Merging Live Video Feeds for Remote Monitoring of a Mining Machine

ANDREW FLANGAS, JAVAD SATTARVAND, SERGIU DASCALU, and FREDERICK C. HARRIS, JR., University of Nevada, Reno, USA

This research entails using virtual reality to interpret video recordings in Unity from cameras on an unmanned machine used for mining excavations. The purpose of using a machine of this nature is to send it into hazardous mining environments rather than sending workers and having their lives jeopardized. This work is significant because it demonstrates how two separate fields, such as virtual reality and robotics, can be combined to complete useful tasks. It also illustrates how machines can be used to replace workers in hazardous conditions not only in the field of mining, but in other fields as well. The main contribution of the work presented in this paper is the creation of a panorama of live video feeds captured by several webcams, which can be seen using a VR headset. As also described in the paper the software developed for this engineering application has been created using appropriate software engineering techniques and tools. Results of merging live video feeds and testing camera placements are also presented and planned directions of future work are outlined.

CCS Concepts: • **Applied computing** → **Physical sciences and engineering**; • **Software and its engineering** → **Software notations and tools**; • **Human-centered computing** → **User interface toolkits**.

Additional Key Words and Phrases: Virtual Reality, Unity, Mining, Software Requirements, Use Case Modeling, Software Design, HCI Tool, Live Video Feeds

ACM Reference Format:
Andrew Flangas, Javad Sattarvand, Sergiu Dascalu, and Frederick C. Harris, Jr.. 2021. Merging Live Video Feeds for Remote Monitoring of a Mining Machine. 1, 1 (September 2021), 12 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

For this research, we have experimented with two different methods that allow the user to use a VR headset to view live video feeds streamed by webcams connected to a computer via a USB cord. For the first method, video streams from four webcams were merged together into a panoramic frame, which can then be viewed using a VR headset in the Unity gaming engine along with the skybox feature. The second method consisted of importing the live video feed from two webcams directly into Unity and attaching the feed onto polygons. Each of the polygons that have video streams attached to them were aligned next to each other to form a panorama. The purpose of analyzing these two methods is to discern which method is superior so that it could be used for operating mining machines.

We hope that one day we could use VR and other joystick technology [17] to control mining machines and increase visibility so much so that there will be no need to send human workers into hazardous mining conditions. The type of machine we wished to design the software for was initially going to be an excavator, which consists of a crane with a bucket attached. A 3D model of the excavator can be seen in Fig. 1(A) and the drill rig in Fig. 1(B). The excavator is

Authors' address: Andrew Flangas, andrewflangas@nevada.unr.edu; Javad Sattarvand, jsattarvand@unr.edu; Sergiu Dascalu, dascalus@cse.unr.edu; Frederick C. Harris, Jr., Fred.Harris@cse.unr.edu, University of Nevada, Reno, Reno, NV, USA.

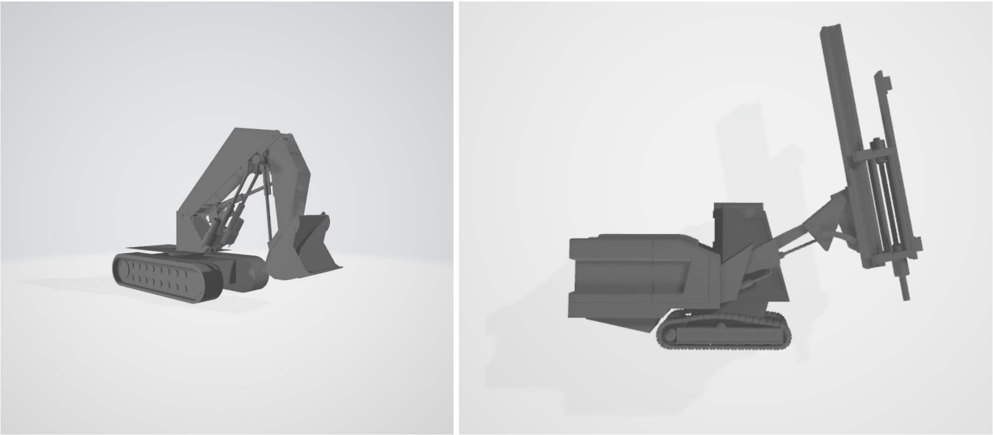


Fig. 1. (A) Model of the excavator mining machine (provided by Mining Engineering at The University of Nevada, Reno). (B) Model for the mining drill rig, (retrieved from [8]).

designed to load, haul, and dump debris created from blasting new holes into the earth. Due to time constraints on when the excavator will be completed, the currently targeted machine for the software has been the mining drill rig.

For visualizing how the cameras should be placed on the excavator machine, we used a 3D model provided by the Mining Department at the University of Nevada, Reno and imported it into Unity [9]. We have been contacted by the professor in the Mining Department in charge of the Immersive Teleoperation of Mining Machines (ITMM) project, and asked to create the software for another type of machine instead. The new machine for which we have developed the software is known as a rotary mining drill rig. Switching machines was necessary because of the it will take for the Mining Department to finish constructing the excavator. The rotary mining drill, however, is a machine that already exists, which allowed the immediate design and implementation of the related software.

