7 May 2015

Christian Ruiz, Christopher Harris, Jake A. Cole

University of New Haven

300 Boston Post Rd

West Haven, CT 06516

Tagliatela College of Engineering

University of New Haven

300 Boston Post Rd

West Haven, CT 06516

To whom it may concern:

Attached is a formal proposal for our senior design project. Titled "RobOculus: Oculus Rift Real World Implementation", explains the specifications and fundamentals of our proposed senior design project. In summary, the goal of this project is to implement the Oculus Rift, a virtual reality headset, to see through a robot's "eyes" as the user navigates said robot with a controller. This will help immerse the user into the robot's environment while keeping the user safe and comfortable. Applications may include using this in dangerous situations or situations where human presence and interaction may not be possible.

Sincerely,

Christian Ruiz

Christopher Harris

Jake A. Cole

Enclosure: Senior Design Proposal

ROBOCULUS

Oculus Rift Real World Implementation

Christian Ruiz, Christopher Harris, Jake A. Cole
Dr. Bijan Karimi

May 7, 2015

Executive Summary

The following project will use various hardware and software tools to implement a new immersive method of perceiving an environment through detached eyes. A virtual reality kit is used to view an environment and change the field of view through head movements. The environment is traversed by a robot, which can move around on wheels and send video information to the virtual reality headset through a computer mediator.

Acknowledgements

We would like to formally thank the Tagliatela College of Engineering as well as the University of New Haven for sponsoring our project and allowing us to use their facilities to work on the project. We would like to formally thank Doctor Bijan Karimi for advising our project. We would finally like to formally thank Mark Morton for helping us determine and retrieve the parts needed to build our prototype.

Table of Contents

Introduction

The following project implements a design for a robot in which a user can view the environment through the robot's "eyes". The Oculus Rift, a virtual reality headset, will immerse the user in the robot's point of view. The robot's field of view will change with the user's natural head and body movements. The user will use a game controller or keyboard to move the robot around its environment. Applications of this include but are not limited to exploration, medical surgeries, and interaction with an environment that is dangerous for human beings.

Project Definition

The Oculus Rift is a virtual reality headset that immerses the user into a virtual reality world. The headset serves as a display that renders the graphics at two different angles depending on each eye. This provides an immersive experience using infrared positional tracking (body movement) and internal tracking (head movement). The Oculus Rift uses stereoscopic 3D view to provide depth, scale and parallax with a large field of view to provide more immersion into the presented environment. The high definition screen in the Oculus Rift clearly displays the environment. Motion blur and judder is eliminated to remove simulator sickness. [5]

A robot is created to work parallel to the Oculus Rift. This robot will have two identical webcams and the ability to move according to the positional and internal tracking of the Oculus. The two webcams will stream video at two different angles to display on the Oculus' display to immerse the user in the physical world through the robot's "eyes" by using stereoscopic 3D. The head and body movements of the user is translated using the internal and external tracking systems of the Oculus to shift the field of view of the robot. An Xbox 360 controller is used to control the robot and move it around the environment.

After a brief patent search, there seems to be no patents for this type of project, where a robot interacts with a virtual reality headset to display the environment in a different perspective.

Evaluation of Alternatives

Various video and graphic handling libraries and tools have been tested to see what works well. For example, at first we looked at the Unreal Engine 4 to handle the Oculus Rift graphic handling (including webcams). We have settled on using OpenGL and OpenCV for graphics and the video feed. We have decided to go with a Lithium Ion battery instead of a Lead battery. Lithium Ion batteries seemed to be the best decision overall and much lighter than its alternative.

Design Approach

Various technical specifications for the Oculus Rift help benefit the project. Internal tracking is done using its internal Gyroscope, Accelerometer and Magnetometer with an update rate of 1000 Hz. Internal tracking is done with pitch, yaw and roll rotation. Pitch corresponds to the X-axis, yaw the Y-axis and roll the Z-axis. Pitch is when looking up and down, yaw is when looking left and right and roll is when tilting on the XY plane.

