Enhancing Triage Training for Mass Casualty Incidents with Virtual Reality and Artificial Intelligence

Peng Xia
Auckland University of Technology
Peng.xia@aut.ac.nz

Ji Ruan
Auckland University of Technology
Ji.ruan@aut.ac.nz

David Parry
Murdoch University
David.parry@murdoch.edu.au

Jian Yu
Auckland University of Technology
Jian.yu@aut.ac.nz

Sally Britnell
Auckland University of Technology
Sally.britnell@aut.ac.nz

ABSTRACT

Mass casualty incidents (MCIs) occur with natural or man-made disasters. Training emergency staff for combating MCIs is essential, but the cost can be high as such incidents rarely occur, and a physical simulation is resource-intensive. Triage is a critical task in dealing with MCIs. In this paper, we propose to use Virtual Reality (VR) and Artificial Intelligence (AI) technologies to build a low-cost, high-efficient system for MCI triage training. Our system captures more comprehensive training data and utilizes state-of-the-art AI evaluation methods.

Keywords

Mass casualty incidents, triage training, virtual reality, artificial intelligence.

INTRODUCTION

Mass casualty incidents (MCIs) occur with natural or man-made disasters. Training emergency staff for combating MCIs is essential, but the cost can be high as such incidents rarely occur and physical simulation as a training tool is resource-intensive. In this paper, we present the design and methods of data analysis for a Virtual Reality (VR) training system that uses Artificial intelligence (AI) to evaluate the success of Mass Casualty Incident (MCI) triage of patients using a simulated environment.

Triage assesses the degrees of urgency to wounds or illnesses to allow clinicians to determine the treatment order for many patients or casualties. Given the limited time and resources during MCIs, the first responders must perform the triage swiftly and accurately. Traditional training methods for MCI triage skills include large-scale exercises (M. J. Reilly, 2011) and paper-based simulations (Fromm, 2018). Large-scale activities have the advantage of being realistic as they involve multiple teams in large open spaces, such as emergency staff, fire bridges and police. The paper-based simulations are easier to conduct but less realistic.

To overcome the disadvantages of the above methods, many researchers have proposed using Virtual Reality (VR) technology to provide an immersive training environment for triage skill training. The decreasing cost of VR devices has recently allowed more research on VR. Andreatta et al. (2010) compared the performance of VR and standardized patient (SP) drills on Simple Triage and Rapid Treatment (START) disaster triage algorithm. The performance was measured using a pretest and posttest triage rating/correctness. The result showed that VR could provide a feasible alternative for training emergency personnel in MCI triage. This is a significant early work in this area. However, the scale is small and the hardware is limited. Mills et al. (2020) present more recent research using VR kit and compared the performance of VR and live simulations in terms of immersion, clinical decision-making, satisfaction and cost. They concluded that VR can provide nearly identical simulation efficacy with much less cost than live simulations. However, this study is also small and further research is required to validate this finding. Lowe et al. (2020) studied the feasibility of VR training for MCIs using HTC VR platformat a larger scale. It measured the accuracy of triage and intervention and the experience rating for 207 participants. The work confirmed the feasibility of VR and the very positive experience by the subjects. Berndt et al. (2018) use a human-centred design process for building a VR training simulation for MCIs. The assessment was conducted on the presence and triage correctness. They concluded that VR training was useful, allowed a moderate level of presence and could be further improved. Bilek et al. (2021) presented a similar VR training application for an MCI on a highway.

Mossel et al. (2020) presented the VROnSite platform, which can support the immersive training experience for first responder squad leaders. The research evaluated two virtual disaster environments with two different navigation means using quantitative and qualitative measures. Koutitas et al. (2019) presented a VR-based training software to help emergency personnel get familiar with the bus-sized ambulance AmBus. The key tasks trained were on the locations of items on the bus, measured by time and accuracy. VR was shown to be more effective than the traditional training method. Caballero & Niguidula (2018) presented a case-driven approach to building VR training for emergency preparedness. Bjørn et al. (2021) provided technical guidance in creating a collaborative work environment in VR.

