A Virtual Reality Mining Training Simulator for Proximity Detection

44

Erik Marsh, Joshua Dahl, Alireza Kamran Pishhesari, Javad Sattarvand, and Frederick C. Harris Jr.

Abstract

Many applications in industrial mining rely on large manually operated trucks to transport materials around the mine. These trucks are often enormous, with very limited visibility for the driver. The combination of limited visibility and a truck with a substantial amount of weight is a recipe for accidents resulting in severe property destruction or even loss of human life. To solve these issues, we implement a simulation of a LIDAR-based proximity collision detection system to notify drivers of an imminent collision with something (or someone) outside of their field of view. We have developed a virtual reality training simulation to help train miners how to use this new system. Our simulation focuses on two main participant types: truck drivers and ground workers. This separation allows both parties to gain experience operating around the other in a low-risk virtual environment. Our final result aggregated useful features from a variety of past works and enhanced them with more immersive input and output devices.

Keywords

Driving training · Hazard awareness · Immersive locomotion · LIDAR · Mining safety · Multiplayer · Open pit mine · Photogrammetry · Serious games · Unity

44.1 Introduction

The use of virtual reality (VR) in the mining industry has been explored in depth over the years. One key use case for VR in this industry is training simulators. Traditionally, training in mines requires that the employees use the physical mine and its equipment, resulting in lower productivity and, therefore, lower profit [1,2]. The mining industry also makes use of heavy machinery that is often dangerous and can be potentially fatal if the workers are not properly trained to operate the machines [1,3,4]. VR systems provide a solution to this issue: instead of training procedures taking place in the real mining environment (and hence wasting resources), the training can instead be performed in a separate collaborative virtual environment (CVE). This allows the mine to continue operating while employees are being trained to properly and safely use the equipment.

This paper documents a VR system for the training of open-pit mine employees who need to drive mining dump trucks. These trucks are huge, giving the driver little visibility of their surroundings. Workers that are walking on foot around the truck may not be seen by the driver, creating a situation that is potentially fatal to the on-foot workers. Even larger obstacles, such as conventionally sized vehicles, are easily missed by the dump truck driver. Our system proposes

This material is based in part upon work supported by the National Science Foundation under grant numbers OIA-2019609 and OIA-2148788 and by the Center for Disease Control and Prevention and the National Institute for Occupational Health and Safety under contract number 75D30119C06044. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation, the Center for Disease Control and Prevention, or the National Institute for Occupational Health and Safety.

E. Marsh · J. Dahl · F. C. Harris Jr. (✉)
Computer Science and Engineering, University of Nevada, Reno, Reno, Nevada
e-mail: erik.i.marsh@gmail.com; joshuadahl@nevada.unr.edu; fred.harris@cse.unr.edu

A. Kamran Pishhesari · J. Sattarvand
Mining and Metallurgical Engineering, University of Nevada, Reno, Reno, Nevada
e-mail: akamranpishhesari@unr.edu; jsattarvand@unr.edu

S. Latifi (ed.), *ITNG 2023 20th International Conference on Information Technology-New Generations*, Advances in Intelligent Systems and Computing 1445,
https://doi.org/10.1007/978-3-031-28332-1_44

a simulation of a LIDAR-based proximity detection system that alerts the driver of entities that draw near to the truck, even if these entities are in one of the driver's many blind spots. To add realism to the simulation, we allow another user to roam the CVE as an on-foot worker who may wander into the blind spots of the truck driver.

The remainder of this paper is structured as follows. Section 44.2 discusses the hardware and software contexts of the project, as well as a summary of the features of systems similar to ours. Section 44.3 covers the design goals and implementation details of the project. Section 44.4 compares the features of our system to the features of the related systems outlined in Sect. 44.2. Section 44.5 overviews the conclusions drawn from our work and identifies several directions that the project can be taken in the future.