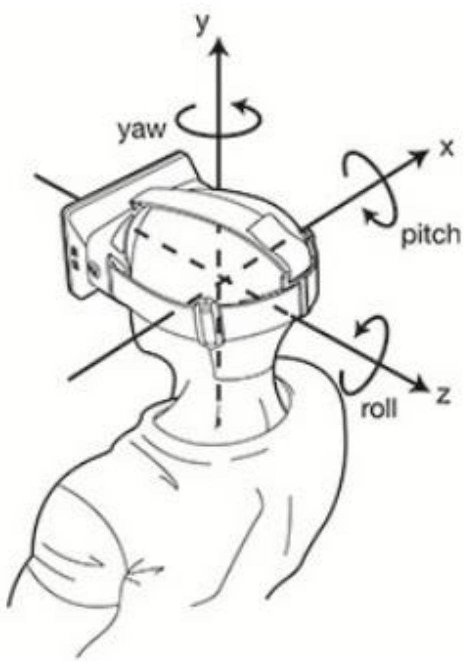


Figure 1: Oculus Rift coordinate system for internal tracking. [1]

The positional tracking is done using a near infrared CMOS sensor with a refresh rate of 60 Hz setup similar to a webcam. The Oculus Rift headset has various infrared dots the infrared sensor can pick up to determine the origin of the headset. As one moves their body and head the origin shifts providing more immersion. This helps head tracking overall, where values from the internal components are more consistent.

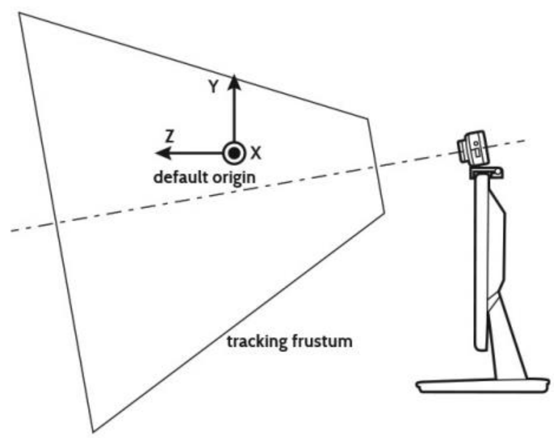


Figure 2: Position tracking camera and tracking frustum. [2]

The Oculus sports a 100-degree field of view, which gives the user a good sense of his or her surrounds of the presented virtual world. The Oculus provides high definition resolution of 960 x 1080 per eye with a refresh rate of 75 Hz. Low persistence in the Oculus eliminates motion blur and judder by turning the display on and off rapidly unnoticeable to the naked human eye. This process helps remove motion sickness. This is done by presenting unique images for each eye. These images are of the same environment at a slightly different angle, the same way our eyes naturally perceive images in the real world. [4]

The software for this project is written in C++. The Oculus SDK is used to interact with the Oculus Rift to read the headset's position. The OpenCV library is used to receive the video streams from the camera. OpenGL is used to process, render, and display the video feed from the OpenCV library. [6]

The final design includes a robot that can move in real time with fluidity. The cameras on the robot will send a video streams to the Oculus Rift in real time through a computer mediator. The transmission of the video stream and various commands are sent directly through a USB extender connected to a USB Hub on the robot. The software has been written to easily use and start up, which start up the real time multi-threaded processes and interact with the robot. The robot is controlled using an Xbox 360 controller connected directly to the user's computer. Interaction between the main program and the robot is possible by serially sending data to two Arduino Uno microcontrollers connected to the USB Hub. The video feed from the webcams are sent from the USB Hub to the main C++ program to send to the Oculus Rift. Two servos oriented on the yaw and pitch axes are used to move the platform holding the two cameras.

