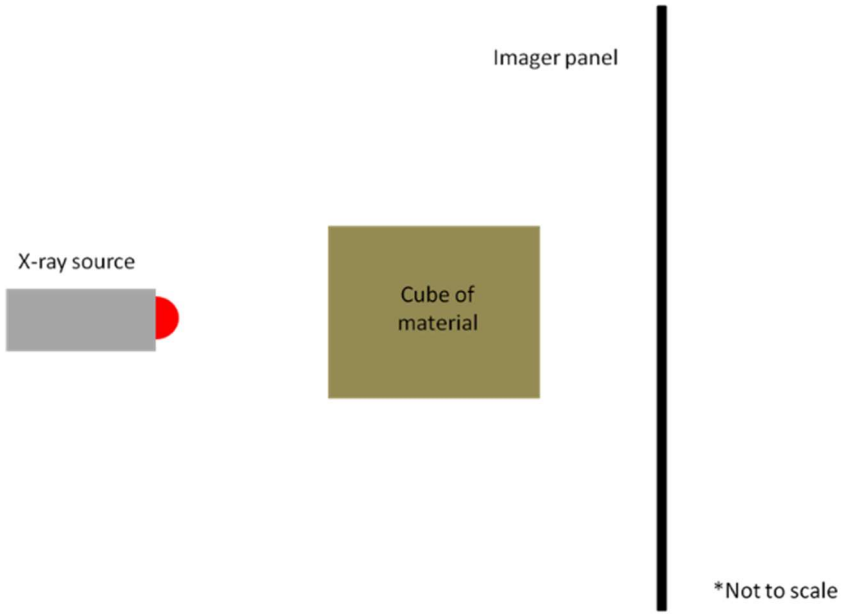

Figure 10. Scenario setup illustrated in 2d for benchmark comparison between UPK-X and calculations in Microsoft Excel

.

The cube was designed with a density of 1 g/cm^3, representing water. An intensity of 1 with a single decay energy of 100 keV was used with a 100% probability of emission - iterating through the entire x-ray spectrum was unnecessary.

In Unity, a simple script was used to determine the measured particle fluence at the screen for the centerline slice in 1 cm steps. This would then be compared with the particle fluence to be calculated from excel. Only the raw particle fluence needs to be compared as converting with dose conversion factors in both calculations would not impact any comparison of the final result. Figure 11 shows the setup in Unity

where the green cube is the shielding object, the white sphere is the source of the x-rays, and the gray canvas is the imaging plate.

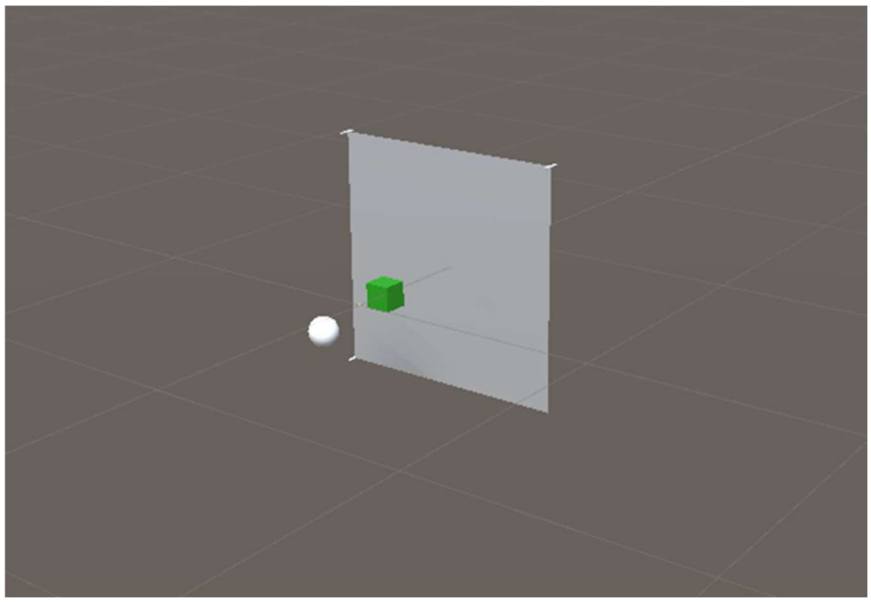

Figure 11. Unity setup for benchmarking experiment

Using a slightly modified version of the UPK-X code, the center line fluence values were returned into a computer text file (as opposed to an image). In excel three separate solutions to three potential scenarios were needed to be solved as depicted in Figure 12.

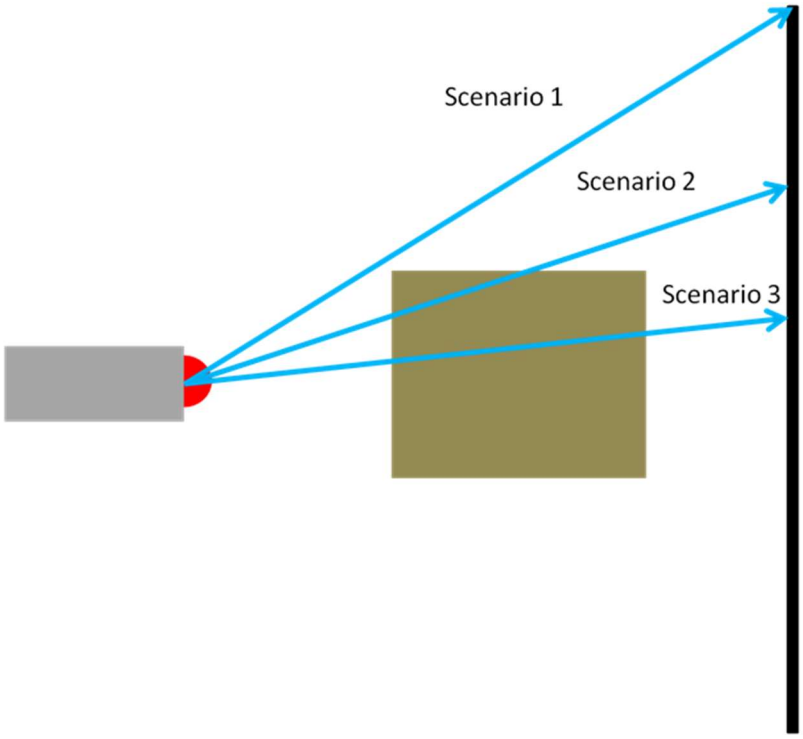

Figure 12. Three potential scenarios for x-rays to make it to the imager panel in the benchmarking experiment

Scenario 1 represents the case where the particle fluence is not impacted by any shielding provided by the cube and therefore the particle fluence is simply calculated using inverse square law. If the location of the source is at x_s and y_s and the location of the point on the screen where the fluence is to be measured is at x_{sc} and y_{sc} and P_i, E_i and A are all equal to 1 as described earlier:

$$\phi_{unshielded} = \frac{P_i E_i A}{4\pi \cdot d^2}$$

Equation 1

Where $d = ((x_s - x_{sc})^2+(y_s - y_{sc})^2)^{0.5}$ and P_i, E_i and A are all equal to 1, therefore the particle fluence on the detector screen for the unshielded scenario can be calculated as:

$$\frac{1}{4\pi \cdot (((x_s - x_{sc})^2 + (y_s - y_{sc})^2)^{0.5})^2}$$

Which simplifies to:

$$\frac{1}{4\pi \cdot ((x_s - x_{sc})^2 + (y_s - y_{sc})^2)}$$

For the case of the cube being between the x-ray source and the imager panel the impact of the shielding provided by the cube needs to be taken into account. Both scenarios are identical only requiring acknowledging if the path between the source and the panel exits the top side of the cube or the rear of the cube as shown in Figure 13 and 14.

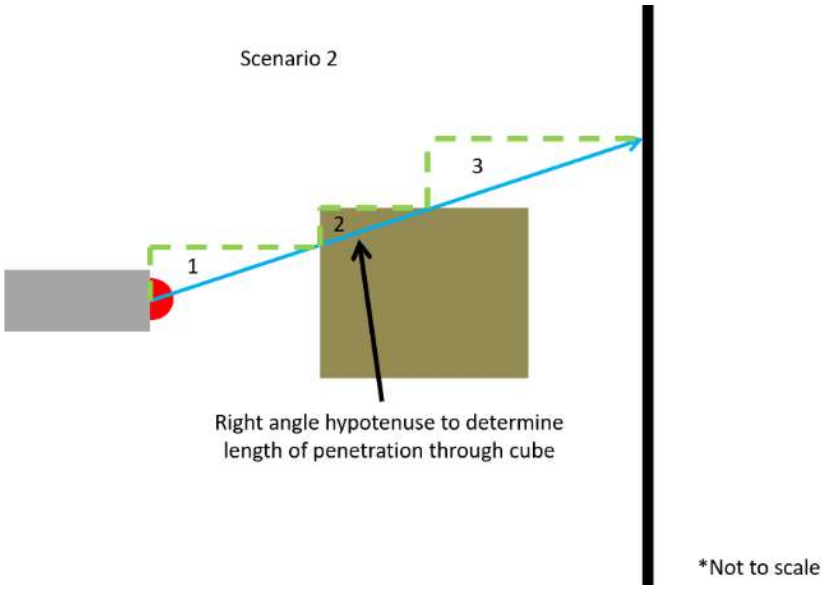

Figure 13. The x-ray exiting the top of the cube in Scenario 2

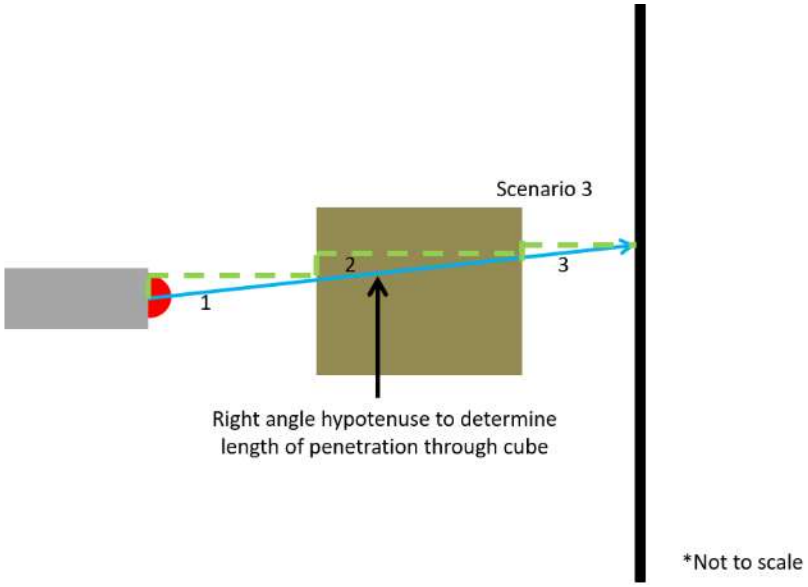

Figure 14. The x-ray travelling through the cube in Scenario 3

The Excel model was setup to conduct this calculation moving upwards at 1 cm increments until the cube was no longer intersected by 10 measurements. The Unity

model was setup with similar geometry, as previously visualized. The results of the comparison are shown in Table 3. Based on this validation, the research proceeded to the next stage, developing operational virtual reality scenarios and scenes.

Height	Excel 1D Model	Unity Code	Delta Percentage
1.5	0.088888889	0.088888889	0.00%
1.6	0.08650519	0.08650518	0.00%
1.7	0.084104289	0.08410428	0.00%
1.8	0.081699346	0.08169935	0.00%
1.9	0.079302141	0.07930214	0.00%
2	0.076923077	0.07692308	0.00%
2.1	0.074571216	0.07457122	0.00%
2.2	0.072254335	0.07225434	0.00%
2.3	0.069979006	0.069979	0.00%
2.4	0.067750678	0.06775068	0.00%
2.5	0.06557377	0.06557377	0.00%
2.6	0.063451777	0.06345178	0.00%
2.7	0.061387354	0.06138735	0.00%
2.8	0.059382423	0.05938243	0.00%
2.9	0.057438254	0.05743825	0.00%
3	0.055555556	0.05555556	0.00%
3.1	0.053734551	0.05373455	0.00%

Table 2. Benchmark comparison results for selected heights expressed in photons per cm^2 at a distance from the x-ray source

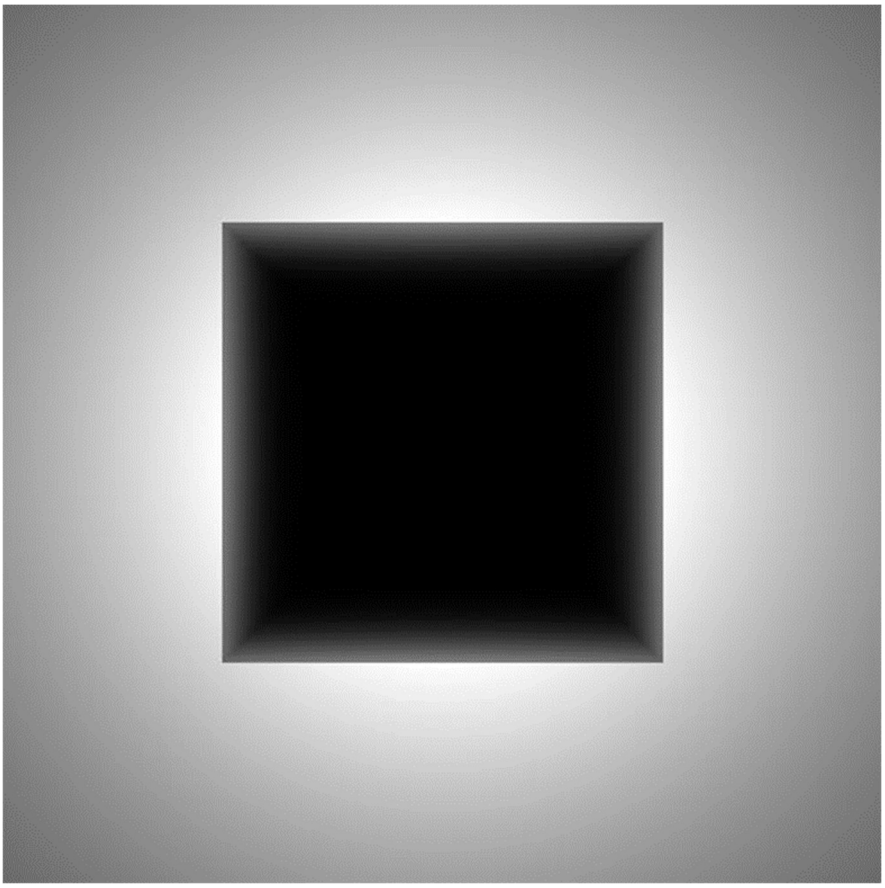

Figure 15 UPK-X image output of the cube validation scene in Unity

4.5 Limitations of the approach

The UPK-X model is optimized for the objectives described above. It does have limitations in theoretical and operational aspects, which are outweighed by the benefits. The solution is an acceptable representation of digital x-ray files but does not fully mimic the behavior of physical x-ray film, especially for extremely low fluence x-rays that would impact a physical x-ray film. Because the x-ray film is a single Unity canvas game object, it is not possible to introduce damaged pixels or components of the x-ray panel except in manual post processing with a defined rate of value modification/variance.

It is also difficult to represent wrap-around x-ray aprons or panels, such as those used in pipeline inspections. Creating a 3D x-ray panel would require redeveloping the detector component of the code which is currently designed to move pixel by pixel through a 2D canvas. Finally, because of the design of the x-ray transport code and the way Unity game objects are designed, the solution does not behave properly for objects that are intersected twice, for example a pipeline which would be entered, exited (inside the pipeline), entered again, and exited again before reaching the x-ray panel. This can be overcome with savvy 3D modelling, for example making two half pipelines and perfectly aligning them in Unity but this exceeds the scope of the research.