44.2 Background and Related Work

44.2.1 LIDAR

LIDAR systems have been applied to facilitate obstacle detection around autonomous vehicles [5]. These systems must construct a virtual representation of the vehicle's surroundings and analyze them in real-time to extract the locations of obstacles in order to avoid collisions. Obstacles in this virtual representation must be distinguished from drivable ground as well. While our system is not an autonomous one, these concerns still apply—the sheer scale of the trucks used in surface mines creates conditions where the driver has extremely low visibility of their surroundings. A LIDAR-based proximity detection system may be an effective countermeasure to this visibility issue.

44.2.2 Software and Hardware

44.2.2.1 Unity and MuVR

This project was built in Unity [6], which is a real-time, 3D rendering platform. A user of Unity may create scenes consisting of 3D GameObjects. These GameObjects may be scripted to respond to user input and other GameObjects in the scene using the C# programming language. Other common systems, like physics and lighting, are handled by Unity as well.

To facilitate our VR interaction capabilities, we use a Unity library we developed called MuVR [7]. MuVR is a multiuser VR framework designed for implementing networked VR experiences that allow various users to interact in the same CVE even if each user has a different input system. Each user in the CVE is represented as an avatar with twelve slots. Each slot represents a position and location of an aspect of the avatar and is transferred across the network. This slot-based system is typically applied to represent humanoid user avatars.

If an arbitrary model can be represented within the constraints of the twelve pose-slots (i.e. a humanoid avatar), it can be represented in a MuVR system. Unfortunately, our application requires the use of an avatar that does not fit well within these constraints: a truck. The points of interest are instead the body and wheels of the truck. We had to modify the MuVR library slightly to accommodate our truck avatar.

44.2.2.2 Hardware Overview

The system presented in this paper makes use of several hardware components. Our VR functionality is facilitated primarily through the use of the *HTC Vive* [8]. This system consists of a head-mounted display (HMD) and two wand-style controllers. All of our visuals are shown to the user via the *HTC Vive* HMD display to provide an immersive VR experience.

Our input system primarily consists of two primary locomotion-focused input devices. The first is a *Virtuix Omni* [9] which is a particular brand of unidirectional VR treadmill, used for on-foot workers. The user is placed on a gait-negating treadmill that allows them to walk in the CVE without moving in reality.

For truck-driving users, a *Logitech G923 TRUEFORCE Sim Racing Wheel* [10] is used to steer the truck. The wheel comes with a set of pedals that are used for the truck's acceleration and braking. Additionally, a *DOF Reality H6 Consumer Motion Simulator Platform* [11] racing chair serves as an output device. The chair is mounted on a hydraulic system that tilts the chair in response to the movements of the truck in simulation. For example, if the truck drives up a hill in the CVE, we can relay the current angle of the truck to the racing chair. The racing chair then replicates this angle in reality, giving the user the impression they are at the same angle as the truck. The chair's hydraulic system is also leveraged to provide vibration feedback to the user when they collide with an object. The entirety of the truck driving setup can be seen in Fig. 44.1.

44.2.3 Existing Systems

While VR has a long history of being used for training purposes in the mining industry, the authors were not able to find many examples of VR training simulators for the operation of surface mine dump trucks that integrate proximity detection systems into their training.

Lucchesi et al. [1] present a simulator for the operation of surface mine dump trucks. This provides the advantage of providing training to employees without the company having to suffer the cost of training using a real truck. Their system

Fig. 44.1 The truck driving input and output system with a user seated inside

provides a set of immobile and mobile hazards for the driver to navigate around. Much attention is paid to the driving of the truck: the amount of material loaded onto the truck will have an effect on how the truck handles. Additionally, truck handling will differ depending on weather conditions (wet vs dry). Differing visibility levels are also implemented. The user is also able to customize the CVE that they will be driving in.