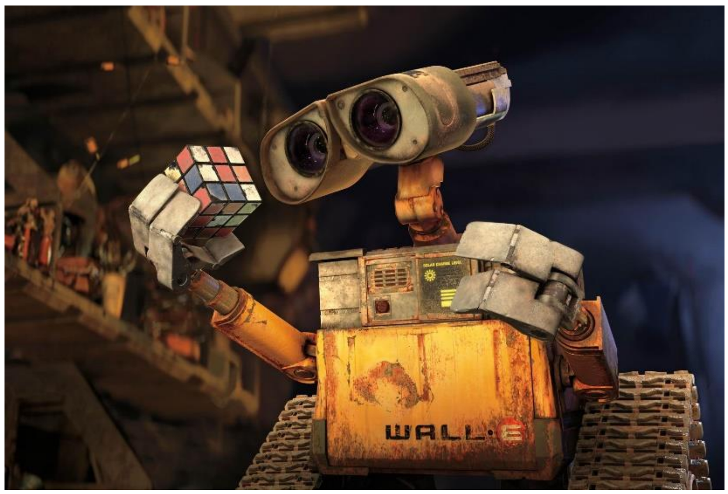


Figure 3: Conceptual idea of how one should think of the robot. [3]

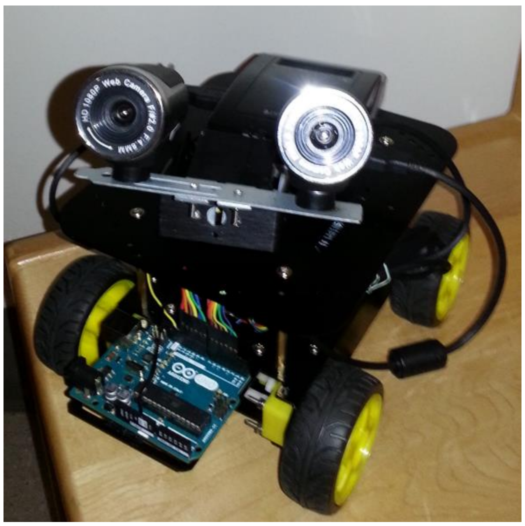
<u>Design Narrative</u>

For the software design, the Oculus SDK and video processing libraries, including OpenCV and OpenGL, are synched up to render the video feed to the Oculus Rift. OpenCV is used for the webcam image feed and OpenGL is used to render and display to the Oculus. OpenGL is created as a context. Wrappers are then used to access OpenGL functionality. GLFW is used to create an OpenGL context which is an object. The GLEW is used to bring in OpenGL functions such as texturing.

For the hardware design we decided to use two TeckNet 1080p HD webcams which are connected via the USB Hub to the computer. These cameras are mounted to the Lynx-motion pan and tilt kit. These cameras feed video into the computer which is interpreted by the C++ program. This pan and tilt kit is attached to the Arduino mobile platform. The Arduino Uno microcontrollers are mounted on the mobile platform. One Arduino Uno microcontroller is used to control the servos movement. The other microcontroller controls the robot's movement throughout its environment. These microcontrollers have code written in C that help process the serial data from the main C++ program to either move the wheels or move the servos.The Lithium Ion battery is then connected to both motor drivers. Each motor driver has an on-board voltage regulator that outputs a steady stream of 5 volts. The microcontrollers, servos, and webcams are powered by the USB Hub, which is receiving power from the computer.

The robot's movement throughout the environment is controlled with an Xbox 360 controller. The left analog stick controls the direction, the A button is used to drive forward, the B button is used to drive in reverse, the button X is used as a hand break, and the Y button is used to reset the current orientation of the Oculus Rift. The speed and direction data is sent serially to a separate Arduino Uno microcontroller, which processes the data and sends the digital PWM signals to the motor drivers.