The methodology used to generate the x-ray image is designed such that the 'darkest' colour is chosen to be the highest measured fluence of x-rays in the detector array and the lighted colour is lowest. This produces visuals which are optimized in terms of maximizing 'contrast' within a 256-scale linear gradient from white to black, for the user to inspect the internal contents. Operation of an x-ray device in the field during a deployment not only has to content with the physical positioning of the detector and screen, but also has to optimize the rate and number of pulses produced by the generator. If too many or too few x-rays are generated when taking the image, the operator will need to adjust these settings in order to ensure the contrast is optimized and the resultant image is not over or under exposed as illustrated in Figure 16 (Golden Engineering, Inc., 2017). As the focus of the development of this training was to focus on the physical geometry and ease of rearrangement made possible by

virtual reality technologies, this specific aspect of x-ray deployment was determined to be unnecessary.

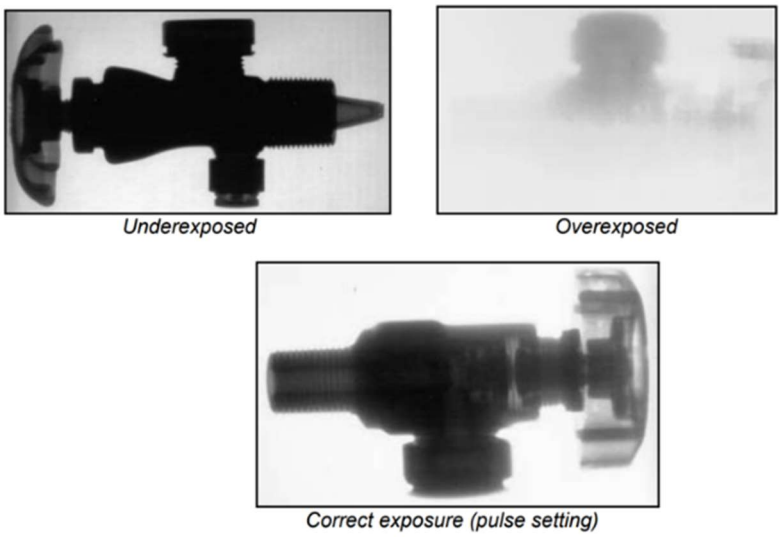

Figure 16. Examples of underexposed, overexposed, and correctly exposed x-rays

Finally, as x-ray devices are designed to have specific field of views which vary by manufacture and model, the UPK-X model was instead designed to work with an ‘infinite field of view. The field from the x-ray source in UPK-X is not given a limited directionality via the opening of the x-ray tube. This was intended to be adaptable to a specific make and model in the future as needed which could easily be done by covering the x-ray source with shielding in a geometry that mimics the make and model of the x-ray device to be simulated.

Chapter 5. Virtual reality scenarios

5.1 Testing x-rays of realistic objects

To transition the x-ray transport code into realistic Unity scenes, a step-by-step approach was developed to determine critical factors in 3d models that could be developed or acquired from the Unity Asset Store that would affect the overall success of the research. The initial approach included generating standard Unity game objects in the virtual environment, such as spheres and cylinders, to apply the UPK-X code a complex geometry of objects with various sizes, shapes, and comprised of different materials with corresponding densities and attenuation coefficients.

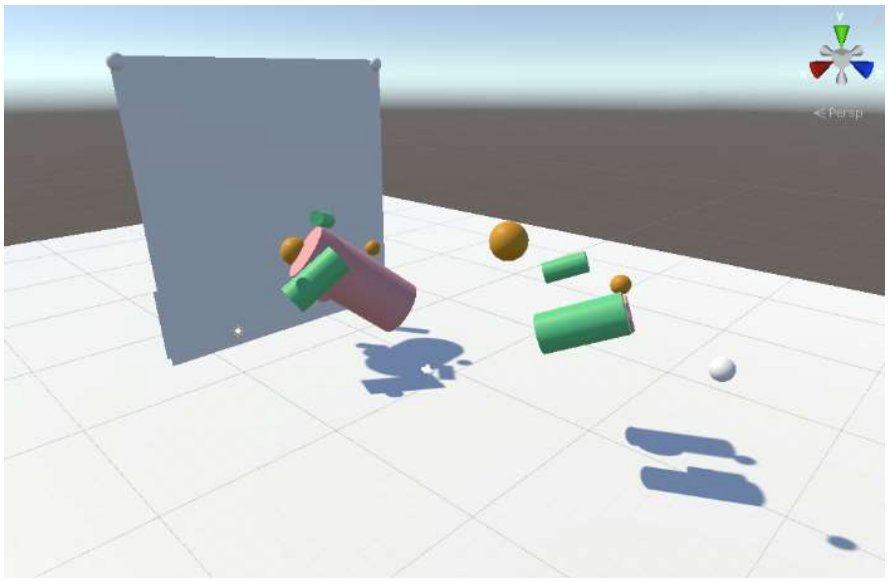

Figure 17. Complex geometry proof of concept in Unity.

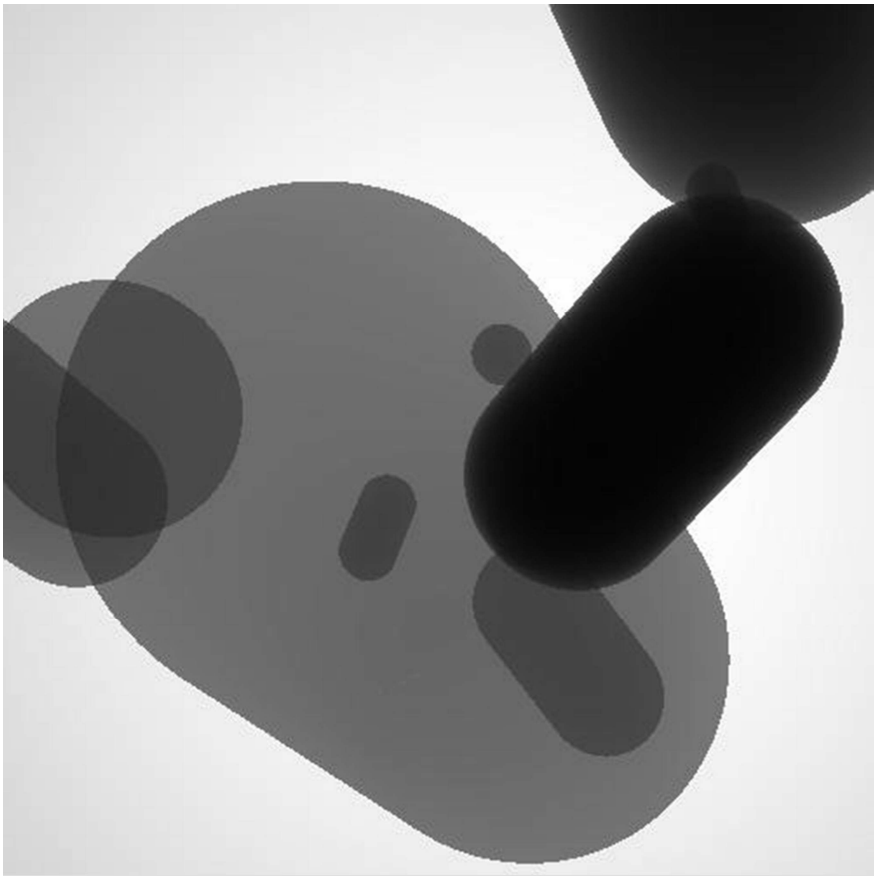

Figure 18. X-ray image of the complex geometry.

The complex geometries were applied successfully, however, the first of numerous idiosyncrasies of 3D models were noted. In this case, it is observed that the Unity game engine generates visual cylinders but actually implements invisible meshes and colliders that mimic a capsule or pill shape – which is reflected in the x-ray outputs above. This is inherent in the Unity game engine and cannot be avoided without custom 3D models of objects to be x-rayed.

Following this success and lesson learned, the research tested commonly available 3D objects available from the Unity Asset Store to determine feasibility of using

these objects for x-rays. A free model of a printed circuit board was selected and successfully inserted into the model to be x-rayed[3].

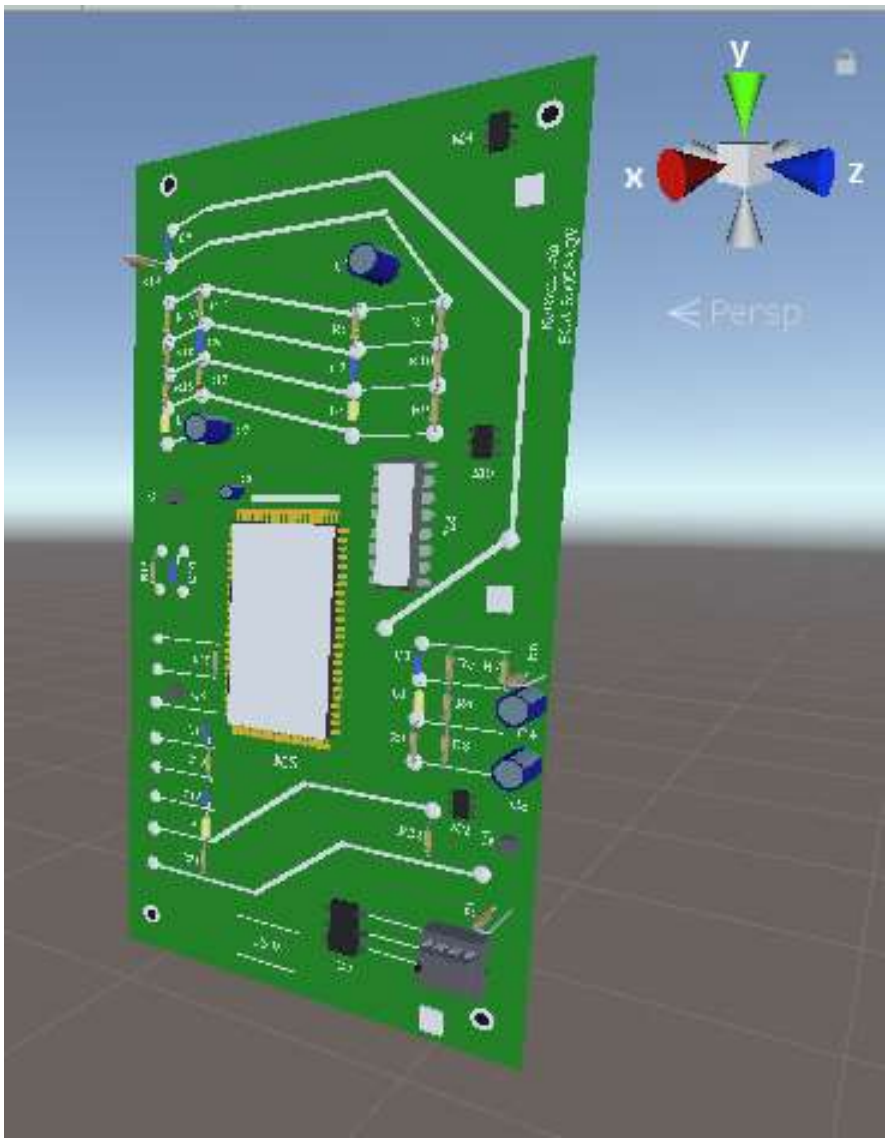

Figure 19. 3D printed circuit board model in Unity

[3] The circuit board was downloaded from the Unity Asset Store (https://assetstore.unity.com/), however, the model was no longer available at the time of publishing.

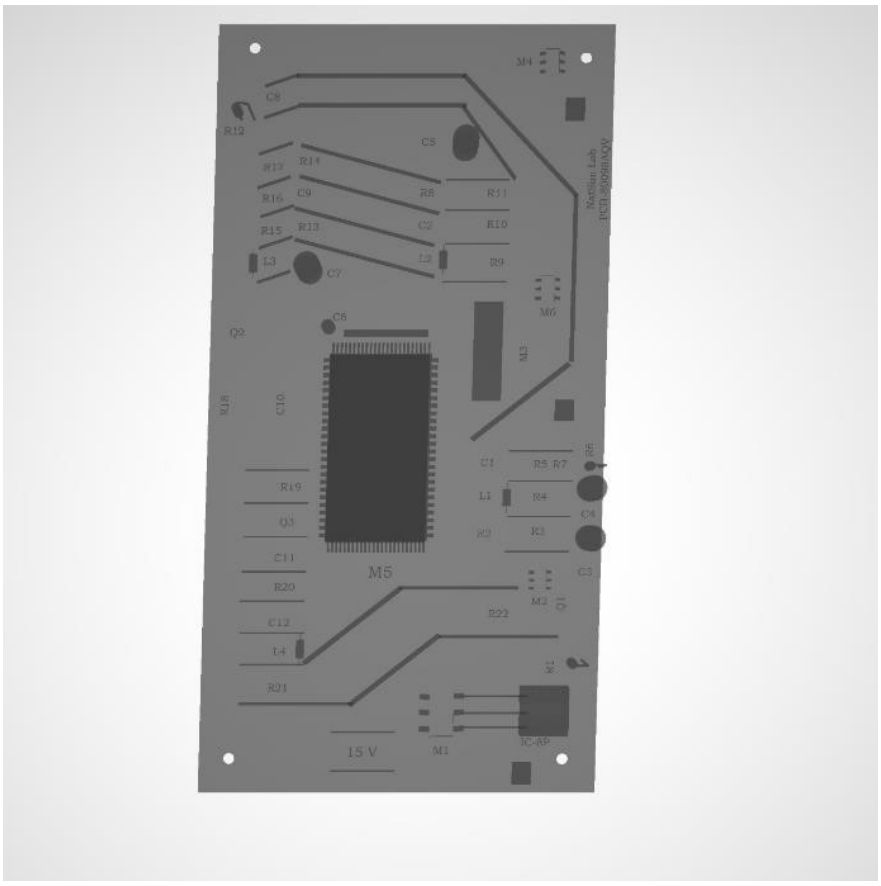

Figure 20. The x-ray image of the 3D printed circuit board

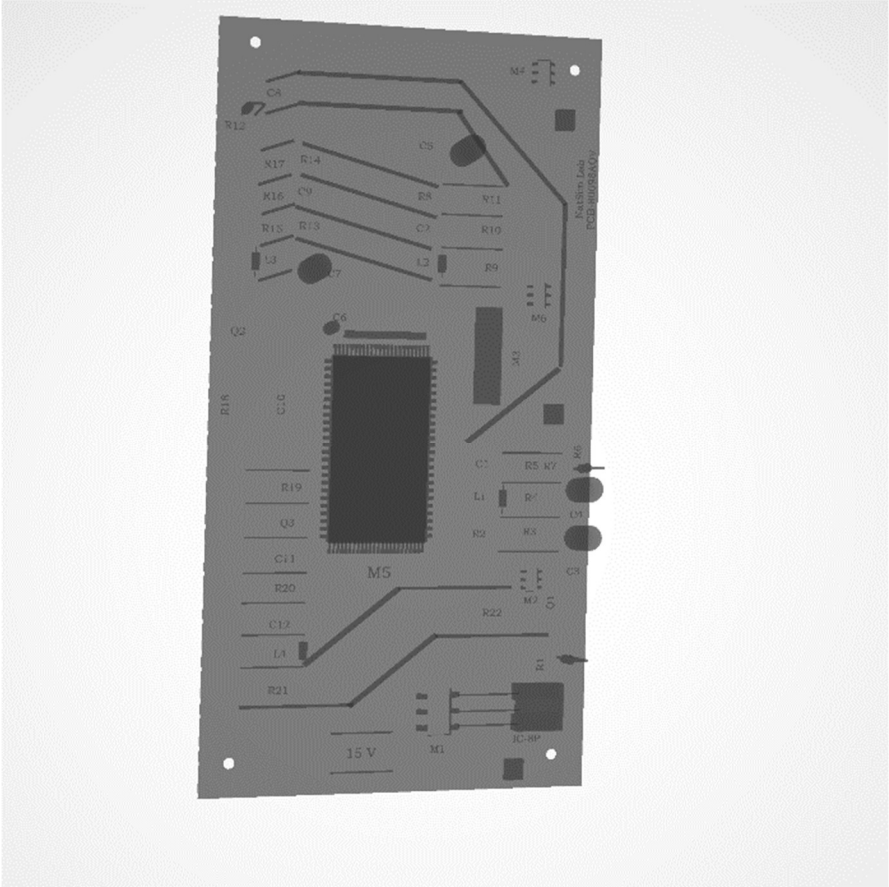

Figure 21. The same object offset 75 degrees

However, the vast majority of available 3D objects turned out to be unsuitable for x-ray scenarios for two main reasons:

- Many 3d objects for virtual reality environments are modeled only for their external surfaces. Objects are developed to be seen and interacted with only at the surface level. In this case, x-rays that penetrate the game objects return results indicative of objects that are 1 mm or 1 pixel thick.
- Many objects are modeled for visual accuracy but not realistic physical characteristics. In many cases the construction of the model uses additional

polygons or combinations of sub-objects that appear correct at the surface but have sub-surface inaccuracies. For example, in many models, surfaces actually continue beneath the next segment of the surface indefinitely whereas physical objects with different surfaces, for example two metals welded together, would only have minimal overlap.

