

First Person Virtual Reality for Evaluation and Learning of Construction Site Safety

Thomas Hilfert¹, Jochen Teizer² and Markus König¹

¹ Computing in Engineering, Ruhr-Universität Bochum, Bochum, Germany

² RAPIDS Construction Safety and Technology Laboratory, Ettlingen, Germany

E-mail: thomas.hilfert@rub.de, jochen@teizer.com, koenig@inf.bi.rub.de

Abstract –

Early hazard recognition in the construction environment plays a pivotal role in accident prevention. Modern technological advances in the field of visualization and display hardware enable unrivaled situational experiences when using head-mounted displays. Low-cost hardware, high refresh rates and increasing resolutions of devices enable immersion into a virtual world with useful safety applications for the architecture, engineering, and construction (AEC) industry. Realistic particle effects and audio simulation support the perception of human test subjects experiencing a construction site from a first person view. Such test infrastructure and environment generate more accurate reactions to dangers emerging from hazardous site conditions. The scope of the presented work is developing the framework for gathering scientific data that, once analyzed, defines new approaches towards situational awareness and learning. The particular application we investigate is the understanding of the safety behavior of pedestrian workers performing tasks near heavy equipment. To create a realistic virtual reality environment, we are importing building information modeling (BIM) data into the Unreal Engine 4 for visualization with modern, low-cost HMD hardware. The developed first person virtual reality infrastructure allows us to test human behavior in several unsafe work scenarios that are common in practice. While the test subjects experience the hazards in a safe test environment, their learning is made analyzable through the use of computational logging operating on a multitude set of data records collected in the virtual environment. Results are presented that show the potential impact of the developed method on existing construction safety applications, including but not limited to rapid hazard evaluation and learning.

Keywords –

first person view, construction safety, equipment, hazard recognition, learning, situational awareness, virtual reality.

1 Introduction

Construction site safety evaluation is an important part when planning projects. Since many workers are working in confined spaces for a manifold of different tasks, risks can be associated to the work tasks. Especially when operating in the vicinity of heavy machinery, such as mobile cranes, excavators and loaders, distracted workers are prone to serious and / or fatal injuries. Construction site safety should therefore be integrated from an early on point in the planning phase to minimize costs and overhead in the whole building project [1]. Ideally, safety engineers and workers should be able to review construction site plans at this stage. However, conventional 2D site layout plans are lacking intuitive information about safety and health considerations [2]. Virtual-reality (VR) with head-mounted displays (HMDs) enables users to view the construction sites from a First Person View (FPV) to intuitively perceive the impressions in a lifelike virtual environment.

While Virtual Reality (VR) hardware and specific software was very expensive and complicated to setup and use in the past, recent changes in commercialized products allow for a broader application of VR in engineering and construction. The rise of a new generation of head-mounted displays (HMDs) enables even private users to own and use them at a reasonable cost. Targeting home users and gamers is beneficial, as this means the products are fit for consumer hands-on experiences and are not as complicated to setup as developer versions.

In this paper, we propose a VR testbed that enables the users to assess the reactions of workers in simulated critical situations on construction sites while interacting with heavy machinery. Using realistic rendering in the Unreal Engine 4 (UE4, [3]), we are able to create construction site scenarios with the support of importing existing Building Information Modeling (BIM) geometry. Audio effects can be simulated to create an ambient and localized noise source for distraction and simulation of realistic environments. Data collection is available to further analyze the users' perception of

heavy machinery and to test the availability of different warning systems for pedestrian workers and operators.

2 Related Research

Wang et al. [4] introduced the early concept and design of a virtual environment for excavator operator training using the Unity3D game engine. They mentioned the value of having a VR simulator for tasks that are “[...] expensive, logically difficult [or] dangerous [...]”. They showed, among other researchers [5], the complexity and programming effort that needs to be taken in account when designing a construction VR simulator environment is highlighted. The benefit in having a realistic physical simulation engine is mentioned, when dealing with excavation tasks. However, they fall short on describing an actual implementation using VR hardware, such as HMDs or stereoscopic displays and their impact on the presence of the system to the user.