These studies have shown that VR can provide a viable solution for emergency staff training. However, these studies use some forms of tests, questionnaires, or interviews for evaluation, which are somewhat limited. We propose to improve the evaluation by collecting more comprehensive data on the training and processing them with state-of-the-art AI algorithms. More specifically, we will utilize multiple VR device sensors to collect the VR controllers' timestamps and coordinates. This allows us to detect and quantify the events and completion of the tasks during the training. We will also process the videos captured by cameras using AI-based body and action recognition algorithms to better understand the behaviour patterns. Moreover, the audio data will be processed using AI-based speech recognition algorithms. We will measure the semantic similarity of the participants' descriptions with a pre-defined description of the scenario information, which provides another way to evaluate performance. The following provides a more detailed description of our system design and the individual components, evaluating these as future research.

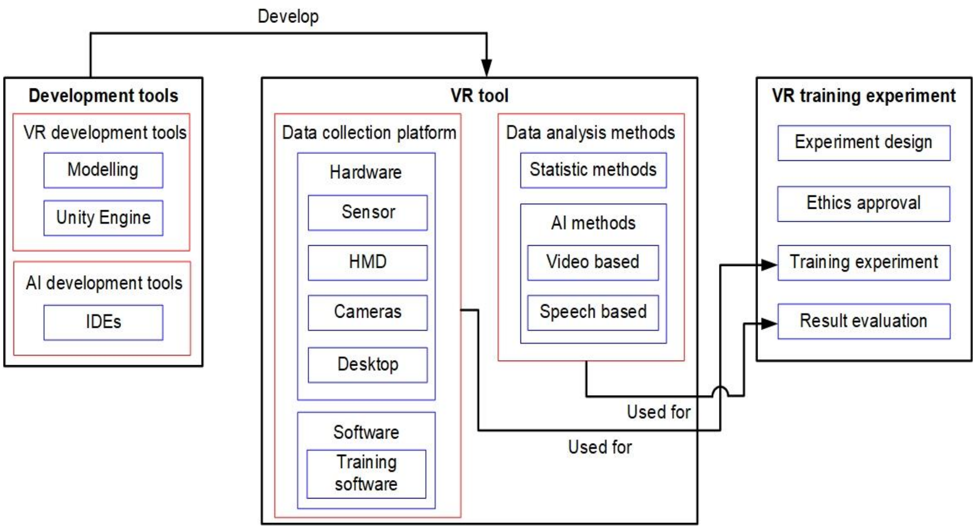

Figure 1: Research Framework

SYSTEM DESIGN

Our research framework has three major components (see figure 1). The central one is the VR tool that have built. It consists of a data collection platform and a set of data analysis methods. The left is the VR and AI development tools for developing our VR tool; the right is the component of the VR training experiment for evaluation. In particular, the data will be collected from the training experiments using the data collection platform and the results will be evaluated using the data analysis methods.

Main Component: Development Tools

We have two types of development tools. One is for VR, and the other is for AI. The VR development tools consist of the Modelling tools and the Unity Engine. We use the modelling tools to build the objects and assemble those objects in the Unity Engine to create a VR scenario. AI development tools include various development environments, and we implemented AI algorithms based on those environments. In the following, we will present details of those development tools.

Modelling tools

As VR modelling contains a complete process and involves various tools, we demonstrate our modelling process in Figure 2.

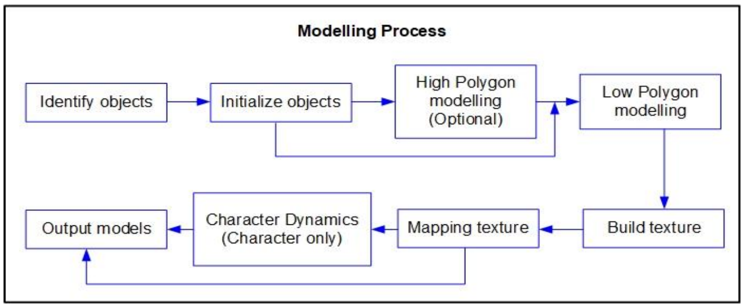

Figure 2: Modelling process

The first step in modelling is identifying objects such as cars, buildings, and humanoid characters. The objects are classified as either static or dynamic objects. We set up a target model for each object and find the reference appearance via different sources. Then, we start to initialize the target model. *Blender* and *Google Sketchup* are two draft modelling tools involved in this step. Some static objects (e.g. cars, street lights) are built by using those tools. Besides the static objects, dynamic objects such as humanoid characters will have a further processing step called Polygon optimization. This project deals with two types of humanoid characters in modelling. One type is normal characters that use default joint mapping, and the other type is patient characters that contain special joint connections or meshes for simulating patient conditions (e.g. Trauma, bone break). For normal humanoid character modelling, we directly build the low polygon model. For the special humanoid character modelling, we firstly build the high polygon model, carving the particular part (e.g. scars) on the high polygon model. And then, we convert the high polygon model to a low polygon model. In the polygon modelling steps, *Adobe Fusion* (for low polygon modelling), *Zbrush* (for high polygon modelling) and *Autodesk Maya* (for high/low polygon modelling) are used.