The rest of this paper is structured as follows: The concept behind this engineering application is described in Section 2. Software requirements are presented in Section 3. Use case modeling is outlined in Section 4. Softwrae design and implementation are detailed in Section 5. Merging live video feeds is illustrated and analyzed in Section 6. Related Work is discussed in Section 7, and conclusions and future Work are presented in Section 8.

2 CONCEPT

2.1 The Problem

The problem addressed through the implementation of this application is the need to send human workers into hazardous mining conditions. Shortly after blasting a hole in order to create a new mining tunnel, the conditions generated for miners to move into those newly blasted holes are often unfavorable or even dangerous. The soil that makes up the floor and ceilings of that newly blasted tunnel can be unstable, and therefore the tunnel is prone to cave-ins of both the ceiling and the floor. In order to remedy this situation, a solution needed to be implemented to replace the workers meant to move into these tunnels with something that can be subjected to unsafe environments in their place – a solution that can also provide good levels of visibility for a remote observer. Further details are presented throughout the paper of how the software we developed and the technology involved can be used effectively to address the aforementioned problem.

2.2 Main Goal and Approach

The main goal of our contribution to the ITMM project has been to merge live video feeds from cameras placed on a mining machine. Once the feeds could be assembled in a panoramic frame, the next step was to develop software that allows the use of a virtual reality headset to view the panoramic video of live footage taken from the excavation machine. The tools used to develop this software were the Unity gaming engine [9] and Blender [1]. Creating the software in Unity also required the use of the C # programming language. The library and language we used for the first method to create the panorama was OpenCV for Python, which was also useful for stitching the video recordings together.

2.3 Intended Users

The intended users of the mining machine are typical mining workers at excavation sites. An important human-computer interaction goal was to make the software easy enough to use so that the technology can be utilized by anyone with minimal training.

The users will benefit from this application by not having to go into hazardous mining conditions, such as a recently blasted site and instead send in the machine to complete tasks. It is also helpful to create a virtual training environment for new operators to practice what it would be like to use the machine in real-life. The goal of the ITMM project is to eventually allow the users to operate the machine using a virtual reality headset along with a joy-stick controller. Not only does this technology have the potential to save lives in the field of mining, but it can also be used to accomplish dangerous tasks in other environments and situations, such as deep sea exploration or repairing levies during hurricanes. This technology can also be useful for space exploration, in which the user can experience a more realistic view of stars and the surface of asteroids by seeing them in virtual reality instead of on a video recording terminal. It could also be an exhilarating experience to be able to operate a machine that is located on the surface of, for example, Mars with the sensation that you, the user, are actually there performing various robotics tasks.

3 REQUIREMENTS

Stakeholder interviews were conducted early on with two researchers in the Mining Engineering Department at the University of Nevada, Reno in order to get a better idea of how the software should function.

These interviews were used to create the software requirements for our application. The levels of requirements indicate their significance and priority and consequently the strictness of planning and completing their implementation. For example, all level 1 (highest priority) requirements were implemented and tested, while level 3 (lowest priority) requirements were left for the future extensions of our software.

Level 1 functional requirements: Video camera objects should be added to virtual model of machine. Video recordings should be clipped together into one panoramic frame. User should be able to see the frame through a VR headset in Unity.

Potential challenges and limitations for level 1: Position of the video cameras on the machine may affect how video recordings are processed etc. one camera will be placed at a higher level than the others. Real-life machine may not function similarly to the machine model in Unity. Quality of the recordings may be affected by telecommunication. Unity may not be able to simulate real-life environment accurately.

Level 2 functional requirements: Locomotion should be attached to a joystick similar to the one mentioned in the interviews. Video recordings should be rendered more clearly based on which direction the user is facing by having the cameras on the machine rotate in that direction.

Potential challenges and limitations for level 2: The behavior of the rotating cameras in real-life might not be similar to the functionality of those available in OpenCV. Locomotion on the actual machine may experience a lag effect due to telecommunication, no way to know for sure until machine has been fully developed.

Level 3 functional requirements: The rotating cameras should change position through the use of eye-tracking.

Potential challenges and limitations for level 3: Eye tracking may be too new of a technology to implement for the use of this kind of system i.e. VR telecommunications AND operation of a machine.

4 USE CASE MODELING

The use case diagram of the software is shown in Fig 2 and the detailed use cases are as follows:

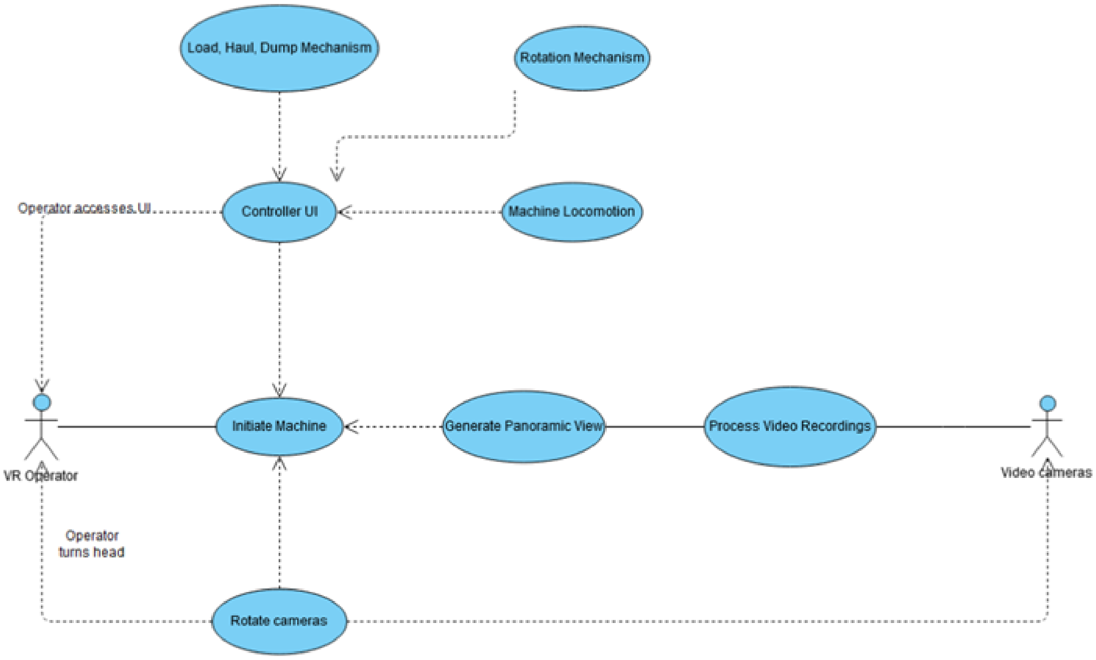


Fig. 2. Use case diagram of the proposed software.

Initiate Machine: This process is activated when the user initiates the software of the machine by powering on the VR headset. All system components are then activated such as the cameras and then the machine awaits further instruction.

Process Video Recordings: This process is activated once the machine has been initialized and the camera actor begins recording information. The recordings are then processed which then activates the panoramic view process.

Generate Panoramic View: This process is activated once the video cameras begin recording and the recordings are then processed. Once the recordings have been processed, the panoramic view process generates the recordings into one 360 degree frame. This chain of process is dependent on whether or not the machine has been initiated.

Rotate Cameras: This process is only activated if the operator wearing the VR headset turns his/her head, the machine has been initiated, and the video cameras are operational. If all of these parameters are met, then the cameras will rotate their position.

Controller UI: This process is activated if the operator accesses the UI, and the machine has been initiated. If both of these parameters are met, then the machine awaits further instruction from the UI. Machine Locomotion: Once the controller UI process has been activated, the machine locomotion is then an operation accessible through the UI. This process governs the overall horizontal movement of the machine based on the inputs to the UI.

Rotation Mechanism: This process governs the rotation of the machine about the vertical axis. It is useful for rotating the machine when it is stationary. This process can only be accessed if the controller UI process has been activated.

Load, Haul, Dump Mechanism: This process governs the movement of the bucket and crane attached to the machine. This mechanism is used to load the bucket with debris or dirt, haul it to another location through the use of the locomotion process, and to dump the debris or dirt. It is only accessed if the controller UI process is activated.

5 DESIGN AND IMPLEMENTATION

5.1 Block Diagram

To provide a high-level view of the proposed system's design, a block diagram was constructed and is shown in Fig. 3.

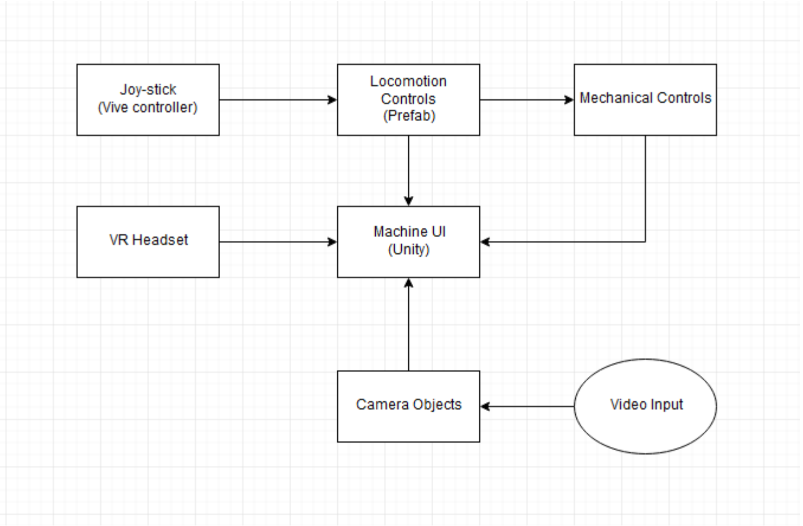


Fig. 3. Block diagram showing how the system is supposed to function. The operator uses a VR headset and HTC-Vive controller to operate the machine. The VR headset connects directly to the UI, whereas the joystick connects to the locomotion prefab, which can alternatively control the mechanical aspects of the machine. The camera objects read in video inputs and send them to the machine UI.