Li et al. [6] presented a virtual safety assessment system for training workers on construction sites. It highlights the use of a VR environment for simulating activities that are bound to high risks for the workers. The users are trained in certain scenarios and may answer questions about mistakes in their workflow and how to correct them. Especially tasks associated with unsafe site conditions, unsafe work behaviour or unsafe construction methods are candidates for training in construction. The visualization is done using Unity3D, but falls short on an actual implementation using VR hardware. Also, this system has its main focus on retraining work behaviour, instead of analysing workers reactions towards certain site layouts, similar to [7], where Nintendo Wii controllers are used as 3D input devices. Users have to fulfil certain tasks, while working on a virtual building platform. Physical errors, such as lifting objects with too much weight, will be relayed via optical cues. They also mention the potential of using open-source game engines to extend the potential beyond only the presented use case, as it allows developers for extension of the engine code.

Cheng et al. [8] described a system for tracking trainees’ safety and productivity performance of ironworkers. They utilized Ultra Wideband (UWB) technology to track real workers in an instruction environment and map the results in a VR environment, enabling immediate analysis of the actions in the real world. On-site feedback, e.g. on the violation of defined hazard zones, has shown possible. Similar to [8], the authors in [9] show a system capable of real-time location tracking on a construction site. Laser scanning was used to create a topological realistic virtual scene and the Global Positioning System (GPS) and / or UWB were used to track pedestrian workers and machinery. The application can be used to enable stakeholders to

make “informed decisions in shorter times and at lower cost”. However, as the virtual environment is based on real actions in both cases ([8], [9]), simulation of potential dangerous activities is not possible.

A virtual crane simulation system is mentioned in Juang et al. [10], where stereoscopic displays are used. With the addition of kinaesthetic tracking, to alter the virtual view point of the user, it gives the impression of looking through a window. In the results of testing the performance on crane operation, only a marginally improvement was observed. This is due to users having to get used to the shutter glasses and / or refresh rates, according to the authors, and the complexity of the human vision system. While having a kinaesthetic system is an improvement over stereoscopic displays, modern HMDs are far more suitable for using such a system, as they translate head movements with high refresh rates to the virtual world. Also, displaying the world through a monitor is hindering, as static reference points around the screen exist and confuse the viewer, in contrast to a HMD.

While multiple concepts and applications of safety simulators, training tools and assessment exist ([11], [12]), few of them utilized modern, low-cost HMD technology. Therefore, we see benefit in creating a test bed for researching human behaviour with construction site safety and construction site layouts in mind, especially designed for HMD usage.

3 Concept

Our proposed concept for a testbed for construction site safety to analyse human behaviour consists of a hardware and software side.

As several major companies developing VR HMDs recently went out of development status with their products, consumer headsets are available from March / April 2016. The Oculus Rift Consumer Version 1 (CV1, [13]) and the HTC Vive ([14]) both promise to deliver visual fidelity and input solutions for our proposed approach. The Vive includes two hand controllers, which are tracked in a room-scale environment via passive laser diodes, while the Rift will sell their optical tracked Touch controllers ([15]) later in 2016. Both enable users to use grab gestures for picking up objects and interacting with their surroundings. Additionally, the Leap Motion controller ([16]) can be affixed to the front of the Oculus Rift to detect the hands via a stereoscopic camera setup. Leap’s new Orion software release further improves hand detection and reliability on occluded parts of the hands to a usable degree for normal users. Support for both the HTC Vive and the Oculus Rift are included in most of the major game engines, including Unity3D ([17]) and UE4.

On the software side, our system relies on UE4 to create an interactable world for the user. As UE4 is free

of charge (except commercial royalties, which apply upon exceeding a certain sale margin per year) and open source, we can modify the engine to our needs. Additionally, UE4 delivers an intuitive logic building block approach on programming, the visual programming language Blueprint. Therefore, it is easier to maintain for people not used to program in C++.

Figure 1 shows the proposed system in use. The user faces a traditional PC workplace, while wearing the Oculus Rift on his head. Scenario operators can watch a mirrored version of the VR rendering on the screen, so instruction is possible at any time. The Oculus positional tracker for absolute calibration of the headset position is mounted on top of the monitor. While the head tracking determines the view inside the VR world, a gamepad can be used to move around, interact with objects and override turning direction, if necessary.