When building the polygon models, we also create textures for those models. Similar to special humanoid character modelling, we customize the texture for the virtual patient. Adobe Photoshop is used in this step. Finally, we use Maya and Unity to integrate the texture with the polygon models. All humanoid characters must add dynamics or so-called 'bones' or 'joints' to present animation in the scenario. Each character contains 20 joints group to simulate body action. For instance, the 'Shoulder' joint group includes two joints, connecting with the 'Upper Arm' and 'Chest' joint groups to simulate movement and rotating angle of the human shoulder. We use Maya and Unity Engine to build such dynamics for the humanoid characters. In the end, we output those models as *.FBX files and import them into the Unity Engine.

Unity Engine

We used the Unity Engine to develop the VR tool. Unity Engine is a game engine that consists of different essential development components (e.g. Physics, materials, animation bakin). Developers can use Unity Engine to initiate prototypes and conduct further development efficiently. Figure 3 shows the development process using Unity Engine. The first step is to build the environment. Then, we combine objects from the Unity asset market and custom-made sources to construct a small city-style environment. After that, we added several static objects, such as a car, stone, and fire flame, to present an MCI scenario. Currently, there is no interaction between the trainee and those static objects.

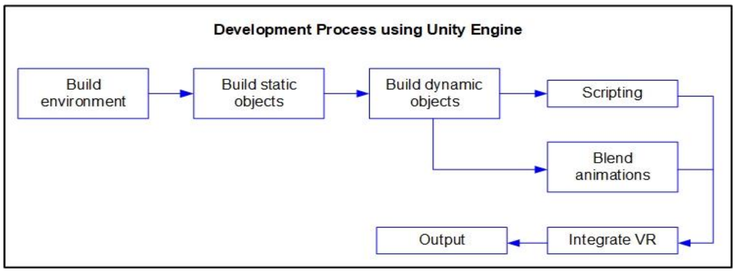

Figure 3: Development process of Unity Engine

The third step is building dynamic objects such as humanoid characters. The properties such as position, interaction method, and behaviour logic of those dynamic objects must be addressed carefully. We use two significant sub-steps, Scripting and Blend animations, to control the performance of the involved objects. We use Microsoft Visual Studio in the scripting step and focus on the dynamic object's interaction event and behaviour logic. An example is given below to indicate how the script works. In the prototype, we have built several scripts to control the vital features of the virtual patient. Those scripts can generate an arbitrary number of virtual patients with their vital features, and different vital features will also affect the visual expression of virtual patients. After reviewing the Australasian Triage Scale (ATS), we extract a triage feature schema. Figure 4 gives an abbreviated version of the triage feature schema. We notice that some features in the triage schema could be quantified. For instance, according to the ATS guidelines, a heart rate of 80 beats per minute is a vital threshold for determining if the patient belongs to ATS-1 or ATS-2. If needed, we can generate data for each quantifiable feature.

Major Features	Component	Features
	Respiratory	Respiratory Rate (RR)
		Respiratory Status (e.g. Extreme respiratory distress)
	Blood Pressure	Blood pressure (BP)
	Glasgow Coma Score	Glasgow Coma Score (GCS)
	Seizure	Seizure
	Circulation	Skin Color
		Heart Rate (HR)
		Blood loss
		Blood Glucose
	Pain	Chest
		Other pains
Other Feature	Other features include psychological features, and trauma, etc.	

Figure 4: Triage feature schema

AI development tools

The AI development tools for developing Machine Learning (ML) algorithms include video and speech-based algorithms. We use Pycharm as the primary IDE and Ubuntu OS to build and test the AI algorithms. Most ML algorithms and models use Python as a major development language. Python has several advantages that make it popular in academia and industry. Those advantages include 1) being Supported by a large number of third-party add-on toolkits and libraries; 2) Highly readable programming style; 3) High portability and extensibility making it quickly adaptable to the different environments; 4) Free to use. Therefore, Python was the primary programming language to develop the AI methods, with Pycharm as the primary IDE because it makes Python development more efficient. Besides the Pycharm, we also use Microsoft Visual Studio to compile and test some C++ ML projects. Furthermore, the Linux core-based operation system, such as Ubuntu OS, has a higher capacity than Windows, which can also increase development efficiency.