When contrasted with a circuit board x-ray taken with the Golden XR200 as depicted in Figure 223 (Waller, 2016), the results appear very promising even with this model.

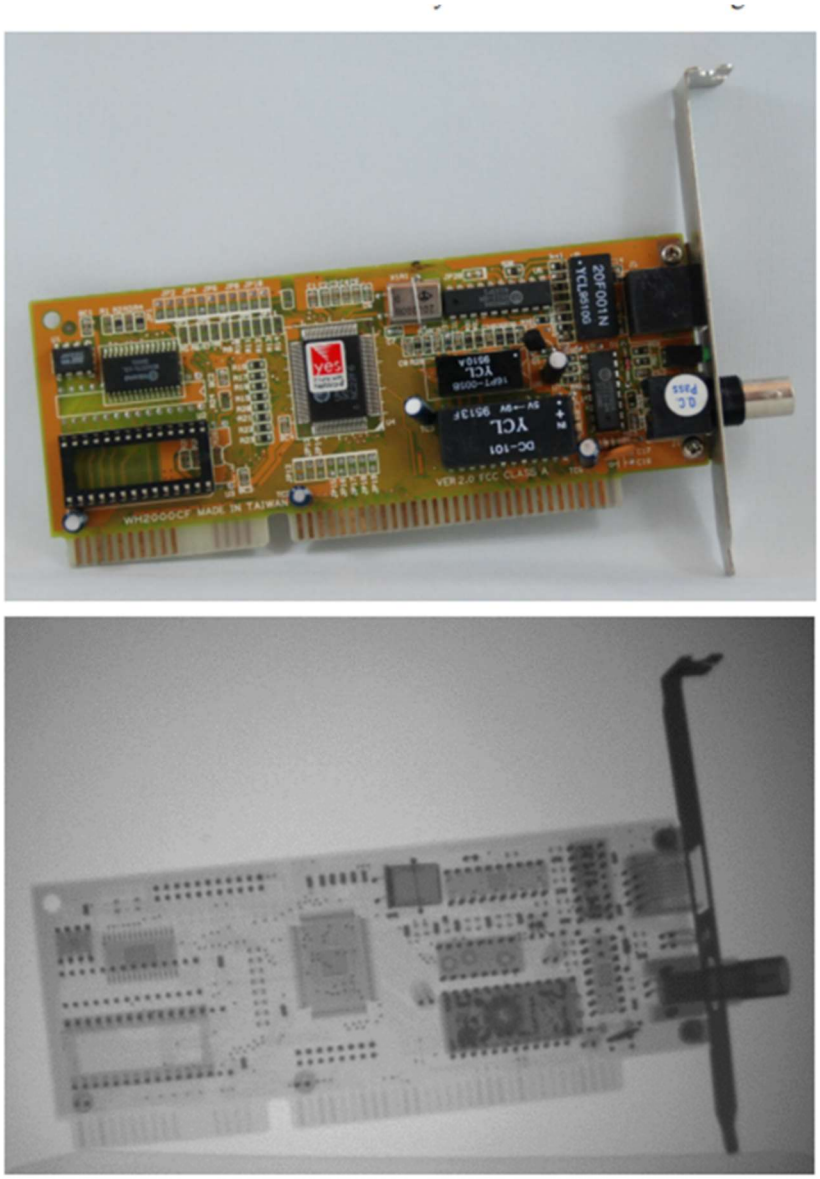

Figure 22. images of a circuit board taken with a Golden XR200

The results validated the approach but raised caution about the applicability of many commercially available object models due to the fact that many common models are designed for computational efficiency in a virtual environment and modelled only at the surface rather than as solid objects. This resulted in the need to develop custom

3D models of x-ray target objects and intermediate shielding objects to obtain accurate results.

5.2 Developing scenarios using the Systematic Approach to Training

The Systematic Approach to Training (SAT) is a methodology used to ensure that training is conducted in such a way as to effectively meet an organization's needs and that the training itself drives performance improvement. There are multiple definitions and applications of SAT but they all involve variations of the same five phases to training: 1) Analysis, 2) Design (including training objectives), 3) Development (including training sequence), 4) Implementation, and 5) Evaluation. Feedback is provided during each phase to ensure that input is captured at each phase.

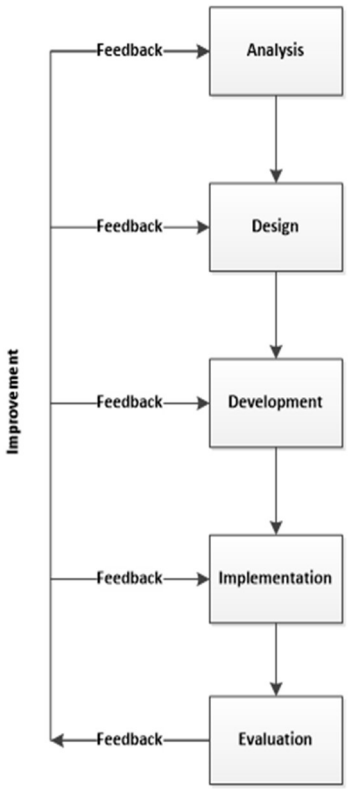

Figure 23. The basic structure of training development using the systematic approach to training

By utilizing SAT, an organization can ensure that the training is developed not based on the input of individual experts, lecturers, or instructors, but instead on the measurable performance of the persons being trained in their expected tasks, before and after training.

SAT is used widely in the nuclear industry to ensure that individuals are able to perform expected tasks to the expected level, usually in accordance with regulatory guidance. The use of virtual reality scenarios for x-ray operator training supports a gap identified in the Design phase and Development phase, as described in the Problem Analysis section of this research, which has not been previously addressed.

Two scenarios are described below, including a background and the 5 phases of SAT. Each scenario utilizes the virtual reality approach described previously to overcome a limitation of existing training for mobile x-ray device operators. To complete the SAT, each scenario would need to be integrated into an organization's overall training programme to ensure consistency across training environments. For the purposes of this research, it is assumed that each scenario would be integrated into an organization's graded approach including theoretical classroom training, on-the-job training, and certification exams described in the previous section on Operator Training.

5.3 Explosive ordnance disposal

Explosive ordnance disposal (EOD) technicians are called upon to investigate and render-safe potential threat devices ranging from pipe bombs and homemade

explosives to military ordnance. In many public safety organizations such as local or provincial police departments, serving as an EOD technician is a part-time duty conducted in addition to other duties such as patrol, investigation, etc. This can provide limits on the number of hours available for training each year. In other cases, such as military or defence settings, EOD technicians can be fully dedicated to the EOD mission, which typically involves responsibility not only for threat devices but also the explosive ordnance of the organization.

In this scenario, the participant is in the role of an EOD technician deployed alongside a special weapons and tactics (SWAT) team to investigate a suspicious package discovered during a raid on a suspected terrorist hideout in an industrial area. After the initial tactical raid, the SWAT team secures the premises and asks the EOD technician to determine if the package is a threat, and if so, render it safe.

The learning objectives of this scenario are to:

- Generate x-ray images of the threat device using standard procedures
- Identify limitations in image interpretation based on these procedures
- Identify which variables limit the quality of the image for analysis
- Iterate and alter the limiting variables (e.g., proximity to the wall, shielding objects, orientation of the threat device) and generate new x-ray images
- Discuss the best course of action to render-safe the device, if applicable

This research was designed to emulate the deployment of a handheld, portable x-ray device, similar to the Golden XR200, a 20V X-ray generator (Golden Engineering, Inc., 2021). As stated in the user manual:

"The XR200 is a small, lightweight x-ray generator that operates on its own removable battery pack. The XR200 is a pulsed X-ray device that produces X-ray pulses of very short duration (50 nanoseconds). It produces a low dose rate comparable to a 0.5 ma constant potential machine. The energy produced by the XR200 is up to 150KVP, which makes it possible to radiograph up to one (1/2) inch (1.27 cm) of steel." (Golden Engineering, Inc., 2017)

The Golden XR200, depicted in Figure 24 (Golden Engineering, Inc., 2017), is a typical unit which can be found deployed infield for industrial, police, and defence purposes in either mobile or fixed environments.

Figure 24. The Golden XR200

The University of Ontario Institute of Technology (UOIT) operates an XR200 in one of the radiation laboratories, which has been used to x-ray objects for related research purposes. Figures 25 and 26 depicts the current setup, which is limited to fixed

operations due to regulatory authorization requirements. Figures 28 and 29 depict x-rays generated from the UOIT XR200 (Waller, 2016).

Figure 25. An image of an XR2000 in a UOIT radiation laboratory

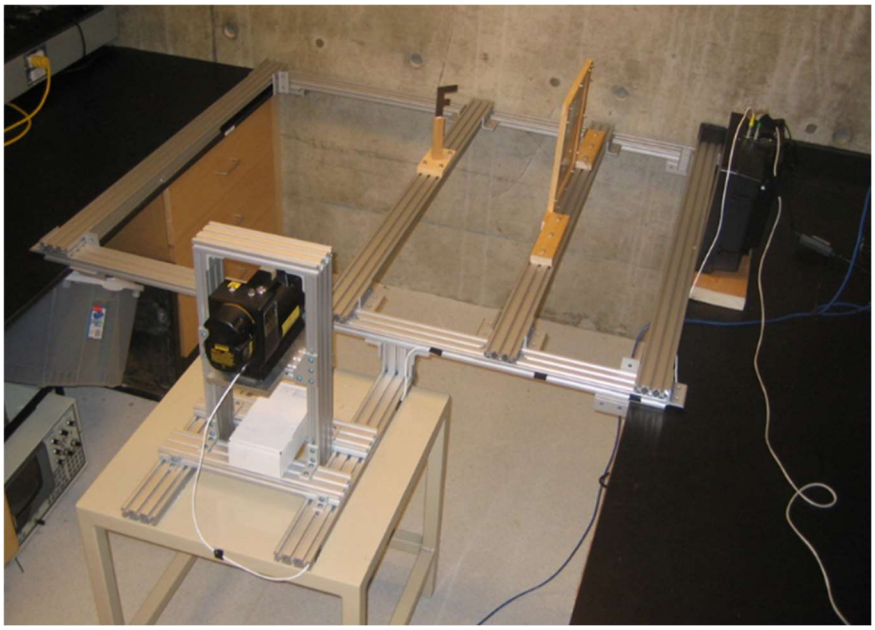

Figure 26. An XR200 in a fixed setting in a UOIT radiation laboratory

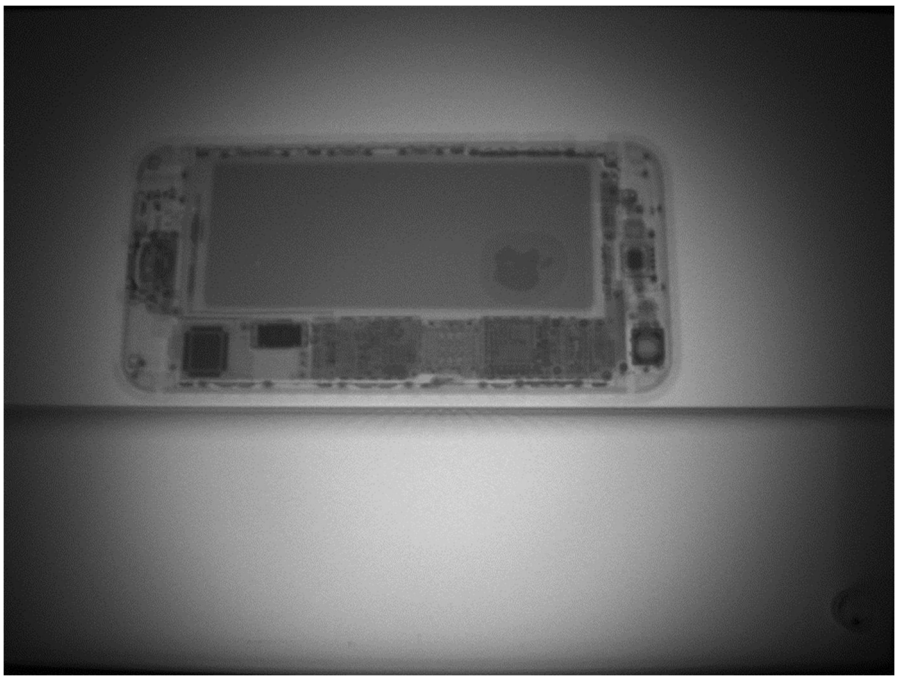

Figure 27. An x-ray of an Apple iPhone taken in the UOIT laboratory

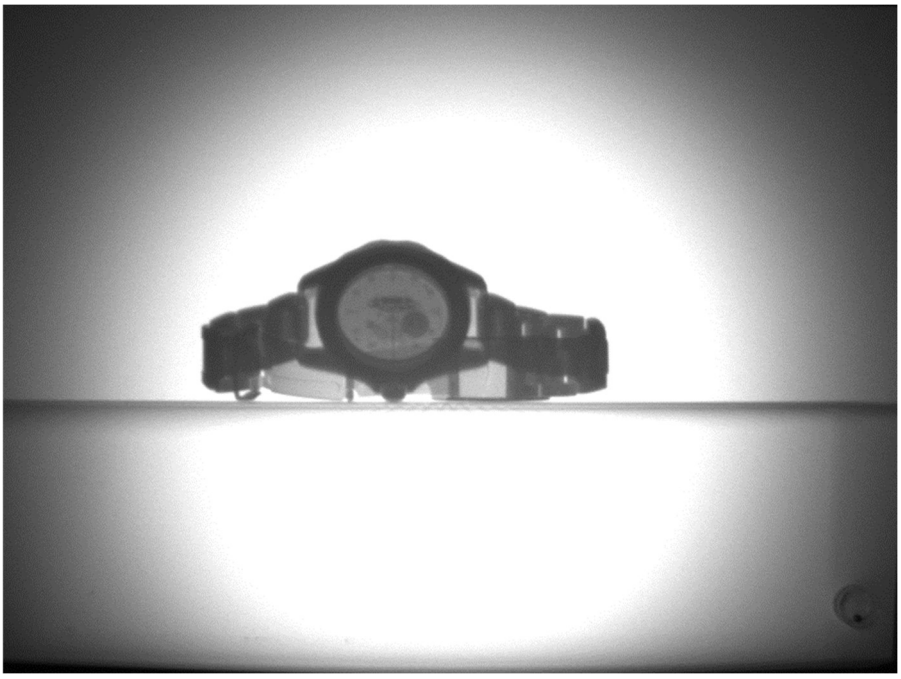

Figure 28. An x-ray of a watch using the UOIT XR200

The x-ray generator works following the principals described in Chapter 1 and the specific user manual details the settings and instructions for its usage (Golden Engineering, Inc., 2017).