Figure 1: VR test bed in use

3.1 World building

To create meaningful surroundings for construction site safety, the scene should resemble the later construction site in the real world. As the adoption of Building Information Modelling (BIM) in the AEC industry is rising internationally, we can leverage the existing models for our purpose. Existing map services, such as OpenStreetMaps can be used to further extract geometries of surrounding buildings and to gather the digital elevation model, relevant to the scene. Manual corrections are possible through the UE4 editor application, which is used to build environments in conventional games.

3.2 Data Logging

To analyse all interactions of a worker inside the VR scenarios, different log file information should be collected. As one of the main foci of this concept lies in the detection and perception of heavy machinery around pedestrian workers, data should be saved when a

participant is able to visually process information about machinery in his / her vicinity.

As most game engines only render objects when they are inside the visible part of the world (the viewport), we can use this data to detect when certain objects are visible to the user in his peripheral vision (simplified by a grey arc in Figure 2). As the participant will wear a VR HMD, his head movements are translated to the viewport in the virtual world. The natural reaction is to turn one's head towards the location of a stimulus to further analyse the situation. Therefore, we are able to log when the user is able to react to avoid hazards. Non-reaction logging is also possible, enriching the learning process once data is processed to valuable safety information. While this only filters data on visual clues, users are expected to almost always react on auditive clues in their surroundings with looking towards them.

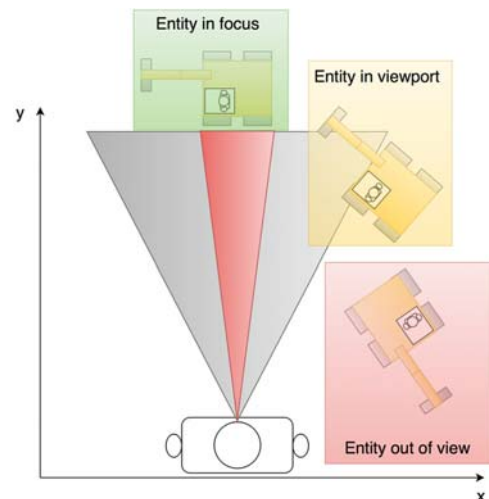


Figure 2: Schematic view of the user's perspective inside a VR environment

The second step of acquisition should be when the subject focuses on certain parts of the scenery that is associated with a-priori known hazards. As the VR HMDs contain Fresnel lenses, which will have a 'sweet spot' in the middle of the field of view (Figure 2, red arc), eye movements will be reduced in favour of head movements, such that the eyes maintain fixed in a forward looking position. We can use this downside of the HMDs to do an approximate 'eye tracking' with the middle of the camera frustum. If an object is in the middle of the view for multiple milliseconds, we can assume it has gotten the viewer's attention.

For further understanding the behaviour of the participant in the scene, it would be beneficial to have a replay functionality, which allows for the recording and

playback of all events inside the scene. This will allow later in-depth analysis of all circumstances that led to certain decisions.

All of these data should be processable in an interchangeable format, for example character separated value (CSV) files, except the replay functionality. Logging all the scene data outside of the engine is not intuitive to analyse. Most game engines offer this feature in an own format, native to the engine.

3.3 Rule set

For users immersing themselves into a prepared scene, certain game logic rules have to be applied in order to deliver a realistic test bed environment.

Foremost, all of the interactions between dynamic participants (heavy machinery, physics bodies, the users) that lead to fatal or major injuries should result in the scene coming to a halt and giving feedback about the error. A supervisor should be in range to support the setup and resetting of the test run. If the users are given a specific task to fulfil, additional rules would apply.

One possible task for the participants would be to steer a wheelbarrow down a certain path. The game rule set would impose certain restrictions that will lead to failure, such as diverging outside of certain bounds away from the path or tilting the barrow above a predefined degree constraint. These restrictions would be specific to the feature set of the scene and therefore have to be programmed for each application.