McMahan et al. [4] present two systems for open pit mines: one for training individuals how to drive dump trucks and another for training individuals how to operate conveyor belts. Of interest is the truck simulation. This is divided into three separate systems. The first is a tour of the truck, which shows the user several parts of the truck important for inspection. The second is a pre-shift inspection, where the user must navigate to the points shown in the first system, checking for defects. The third is a non-interactive driving simulation that shows the user either a good outcome or a bad outcome depending on if they corrected all defects or not. The VR effect is achieved using a CAVE, a large cube-shaped room where a CVE is projected onto the walls and floor.

Bellanca et al. [12] created a Unity framework for the creation of CVEs for underground mining for use in training mine employees how to escape in the event of an emergency. Of interest is their implementation of proximity detection systems. They choose to implement three types of proximity detection: standard Unity box colliders, "shells" mimicking the appearance of magnetic fields, and deformable magnetic field shells to replicate magnetic interference. Additionally, the system provides robust data logging capabilities about the state of an object at a certain point in time.

Anderson et al. [2] presents a system similar to the previous system. It is again focused on the idea of training underground mine employees on how to evacuate in the event of an emergency. Their system was also implemented in Unity and uses the *HTC Vive* to provide a VR experience. The feature set of the project had a lot to do with locomotion, providing the user with three different options: Movement via the touchpad on the *HTC Vive* controllers, walking in place with sensors placed on the user's ankles, and a *Virtuix Omni* gait-negating treadmill. One of the other key features is that this system provides networking capabilities: one user acts as a guide with access to the mine layout to help the other users evacuate the mine.

Finally, while not explicitly related to mining, Golovina et al. [3] explore a VR training simulation designed to teach its users about the dangers of being too close to construction machinery. When a user approaches a machine too closely, their location is marked on the ground to highlight the mistake. The principal use case is to collect information about how the users navigate the CVE as a way to study potentially deadly workplace accidents without bringing harm to the subjects. The system is implemented in Unity and collisions are determined through the use of Unity's built-in colliders. Two levels of collisions are implemented: a close-call collision with the machine and physical contact with the machine. To achieve a VR effect, an *HTC Vive Pro* is used. The user moves around in the scene using the touchpad present on the *HTC Vive* controllers.

44.3 Design and Implementation

The principal design aspect of the VR training simulator introduced in this paper is the interaction between two types of users. The first type is the truck driver and the second type is the worker. Both types of users have unique input systems, can coexist in the CVE, and are depicted in Fig. 44.2.

The primary activity given to the truck-driving user is the ability to drive a truck around the CVE. This is designed to be immersive: the truck driver is placed within a racing chair and a VR HMD. The use of the HMD gives the truck driver

Fig. 44.2 Picture of the truck and worker walking around the truck

a one-to-one mapping from their head movements to their viewport in the CVE. The truck driver is also given a steering wheel and a set of pedals to control the truck with. These input devices behave as one would expect: the steering wheel controls the turning rate of the truck, the accelerator pedal accelerates the truck, and the brake decelerates the truck. A Unity library [13] is used to handle the inputs from these devices and relay them to the truck.

To further immerse the truck driver in their driving experience, the racing chair that they sit in responds to the truck's position. The racing chair is mounted on several mechanical cylinders that allow the chair limited movement. We limit the chair's movement to two degrees of freedom: the pitch angle (forward/backward tilt) and roll angle 0 (left/right tilt) of the chair. These angles are restricted to the domains of $[-80°, 45°]$ and $[-90°, 90°]$ respectively.

An API was developed for the racing chair [14] that allows it to be controlled programmatically. This API is designed for Unity and provides the user with functions that send API data (namely the target yaw, pitch, and roll of the chair, along with some additional configuration parameters) through UDP over the loopback interface. A separate application provided by the API authors is used to receive this information and move the racing chair accordingly.

It should be noted that our racing chair had issues where, at certain angle configurations, the chair would collide with its base. This may have been a result of an error in building the chair. The collisions would interrupt the smooth motion of the chair and create a loud noise, interrupting the immersion of the system. The reason for our angle restrictions is to prevent the chair from entering a state where it would collide with itself.