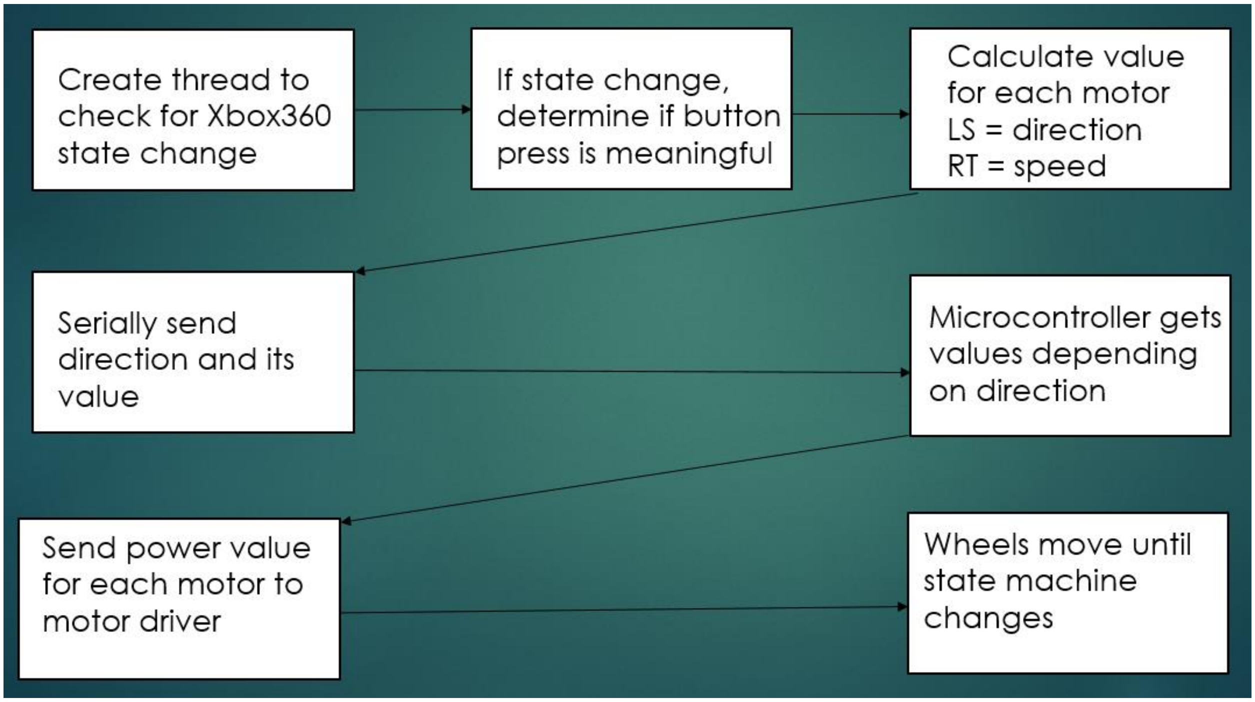
The head-tracking hardware controls pan and tilt servos. To accommodate all of the USB ports the robot has a USB hub that is connected to a 30ft USB extender. This connects the two cameras as well as the microcontrollers to the computer.