The training sequence for this scenario begins with the instructor providing an overview of the VR setup to the trainee and explaining the basic technology. The instructor then has the student enter the virtual reality environment which is designed to simulate an industrial area with a suspicious item identified by first responders.

Figure 29. A suspicious package in the Unity environment

The object was custom constructed for the research and includes the following components:

Object	**Material**	**Density (g/cm^3)**
Outer container (6)	Wood	0.71
Wires (3)	Aluminum	3.95
Power supply	Aluminum	3.95
Detonator	RDX	1.806
Explosives	TNT	1.654
Circuit board	*Various*	*Various*

Table 3. Materials and Densities in the Unknown Package

The power supply was originally designated as lead, however, due to the size of the object, the density of the lead caused distortion in the rest of the image when normalized for minimum and maximum values, so aluminum was substituted for research purposes.

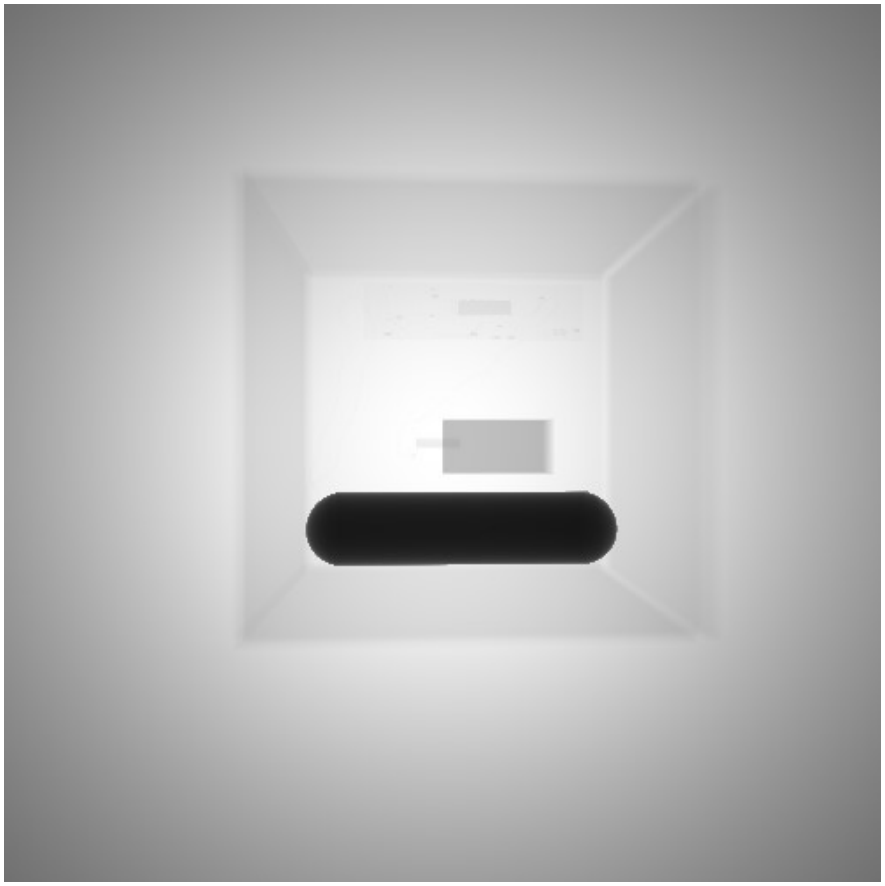

Figure 30. Distorted image with lead as the power supply

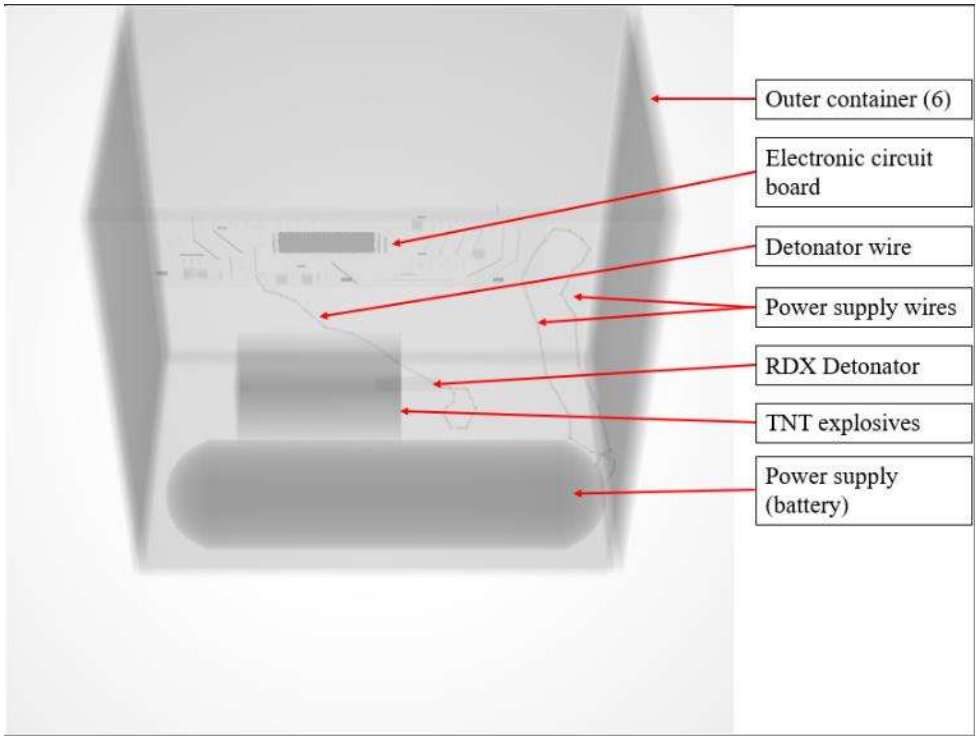

Figure 31. Labeled components of the threat device in Unity

The student can move around the virtual environment freely to determine the optimal angle to begin their diagnostics of the package. Common standard operating procedures are to take a direct orthogonal x-ray before determining follow-on interrogation angles. In this case, an x-ray would result in the following image:

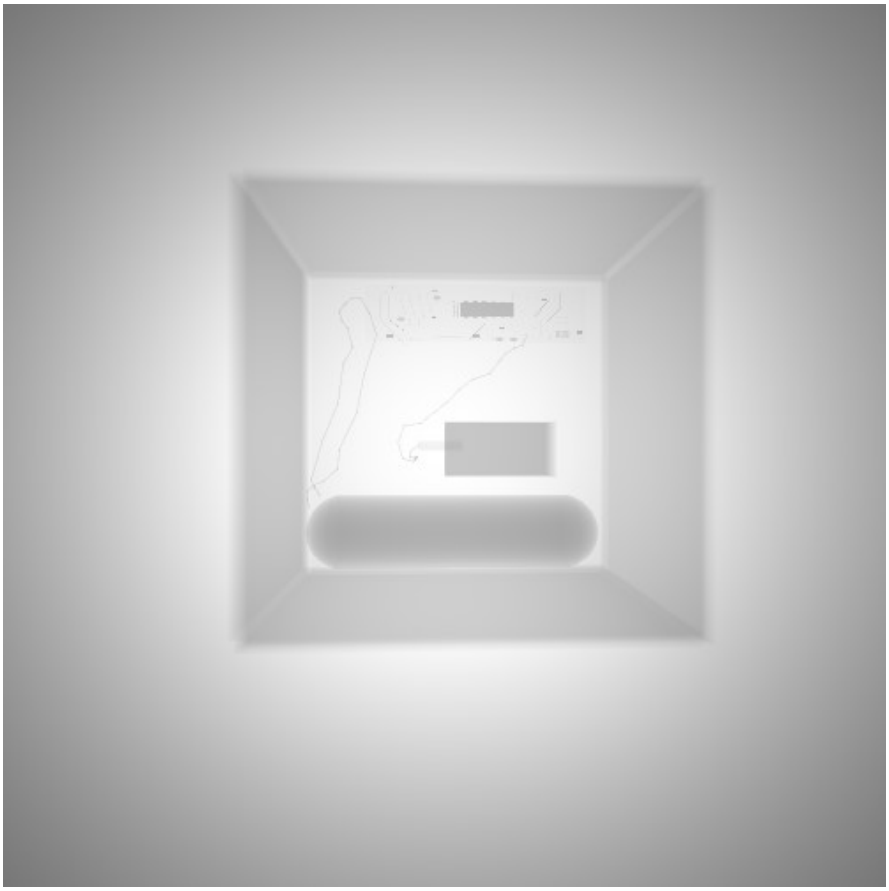

Figure 32. An orthogonal x-ray of the suspicious package using UPK-X

The initial x-ray indicates the potential presence of electronic components, wires, and forms resembling a power supply, detonator, and explosives. In this case, the wires are overlapping making wire tracing and circuit diagramming difficult. In situations like these, technicians are trained to obtain alternate angles that can provide additional details on the internal configuration of the device.

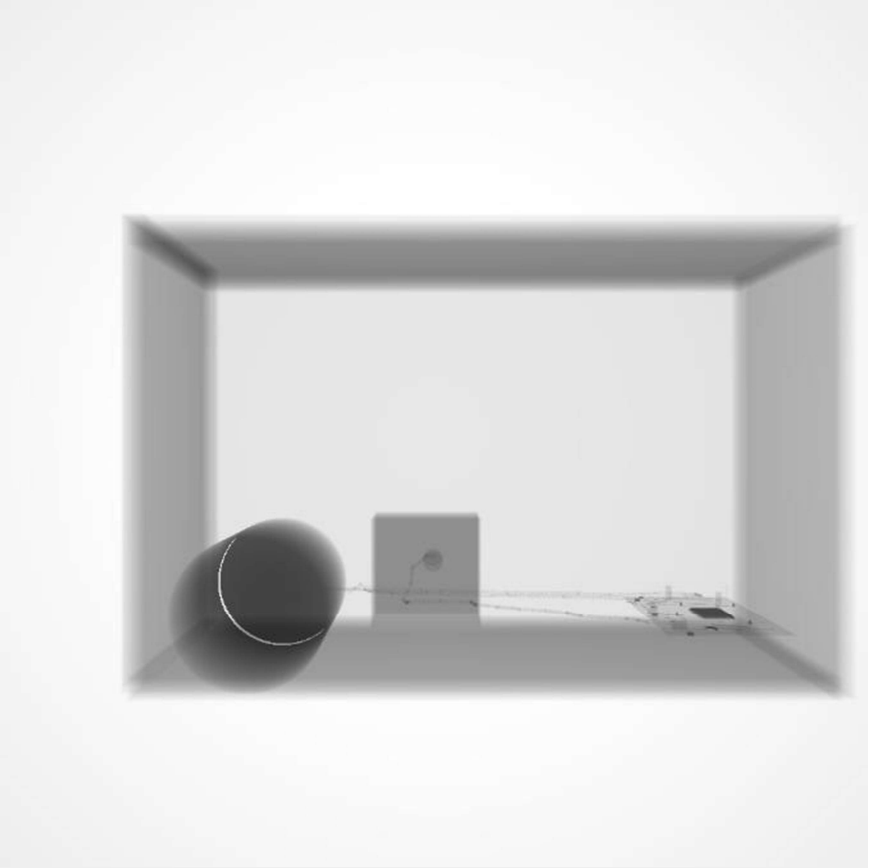

Figure 33. A rotated orthogonal x-ray of the suspicious package

In this case, the non-orthogonal x-ray provides additional clarity on the internal configuration of the device. To complete the imaging, a non-orthogonal image can be quickly discussed between the student and instructor and configured directly in VR by manipulating the x-ray device and panel positioning.

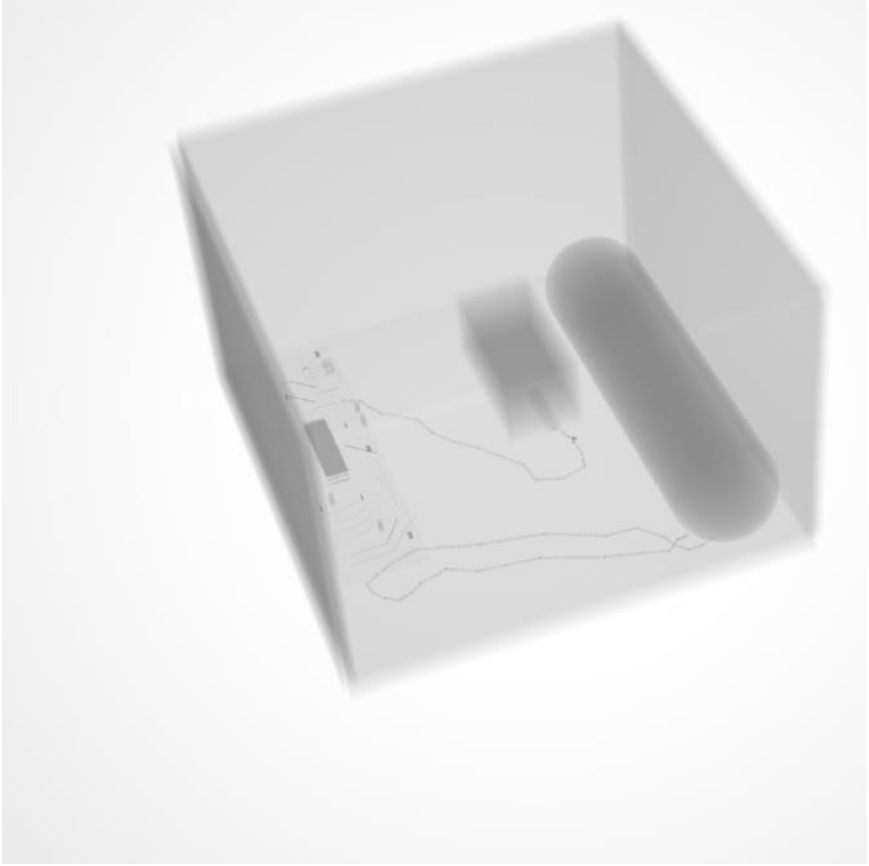

Figure 34. A secondary x-ray of the suspicious package using UPK-X

The UPK-X code automatically saves and exports lossless PNG image files to enable seamless integration into non-virtual training environments. In the case of EOD technicians, x-ray images are commonly imported into specialized software such as X-Ray Toolkit (XTK). Integrating this feature into the UPK-X code allows users to complete standard training without any artificialities associated with virtual reality environments and allows the student and instructor to complete normal training tasks as part of the certification or licensing process.