3.4 Non-player character (NPC) setup

All interactive elements in the scenes, that will move and are relevant to the fulfilment of the task given, need to be prepared for the scenario. While 3D models of equipment are widely available on the internet (paid and free), most of them lack animations or the prerequisites to be animated. Usually, these preparations include binding the triangles of the meshes to certain handles (bones), which are then later animated in a meaningful fashion. This part of the setup cannot be skipped, as it will yield in a much more realistic environment for the users. Apart from using animations, textures are another requirement to be considered when designing realistic NPCs. For prototyping scenarios this may be neglected to a certain degree, but for the final presentation and test runs, animations and textures should be reworked and polished.

The movement of the NPCs inside the scene can usually be steered by setting up navigation nodes, which define the way a certain actor moves. Therefore, setting up a realistic testing environment additionally consists of planning vehicle movement throughout the construction site. Collisions between NPCs in this scenario should be avoided, as they may happen with

superficial planning of navigation routes, to further increase immersion inside the scene.

As most of the models for NPC simulation should be prepared for physics interaction, it is easy to add player controls to them, enabling users to also use the test bed for vehicle driving. The same mechanisms used for processing the viewport to detect recognition of vehicles may be applied vice-versa to the detection of workers in a construction site environment. These versatile applications reduce the amount of preparation needed for multiple use cases, as the environment of the test bed would have to be prepared once and only additional scenarios have to be set up.

4 Implementation

As mentioned in the previous chapter, we are using UE4 for creating the software side of the VR test bed. To keep the system extensible to non-programmers, Blueprint visual scripting language is used for setting up the major interaction rules and scenarios. This also keeps the project compatible to additional C++ code, if needed, as projects support both languages at the same time.

4.1 General setup

UE4 comes with multiple project templates, enabling developers to swiftly create prototypes for game creation. These prototypes consist of control input schemes, a starting environment, and additional starter content, such as materials and elemental building blocks.

We chose the First Person template as a basis, as it includes most of the features needed for the creation of a VR test bed. Controls of the player character are done using the computer mouse and keyboard, or by using a gamepad, which is more intuitive to use while wearing a VR headset.

Setting up scenes in the UE4 editor is intuitive, as it has drag and drop functionality for adding models to the environment. Landscape sculpting tools are available to shape the ground of a construction site and painting different textures on the landscape is directly possible. As soon as basic placement of all necessary entities (scene starting point, lighting, landscape) is done, a preview version of the current scene can be tested inside the editor itself, or as a standalone running process.

4.2 Construction site layout

As the building to be constructed is the main focus on the site, attention should be set on modelling an authentic representation. Many buildings nowadays are already planned using BIM, therefore we are using the import plugin for UE4 described in [18]. This allows us to connect to an instance of the open source BIMserver

to import geometry for the building. After importing a BIM model, textures need to be set in order for it to be realistically rendered. UE4 delivers some materials already included in the starter content, for example, concrete. Individual parts of the building can be later selected and hidden to reflect the current state of the construction process. Additionally, direct import of the Autodesk FBX file format is also possible.

Landscapes should be modelled according to the site layout plan. This can be done using UE4 landscape sculpting tools (see Figure 3), which allow on-the-fly modification of the ground. A landscape in UE4 is always assigned to a material, which can consist of different textures that blend into each other. During the design process of the scenario different textures can be painted directly on the landscape meshes in the editor. Raw height map files are also importable into an UE4 scene, which gives the possibility of automatically grabbing height information from geographical databases and use them as a basis for landscape building.

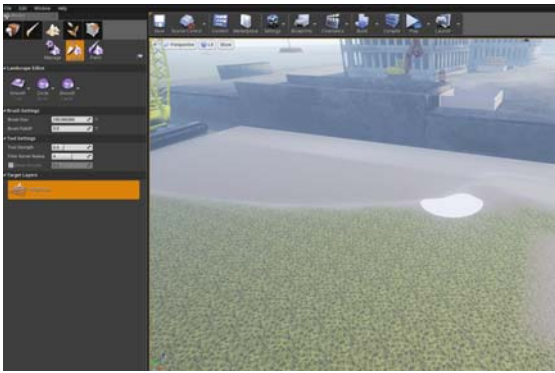


Figure 3: UE4 Landscape sculpting tools

Map services, such as OpenStreetMaps, deliver basic models of the buildings surrounding the construction site. Most of the times they are missing the height of the buildings in question, but the outline in relation to the streets and other buildings is correct. This gives some basic context to the construction site, in contrast to having only a flat landscape around. However, texturing these buildings should be considered regarding to the complexity of the scene setup, as this will be a time consuming process. For basic shapes it would be advisable to use distance fog to cover up the unrealistic textures and not break immersion, while keeping the setup time to a minimum.