Main Component: VR tool

The VR tool consists of a data collection platform and a set of data analysis methods.

Data collection platform

The data collection platform has both hardware and software. The hardware consists of four sub-components: 1) Motion sensor; 2) Head-mounted device (HMD), 3) camera, and 4) desktop computer.

1) Motion sensor

   As the VR devices will communicate with the software and exchange sensor data such as device coordinates during run-time, we developed several scripts to record sensor data in a specific circumstance (e.g. start doing the task). After building sequences for the sensor data, we can reconstruct and evaluate the user's behaviour in the training session.

2) HMD

   With the rapid development of VR devices in the last ten years, commercial-level HMD can now provide an immersive experience for customers with small transmitting latency. However, most HMDs focus on 'display and experience VR scenario' rather than 'interaction with objects in VR scenario', which means they do not have any interaction functionality. In this project, we plan to use Oculus Rift VR HMD as the HMD and the Oculus motion sensor and controller kit to achieve interaction requirements.

3) Camera

   We use several cameras to collect video and audio data from the trainees. Those cameras will be hosted at different positions on the training ground. One wide-angle camera will be set in front of the trainees at a height of 2000mm. In addition, two side cameras will be placed on the left and right sides of the user, with chest-level height (1600mm). Additional voice recording devices will be prepared in case of that the digital camera may not be able to collect clean sound while training.

4) Desktop

   We use a desktop computer to host the training software and collect the experimental data. The desktop configuration includes an i7 8-core CPU, Nvidia GTX 1070 graphic card, 32G RAM, and 500G SSD.

In terms of software requirements, we have the following three critical aspects:

1) MCI skill training functionalities

   Based on our literature review, we built VR MCI scenarios (e.g. a car crash scenario). There are some virtual patients with various conditions in the scenario, and the trainee needs to interact with those patients by performing tasks such as measurement of vital signs using the information gathered to assign triage categories for each patient. During the training session, the training software collects the quantified data (e.g. timeframes between events and sensor coordinator information) and, in the future, combine it with other unstructured data (e.g. video data and audio data) for evaluation.

2) Modularity

   The proposed training software should be able to extend its functionalities in the future at a low cost, and the modularity of the functionalities could make it feasible. We have done the following to achieve modularity: a) reviewing the add-on tools to make sure they can embed the training software and work independently; b) encapsulating some fundamental functions (e.g. data input class) and make sure they are independent of the business logic functions; c) optimizing the software architecture to follow the high cohesion and low coupling principle. The goal is to minimize the impact of adding or changing functionalities in the future.

3) Supporting multiple MCI triage scenarios

   One main advantage of VR tools is the ability to change to a different scenario at a relatively low cost. In our implementation, we have provided two scenarios (see the details below).

In this VR tool, we have built two MCI scenarios: one is a car crash scenario, and the other is an earthquake scenario. In these scenarios, the trainee will be asked to complete several tasks, which include: a) measuring the vital signs of each patient; b) triaging patients; and c) reporting the patients' vital signs and categories. The details of the two scenarios are given below:

1) Car crash scenario:

Figure 5 shows the car crash scenario of the training software. There are ten patients (three Immediate, four urgent and three delayed) and a few bystanders in the scenario.

Figure 5: Car crash scenario

We designed a category indicator for each patient, see Figure 6 (Left). Those indicators are located above each patient and use a different colour to indicate the true category of a patient. The colours refer to the triage standard: Black – deceased, Red - Immediate, Orange - Urgent, and Green - Normal. The indicators can be set to either visible or invisible. To train the MCI triage skill, we also designed two measurement tools to measure the patient's vital signs and a triage tool to assign the different patient categories. Figure 6 (Right) shows the triage toolkit. The trainee can use controllers to set different triage categories on the wristband for the virtual patient, and colours also represent those categories. The training software will compare the triage category assigned on the wristband with the actual triage category based on the ATS guidelines (reference here) and then output the correctness of each decision.

Figure 6 (Left): Triage Indicator, (Right): Triage toolkit

2) Earthquake scenario:

The second scenario is an earthquake scenario with seven patients. We add an extra task that the trainees must ensure their safety. A few patients are in a dangerous situation (e.g. under collapsing construction), and the trainees must check the environment. The earthquake scenario includes two immediate, three urgent and three delays and reused several components from the car crash scenario, which reduces development time. Figure 7 shows the screenshot of the earthquake scenario.