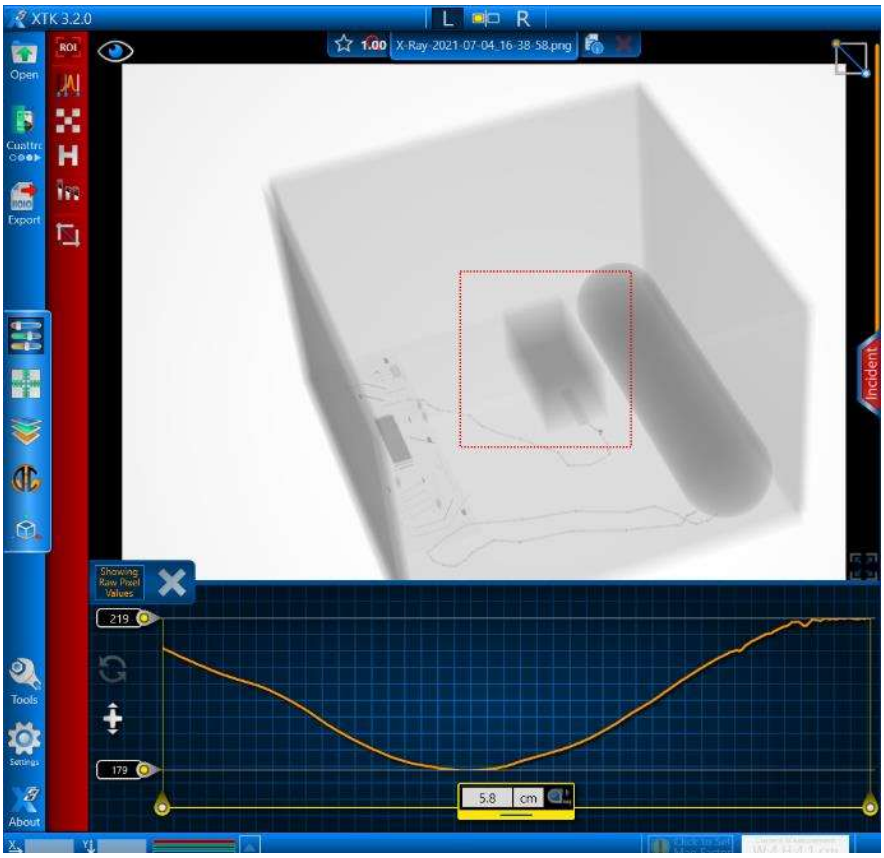

Figure 35. The UPK-X image file imported into XTK with a region of interest selected

The above image shows the sample x-ray imported into XTK with a region of interest, around the simulated explosives and detonator, highlighted as a region of interest. The image can be manipulated and analyzed in the same process as any digital x-ray file obtained through conventional training. The below image shows the same file inverted and with maximized contrast to highlight key features of the material.

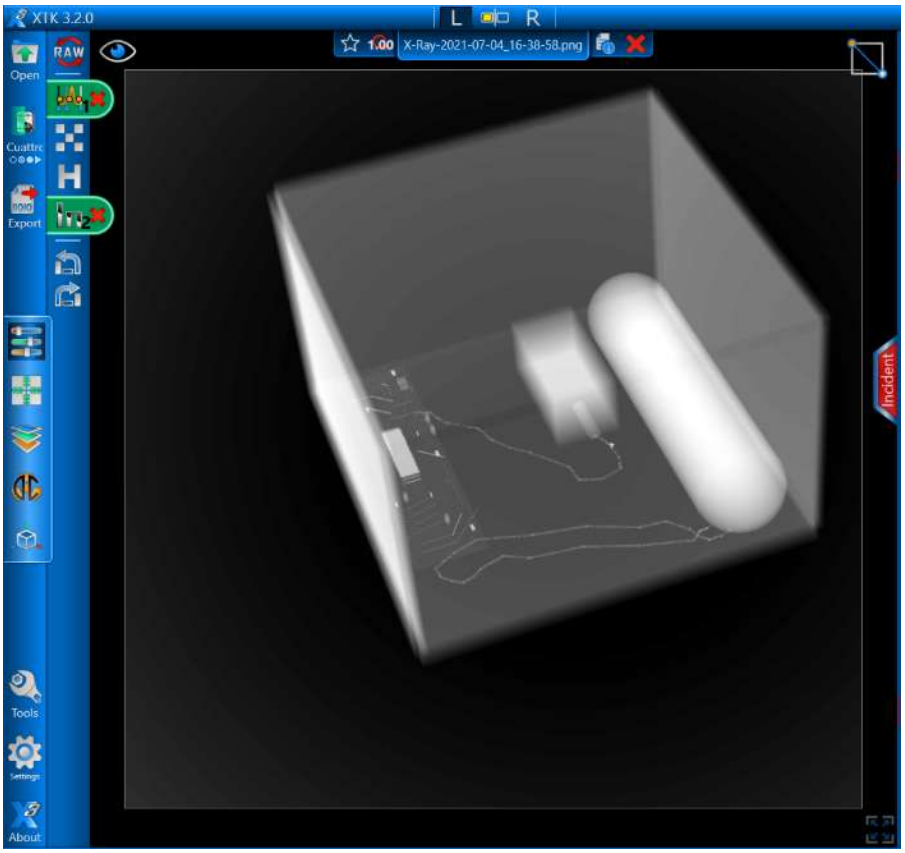

Figure 36. The UPK-X image inverted and contrast adjusted in XTK

At the conclusion of the training session, the instructor can easily remove object layers in the Unity engine with one click to reinforce training points with the student. In the image below, the instructor has removed two sides of the package to discuss the internal configuration of the device with the student following the completion of the training cycle.

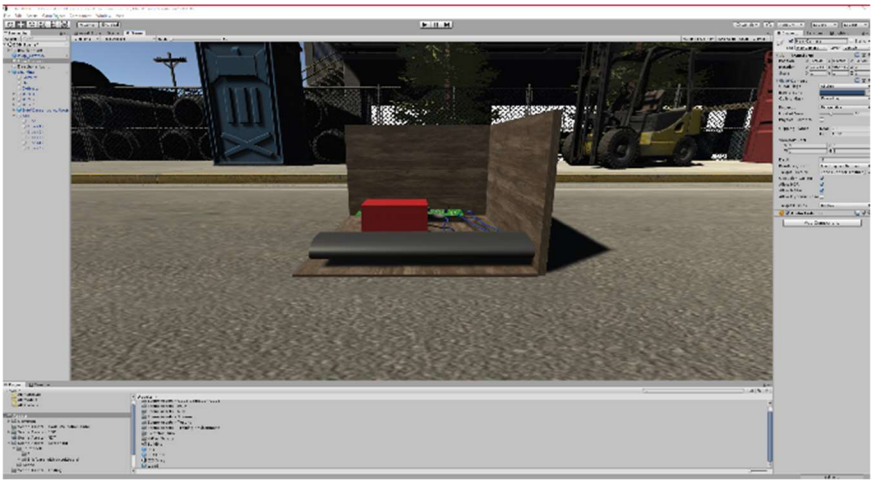

Figure 37. The Unity Game Engine Inspector with two sides of the suspicious package removed

The above scenario demonstrates that the UPK-X code can meet all learning objectives for the selected training and can increase the efficiency and effectiveness of the training by allowing rapid iteration of device configurations, easy manipulation of the training environment to reinforce specific learning objectives, and seamlessly integrate into other software and tools, allowing a blended virtual and conventional training.

5.4 Security checkpoint screening

Security screening using x-ray devices is routinely conducted at a number of sensitive locations including airports, train station, border crossings, stadiums, and other locations where members of the public bring personal belongings into a secure area. In this setting, fixed location x-ray screening machines are used to assess and clear hand-carried items such as purses, backpacks, and other bags or packages prior to entering the security perimeter. In this scenario, the student has the role of a front-line officer conducting security screening operations, with responsibility for allowing

objects into the secure environment, rejecting them, or sending them to secondary screening.

The learning objectives for this scenario are to:

- Generate and interpret x-ray images in a fixed time interval
- Identify which packages can proceed, which packages are threats, and which packages must be re-screened (noting which limitation, such as positioning, limited the interpretation of the image)

The training sequence for this scenario begins with the instructor providing an overview of the VR setup to the trainee and explaining the basic technology. The instructor then has the student enter the virtual reality environment which is designed to simulate an airport security screening.

Figure 38. The security screening scene in Unity

The student can then navigate their way to the security screen station of their choice to being security screen operations.

Figure 39. The security screen station setup in Unity

Any number of appropriately modelled objects can be inserted into the screen sequence. In this case, three briefcases have been queued for the student to screen.

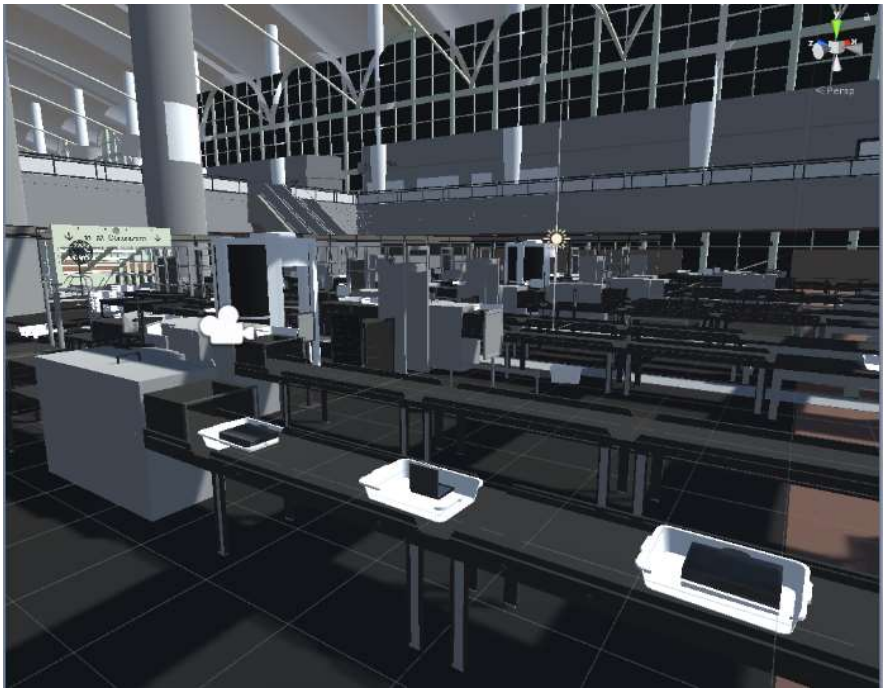

Figure 40. The objects to be screened using UPK-X

The scene is configured to allow the participant to move freely through the environment to visually screen items and to manipulate the x-ray machine with four actions: move the belt forward, initiate the x-ray, move the belt backward (for re-screening), and rotate the screened object to generate a different angle. The student can complete these actions in any sequence until all objects are passed through the security screening or rejected and removed from the scene.

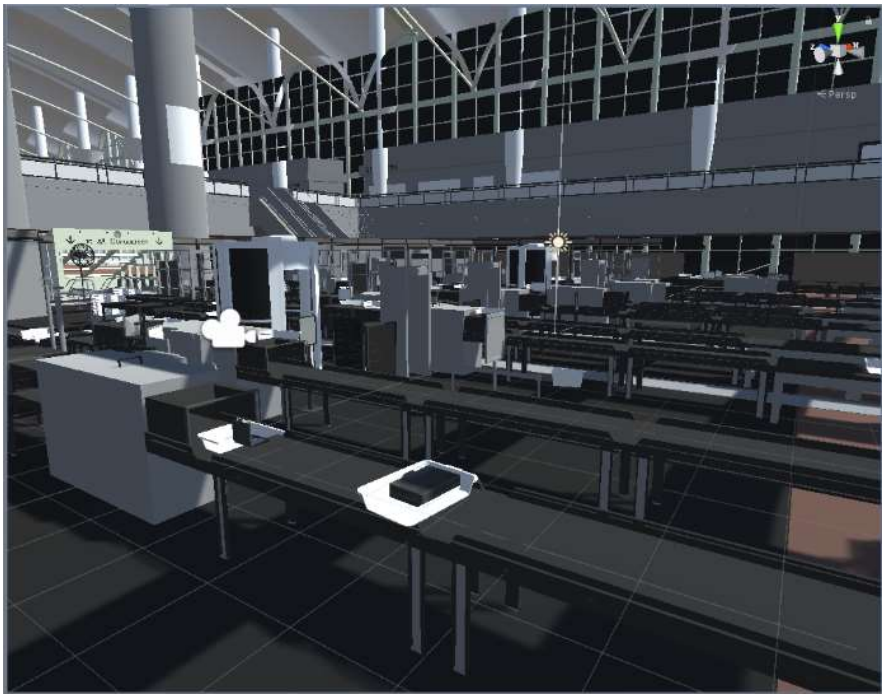

Figure 41. The screening station with 2 items screened and a third item rotated to generate a different angle

Each time the user generates an x-ray, the image is displayed in-game for realistic feedback on the simulated screens.

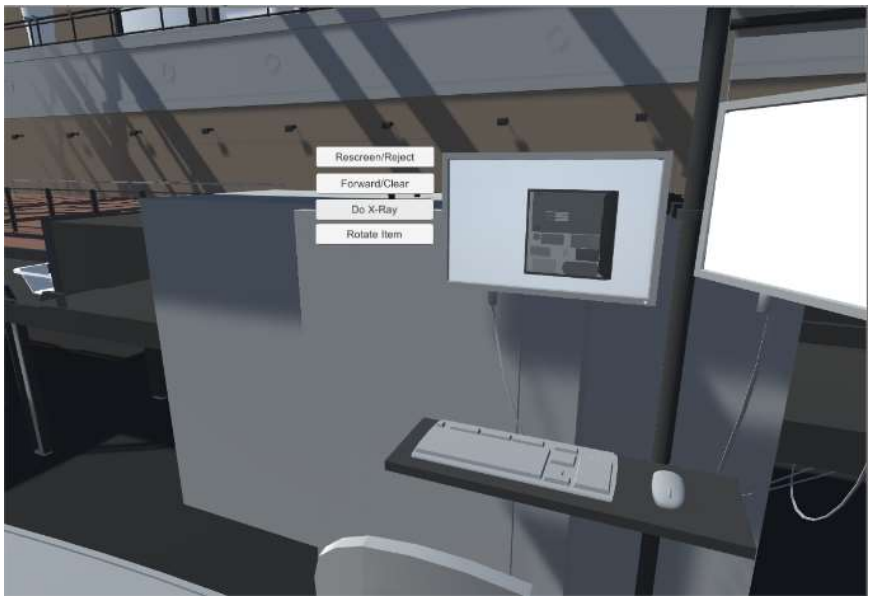

Figure 42. The UPK-X image displayed real-time in the Unity environment

When a suspicious item is noted, the participant can verbalize this to the instructor and choose their next action: reject it (for notional secondary screening or law enforcement action), rotate it, and allow it to pass.

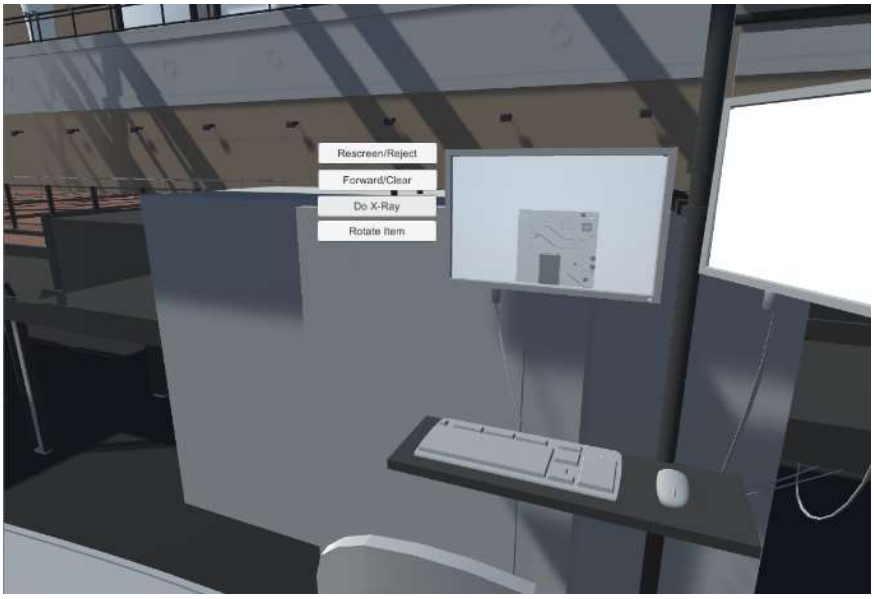

Figure 43. A suspicious device identified inside a briefcase being screened

In addition to being displayed in virtual reality on the screen, all images are exported as lossless PNG files for further analysis using industry standard software or follow-on discussions between the instructor and student.