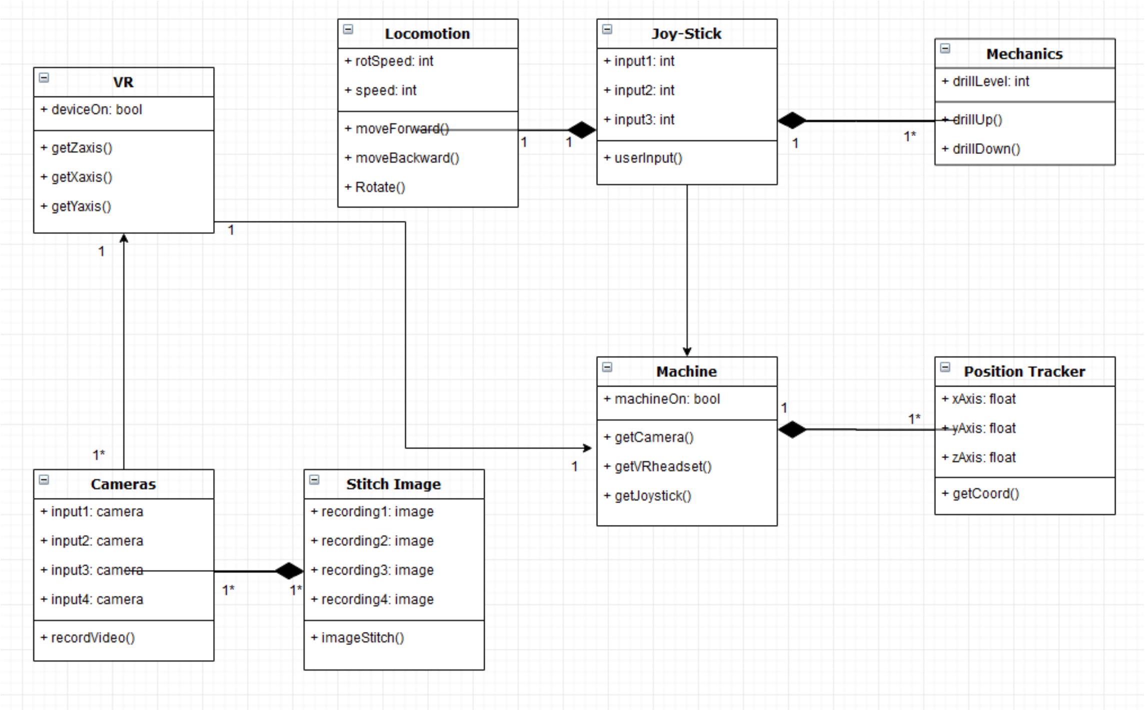


Fig. 4. Class diagram of the developed software. This diagram was created using [6].

5.2 Design Class Diagram

The class diagram of the software is shown in Fig. 4 and a description of classes is as follows:

VR: This class records the direction in which the operator is looking, and then applies the images gathered by the cameras to correspond to that direction. It acquires the direction through the get methods present in the class.

Cameras: This class uses four camera objects to take video recordings via the recordVideo method.

Stitch Image: This class takes the four recordings from the camera in the form of images, then stitches those images together into a panoramic frame with the imageStitch method.

Joy-Stick: This class records three input values in the form of integers, whichever input is chosen depends on the userInput method.

Locomotion: This class is determined by the type of input selected from the Joy-Stick class. The input from the Joy-Stick determines whether the moveForward, moveBackward, or Rotate method is called. The speed and rotSpeed are recorded as integer values in case one wishes to increase the speed of the locomotion.

Mechanics: This class governs the up and down motion of the drill animation attached to the machine. An integer value records the current height or level of the drill, whereas the drillUp and drillDown methods control the actual movement of the drill.

Position Tracker: This class determines the current location of the machine, this feature will come in useful when the actual model of the machine is out in the field. Three float values record the x, y, and z coordinates of the machine. The getCoord method returns the exact coordinates of the machine by combining the three float values.

Machine: This class is the central interface for all of the other classes. If the bool value of the machine is true, then the getCamera, getVRheadset, and getJoystick methods are then activated. This class governs the control of the machine model in Unity.

5.3 Data Design

The processes that make up the software run in a manner in which processes are being activated one after the other. Once the processes involved with locomotion are reached, they should be the last and most consistent processes that are continually executed. In order to best implement this behavior, a stack data structure is the most suitable choice for this software. All other activities are being pushed onto the stack, and once the locomotion activities are reached, they will continually be popped on and off.

6 MERGING LIVE VIDEO FEEDS AND TESTING CAMERA PLACEMENT

6.1 Using OpenCV for Python to Merge Webcam Footage

For merging the live video feeds of four webcams into a panoramic frame, two methods were used and analyzed to choose which one would be best in terms of visual quality. The first method consisted of using openCV for Python and accessing the four webcams by USB port in a Python script. The method for merging the video feeds together using Python relies on information and code provided by [15], which utilizes a data structure to detect the homography matrix between two images. Once the hogography matrix between two images is established, the two image frames along with the homography are then passed to the openCV warp perspective function to create a combined image from the two separate ones. The code is structured in such a way that the homography is not computed between video frames, but on single images that are run through a while loop so that the frames are constantly updated.