Figure 7: Earthquake scenario

The training software will collect sensor and performance data related to triage decisions during training sessions. Table 1 shows the triage data items and descriptions.

Table 1. Triage data items

Data Item name (Type)	Description
Patient ID (string)	Virtual patient's index
Assigned category (int)	The triage category assigned by the trainee
True category (int)	Category which pre-defined
Result (bool)	Comparison results of Assigned category
Start time(string)	The start time of the current training scenario
Triggering time (string)	The time of triggering the current recording event
Timespan (string)	Subtraction of the current time and start time

Data related to vital sign measurement is also collected, for example, when the trainee attempts to measure the patient's vital signs. In data evaluation, we compare the trainee's report with the pre-defined vital signs to determine the accuracy of triage decisions. Table 2 shows the vital sign data item descriptions.

Table 2. Vital Sign data items

Data Item name (Type)	Description
Patient ID (string)	Virtual patient's index
Var (1…n)	variables of a vital sign depend on different measurement types
Start time(string)	The start time of the current training scenario
Triggering time (string)	The time of triggering the current recording event
Timespan (string)	Subtraction of the current time and start time
Measurement type (string)	Indicates which type of measurement is conducted
Moving distance (unit)	Trainee's moving distance in the virtual world

Data analysis methods

We use quantitative analysis as our main research method for data analysis. We use traditional and AI-based methods to evaluate the training efficiency. To evaluate those methods, we collect four types of data from the experiment: personal *information, sensor data, video data and speech data*.

The personal information includes the participant's character features, such as age, gender, and previous VR experience. It aims to use this information to determine the influence of the trainees' background on their performance in the VR environment (for example, time to task or decision and movement in the scenario).

The sensor data includes quantified performance information such as timestamps and coordinates of the VR controllers. In each VR session, the participants will perform essential tasks; one is measuring the vital signs and the other is assigning triage categories. We developed a triggering module to trigger the sensor data recording when a task is performed. The module sets a triggering range for each virtual patient. When the participant accesses the controller's triggering range, the module starts collecting sensor data from the controller. The sensor data can provide the availability to derive time and distance for individual tasks (e.g., vital sign measuring) performed on specific patients. Thus, we can assess the accuracy of the triage in a precise way. By analyzing the raw data collected from the sensors, not only can the data items in Tables 1 and 2 be acquired, but also information regarding the timespan can be calculated. Timestamp-based information can indicate 1) when the participant grabs/picks up a measurement tool (e.g., thermometer), 2) when the participant uses this measurement tool to measure a vital sign, and 3) when the participant releases a measurement tool. Also, individual performance regarding the triage decision-making can be derived, which includes 1) duration of holding a measurement tool, 2) orders of measuring different virtual patients and 3) time spent between vital sign measurement and making triage category decision. In addition, we can measure the distance of the joysticks for each task (this helps to understand the cost of movements) in the virtual and physical worlds.

The video data from the cameras will be fed into an AI algorithm for body action recognition. Our system will recognize the human body skelton, and based on the movement pattern of joints, different actions will be labelled as actions such as 'stand', 'turn around', and 'squat'. This allows us to 1) precisely capture the action patterns of the participants and 2) understand the behaviour patterns for evaluation. We plan to use Spatial-Temporal Graph Convolution Networks (ST-GCN) (Yan, 2018) as the AI algorithm. Once the video data is collected, the body action pattern can be extracted by ST-GCN. ST-GCN is designed on a state-of-the-art body recognition algorithm named Openpose. Openpose recognizes the human body by identifying the coordinates of important joints and provides a visualization of the skeleton. After which, different action labels such as 'stand', 'turn around', and 'squat' are added to the proceeded video based on the movement pattern of joints.

The audio data will be used for speech recognition and evaluation. We plan to observe how the participants would report the scenarios and how they would make the decision. We propose to use Pytorch-Kaldi (Ravanelli, 2019) and Bert for the speech-based method. We have established the Kaldi toolkit and successfully proceeded with some examples. During the evaluation, we expect to use Kaldi to recognize the participant's speech and translate it to plain-text format as participant description. After receiving the participant description from Kaldi, we plan to use Bert (Devlin, 2018) to read its semantics and compare it with a pre-defined description of the scenario information. We expect to observe the training efficiency by evaluating the comparison result.