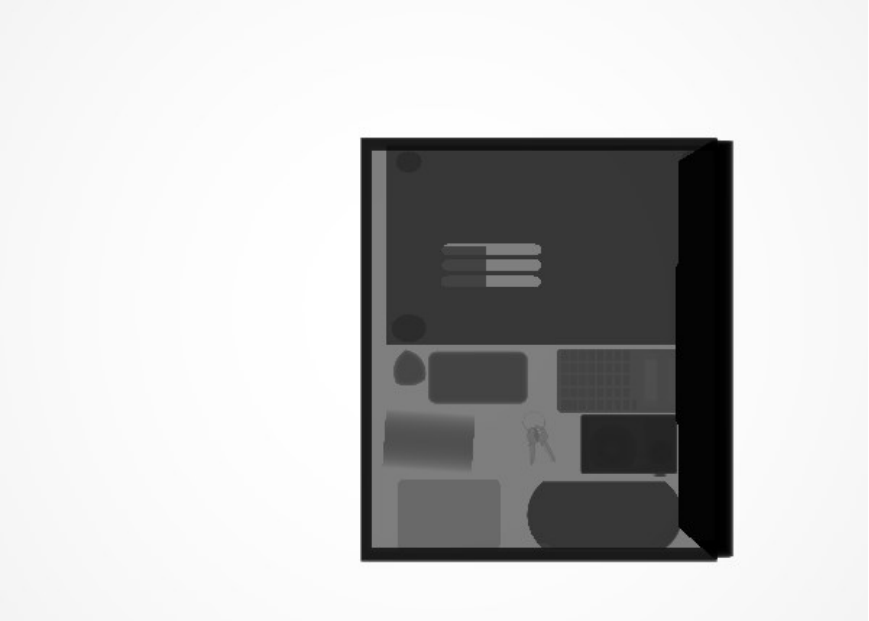

Figure 44. The UPK-X image of routine objects in the screening process

The above scenario demonstrates the ability of the UPK-X code to meet all the learning objectives and accurately simulate non-threat and threat devices in a realistic screening environment.

5.5 Alternative scenario: Nondestructive testing

A third scenario was considered for development but ultimately abandoned. A non-destructive testing scenario was designed using the systematic approach to training previously described. Industrial and commercial radiography is widely used across many industries as a form of non-destructive testing (NDT) during routine operations. This use is often cost effective and operationally efficient in identifying defects in industrial and commercial equipment. One such application is for Certified Exposure Device Operators (CEDO) to test the welding seams in pipelines. In this application, a CEDO is called in with a mobile x-ray device to conduct spot checks on welds to provide definitive analysis of their stability and effectiveness prior to resuming operations.

The learning objectives included:

- Generate x-ray images of the weld seam using standard procedures
- Interpret and evaluate results, identifying limitations (e.g., pipeline located near a wall)
- Identify multiple variable faults (e.g., weld seam error, x-ray device error) iterated in VR

The objectives for this scenario paralleled those of the explosive ordnance disposal scenario but applied in a civilian environment. Two challenges prevented the final development of this scenario. First, many NDT applications in this environment use wrap-around x-ray panels that can detect a source placed inside a pipeline. The design of the x-ray panel in Unity was based on a canvas game object which is inherently

two dimensional. Transferring the pixel movement on the detector panel through a 3D shape proved impractical and would have necessitated a new approach to the detector panel code. Second, for the devices which do use 2D x-ray panels, referred to as 'pipeline crawlers', the 3D modelling to generate a realistic training environment was prohibitive. Placing a generic object as a substitute for a realistic pipeline crawler resulted in a training environment with only limited differences from the EOD scenario.

Chapter 6. Conclusions

6.1 Conclusions

Current training for mobile x-ray device operators involves time- and labor-intensive licensing and certification processes that rely on utilizing finite equipment resources and complex training aids to increase user understanding of the placement of x-ray devices and panels, and how to interpret the end results. This training also has inherent risks when operating in a physical environment with student operators.

The UPK-X code is a mathematically validated mass attenuation coefficient x-ray transport code that allows realistic training in a virtual environment, including accurate simulation of hazardous materials such as explosives. The research demonstrated that the UPK-X code can be applied using proven instructional system design methodologies and concepts to increase the efficiency and effectiveness of the licensing and certification process.

When properly designed and conducted in line with existing systematic approach to training concepts, virtual training can decrease the time required to obtain knowledge or certification by allowing rapid iteration of training environments and configurations and reducing setup and cleanup time. It can also increase the quality of training by allowing immediate reinforcement of learning objectives through the manipulation and reconfiguration of the virtual environment. An additional benefit is the ability to train with dangerous goods or hazardous materials, or in hazardous environments, to obtain realistic results without the inherent risk or permitting process.

The UPK-X code and Unity training environments were developed to allow seamless integration with conventional tools and software that are currently used for x-ray diagnostics and work processes. This allows the benefits of virtual training to be realized while using existing tools where appropriate, reducing one of the most frequent limitations of virtual training when isolated environments do not permit external analysis or integration, which in turn increases the work necessary to create fully realistic simulated environments and tools.

The current UPK-X code demonstrates a realistic technical proof-of-concept and successful completion of learning objectives, in line with the research goals.

6.2 Future research areas

The research could be continued in two main areas: 1) optimizing the code for new technology and increased efficiency, and 2) increasing the realism of the x-ray model and imaging.

Since the initial development of UPK-X, the Unity engine has increased the utility of shaders, which are programs that can run directly on a computer's graphics processing unit (GPU). One of the three main types of shaders is specifically designed for ray tracing, which is the main distance and shielding detection component of UPK-X. This would separate the x-ray panel calculations from the overall VR environment processing and allow the imaging to run directly on the GPU and change the render state for each pixel in real time. This could perhaps be too efficient – the process would have to be slowed down artificially to not render each pixel in real time before the entire image is processed.

Another alternative to increase efficiency would be to implement the Unity Data-Oriented Technology Stack (DOTS), a new technology that was made available after the initial UPK-X development. DOTS contains a jobs system that allows multithreaded processing – for this application to be considered equivalent to parallel processing – which would allow the x-ray panel pixels to be rendered quickly based on the capacity of the computer running the code.

The x-ray code itself could be enhanced by adding functionality for dual-energy x-rays, which are commonly used nowadays in airport security screening. These x-rays can differentiate and color different materials (Dmitruk, Denkowski, Mazur, & Mikołajczak, 2016). Most commonly, this is depicted as denser inorganic materials showing as blues and less dense, organic materials as reds, and green representing a mix of potentially organic and inorganic material. This would be directly applicable to both scenarios represented in this research including the organic explosives in the EOD scene and the inorganic threat devices in the security screening scene.

References

Akhter, S., Javed, M., Shah, S., & javaid, A. (2021). Highlighting the Advantages and Disadvantages of E-Learning. *PSYCHOLOGY AND EDUCATION*.

Azemin, M. H. (2020). COVID-19 Deep Learning Prediction Model Using Publicly Available Radiologist-Adjudicated Chest X-Ray Images as Training Data: Preliminary Findings. *International Journal of Biomedical Imaging*.

Brown, J. G. (2013). *X-Rays and Their Applications.* Springer US.

Călin, R. (2018). Virtual Reality, Augmented Reality and Mixed Reality - Trends in Pedagogy. *Social Sciences and Education Research Review*, 169-179.

Canadian Institute for Non-destructive Evaluation. (2021). *Radiography Level 1 & 2*. Retrieved from https://www.cinde.ca/courses/rad_lev_1-2.php

Canadian Nuclear Safety Commission. (2020, 06 23). *Exposure Device Operator On-the-Job Training Attestation Form.* Retrieved from http://nuclearsafety.gc.ca/eng/nuclear-substances/exposure-device-operators/edo-candidates/on-the-job-training-attestation-form.cfm

Canadian Police College. (2021, 03 18). *Police Explosives Technicians Course*. Retrieved from https://www.cpc-ccp.gc.ca/programs-programmes/explosives-explosifs/petc-cpte-eng.htm

de Beer, F. (2018). *Neutron and X-Ray tomography as research tools for applied research in South Africa.*

Dmitruk, K., Denkowski, M., Mazur, M., & Mikołajczak, P. (2016). Sharpening filter for false color imaging of dual-energy X-ray. *Springer*.

GitHub. (2021). *OpenVR SDK*. Retrieved from https://github.com/ValveSoftware/openvr

Glasser, O. (1993). *Wilhelm Conrad Roentgen and the Early History of Roentgen Rays* . San Francisco: Normal Publishing.

Golden Engineering, Inc. (2017, June). *XR200 20V X-Ray Generator Operator's Maual*. Retrieved from http://www.goldenengineering.com/wp-content/uploads/2018/03/XR200-20V-Manual-layout_032318.pdf

Golden Engineering, Inc. (2021). *XR200*. Retrieved from http://www.goldenengineering.com/products/xr200/

Government of Canada. (2016, 12 12). Radiation Emitting Devices Act (R.S.C., 1985, c. R-1). Ottawa, Ontario, Canada: Government of Canada.

Government of Canada. (2021, 03 31). Radiation Emitting Devices Regulations (C.R.C., c. 1370). Ottawa, Ontario, Canada: Government of Canada.

H. Salehinejad, E. C. (2019). Synthesizing Chest X-Ray Pathology for Training Deep Convolutional Neural Networks. *IEEE Transactions on Medical Imaging, vol. 38, no. 5*, 1197-1206.

Health Canada. (1993). Requirements for the Safe Use of Baggage X-Ray Inspection Systems - Safety Code 29. Ottawa, Ontario, Canada: Health Canada.

Health Canada. (2003). Safety Code 34. Radiation Protection and Safety for Industrial X-Ray Equipment. Ottawa, Ontario: Health Canada.

International Atomic Energy Agency. (1998). *Safety Report Series No. 7: Lessons Learned from Accidents in Industrial Radiography.* Vienna: IAEA.

International Atomic Energy Agency. (2014). *Diagnostic Radiology Physics: A Handbook for Teachers and Students.* Vienna: IAEA.

International Atomic Energy Agency. (2015). *General Safety Requirements No. GSR Part 7: Preparedness and Response for a Nuclear or Radiological Emergency.* Vienna: IAEA.

International Atomic Energy Agency. (2016). *IAEA Safety Standards Series No. GSR Part 1: Governmental, Legal and Regulatory Framework for Safety (rev. 1).* Vienna: IAEA.

International Atomic Energy Agency. (2017, 10 31). *IAEA Closer to Finalising New Safety Guide for Nuclear Transport Emergencies*. Retrieved from https://www.iaea.org/newscenter/news/iaea-closer-to-finalising-new-safety-guide-for-nuclear-transport-emergencies

International Atomic Energy Agency. (2017, 12 08). *IAEA Hosts Workshop on Nuclear or Radiological Accident Assessment and Prognosis*. Retrieved from https://www.iaea.org/newscenter/news/iaea-hosts-workshop-on-nuclear-or-radiological-accident-assessment-and-prognosis

International Atomic Energy Agency. (2019). *Nuclear Security Series No. 11-G (Rev. 1): Security of Radioactive Material in Use and Storage and of Associated Facilities.* Vienna: IAEA.

International Atomic Energy Agency. (2020). *IAEA Safety Standards No. SSG-55: Radiation Safety of X Ray Generators and Other Radiation Sources Used for Inspection Purposes and for Non-medical Human Imaging.* Vienna: IAEA.

Jacob, K., Vivian, G., & Steel, J. (2004). X-ray dose training: are we exposed to enough? *Clinical Radiology*.

Kamińska, D., Sapiński, T., Wiak, S., Tikk, T., Haamer, R., Avots, E., . . . Anbarjafari, G. (2019). Virtual Reality and Its Applications in Education.

Michette, A. P. (1996). *X-rays: the first hundred years.* United Kingdom: John Wiley and Sons.

Natural Resources Canada. (2019, 05 13). *XRF Analyzer Operator Training Requirements*. Retrieved from https://www.nrcan.gc.ca/science-and-data/non-destructive-testing/about-certification-ndt/learn-about-applying-xrf-analyzer-operator-certification/xrf-analyzer-operator-training-requirements/19581

Natural Resources Canada. (2021). *X-Ray Fluorescence Analyzer Operator Certification*. Retrieved from https://www.nrcan.gc.ca/science-and-data/non-destructive-testing/x-ray-fluorescence-analyzer-operator-certification/19572

Oak Ridge National Laboratory. (2014, 05 01). Rad Toolbox v. 3.0.0. Oak Ridge, TN, USA.

Province of Ontario. (2021, 06 07). Occupational Health and Safety Act R.R.O. 1990, REGULATION 861 X-RAY SAFETY. Ottawa, Ontario, Canada: Province of Ontario.

Rizzutto, M., Kajiya, E., H.O.V. de Campos, P., & Almeida, P. (2013). RADIOGRAPHY AND X-RAY FLUORESCENCE USED TO ANALYZE CULTURAL AND ARTISTIC ARTIFACTS. *2013 International Nuclear Atlantic Conference - INAC 2013*.

Russo, P. (2017). *Handbook of X-ray Imaging Physics and Technology.* CRC Press.

S. Michel, S. M. (2007). Computer-Based Training Increases Efficiency in X-Ray Image Interpretation by Aviation Security Screeners. *41st Annual IEEE International Carnahan Conference on Security Technology*, (pp. 201-206).

U.S. Department of Justics. (2007). *Portable X-Ray Systems for Use in Bomb Identification.* Washington D.C.: U.S. Department of Justice.

Unity Technologies. (2021). *Unity Real-Time Development Platform*. Retrieved from https://unity.com/

Valve Corporation. (2021). *SteamVR*. Retrieved from https://store.steampowered.com/steamvr

Virtual Medical Coaching. (2021). *Virtual Reality*. Retrieved from https://virtualmedicalcoaching.com/virtual-reality/

Vive. (2021). *Vive United States*. Retrieved from https://www.vive.com/us/

Waller, E. (2016). *X-Ray Analysis of Common Commercial and Industrial Devices.* Oshawa: University of Ontario Institute of Technology (unpublished).

XVRSim. (2021). *XVRSim*. Retrieved from https://www.xvrsim.com/en/

Appendix A: C# Code for UPK-X integration in Unity

This appendix contains the C# code developed for this thesis for the calculation of the x-ray images in Unity. This is comprised of the following scripts as described in Section 5:

- XRaySource.cs: This script is used as a Unity attribute on the x-ray source. When a scene is loaded this script will automatically import all required energies from a library file and will normalize the energies based on the probability. Creating a separate script increases optimization by separating the x-ray calculation code from the Unity game object. Linking the energy and probability of the x-ray device permits straightforward modifications of the x-ray source for other x-ray devices with different energies.
- ShieldObject.cs: This script is used as an attribute on any object within a scene that functions as a radiation shield when directly between the x-ray device and the x-ray panel. 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 customized densities. When a scene is loaded this script will automatically import the mass attention coefficients related to that material from a library file. 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 attention 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.