Once a panorama comprising video feeds from four separate webcams was established, we then needed to find a way to get the resulting panoramic frame into Unity. We came up with the solution to load the resulting panoramic frame to a web page, and then from that web page into Unity. A method for loading a single webcam frame to a web page using a Python script along with the web framework Flask is provided by [15], so combining the scripts for merging the webcam frames and the flask web server script gave us the result we were hoping for. The most crucial part of the Python script that merges the video frames and prepares them to be output to a web-page [Code details are available upon request] After specifying the proper IP we wanted to send the panoramic frame to so that it could be accessed by other machines on the same network. This gave us the result shown in Fig. 5, which is a panorama of live video feeds from four separate webcams. This is not as clear of a picture as it could be because different webcam models were used to capture the live feeds.

Because the resulting panoramic video frame is technically a JPEG image due to the while loop described earlier, the Unity video player feature is unable to load it using the URL method. After doing research on what the most efficient way to load the panorama into Unity would be, we discovered that sectioning off and live streaming the webpage to Vimeo, and then using the Vimeo Unity SDK would allow us to load the page into Unity and then use the video stream as the background for a skybox. Unfortunately, this method requires a Vimeo pro subscription which happens to be very expensive during the time of writing this paper. Another possible option would be to use the Embedded Browser Unity asset, but again that would require spending a good amount of money. In order to analyze what the result would look like if we could successfully load a live panoramic feed into Unity, we decided to take a video recording of the webpage the same way Vimeo would do, and we used the recorded video clip as the background for the skybox. The result of

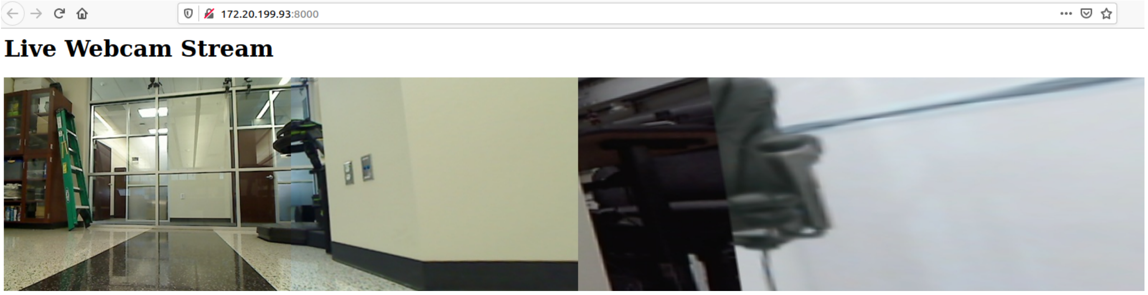
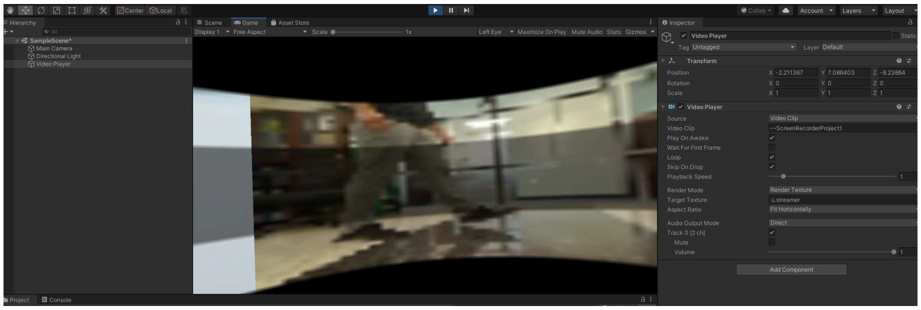


Fig. 5. Image showing the result of merging four webcams together and loading the resulting frame to a web server.

this is displayed in Fig. 6, showing a live video feed of someone walking through a room (however, the resolution is of low quality).



Video Player ☐ Static
Tag Untagged Layer Default

Transform

Position	X -2.211367	Y 7.086403	Z -8.23864
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

☑ Video Player

Source	Video Clip
Video Clip	--ScreenRecorderProject1
Play On Awake	☑
Wait For First Frame	☐
Loop	☑
Skip On Drop	☑
Playback Speed	1
Render Mode	Render Texture
Target Texture	streamer
Aspect Ratio	Fit Horizontally
Audio Output Mode	Direct
Track 0 [2 ch]	☑
Mute	☐
Volume	1

Fig. 6. The result of recording a video clip of the panoramic web page and using it as the skybox background in Unity. However the resolution is of low quality.

6.2 Loading Webcam Footage Directly to Unity

The second method for merging four webcam frames is less complex, but surprisingly it yields a far better result in terms of resolution. This method entails using a C# script to identify the webcams that are connected to the computer, and then loads the live feed from the webcams directly into Unity. The script is attached to whatever object the live feed is displayed on, and it uses the render texture feature to accomplish this. The script automates the whole process of creating and attaching a render texture to a polygon, after which it is up to the programmer to place the polygons next to each other in a careful manner to get a panoramic image from the four cameras. The result of this method is shown in Fig. 7, and as you may see, the resolution is of significantly better quality than the previous method.