The primary role of the worker is to walk around the CVE. In the context of the system, the worker is a moving, low-visibility obstacle for the truck driver to avoid. Similar to the truck driver, the worker wears an HMD to map their head motion to their viewport in the CVE. The worker's primary input device is the *Virtuix Omni*. While in the device, the worker's foot movements, hip orientation, and HMD orientation are tracked as input. A library [9] mapping these device inputs to Unity inputs was provided by request to the authors.

44.3.1 Multiuser Virtual Reality

We began the implementation of this project with a preliminary version of the MuVR library [7]. MuVR is designed to represent users as a humanoid avatar with 12 points of interest. These points of interest include information such as head position, hand position, and foot position which are used to determine what pose the avatar should be in. However, we were not able to represent the truck driver and worker as such an avatar. It is clear to see that the truck itself does not play well with the avatar system, since a truck is not a humanoid figure. The less clear case is related to the implementation of the worker. For our purposes, the worker is simply a static avatar that slides along the ground. The only information about the worker being sent and received across the network is their world-space position and rotation in the CVE. Since our colleagues only requested this functionality, we deemed it unnecessary to utilize the full system.

However, the preliminary version of MuVR was not designed to support these deviations. Thus the library was enhanced to support a dynamic set of pose-slots, allowing the addition and removal of pose-slots.

44.3.2 Virtual Environment

Creating the CVE proved to be difficult due to limited access to data about the mine. Our colleagues in mining gave us a relatively low-resolution mesh considering the scale of our environment (~400,000 triangles representing an approximately one square kilometer area) as well as a low-resolution image (4096 square pixels), of the South Arturo mine that they extracted from drone footage using photogrammetry [15]. This mesh while being relatively low resolution from the perspective of a 1×1 km area still wound up using more computing resources to render than we were willing to allocate. Unfortunately, the processes we took to solve these issues are not generalizable.

Unity provides a built-in terrain system that given a heightmap will create a mesh conforming to it that is automatically tessellated based on proximity to the camera. Thus, we just needed to convert our terrain's polygonal mesh into a heightmap. Blender [16] provides utilities to perform this task and we output a 16,384 square pixel heightmap based on a smoothly subdivided version of the initial mesh. However, this heightmap had several issues: first Unity's terrain rendering system only supports heightmaps with a

Fig. 44.3 Virtual depiction of the Arturo mine

resolution of up to 4096 square pixels, this was remedied by breaking the terrain up into 16 tiles; second Unity requires a very particular "RAW" data format for its heightmaps, every PNG to RAW converter at our disposal resulted in very noticeable terracing artifacts on the terrain. Unfortunately, terracing on the mine's existing terracing was not desirable and we eventually were forced to accept that we could not automate this process without developing custom tooling and decided to manually smooth out this terracing using the terrain modification utilities provided by Unity. This manual smoothing process has resulted in a terrain that maintains the same forms as the original terrain, however, a bit more smooth as depicted in Fig. 44.3.

The low-resolution visual image issue was much easier to solve. The authors were already familiar with an SR-CNN [17] based deep-learning image super-resolution model called waifu2x [18]. This model utilizes a convolutional neural network to generate additional detail in a low-resolution image. We up-scaled our source image to 16,384 square pixels and then broke it up into 16 smaller images, one for each terrain tile.

44.3.3 Collision and Proximity Detection

We wanted to provide an emulation of a LIDAR-based proximity detection system that scans around the truck within Unity. If an object enters a certain distance around the truck, it will trigger an audiovisual alarm notifying the driver. This was trivial to implement. Unity provides several shapes of colliders that trigger an event when another entity enters the collider. Our implementation uses a capsule-shaped collider centered around the truck which extends five meters past the bounds of the truck (Fig. 44.4). When a "foreign object" (in our case a smaller truck or a worker) enters the capsule collider, an alarm system will be triggered within the truck to notify the driver of an incoming collision. By default, every object within the CVE is considered a foreign object. Users may mark certain objects (e.g. terrain) in the CVE as non-foreign if they want the alarm system to ignore them.