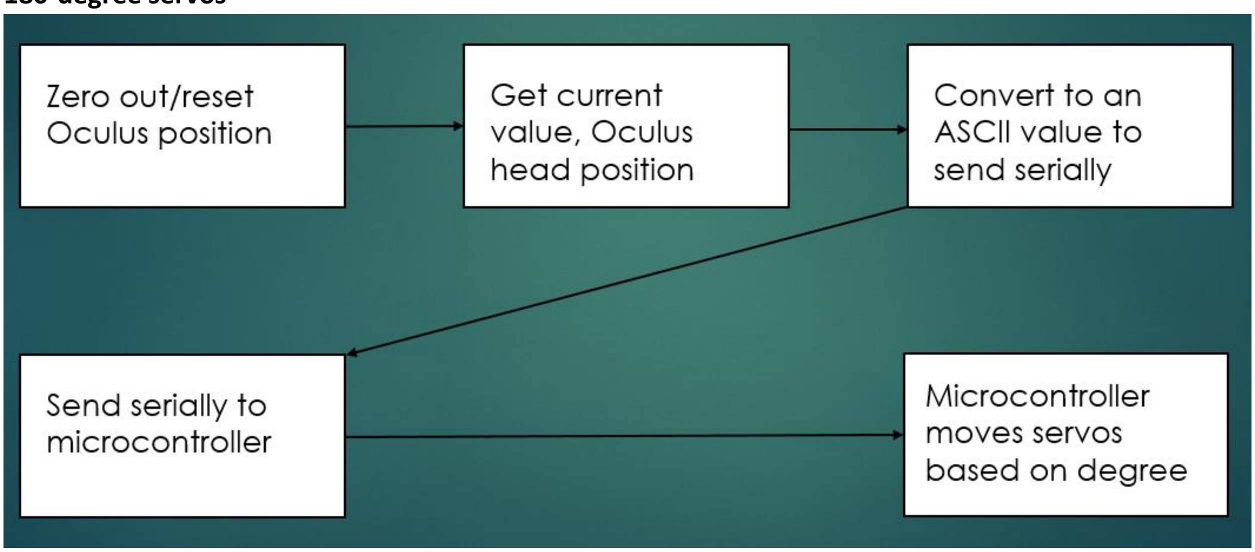
Design Implementation

1. Set up Oculus SDK and video processing libraries (OpenCV, OpenGL).
2. Configure the Oculus Rift correctly to send images out to the display.
3. Set up two cameras at different angles
4. Read and process the video stream using OpenCV and OpenGL and send the video stream to the Oculus Rift.
5. Set up and implement movement of the camera platform to correspond with the internal tracking (pitch and yaw). Positional tracking corresponds with the change of the default origin of the Oculus Rift.
6. Set up and implement environmental movement of the robot using four wheels. Movement is controlled using the Xbox 360 controller.

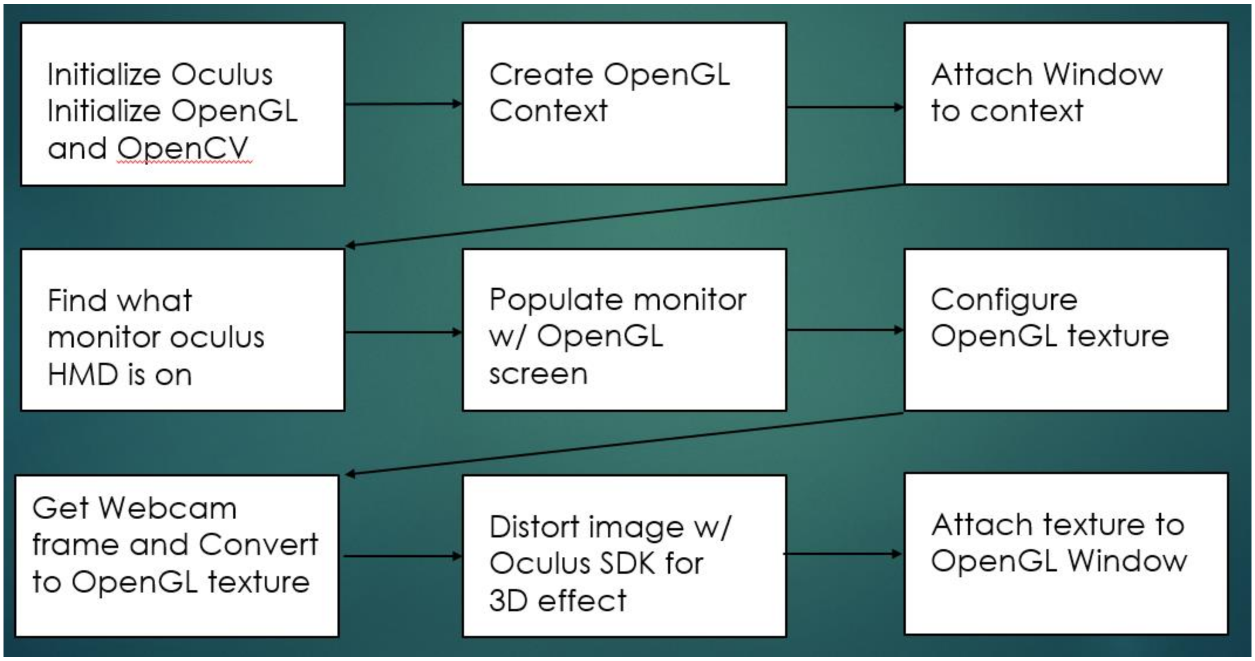
Mobile Platform and Motors



180-degree servos



Oculus Rift (OpenGl/OpenCv)



Performance Evaluation

The robot moves intuitively around the environment. We looked at various racing games to see various ways on controlling a car. We settled on using Mario Kart 8 controls, since these controls are simplistic, intuitive, and easy to pick-up. The servos move intuitively using the tracking data from the Oculus Rift.