Main Component: VR Training Experiment

We will conduct experiments to evaluate the efficiency of the VR tool. The experiments will involve participants with some basic prior knowledge of triage. The expected participants would be paramedic students or professionals. For each participant, a training session with a length of 15–30 minutes will be conducted. In the session, the participant will finish both scenarios described above.

CONCLUSION

In this paper, we present the system design of our VR tool to train and assess the emergency staff for MCI triage skills. Compared with traditional training methods, VR technology allows for building multiple scenarios with many virtual patients at a lower cost and in less time. Another clear advantage of VR technology is bringing the participant an immersive experience. Compared with other VR training systems, the benefit of our system will be a more comprehensive collection of training data and the utilization of state-of-the-art AI evaluation methods.

REFERENCES

Andreatta, P. B., Maslowski, E., Petty, S., Shim, W., Marsh, M., Hall, T., . . . Frankel, J. (2010). Virtual reality triage training provides a viable solution for disaster-preparedness. *Academic Emergency Medicine, 17,* 870-876. doi:10.1111/j.1553-2712.2010.00728.x

Atlantic Life Safety Services. (2018). *MCI Command Village 'Putnamville'*. Retrieved from https://www.atlanticlife safety.com/mci-command-village

Berndt, H., Wessel, D., Mentler, T., & Herczeg, M. (2018). Human-centered design of a virtual reality training simulation for mass casualty incidents. doi:10.1109/VS-Games.2018.8493427

Bhagavathula, R., Williams, B., Owens, J., & Gibbons, R. (2018). The Reality of Virtual Reality: A Comparison of Pedestrian Behavior in Real and Virtual Environments. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *62*, pp. 2056–2060.

Billis, A., Papadimitriou, G. I., Douligeris, C., Papavramidis, T. S., Krizea, M., Bamidis, P. D., . . . Kallergis, D. (2019). Functional and non-functional requirements of a smart triage system for Emergency Departments: The case of IntelTriage project. doi:10.1109/SEEDA-CECNSM.2019.8908320

Devlin, J. C. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv*.

Fertleman, C., Aubugeau-Williams, P., Sher, C., Lim, A.-N., Lumley, S., Delacroix, S., & Pan, X. (2018). A discussion of virtual reality as a new tool for training healthcare professionals. *Frontiers in public health, 6*, 44.

Fromm, A. (2018). Training und Simulation von MANV-Lagen. *Im Einsatz*, 42-45.

Gunn, T., Jones, L., Bridge, P., Rowntree, P., & Nissen, L. (2018). The use of virtual reality simulation to improve technical skill in the undergraduate medical imaging student. *Interactive Learning Environments, 26*, 613–620.

Hill, M. G., Sim, M., & Mills, B. (2020). The quality of diagnosis and triage advice provided by free online symptom checkers and apps in Australia. *Medical Journal of Australia, 212*, 514-519. doi:10.5694/mja2.50600

M. J. Reilly, D. M. (2011). *Health Care Emergency Management. Principles and practice.* MA: Jones & Bartlett Learning.

Mills, B., Dykstra, P., Hansen, S., Miles, A., Rankin, T., Hopper, L., . . . Bartlett, D. (2020). Virtual Reality Triage Training Can Provide Comparable Simulation Efficacy for Paramedicine Students Compared to Live Simulation-Based Scenarios. *Prehospital Emergency Care, 24*, 525-536. doi:10.1080/10903127.2019.1676345

Mossel, A., Schoenauer, C., Froeschl, M., Peer, A., Goellner, J., & Kaufmann, H. (2020, December). Immersive training of first responder squad leaders in untethered virtual reality. *Virtual Reality, 25*, 745–759. doi:10.1007/s10055-020-00487-x

Povey, D. a. (2011). The Kaldi speech recognition toolkit. *IEEE 2011 workshop on automatic speech recognition and understanding*.

Ravanelli, M. P. (2019). The pytorch-kaldi speech recognition toolkit. *IEEE International Conference on Acoustics, Speech and Signal Processing*, 6465-6469.

Yan, S. X. (2018). Spatial temporal graph convolutional networks for skeleton-based action recognition. *Thirty-second AAAI conference on artificial intelligence*.

Zachariasse, J. M., van der Hagen, V., Seiger, N., Mackway-Jones, K., van Veen, M., & Moll, H. A. (2019). Performance of triage systems in emergency care: a systematic review and meta-analysis. *BMJ open, 9*, e026471.