6.3 Testing Placement of Cameras on the Machine Using Unity

In order to establish the best positions for placing the cameras on the machine to obtain a clear 360 degree recording, a model of the mining drill rig machine was loaded into Unity and camera objects were placed in strategic locations on the machine. To get a clear perspective of what the cameras were capturing, render textures corresponding to each of the camera objects' footage were attached to 3D planes in such a way to show an outline of an unfolded skybox. This is presented in Fig. 8.

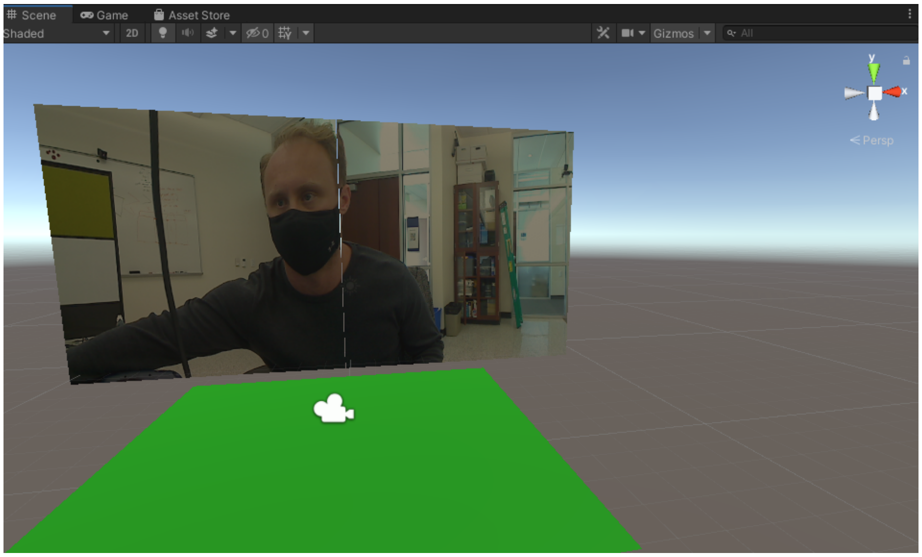


Fig. 7. The result of loading the webcam feeds directly into unity and attaching them to polygons in order to create a panorama.

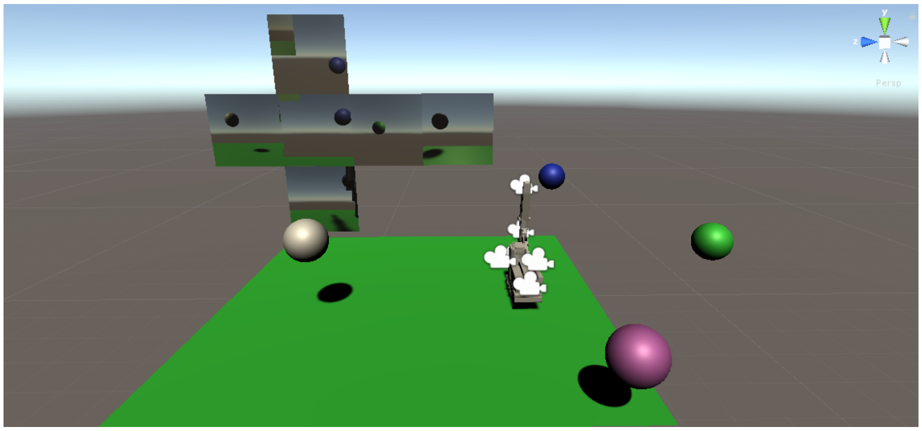


Fig. 8. Scene view in Unity prototype. It shows the the mining machine model with the camera objects attached to it, along with the render textures set up in order to create a panoramic frame.

7 RELATED WORK

7.1 VR and Robotics

In our search to find projects related to using VR to operate machines, we came across a conference paper [13] that describes a method that uses VR to train operators to control waste management robots. To train these operators, the participants are placed in a VR simulation that mimics how controlling the robots in real-life would be. To do this, the operators use voice commands and hand gestures in order to create a realistic and natural experience. The robots that the operators are being trained to use are for cleaning up trash in toxic environments.

Another very useful paper is one that describes three areas where VR and robotics can be successfully integrated to solve certain tasks [2]. According to the authors, robotics and newer VR research communities have been working in isolation. Both of these fields can benefit greatly when collaborating with each other. Areas where an integration of the two fields would complement one another is VR-enhanced CAD design, simulation of a plant layout, and robot programming. Other areas that would also see benefits include VR for supervisory teleoperation, collaborative control of a multiple operator-single robot systems, and single operator-single robot systems. [2] also discusses how robotics can benefit VR in general pertaining to the ability of robots to serve as force feedback interfaces to a VR simulation.

7.2 Relevant Projects and Applications

A relevant project that pertains to this field of study is the work being done on the Spherical Fish Tank Virtual Reality Display (SFTVR) [19]. What makes this project relevant is its focus on displaying 3D spatial effects by using head-tracking to render an image to the perspective of the viewer. It combines Fish Tank Virtual Reality with a spherical display, which can have promising effects on increasing the user's performance for 3D tasks. The advances made as a result of optimizing such a project are clearly useful for future work in our own project. One application of SFTVR could be designing a UI [12] to control the mining machine within the virtual environment.