4.3 Data logging components

For later assessment of the users' performance inside the VR test bed, data logging components, as mentioned in Concept, have to be implemented inside UE4. We are

using Blueprints for modelling the logic, as this is easily extensible.

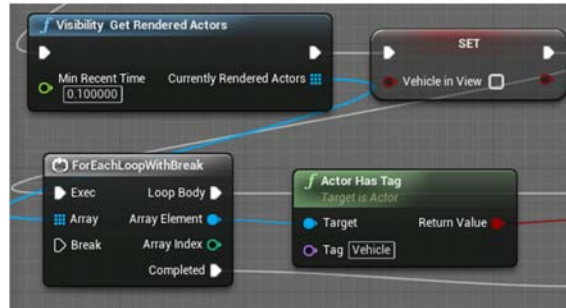


Figure 4: Checking if a vehicle is inside the viewport (Blueprint)

Figure 4 illustrates the Blueprint for detecting if a vehicle is inside the user's viewport. As this functionally is not included in the base UE4 classes, we are using the Blueprint extensions from [19]. We are then iterating over all actors, which are the base class for displayed entities, inside the viewport and check for an attached property tag. If it equals to "Vehicle", we can assume a vehicle actor has been rendered and can further process this input for analysis.

Figure 5 shows the process of detecting if a vehicle is inside the center of the viewport, which is more complicated. The first person camera is used as a reference and its location and rotation are extracted. Then we derive the forward vector from the rotation and extend it by a certain amount of distance, which varies from scene to scene. We can then use UE4's *LineTraceByChannel* function to check for a ray hit test result with any entity in the world. As the "visibility" channel is selected, only visible entities, not the ones occluded by other objects, will be returned. After this process, UE4 returns an array of actors to be checked for the aforementioned property tag "Vehicle".

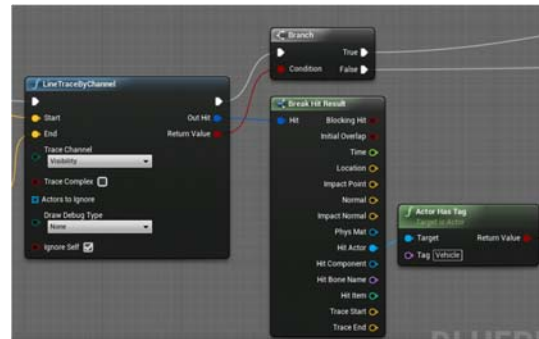


Figure 5: Line-tracing a vehicle in the center of the viewport

Both methods of detection deliver a reference to the actor in question, enabling to access all of its properties, e.g. the location, speed or other state variables. This information may be retained in the CSV log files, as it will enable complete analysis of the scene and events. As seen in Figure 6, we can use the FileIO Blueprint functions to write text to the file system. Normally the library allows only for overwriting the same file, so we did extend the functionality via C++ to being able to append text instead. As a reference value for the time elapsed, we can use the *GetRealTimeSeconds*, which references the time in milliseconds since the scene has been started. This is combinable with a custom console command, such that the time will be only measured from starting the scenario with the user already in place, instead of referencing only the time of the engine startup.