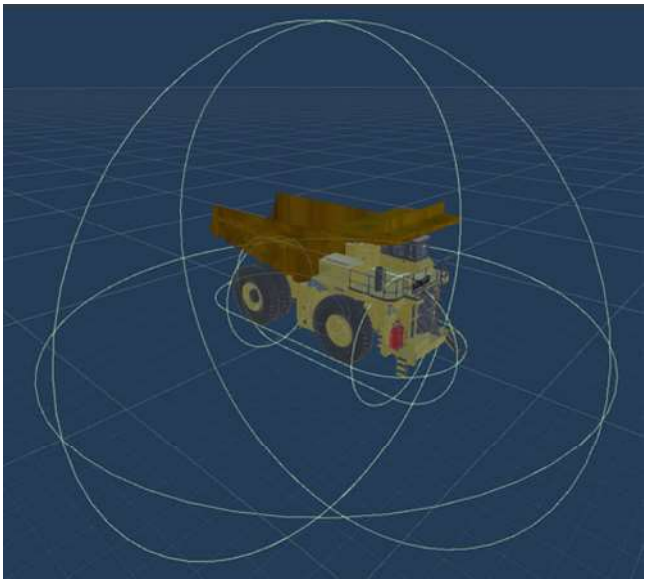

Fig. 44.4 The capsules we used to emulate LIDAR (outer) and direct collisions (inner)

This alarm consists of an alarm noise and a flashing red light within the cabin of the truck. If one of these foreign objects collides with the truck itself, the alarm system will stay triggered and the racing chair that the user is seated in will begin to vibrate to indicate the collision to the user. This vibration is achieved by applying small perturbations to the lateral velocity and vertical acceleration parameters of the racing chair's API. These perturbations are randomly selected from a range of $[-10, 10]$ and $[0, 10]$, respectively.

It is possible that a foreign object that poses no immediate safety concern may trigger the proximity detection system. For instance, a small rock could fall out of a truck with a full load of material and fall within the range of the proximity detection system. To account for this, the alarm system is designed to occasionally give false alarms, in which the alarm system is triggered without a foreign object being within the required range of the truck. A false alarm has a 1 in 35 (2.85%) chance of happening every second and will have a duration that is randomly selected from a uniform distribution across a range of 3–5 seconds.

44.4 Feature Comparison

To show the utility of our system, we will compare our feature set to that of the other existing systems shown in Sect. 44.2.

The hardware requirements of these systems contribute to their financial accessibility and immersion capabilities. More immersive hardware is generally more expensive. Our system, Anderson et al. [2], and Golovina et al. [3] implement VR capabilities with versions of the *HTC Vive*. McMahan

et al. [4] allow their system to be used with a CAVE and a standard desktop monitor, while Lucchesi et al. [1] only show their CVE through a standard desktop monitor. While standard desktop monitors provide less immersion, they are cheaper than a VR HMD or a CAVE.

Hardware used for locomotion should also be considered. Golovina et al. [3] opt for movement using the *HTC Vive* controller touchpads exclusively, while Anderson et al. [2] use touchpad movement, walk-in-place movement using *HTC Vive* trackers, and a *Virtuix Omni*. Our locomotion hardware may be the most prohibitively expensive, as we use a *Virtuix Omni* in addition to an immersive driving simulator chair, steering wheel, and pedals. On the other hand, ours has the most potential to be immersive.

Hardware requirements aside, software features are of more importance, since they determine what the system is useful for. An important distinction to make is the mining context. Our system and the systems of McMahan et al. [4] and Lucchesi et al. [1] are based on open-pit surface mines, whereas Anderson et al. [2] and Bellanca et al. [12] deal with underground mines. Golovina et al. [3] present a system that has nothing to do with mining, but still has other relevant features, such as a focus on hazard awareness. Our system also deals with hazard awareness, as does McMahan et al. Our system and Lucchesi et al. also deal with driving simulation, as well as taking into consideration moving and stationary obstacles to navigate around.