Another useful application is the use of visual manipulation and kinematic data collection in VR [17]. This journal article aimed to validate the use of the HTC VIVE VR system for collecting kinematic data by assessing the accuracy of its orientation signals and position. The evaluation was conducted by taking the VIVE controller, as well as the tracker, and comparing them to a Polhemus Liberty magnetic tracking system sensor. This evaluation concluded that the HTC VIVE system can be used as a reliable means of collecting kinematic data. This information could be useful for our project, in particular for creating a UI in the virtual environment. The HTC VIVE [5] controllers can also be used for tracking the user's hands on the 3D UI, or even be utilized to control the machine without the need of another UI.

7.3 Other Related Areas

There are a host of other resources pertaining to this field of research in the form of other major VR research labs, as well as patents for new devices that can aid in future work for my project. One of these major labs includes the NYC Media Lab [11], which is currently working on 14 new exciting projects. These projects not only enhance the way people experience entertainment, some of them also have the potential to help workers of multiple different fields. The project Acculis for example, is being developed for the construction industry to facilitate the transfer of information between designers and construction workers.

Another VR lab that appears to be conducting extensive research in the subject is the CITL'S Virtual Reality Lab [7] located in Illinois. The purpose of this lab is to serve as a research and testing grounds for students and faculty who share an interest in exploring the educational potential VR has to offer. They are multiple different headsets available at this lab including the HTC VIVE Pro, the Oculus Rift, as well as the Oculus GO. The innovation studio located in this lab is designed to observe the learning strategies and behaviors of individuals using the devices in order to use them as mediums to further explore educational content.

One of the patents we located [3] describes a 360-degree virtual reality camera system. This device captures a 360 degree image of a local area, in which a portion of it is in stereo. The camera system includes a first rigid plate, a plurality of axis cameras, and a second rigid plate. The peripheral cameras are aligned in a ring configuration which can be useful for allowing objects past the image threshold to be within fields of view of at least two of the cameras. Another patent [14] discusses the use of data structures and delivery methods that can expedite VR playback. This can be used for both virtual and augmented reality. The concept is that a video stream or augmented reality experience can be stored and then delivered to the user. The patent discusses how the stream is divided into plurality of units based on viewpoint segmentation, time segmentation, as well as orientation segmentation.

Patent [18] discusses a new method for remote virtual reality control of move-able vehicles. This method includes displaying a graphic model of a portion of the vehicle on a remote output device. The vehicle also has one or more move-able partitions. This method involves image processing from one or more devices, and can identify one or more vehicle commands. Yet another patent [10] relevant to our research topic outlines the idea of a three-dimensional scene reconstruction method that uses a set of two dimensional images. The purpose of this is to enable better preservation and sharing of immersive experiences. The set of images that are used are taken from different vantage points, and can process each image to extract depth information and then stitch the images in a multilayered panorama.

Article [4] portrays the capabilities of current, low-cost VR technology. It discusses recent advancements in devices such as the HTC VIVE, the Leap Motion Controller, the Oculus Rift, and the Microsoft Kinect. Each of these devices brings an immersive VR experience into the homes of consumers at a much lower cost and special requirements than in recent years. These devices are now lowering the barrier for the entry of VR engineering applications. Another relevant paper [16] outlines three different input technologies which are vision-based motion controls, gamepads, and the room scale feature, for an interactive VR environment. The system is able to visualize data from several sources by enabling the user to browse and interact with displayed information in an immersive VR environment. This paper includes a user study that assesses the experienced workload as well as the perceived flow of interaction.

8 CONCLUSION AND FUTURE WORK

In this paper we presented an engineering application involving the use of VR for displaying multiple video streams from a mining excavator. Two methods for merging live video feeds were described and compared, together with a detailed description of related software requirements specification, design, and implementation. The work presented lays the foundation for an actual remote control system that will eliminate the need of sending miners inside mines after blasting to loosen rock is performed in these underground mines. Instead, remotely controlled vehicles could be operated in the mine at all times thus significantly increasing safety.

There are other stages for the development of the research we intend to participate in, such as the integration of a joystick controller to operate the machine. Another goal could be to create and implement a 3D UI within the virtual environment that mimics the actual operations of the joystick controller. This 3D UI would allow the users to see which buttons they are pressing on the joystick controller in the virtual environment so they do not have to remove the headset for any reason.

Some other future work considered would be to integrate light detection and ranging (LIDAR) sensors into the mining machine to either complement or replace the cameras. These sensors would use light in the form of a pulsed laser to map the terrain of the environment the machine is present in, and then procedurally generate a graphical representation of that terrain inside of Unity. A 3D model of the vehicle could then be added to the scene with the generated terrain to simulate the position of the real machine with respect to its location in the actual terrain. VR could then be enabled in Unity to give the user a view of the procedurally generated terrain from the perspective of the machine model.

When implementing the use of a joystick controller to direct the locomotion of the machine, we could also allow the controller to manage the load, haul, and dump mechanisms. It would be a convenient feature to link the mechanisms of the joystick controller to the graphical model in Unity, so that whatever operations the machine performs as a result of the joystick controller in the real world will also be simulated in the Unity rendition. To achieve this effect, the joystick device that controls the machine should also control the model in Unity.