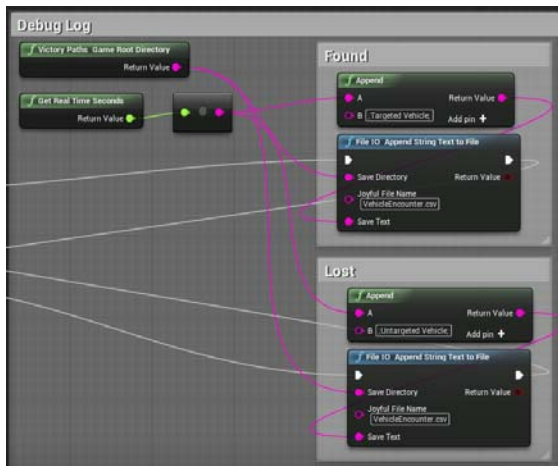


Figure 6: File logging to CSV

While logging the data about vehicles encountered in the viewport and center of view is essential for the benchmark of the performance inside the test bed, it misses some of the context information needed to fully assess the situational awareness of the environment. The test data will include the timing of the participant, but not all data about the view orientation, currently processed tasks at hand, and more. Therefore, it is important to be able to record the scene execution in as much detail as possible. Creating a logging component for all of this would be time-consuming, but UE4 already has a replay and demonstration system available. When designing network-based player experiences and games, the information about all entities in the scene needs to be propagated to all connected clients, such that all of the distributed engine instances are in the same state at all times. This system, called replication, is available inside of UE4 and can be enabled for

individual actors and entities, as, for example, static and unmovable entities do not need to be updated in their position. UE4's demonstration recorder will then use the information originally intended for network play, to store all needed information for later replay. When activating the replay feature, the demo can be paused and slowed down, as needed.

5 Case Study

To demonstrate the feasibility of the developed VR test bed system, we are using the layout, of a real world construction site. The users have to solve a simple task inside the test scenario to distract him/her enough from the surroundings of the construction site. For the selected application they have to carry spherical objects from one point to another. Objects can be picked up using the left mouse button and dropped in the same fashion. The two buckets (Figure 7, position A) are located at the intersection of a wheeled excavator driving path (Figure 7, position B). At a randomly triggered point in time, when the user is behind the building, the excavator will start to drive towards the user's path of operation. As most of the view is obstructed by the building, only audible clues are available to him/her at this time.

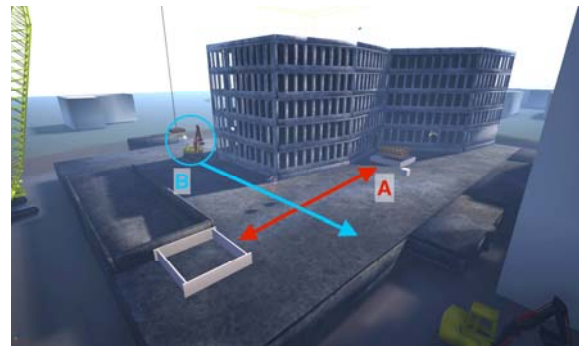


Figure 7: Case study scenario

Events to be logged in this case are: Start of the scenario, excavator in viewport, excavator in focus of viewport, excavator starting to drive, excavator striking user, excavator stopping. The scenario is stopped as soon as the excavator hits the user while on duty or the task is fulfilled. The scenario operator instructs the user a priori the test about the possibility of injuries and the construction site having safety hazards, but will not warn about the imminent danger of the driving excavator.

For the sound design we decided upon an ambient construction site recording which will be looped and is played omnidirectional. Additionally, excavators and cranes emit a motor sound at their respective positions.

Therefore, the user should be able to detect a vehicle with only audible clues. The excavator is set to drive with a top speed of 10 km/h and will start at a random point in time (between 10 and 180 seconds from start).

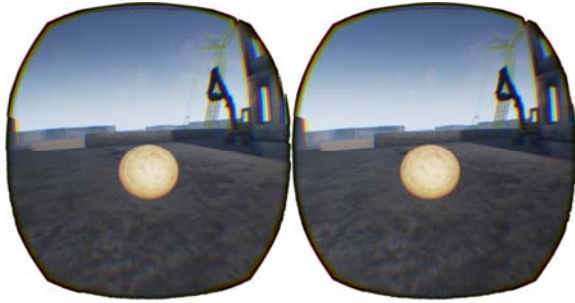


Figure 8: Stereoscopic view rendered to Oculus Rift

Figure 8 shows the view rendered to the user inside the VR headset. While s/he is carrying the spherical object for the task, s/he notices the approaching excavator around the building's corner. As the events on starting are random, it is not guaranteed, that the excavator will hit the participant, but identification on near misses are also important to consider when designing a safe construction site layout.