A key aspect of our system is its proximity detection facilities. We implement a two-stage proximity detection simulation (i.e. close to object and contacting object) using Unity's built-in colliders, as does Golovina et al. [3]. Bellanca et al. [12] also implements a two stage proximity detection system, but also provides three different levels of it: Unity colliders, ideal magnetic fields, and magnetic fields with outside interference.

Of these systems, only ours and the one by Anderson et al. [2] have multiuser capabilities. Not only that, but both systems allow for the distinction between two classes of users: Anderson et al. have mine workers and a mine supervisor, and we have a truck driver and an on-foot mine worker.

While our system implements a driving simulation, it lacks the robustness of the driving implementation of Lucchesi et al. [1] Their system considers the weight of the material loaded into the truck and weather conditions (i.e. wetness of the roads) to determine how the vehicle will handle when driving. They also consider a scenario with fog, reducing visibility. Our system does not implement these features. Other features that we do not implement are the robust data logging of Golovina et al. [3] and Bellanca et al. [12], or the ability to conveniently create different CVEs implemented by Bellanca et al. and Lucchesi et al.

44.5 Conclusions and Future Work

The system presented in this paper provides the user with a VR dump truck driving simulator for open-pit mines with a focus on implementing a simulated LIDAR-based proximity detection system to protect the safety of mining personnel. To add realism to this training simulation, another user may be present in the CVE walking around, serving as a moving obstacle for the driver of the truck. Special care is given so that the system is highly immersive. We make the acts of driving a truck and walking around a CVE more realistic through the use of immersive input devices such as a gait-negating treadmill and a realistic truck driving input system.

The system is a unique combination of several different features present in similar VR applications, such as hazard awareness training, driving training, immersive locomotion, proximity detection, collision detection, and multiuser functionality. While our system brings a lot of features to the table, it lacks some details brought forward by previous work, such as changing the truck's handling as weather and load conditions change, customization of the CVE, and data logging for post-training analysis. Nevertheless, our system provides open pit mines with a low-cost, multifaceted, and immersive training simulation for its workers.

To further validate the system that we developed, we plan to conduct a user study evaluating the usability of the system and whether or not it provides a benefit to the users when compared to typical truck driving training procedures in open pit mines. Comparative evaluations of other systems are another subject of interest for a user study.

Our system does not currently consider variations in the weight of the truck. Real open pit mines utilize the trucks to carry material from one location to another, causing a large variation in the weight of the truck, and therefore, its handling [1]. For example, Lucchesi et al. consider the weight of trucks under a heavy load of material and trucks that are empty. They also incorporate varying weather conditions and their effect on driving. For example, the rain will cause the roads to become wet and muddy, which affects the handling of the truck. Fog is an example of a weather condition that reduces visibility, further stressing the importance of proximity detection systems.

Finally, the technologies used to implement this system may continue to evolve. MuVR plans to implement more elaborate virtual presence facilities which can be leveraged by this system. New VR treadmills and HMDs have been released and will likely continue to be released and supporting these new technologies will eventually become a necessity.