Furthermore, there are two options for integrating remote cameras on the mining machine to use for the video merging methods. One option is to have wireless cameras that are connected to a network, used together with a tool such as Wireshark to identify the cameras' IPs. The other option is to continue using USB cameras but utilize the wireless blue-tooth variation for connecting them to the computer. This way, getting the best video resolution is likely to increase the strength of the blue-tooth signal and not the processing power of the computer.

ACKNOWLEDGMENTS

This work was partially supported by the National Science Foundation under grant number IIA-1301726. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] Blender Online Community. 2019. *Blender - a 3D modelling and rendering package.* Blender Foundation, Blender Institute, Amsterdam. http://www.blender.org

[2] G. C. Burdea. 1999. Invited review: the synergy between virtual reality and robotics. *IEEE Transactions on Robotics and Automation* 15, 3 (June 1999), 400–410. https://doi.org/10.1109/70.768174

[3] Brian Keith Cabral, Forrest Samuel Briggs, Joyce Hsu, Albert Parra Pozo, and Andrew Hamilton Coward. 2016. *US10230904B2 - Three-dimensional, 360-degree virtual reality camera system.* FaceBook Inc. Retrieved June 9, 2021 from https://patents.google.com/patent/US10230904B2/en

[4] Joshua Q. Coburn, Ian Freeman, and John L. Salmon. 2017. A Review of the Capabilities of Current Low-Cost Virtual Reality Technology and Its Potential to Enhance the Design Process. *Journal of Computing and Information Science in Engineering* 17, 3 (September 2017), 031013. https://doi.org/10.1115/1.4036921

[5] HTC Corporation. 2021. HTC VIVE. https://www.vive.com/us/

[6] diagrams.net. 2021. *Flowchart Maker & Online Diagram Software.* Retrieved June 9, 2021 from https://www.draw.io/

[7] Center for Innovation in Teaching & Learning. 2021. Virtual Reality Lab. https://citl.illinois.edu/citl-101/instructional-spaces-technologies/armory-innovation-spaces/virtual-reality-lab (Last Accessed: June 9, 2021).

[8] Free3D. 2021. *Mining Rotary Drill V3 3d model - Free3D.* Retrieved June 9, 2021 from https://free3d.com/3d-model/mining-rotary-drill-v3--675841.html

[9] D. Helgason, J. Ante, and N. Francis. 2019. *Unity - Video Game Engine.* Unity Technologies, Unity Technologies, San Francisco. http://www.unity3d.com

[10] Johannes Peter Kopf, Lars Peter Johannes Hedman, and Richard Szeliski. 2017. *US10038894B1 - Three-dimensional scene reconstruction from set of two dimensional images for consumption in virtual reality.* Retrieved June 9, 2021 from https://patents.google.com/patent/US10038894B1/en

[11] NYC Media Lab. 2017. 14 Virtual and Augmented Reality projects emerging from NYC Media Lab this spring. https://medium.com/@nycmedialab/14-virtual-andaugmented-reality-projects-emerging-from-nyc-media-lab-this-spring-af65ccb6bdd8

[12] Joseph J. LaViola, Ernst Kruijff, Ryan P. McMahan, Doug A. Bowman, and Ivan Pouperev. 2017. *3D user interfaces theory and practice.* Addison-Wesley, Boston, MA.

[13] N. E. Miner and S. A. Stansfield. 1994. An interactive virtual reality simulation system for robot control and operator training. In *Proceedings of the 1994 IEEE International Conference on Robotics and Automation.* IEEE, 1428–1435 vol.2. https://doi.org/10.1109/ROBOT.1994.351289

[14] Derek Pang, Colvin Pitts, and Kurt Akeley. 2017. *US20180020204A1 - Data structures and delivery methods for expediting virtual reality playback.* Retrieved June 9, 2021 from https://patents.google.com/patent/US20180020204A1/en

[15] PyImageSearch. 2021. *PyImageSearch - You can master Computer Vision, Deep Learning, and OpenCV.* Retrieved June 9, 2021 from https://www.pyimagesearch.com/

[16] Nico Reski and Aris Alissandrakis. 2020. Open data exploration in virtual reality: a comparative study of input technology. *Virtual Reality* 24 (2020), 1–22. https://doi.org/10.1007/s10055-019-00378-w

[17] Kate A. Spitzley and Andrew R. Karduna. 2019. Feasibility of using a fully immersive virtual reality system for kinematic data collection. *Journal of Biomechanics* 87 (2019), 172–176. https://doi.org/10.1016/j.jbiomech.2019.02.015

[18] Joseph Whinnery. 2014. *US10249088B2 - System and method for remote virtual reality control of movable vehicle partitions.* Retrieved June 9, 2021 from https://patents.google.com/patent/US10249088B2/en

[19] Q. Zhou, G. Hagemann, D. Fafard, I. Stavness, and S. Fels. 2019. An Evaluation of Depth and Size Perception on a Spherical Fish Tank Virtual Reality Display. *IEEE Transactions on Visualization and Computer Graphics* 25, 5 (May 2019), 2040–2049. https://doi.org/10.1109/TVCG.2019.2898742