- XRay.cs: This script is used as an attribute on the 2d canvas that represents the x-ray panel. It will look up the relevant characteristics from the XRaySource game object and then looks up a table of conversion factors from photon energy intensity to Gy and from Gy to Sv(H*10) operational units and it will linearly interpolate between the two closet conversion factors for each photon energy and store that value in a list. During game play, when the function CalculateXRay(), a raycast is sent from each radiation source to the current point in the x-ray panel 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). The distance to the detector is calculated and if a game object 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 pixels in the x-ray detector panel. As the measurements are taken, the script creates a list of minimum and maximum values and normalizes against the minimum and maximum to produce a PNG image file that contains the final readings in gray scale.

Table 4. XRaySource.cs

```
using System.Collections;
```

```
using System.Collections.Generic;

using UnityEngine;

using System.IO;

public class RadiationSource : MonoBehaviour {

public Radionuclides radionuclide;

public List<string> listRadionuclideEnergy = new List<string>();

public List<string> listRadionuclideProb = new List<string>();

void Start () {

using(var radionuclideenergyreader = new StreamReader(Application.dataPath +
"/../Input Data Files/XRaySpectra.csv")) {

                        while (!radionuclideenergyreader.EndOfStream) {

                                var line = radionuclideenergyreader.ReadLine();

                                var values = line.Split(',');

                                if (values[0] == radionuclide.ToString()) {

                                        listRadionuclideEnergy.Add(values[1]);

            listRadionuclideProb.Add(values[2]);

                                }

                        }

                }
```

```
        }
}
```

Table 5. ShieldObject.cs

```
using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using System.IO;

public class ShieldingObject : MonoBehaviour {

public enum Materials {

// List of materials omitted for brevity. List can include any materials as long as their density is available and added to the CSV.

}

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"));
                using(var materialsreader = new
StreamReader(Application.dataPath + "/../Input Data
Files/MaterialsAttenuationCSV.csv")) {
                        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{
                        listRadionuclides.Add(temp);
                        using(var radionuclideenergyreader = new StreamReader(Application.dataPath + "/../Input Data Files/XRaySpectra.csv")) {
                                while (!radionuclideenergyreader.EndOfStream) {
                                        var line = radionuclideenergyreader.ReadLine();
                                        var values = line.Split(',');
                                        if (values[0] == temp) {
                                listRadionuclideEnergy.Add(values[1]);
                                }
                        }
                }
```

```
            }

    }
```

Table 6. Xray.cs

```
using System;

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using System.IO;



public class XRay : MonoBehaviour {

        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 List<Vector3> MeasurementList = new List<Vector3> ();
```

```
    public int Xsize;

    public int Ysize;

    public Vector3 mPosition;

    public GameObject TopLeftPositionGO;

    public GameObject TopRightPositionGO;

    public GameObject BottomLeftPositionGO;

    public Vector3 TopLeftPosition;

    public Vector3 TopRightPosition;

    public Vector3 BottomLeftPosition;

        public List<GameObject> SourcesList = new List<GameObject>();

        public List<GameObject> ShieldList = new List<GameObject>();

        private double MinQuadRadMeasurement;

        private double MaxQuadRadMeasurement;

    public GameObject digitalxrayscreenprimary;

    public GameObject digitalxrayscreenreject;

  void Start() {

        using(var dcfreader = new StreamReader(Application.dataPath + "/../Input
Data Files/AmbientDoseEquivalentCSV.csv")) {

                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();
      if(listRadionuclides.Contains(temp4)){


    }
    else {
      listRadionuclides.Add(temp4);
      using(var radionuclideenergyreader = new
StreamReader(Application.dataPath + "/../Input Data Files/XRaySpectra.csv")) {
```

```
            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 void doxray() {
    TopLeftPosition = TopLeftPositionGO.transform.position;
    TopRightPosition = TopRightPositionGO.transform.position;
    BottomLeftPosition = BottomLeftPositionGO.transform.position;
    MinQuadRadMeasurement = 10000000f;
    MaxQuadRadMeasurement = 0f;
    MeasurementList.Clear();
    for (float x=0; x<=Xsize; x++) {
        for (float y=0; y<=Ysize; y++) {
```

```
        mPosition = (TopLeftPosition + ((TopRightPosition - TopLeftPosition)/Xsize)*x)+(((BottomLeftPosition-TopLeftPosition)/Ysize)*y);
                Reading = 0.0d;
                foreach (GameObject source in RadionuclidesList) {
            distance = (Vector3.Distance (source.transform.position, mPosition))*100;
            double unnormalizedparticlefluence = 1 / (4 * 3.14159d * (distance*distance));
                        ray = new Ray (source.transform.position, mPosition - source.transform.position);
                        raybk = new Ray (mPosition, source.transform.position - mPosition);
            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 linearattentuation = 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) {
                                    linearattentuation =
shield.GetComponent<ShieldingObject> ().listLinearAttenuationInterpolation[k];
                                }
                                k++;
                            }
```

```
                                    reductionfactor = Math.Pow (2.71828d, (-
linearattentuation*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++;

            }

                    }

        if (Reading > MaxQuadRadMeasurement) {

            MaxQuadRadMeasurement = Reading;

        }

        if (Reading < MinQuadRadMeasurement) {

            MinQuadRadMeasurement = Reading;
```

```
        }
        MeasurementList.Add(new Vector3(x, y,
System.Convert.ToSingle(Reading)));
    }
}

var texture = new Texture2D(Xsize, Ysize, TextureFormat.RGB24, false);

float RedValue;
float tempScaleStep;
int intRedvalue;
byte ByteRedValue;
tempScaleStep = 255 / System.Convert.ToSingle(MaxQuadRadMeasurement);
foreach (Vector3 measurement in MeasurementList) {
    if (float.IsNaN(measurement.z)){
        ByteRedValue = 0;
    }
    else {
        RedValue = tempScaleStep * measurement.z;
```

```
            intRedvalue = Mathf.RoundToInt(RedValue);
            ByteRedValue = System.Convert.ToByte(intRedvalue);
        }
        texture.SetPixel(Mathf.RoundToInt(-measurement.x), Mathf.RoundToInt(-measurement.y), new Color32(ByteRedValue, ByteRedValue, ByteRedValue, 1));
    }
    texture.Apply();
    GetComponent<Renderer>().material.mainTexture = texture;
    digitalxrayscreenprimary.GetComponent<Renderer>().material.mainTexture = texture;
    var pngData = texture.EncodeToPNG();
    string datetime = string.Format("-{0:yyyy-MM-dd_HH-mm-ss}", DateTime.Now);
    File.WriteAllBytes(Application.dataPath + "/../X-Ray Outputs/X-Ray" + datetime + ".png", pngData);
    }
    public void reject() {
        digitalxrayscreenreject.GetComponent<Renderer>().material.mainTexture = digitalxrayscreenprimary.GetComponent<Renderer>().material.mainTexture;
```

```
    }

}
```

- Finally, specific to the airport security screening scene, it was necessary to create a customized script to handle the Unity game objects representing the bins in order to allow the user to manipulate the bins in the screening process. The script currently allows the user to manipulate the bins in three ways:
    - Forward – out of the x-ray machine, simulating a package that has cleared screening and can be returned to the user on the other side of the x-ray machine.
    - Back – Back to the input side of the x-ray machine to simulate a rejected screening requiring additional investigation. It was not feasible to allow a split track screening when the package can be routed to a secondary screening location given the limitations of the Unity 3d model used.
    - Rotate – allowing the user to rotate each bin 90 degrees in a clockwise direction on the x-axis to get a different view for the next x-ray. The bins can be fully rotated by completing 4 rotations. This function could be replicated along the y- or z-axes to fully manipulate the contents of the bins.

Table 7. BinController.cs

```
using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class BinController : MonoBehaviour {

	public GameObject RotateBin;

	bool needtomoveforward;

	bool needtomovebackward;

	bool needtorotate;

	private float fwdmovecounter = 0;

	private float bkmovecounter = 0;

	private float rotatemovecounter = 0;

	void Start () {

	}

	void Update () {

		if (needtomoveforward == true) {

			fwdmovecounter = fwdmovecounter + Time.deltaTime;

			transform.Translate(0,0,Time.deltaTime);
```

```
            if (fwdmovecounter >= 2.04) {
                needtomoveforward = false;
                fwdmovecounter = 0;
            }
        }
        if (needtomovebackward == true) {
            bkmovecounter = bkmovecounter + Time.deltaTime;
            transform.Translate(0,0,Time.deltaTime*-1);
            if (bkmovecounter >= 2.04) {
                needtomovebackward = false;
                bkmovecounter = 0;
            }
        }
        if (needtorotate == true) {
            rotatemovecounter = rotatemovecounter + Time.deltaTime;
            RotateBin.transform.Rotate(0,Time.deltaTime*90,0);
            if (rotatemovecounter >= 1) {
                needtorotate = false;
                rotatemovecounter = 0;
```

```
                        }

                }

        }

        public void Forward () {

                needtomoveforward = true;

        }

        public void Backward () {

                needtomovebackward = true;

                //transform.Translate(0,0,-1);

        }

        public void Rotate () {

                needtorotate = true;

        }

}
```