References

1. B. Lucchesi, N. Oberlander, F.C. Harris, P. Mousset-Jones, Surface mine truck safety training: Scenario setup for a VR driving simulator, in *Proceedings of the 12th International Conference on Computer Applications in Industry and Engineering (CAINE '99* (1999), pp. 62–65
2. K. Andersen, S.J. Gaab, J. Sattarvand, F.C. Harris, METS VR: Mining evacuation training simulator in virtual reality for underground mines, in *17th International Conference on Information Technology- New Generations (ITNG 2020)*, ed. by S. Latifi (Springer International Publishing, 2020), pp. 325–332, ISBN: 978-3-030-43020-7. https://doi.org/10.1007/978-3-030-43020-743
3. O. Golovina, C. Kazanci, J. Teizer, M. König, Using serious games in virtual reality for automated close call and contact collision analysis in construction safety, in *Proceedings of the 36th International Symposium on Automation and Robotics in Construction (ISARC), M. Al-Hussein, Ed., Banff, Canada: International Association for Automation and Robotics in Construction (IAARC)*, May (2019), pp. 967–974, ISBN: 978-952-69524-0-6. https://doi.org/10.22260/ISARC2019/0129
4. R.P. McMahan, S. Schafrik, D.A. Bowman, M. Karmis, Virtual environments for surface mining powered haulage training, in *Extracting the Science: A Century of Mining Research*, ed. by J.F. Brune (2010), pp. 520–528
5. A. Asvadi, C. Premebida, P. Peixoto, U. Nunes, 3D lidar-based static and moving obstacle detection in driving environments: An approach based on voxels and multi-region ground planes. Robot. Autonomous Syst. **83**, 299–311 (2016). ISSN: 0921-8890. https://doi.org/10.1016/j.robot.2016.06.007
6. Unity Technologies, Unity real-time development platform [Online]. Available: https://unity.com/ (visited on 10/05/2022)
7. J. Dahl, E. Marsh, C. Lewis, F.C. Harris Jr, MuVR: A multiuser virtual reality framework for Unity, in *Proceedings of the 31st International Conference on Software Engineering and Data Engineering* (SEDE 2022) (2022), pp. 61–70. https://doi.org/10.29007/jdlg
8. Vr headsets, games, and metaverse life: United states [Online]. Available: https://www.vive.com/us/ (visited on 10/24/2022)
9. Virtuix, Omni by virtuix – the leading and most popular vr treadmill [Online]. Available: https://www.virtuix.com/developers/ (visited on 10/19/2022)
10. Logitech g923 trueforce sim racing wheel for xbox, playstation and pc [Online]. Available: https://www.logitechg.com/en-us/products/driving/g923-trueforcesim-racing-wheel.html (visited on 10/24/2022)
11. Full motion simulator 2,3,6 axis platforms for pc home flight and racing games [Online]. Available: https://dofreality.com/ (visited on 10/24/2022)
12. J.L. Bellanca, T.J. Orr, W.J. Helfrich, B. Macdonald, J. Navoyski, B. Demich, Developing a virtual reality environment for mining research. Mining Metall. Explor. **36**(4), 597–606 (2019). ISSN: 2524-3470. [Online]. Available: https://doi.org/10.1007/s42461-018-0046-2
13. Unity Technologies, Standard assets (for unity 2018.4) [Online]. Available: https://assetstore.unity.com/packages/essentials/asset-packs/standard-assets-for-unity-2018-4-32351 (visited on 10/05/2022)
14. SimRacingStudio, Simracingstudio/srsapi · gitlab [Online]. Available: https://gitlab.com/simracingstudio/srsapi (visited on 10/19/2022)
15. F. Remondino, L. Barazzetti, F. Nex, M. Scaioni, D. Sarazzi, UAV photogrammetry for mapping and 3D modeling-current status and future perspectives, in *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 38, no. 1 (2011), p. C22
16. Blender Foundation, *Home of the blender project - free and open 3D creation software* [Online]. Available: https://www.blender.org/ (visited on 10/17/2022)
17. C. Dong, C.C. Loy, K. He, X. Tang, *Image superresolution using deep convolutional networks* (2015). https://doi.org/10.48550/ARXIV.1501.00092 [Online]. Available: https://arxiv.org/abs/1501.00092
18. Nihui, *Nihui/waifu2x-ncnn-vulkan: Waifu2x converter ncnn version* [Online]. Available: https://github.com/nihui/waifu2x-ncnn-vulkan (visited on 10/17/2022)