Research has shown that intelligent vehicle safety systems, for example the automatic deployment of an air bag, must work within 30 milliseconds of an event [20]. Table 1 shows the CSV results of a sample test run with a male doctoral student. Up to time index 10.413s the subject works on the task of carrying the spheres from basin to basin. The user is expected to see the excavator at its designated starting point, if the subject is looking towards its direction. As soon as the vehicle has started moving at around 11 seconds, the subject saw the excavator driving towards the subject and shifts his attention to it. This is seen by the "Vehicle in center" event that uses the line trace method to check for vehicle actors in the center of the view. He has, however, no time left to react and counter the struck-by event, which is seen at time index 13.297s. The subsequent "Vehicle out of center" was accompanied by an evasion reflex of the head, which was tilted away from the excavator.

The reaction time between seeing the vehicle in the viewport and focusing on it was 1612 ms, too late for a human to react in time. While this is above normal reaction time it has to be considered, that the subject was focused on the work task and it took time to realize the direction of movement of the vehicle. The narrow detection range in the center field is also difficult to interpret, as the movement towards the excavator may have started earlier. The time between "Vehicle in

center" and "Vehicle hit" is merely around 87 ms and also not enough to counter the event. However, the time span between noticing the vehicle and moving the head away was well inside normal reaction margins of a visual stimulus (330 ms, [21]). For further analysis on this topic, the recoding functionality of UE4 can be used to replay the event in detail. The results should vary when modifying auditive and visual clues, such as emitting a beeper tone or signal light when driving, as auditive stimuli are faster reacted upon [21]. However, research has shown that many construction workers ignore them [22], making them less effective [22].

Table 1: CSV results of case study test run

Time (Seconds)	Event
1.036	Vehicle in viewport
1.985	Vehicle out of sight
9.557	Vehicle in viewport
10.413	Vehicle out of sight
11.064	Starting vehicle
11.598	Vehicle in viewport
13.210	Vehicle in center
13.297	Vehicle hit
13.336	Vehicle out of center

6 Conclusions & Outlook

Evaluation and analysis of human behavior at construction sites for safety considerations is an important task. With today's digital building modeling and the commercial availability of VR headsets it is possible to create the impression of on-site presence for the user to train and test inside a VR test bed. With our proposed concept, it is feasible to create such a system at a low cost and reasonable development effort. The resulting data sets are fit for later analysis and recording functionality can be used to examine and replay problems in greater detail than before.

We are looking forward to extend our test scenarios to vehicle driving and crane steering. First person interaction with the machinery may be done using hand detection and the Leap Motion, to enable realistic interactions with the steering controls. We will further elaborate on the presented and proposed scenarios and test the system with a reasonable amount of users, while trying to incorporate more complex tasks, such as building a brick wall or using machinery while trying to keep the user's environmental awareness high. Oculus CV1's Touch and HTC Vive's controllers may be used as an alternative to optical detection of the user's hands. Also, natural day/night light cycles and weather simulation can be incorporated, to create even more immersive scenarios, which distract the users with realistic environmental conditions. Also, it should be tested if an interface towards real-time location tracking

using positioning systems can be implemented, to record and replay events in case of an accident.

While modern game engines like UE4 deliver a high visual fidelity, in terms of lighting, shadows, lens flares, etc., they may fall short in simulating complex physical effects, like windows fogging up. Also, the development of a scene can be complex, as animation and modelling take up a large amount of time. As soon as models are set up in a library, they can be reused for other test scenarios. However, this step should not be underestimated in project planning. Resource-wise the scenes can get also very costly to render fluidly, as the targeted frames-per-second (FPS) should be around 180 (90 FPS per eye), if wide open spaces are used without a level-of-detail mechanism in place.

Nevertheless, the current-gen HMDs for VR offer a feasible approach to run a test bed for a variety of applications, construction site safety being one of them. As more graphics card manufacturers will optimize their product to VR, entry costs will drop and the products will be affordable even for home and remote usage, priming and training users with experiences that overcome current limitations, such as motion sickness.