Appendix B: UPK-X Validation Data

Height	Distance from source	Dose Rate No Cube	Cube intersect	Top or back	Vector Angle Radians	Vector Angle Degrees	Back Exit Case Hyp. of Inner Tri.	Top Exit Case Hyp. of Front Tri.	Top Exit Case Hyp. of Back Tri.	Top Exit Case Hyp. of Back Tri.	Top Exit Case Hyp. of Middle Tri.	One Column Penetration Distance Solution	Reduction Factor	Final result (no conditional formatting)	Final result (conditional formatting)
0	3	0.111111111	Yes		0										
0.1	3.001666204	0.110987791	Yes	Back	0.033320996	1.909152433	1.000555401	1.000555401	-0.4	-12.00666482	14.00777562	1.000555401	10.00555401	0.011092618	[redacted]
0.2	3.006659276	0.110619469	Yes	Back	0.066568164	3.814074834	1.002219759	1.002219759	-0.3	-4.509988914	6.514428431	1.002219759	10.02219759	0.011037446	[redacted]
0.3	3.014962686	0.110011001	Yes	Back	0.099668652	5.710593137	1.004987562	1.004987562	-0.2	-2.009975124	4.019950248	1.004987562	10.04987562	0.010946504	[redacted]
0.4	3.02654919	0.109170306	Yes	Back	0.132551532	7.594643369	1.00884973	1.00884973	-0.1	-0.756637298	2.774336758	1.00884973	10.0884973	0.010821265	[redacted]
0.5	3.041381265	0.108108108	Yes	Back	0.165148677	9.462322208	1.013793755	1.013793755	0	0	2.02758751	1.013793755	10.13793755	0.010663718	[redacted]
0.6	3.059411708	0.106837607	Yes	Back	0.19739566	11.30993247	1.019803903	1.019803903	0.1	0.509901951	1.529705854	1.019803903	10.19803903	0.010476289	[redacted]
0.7	3.08058436	0.105374078	Yes	Back	0.229231933	13.13402231	1.026861453	1.026861453	0.2	0.88016696	1.173555947	1.026861453	10.26861453	0.010261762	[redacted]
0.8	3.104834939	0.10373444	Yes	Top	0.260602392	14.93141718	1.03494498	1.03494498	0.3	1.164313102	0.905576857	0.905576857	9.055768573	0.011455067	[redacted]
0.9	3.132091953	0.101936799	Yes	Top	0.291456794	16.69924423	1.044030651	1.044030651	0.4	1.392040868	0.696020434	0.696020434	6.960204339	0.011645662	[redacted]
1	3.16227766	0.1	Yes	Top	0.321750554	18.43494882	1.054092553	1.054092553	0.5	1.58113883	0.527046277	0.527046277	5.270462767	0.011897366	[redacted]
1.1	3.195309062	0.097943193	Yes	Top	0.351444794	20.13630343	1.065103021	1.065103021	0.6	1.742895852	0.387101889	0.387101889	3.871018893	0.02528805	[redacted]
1.2	3.231098884	0.095785441	Yes	Top	0.380506377	21.80140949	1.077032961	1.077032961	0.7	1.884807682	0.26925824	0.26925824	2.692582404	0.03557382	0.03557382
1.3	3.269556545	0.09354537	Yes	Top	0.408907829	23.42869281	1.089852182	1.089852182	0.8	2.012034797	0.167669566	0.167669566	1.676695664	0.055791502	0.055791502
1.4	3.310589071	0.091240876	Yes	Top	0.43662716	25.01689348	1.10352969	1.10352969	0.9	2.128235832	0.078823549	0.078823549	0.788235493	0.115753321	0.115753321
1.5	3.354101966	0.088888889	No	NA	0.463647609	26.56505118	1.118033989	1.118033989	1	2.236067977	0	NA	#VALUE!	0.088888889	0.088888889
1.6	3.4	0.08650519	No	NA	0.489957326	28.07248694	1.133333333	1.133333333	1.1	2.3375	-0.070833333	NA	#VALUE!	0.08650519	0.08650519
1.7	3.44818793	0.084104289	No	NA	0.515549007	29.53878226	1.149395977	1.149395977	1.2	2.434015009	-0.135223056	NA	#VALUE!	0.084104289	0.084104289
1.8	3.498571137	0.081699346	No	NA	0.5404195	30.96375653	1.166190379	1.166190379	1.3	2.526745821	-0.194365063	NA	#VALUE!	0.081699346	0.081699346
1.9	3.553056181	0.079302141	No	NA	0.564569394	32.3474435	1.183685394	1.183685394	1.4	2.616567712	-0.249196925	NA	#VALUE!	0.079302141	0.079302141
2	3.605551275	0.076923077	No	NA	0.588002604	33.69006753	1.201850425	1.201850425	1.5	2.704163457	-0.300462606	NA	#VALUE!	0.076923077	0.076923077
2.1	3.661966685	0.074571216	No	NA	0.610725964	34.9920202	1.220655562	1.220655562	1.6	2.790069855	-0.348758732	NA	#VALUE!	0.074571216	0.074571216
2.2	3.720215048	0.072254335	No	NA	0.632748835	36.25383774	1.240071683	1.240071683	1.7	2.874711628	-0.394568263	NA	#VALUE!	0.072254335	0.072254335
2.3	3.780211634	0.069979006	No	NA	0.654082724	37.47617956	1.260070545	1.260070545	1.8	2.958426496	-0.438285407	NA	#VALUE!	0.069979006	0.069979006
2.4	3.841874542	0.067750678	No	NA	0.674740942	38.65980825	1.280624847	1.280624847	1.9	3.041484013	-0.480234318	NA	#VALUE!	0.067750678	0.067750678
2.5	3.905124838	0.06557377	No	NA	0.69473876	39.80557109	1.301708279	1.301708279	2	3.12409987	-0.520683312	NA	#VALUE!	0.06557377	0.06557377
2.6	3.969886648	0.063451777	No	NA	0.714090699	40.91438322	1.323295549	1.323295549	2.1	3.206446908	-0.559855809	NA	#VALUE!	0.063451777	0.063451777
2.7	4.036087214	0.061387354	No	NA	0.732815102	41.9872125	1.345362405	1.345362405	2.2	3.288663656	-0.597938847	NA	#VALUE!	0.061387354	0.061387354
2.8	4.103656906	0.059382423	No	NA	0.750929062	43.02506599	1.367885635	1.367885635	2.3	3.37086103	-0.635089759	NA	#VALUE!	0.059382423	0.059382423
2.9	4.172529209	0.057438254	No	NA	0.768450634	44.02897807	1.39084307	1.39084307	2.4	3.453127621	-0.671441482	NA	#VALUE!	0.057438254	0.057438254
3	4.242640687	0.055555556	No	NA	0.785398163	45	1.414213562	1.414213562	2.5	3.535533906	-0.707106781	NA	#VALUE!	0.055555556	0.055555556
3.1	4.313930922	0.053734551	No	NA	0.801790138	45.93919095	1.437976974	1.437976974	2.6	3.618135612	-0.742181664	NA	#VALUE!	0.053734551	0.053734551
3.2	4.38634244	0.051975052	No	NA	0.817645046	46.84761027	1.462114147	1.462114147	2.7	3.700976434	-0.77674814	NA	#VALUE!	0.051975052	0.051975052
3.3	4.459820624	0.050276521	No	NA	0.832981267	47.72631099	1.486606875	1.486606875	2.8	3.784090227	-0.810876477	NA	#VALUE!	0.050276521	0.050276521
3.4	4.53431362	0.048638132	No	NA	0.847816973	48.57633437	1.511437873	1.511437873	2.9	3.867502793	-0.844627047	NA	#VALUE!	0.048638132	0.048638132
3.5	4.609772229	0.047058824	No	NA	0.862170055	49.39870535	1.536590743	1.536590743	3	3.951233339	-0.878051853	NA	#VALUE!	0.047058824	0.047058824
3.6	4.686149806	0.045537341	No	NA	0.876058051	50.19442891	1.562049935	1.562049935	3.1	4.035295666	-0.911195796	NA	#VALUE!	0.045537341	0.045537341
3.7	4.763402146	0.044072279	No	NA	0.889498102	50.9644871	1.587800715	1.587800715	3.2	4.119699153	-0.944097723	NA	#VALUE!	0.044072279	0.044072279
3.8	4.841447375	0.042662116	No	NA	0.902506938	51.70983681	1.613829125	1.613829125	3.3	4.204449562	-0.976781312	NA	#VALUE!	0.042662116	0.042662116
3.9	4.92036584	0.041305246	No	NA	0.915100701	52.43140797	1.640121947	1.640121947	3.4	4.289549707	-1.009305813	NA	#VALUE!	0.041305246	0.041305246
4	5	0.04	No	NA	0.927295218	53.13010235	1.666666667	1.666666667	3.5	4.375	-1.041666667	NA	#VALUE!	0.04	0.04
4.1	5.080354318	0.038744673	No	NA	0.939105692	53.80679269	1.693451439	1.693451439	3.6	4.460798914	-1.073896035	NA	#VALUE!	0.038744673	0.038744673
4.2	5.16139516	0.037537538	No	NA	0.950546841	54.46232221	1.720465053	1.720465053	3.7	4.546943355	-1.106013249	NA	#VALUE!	0.037537538	0.037537538
4.3	5.242906692	0.036376864	No	NA	0.961632861	55.09750438	1.747696897	1.747696897	3.8	4.633428983	-1.138035189	NA	#VALUE!	0.036376864	0.036376864
4.4	5.325410782	0.035260931	No	NA	0.972377433	55.71312302	1.775136927	1.775136927	3.9	4.720250466	-1.169976611	NA	#VALUE!	0.035260931	0.035260931
4.5	5.408326913	0.034188034	No	NA	0.982793723	56.3099247	1.802775638	1.802775638	4	4.807401701	-1.201850425	NA	#VALUE!	0.034188034	0.034188034
4.6	5.491812087	0.033156499	No	NA	0.99289439	56.88865804	1.830604029	1.830604029	4.1	4.894875991	-1.233667393	NA	#VALUE!	0.033156499	0.033156499
4.7	5.575840744	0.032164683	No	NA	1.002691594	57.44899651	1.858613581	1.858613581	4.2	4.98266197	-1.265438004	NA	#VALUE!	0.032164683	0.032164683
4.8	5.660388679	0.031210986	No	NA	1.01219701	57.99461679	1.886796226	1.886796226	4.3	5.070764858	-1.297172406	NA	#VALUE!	0.031210986	0.031210986
4.9	5.745423969	0.03029385	No	NA	1.021421842	58.52316065	1.915144323	1.915144323	4.4	5.159164299	-1.328875653	NA	#VALUE!	0.03029385	0.03029385
5	5.830951895	0.029411765	No	NA	1.030376827	59.03624347	1.943650632	1.943650632	4.5	5.247856705	-1.360555442	NA	#VALUE!	0.029411765	0.029411765
5.1	5.916924877	0.028563268	No	NA	1.03907226	59.53445508	1.972308292	1.972308292	4.6	5.336834203	-1.392217618	NA	#VALUE!	0.028563268	0.028563268
5.2	6.003332408	0.027746948	No	NA	1.047518005	60.01836063	2.001110803	2.001110803	4.7	5.426088907	-1.423867302	NA	#VALUE!	0.027746948	0.027746948
5.3	6.090155991	0.026961445	No	NA	1.05572351	60.48850144	2.030051997	2.030051997	4.8	5.515612973	-1.455508979	NA	#VALUE!	0.026961445	0.026961445
5.4	6.177378085	0.026205451	No	NA	1.063697822	60.9453959	2.059126028	2.059126028	4.9	5.605398632	-1.487146576	NA	#VALUE!	0.026205451	0.026205451
5.5	6.264982043	0.025477707	No	NA	1.071449605	61.3895403	2.088327348	2.088327348	5	5.695438221	-1.518783526	NA	#VALUE!	0.025477707	0.025477707
5.6	6.35295207	0.024777007	No	NA	1.078987151	61.82140989	2.1176509	2.1176509	5.1	5.785724206	-1.550422827	NA	#VALUE!	0.024777007	0.024777007
5.7	6.441273166	0.024102193	No	NA	1.086318398	62.2414594	2.147091055	2.147091055	5.2	5.876249204	-1.582067093	NA	#VALUE!	0.024102193	0.024102193
5.8	6.529931087	0.023452158	No	NA	1.093450944	62.65012422	2.176643696	2.176643696	5.3	5.967005993	-1.613718602	NA	#VALUE!	0.023452158	0.023452158
5.9	6.618912297	0.022825839	No	NA	1.100392064	63.04782107	2.206304099	2.206304099	5.4	6.057987526	-1.645379328	NA	#VALUE!	0.022825839	0.022825839
6	6.708203932	0.022222222	No	NA	1.10714718	63.43494882	2.236067977	2.236067977	5.5	6.149189838	-1.677050983	NA	#VALUE!	0.022222222	0.022222222
6.1	6.7979376	0.021640338	No	NA	1.11372757	63.811899	2.265931253	2.265931253	5.6	6.24059755	-1.708735043	NA	#VALUE!	0.021640338	0.021640338
6.2	6.887670143	0.021079258	No	NA	1.120135001	64.17900803	2.295890048	2.295890048	5.7	6.332212873	-1.740432778	NA	#VALUE!	0.021079258	0.021079258
6.3	6.97782201	0.020538098	No	NA	1.126377117	64.53665494	2.32594067	2.32594067	5.8	6.424026612	-1.772145272	NA	#VALUE!	0.020538098	0.020538098
6.4	7.068238819	0.020016013	No	NA	1.132459767	64.88516511	2.356079606	2.356079606	5.9	6.516032661	-1.803873449	NA	#VALUE!	0.020016013	0.020016013
6.5	7.158910532	0.019512195	No	NA	1.13838855	65.22485943	2.386303511	2.386303511	6	6.608225106	-1.835618085	NA	#VALUE!	0.019512195	0.019512195
6.6	7.249827584	0.019025875	No	NA	1.144168834	65.55604522	2.416609195	2.416609195	6.1	6.700598222	-1.867379832	NA	#VALUE!	0.019025875	0.019025875
6.7	7.340980861	0.018556318	No	NA	1.149805753	65.87901688	2.44699362	2.44699362	6.2	6.793146468	-1.899159228	NA	#VALUE!	0.018556318	0.018556318
6.8	7.43236167	0.018102824	No	NA	1.155304231	66.19405648	2.47745389	2.47745389	6.3	6.885864489	-1.930956708	NA	#VALUE!	0.018102824	0.018102824
6.9	7.523961722	0.017664724	No	NA	1.160668986	66.50143432	2.507987241	2.507987241	6.4	6.978747105	-1.962772623	NA	#VALUE!	0.017664724	0.017664724
7	7.615773106	0.017241379	No	NA	1.165904541	66.80140949	2.538591035	2.538591035	6.5	7.071789351	-1.994607242	NA	#VALUE!	0.017241379	0.017241379
7.1	7.707788269	0.016832183	No	NA	1.171015229	67.09423034	2.569262756	2.569262756	6.6	7.164986278	-2.026460766	NA	#VALUE!	0.016832183	0.016832183
7.2	7.8	0.016436555	No	NA	1.176005207	67.38013505	2.6	2.6	6.7	7.258333333	-2.058333333	NA	#VALUE!	0.016436555	0.016436555
7.3	7.892401409	0.016053941	No	NA	1.180878463	67.65935204	2.63080047	2.63080047	6.8	7.35182597	-2.090225031	NA	#VALUE!	0.016053941	0.016053941
7.4	7.984985911	0.015683814	No	NA	1.18563882	67.93210044	2.66166197	2.66166197	6.9	7.445459836	-2.122135895	NA	#VALUE!	0.015683814	0.015683814
7.5	8.077747211	0.01532567	No	NA	1.19028995	68.19859051	2.692582404	2.692582404	7	7.53923073	-2.154065923	NA	#VALUE!	0.01532567	0.01532567
7.6	8.170679286	0.014979029	No	NA	1.194835373	68.45902408	2.723559762	2.723559762	7.1	7.633134596	-2.186015072	NA	#VALUE!	0.014979029	0.014979029
7.7	8.263776376	0.014643432	No	NA	1.199278472	68.71359489	2.754592125	2.754592125	7.2	7.72716721	-2.21798327	NA	#VALUE!	0.014643432	0.014643432
7.8	8.357032966	0.014318442	No	NA	1.203622493	68.96248897	2.785677655	2.785677655	7.3	7.821257525	-2.249970414	NA	#VALUE!	0.014318442	0.014318442
8	8.450443775	0.014003641	No	NA	1.207870566	69.20588504	2.816814592	2.816814592	7.4	7.915605562	-2.281976378	NA	#VALUE!	0.014003641	0.014003641
8	8.544003745	0.01369863	No	NA	1.212025657	69.44395478	2.848001248	2.848001248	7.5	8.010003511	-2.314001014	NA	#VALUE!	0.01369863	0.01369863
8.1	8.637708029	0.013403029	No	NA	1.216090675	69.67686317	2.87923601	2.87923601	7.6	8.104516176	-2.346044156	NA	#VALUE!	0.013403029	0.013403029
8.2	8.731551981	0.013116474	No	NA	1.220068379	69.9047688	2.910517327	2.910517327	7.7	8.199140275	-2.378105621	NA	#VALUE!	0.013116474	0.013116474
8.3	8.825531145	0.012838619	No	NA	1.223961429	70.12782418	2.941843715	2.941843715	7.8	8.293872643	-2.410185212	NA	#VALUE!	0.012838619	0.012838619
8.4	8.919641248	0.01256913	No	NA	1.227772386	70.34617594	2.973213749	2.973213749	7.9	8.388710222	-2.442282723	NA	#VALUE!	0.01256913	0.01256913
8.5	9.013878189	0.012307692	No	NA	1.231503712	70.55996517	3.004626063	3.004626063	8	8.48365006	-2.474397934	NA	#VALUE!	0.012307692	0.012307692
8.6	9.108238029	0.012054002	No	NA	1.235157776	70.76932762	3.036079343	3.036079343	8.1	8.578689307	-2.50653062	NA	#VALUE!	0.012054002	0.012054002
8.7	9.2027699	0.01180777	No	NA	1.238736859	70.9743959	3.06757233	3.06757233	8.2	8.673825209	-2.538680549	NA	#VALUE!	0.01180777	0.01180777
8.8	9.297311439	0.011568718	No	NA	1.242243156	71.17528998	3.099103813	3.099103813	8.3	8.769053108	-2.57084748	NA	#VALUE!	0.011568718	0.011568718
8.9	9.392017888	0.011336583	No	NA	1.245678782	71.37213682	3.130672629	3.130672629	8.4	8.864376433	-2.603031175	NA	#VALUE!	0.011336583	0.011336583
9	9.486832981	0.011111111	No	NA	1.249045772	71.56505118	3.16227766	3.16227766	8.5	8.959786704	-2.635231383	NA	#VALUE!	0.011111111	0.011111111
9.1	9.581753493	0.01089206	No	NA	1.25234609	71.75414547	3.193917831	3.193917831	8.6	9.055283521	-2.667447859	NA	#VALUE!	0.01089206	0.01089206
9.2	9.676776323	0.010679197	No	NA	1.255581627	71.93952806	3.225592108	3.225592108	8.7	9.150864566	-2.699680351	NA	#VALUE!	0.010679197	0.010679197
9.3	9.771898485	0.010472301	No	NA	1.258754205	72.1213034	3.257299495	3.257299495	8.8	9.246527599	-2.731928609	NA	#VALUE!	0.010472301	0.010472301
9.4	9.867117107	0.010271159	No	NA	1.261865583	72.29957221	3.289039036	3.289039036	8.9	9.342270452	-2.764192381	NA	#VALUE!	0.010271159	0.010271159
9.5	9.962429423	0.010075567	No	NA	1.264917455	72.47443163	3.320809808	3.320809808	9	9.438091032	-2.79647141	NA	#VALUE!	0.010075567	0.010075567
9.6	10.05783277	0.00988533	No	NA	1.267911458	72.64597536	3.352610923	3.352610923	9.1	9.533987312	-2.828765466	NA	#VALUE!	0.00988533	0.00988533
9.7	10.15324458	0.009700262	No	NA	1.27084917	72.81429386	3.384441526	3.384441526	9.2	9.629957332	-2.86107428	NA	#VALUE!	0.009700262	0.009700262
9.8	10.24890238	0.009520183	No	NA	1.273732114	72.97947439	3.416300793	3.416300793	9.3	9.725999198	-2.893397611	NA	#VALUE!	0.009520183	0.009520183
9.9	10.34456379	0.009344921	No	NA	1.276561762	73.14160123	3.44818793	3.44818793	9.4	9.822111073	-2.925735213	NA	#VALUE!	0.009344921	0.009344921
10	10.44030651	0.009174312	No	NA	1.279339532	73.30075577	3.48010217	3.48010217	9.5	9.918291183	-2.958086844	NA	#VALUE!	0.009174312	[redacted]

Table 8. Complete results of the mathematical modeling of the cube in Excel

Appendix C: Impact of the COVID-19 Pandemic

The research was initiated prior to and was conducted during the Covid-19 pandemic, during a time in which the use of virtual reality technology was being adopted more widely. The impact of the pandemic and the attention on public health measures undoubtedly changes some of the dynamics regarding the use and desirability of shared equipment, in particular headsets placed in direct contact with part of the user's face.

Testing the user experience through targeted focus groups proved impractical due to mandatory social distancing requirements and limited access to university laboratories. While the technical capabilities of the virtual reality system have been validated, expert user feedback and benchmarking would provide valuable feedback for future research and development.

The use of virtual reality technology will undoubtedly become more common in the future. Prices of hardware and the computational requirements of the connected computers have rapidly decreased, even during the course of this research. It remains likely that using shared virtual reality headsets will remain impractical for the next few years. The mostly likely scenario to adopt technology demonstrated by this research for specialized training scenarios is to procure the next generation of consumer virtual reality headsets for each trainee, similar to the adoption of individual tablet devices for employees observed over the last decade where each training participant will bring their own hardware with them to such training sessions without the need for sharing.