References

- [1] Hinze J. Safety plus: Making zero accidents a reality, CII Research Rep. Univ. of Texas at Austin, 2002.
- [2] Zhang S., Boukamp F., and Teizer J. Ontology-Based Semantic Modeling of Construction Safety Knowledge: Towards Automated Safety Planning for Job Hazard Analysis (JHA), *Automation in Construction*, 52:29-41, 2015.
- [3] Epic Games, Unreal Engine 4. On-line: www.unrealengine.com, Acces 03/14/2016.
- [4] Wang X., Dunston P.S., Proctor R., Hou L., and So J. Reflections on using a game engine to develop a virtual training system for construction excavator operators. In *Proc. 28th International Symposium on Automation and Robotics in Construction (ISARC)*, Seoul, Korea, 2011
- [5] Kamat V. and Martinez J. Dynamic 3D visualization of articulated construction equipment, *Journal of Computing in Civil Engineering*, 19(4):356-368, 2005.
- [6] Li H., Chan G. and Skitmore M. Visualizing safety assessment by integrating the use of game technology, *Automation in Construction*, 22:498-505, 2012.
- [7] Guo H., Li H., Chan G. and Skitmore M. Using game technologies to improve the safety of construction plant operations, *Accident Analysis & Prevention*, 48:204-213, 2012.
- [8] Teizer J., Cheng T. and Fang Y. Location tracking and data visualization technology to advance construction ironworkers' education and training in safety and productivity, *Automation in Construction*, 35:53-68, 2013.
- [9] Cheng T. and Teizer J. Real-time resource location data collection and visualization technology for construction safety and activity monitoring applications, *Automation in Construction*, 34:3-15, 2013.
- [10] Juang J.R., Hung W.H. and Kang S.C. SimCrane 3D+: A crane simulator with kinesthetic and stereoscopic vision, *Advanced Engineering Informatics*, 27(4):506-518, 2013.
- [11] Chen A., Golparvar-Fard, M., Kleiner B. SAVES: A safety training augmented virtuality environment for construction hazard recognition and severity identification, CONVR 2013, 373-384.
- [12] Sivanathan A. Abdel-Wahab M., Bosche F., Lim T. Towards a Cyber-Physical Gaming System for Training in the Construction and Engineering Industry, ASME 2014 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, 2014, doi:10.1115/DETC2014-34930.
- [13] Oculus, Rift. On-line: <https://www.oculus.com/en-us/rift>, Accessed: 03/14/2016.
- [14] HTC, Vive, On-line: <http://www.htcvive.com>, Accessed: 03/14/2016.
- [15] Oculus, Touch. On-line: <https://www.oculus.com/en-us/touch/>, Accessed: 03/14/2016.
- [16] Leap Motion, Leap Motion, On-line: <https://www.leapmotion.com>, Accessed: 03/14/2016.
- [17] Unity, Unity3D, On-line: <http://unity3d.com/>, Accessed: 03/14/2016.
- [18] Hilfert T. and König M., Low-cost virtual reality environment for engineering and construction, *Visualization in Engineering*, 4(2), 2016, <http://dx.doi.org/10.1186/s40327-015-0031-5>.
- [19] Rama. Victory Blueprint Library. On-line: <https://github.com/EverNewJoy/VictoryPlugin>, Accessed: 03/14/2016.
- [20] Air bags, <https://en.wikipedia.org/wiki/Airbag>, Accessed: 03/18/2016.
- [21] Shelton J. and Kumar G., "Comparison between Auditory and Visual Simple Reaction Times," *Neuroscience & Medicine*, 1(1):30-32, 2010.
- [22] Golovina, O., Teizer, J., Pradhananga, N. Heat map generation for predictive safety planning: Preventing struck-by and near miss interactions between workers-on-foot and construction equipment, *Automation in Construction* <http://dx.doi.org/10.1016/j.autcon.2016.03.008>.
- [23] Hinze, J.W. and Teizer, J. Visibility-Related Fatalities Related to Construction Equipment", *Journal of Safety Science*, 49(5):709-718, 2011.