Testing

Continuous testing is being done for each aspect of the project. There have been various tweaks on how the servos and wheels move. Improvements and optimizations are being worked on continuously to improve overall user experience.

Professional and Societal Concerns

The field of virtual reality is fairly new and still developing. Even though the Oculus Rift is still in development phase, it can still be utilized as a powerful tool in immersing the user in a 3D environment. Research in this field could cause technological breakthroughs that could revolutionize many different industries. For example, advances in virtual reality drastically change the medical field by allowing a doctor to perform an operation hundreds of miles away. It could allow a professional pilot to control an aircraft from his office. It could even allow for search and rescue teams to be immersed in an environment, even though they aren't there.

This robot can be utilized as a stepping-stone into immersion based robot piloting. The Oculus Rift allows for a robot pilot to have the display rendered to look like they are in the robot's environment. It may allow the pilot to focus in on certain details that he or she would have otherwise been unable to due to the stereoscopic vision displayed on the parabolic lenses of the Oculus Rift. This concept could be utilized in dangerous situations or situations where human presence and interaction may not be possible. Some situations may include firefighting, hostage recovery, search and rescue, low or no oxygen environments, deep sea and space exploration, and medical operations.

Materials and Costs

- Oculus Rift Development Kit 2 - $350
- (2) TeckNet 1080P HD Webcams - $80
- Lynx-motion Pan and Tilt Kit - $30
- (2) Arduino Uno Microcontrollers - provided by university
- (2) L298N Dual-H Bridge Motor Drivers - $10
- DEF-Robot 4WD Arduino Mobile Platform - $45
- Monoprice USB extender - $25
- Monoprice Aquagate USB Hub - $25
- Stereoscopic Webcam Mount - provided by university
- 11.1V Lithium Ion Battery - provided by university
- Xbox 360 Controller - personal controller
- (2) USB A to USB B Cables (Printer Cables) - provided by university

The Oculus Rift Development Kit 2 costs $350.00 [7]

The robot parts cost approximately $215.00 and were ordered off amazon.com as well as robotshop.com.

Overall, the prototype cost approximately $565 excluding the items provided by the university. If all items had to be purchased the cost would increase to approximately $600 to $650.

Economic Evaluation

Over the next few years, virtual reality headsets will become a more common household item and a good investment for entertainment. Therefore, the overall cost of an investment for a product like our RobOculus robot will be minimized. If this continued to be developed and optimized, it would be a good and economically friendly investment for various applications.

Discussion

In summary, there are three main aspects of the project. Robot movement through the wheels, servo movement for the cameras, and video processing and rendering to the Oculus Rift. Code is written in either C++ or C. Various libraries are used to handle video processing, video rendering, controller input, Oculus interfacing, and serial communication. All aspects work together using multi-threading to allow for asynchronous control of the various components of the robot.

Conclusions and Recommendations

To conclude, we were able to develop and implement a prototype of our robot. The robot’s field of view is constrained to two axes. Servo movement is done in the horizontal (yaw) and vertical (pitch) axes. The tilt axis (roll) is excluded due to the lack of some necessary components; however, good user experience is maintained. Movement of the robot is intuitive, where speed increases or decreases when you hold or let go of the acceleration buttons on the Xbox 360 controller. The robot is tethered to the computer with a 30ft USB extender.

Further development could be done to improve the overall user experience and add more functionality. A 270 degree or 360 degree servo could be used instead on the horizontal (yaw) axis and a 180 degree servo could then be used for the user's head tilt (roll). There could be a graphical HUD to provide useful information. Augmented reality could be explored to add various elements virtually to enhance the user's experience. Arms could be added to the robot to help interact with the environment. The vertical level of the camera mount could be made to rise or fall. Microphones could be used to listen to the environment with 3D sound. Speakers could be used for various alerts.

References

[1] http://static.oculusvr.com/sdk-downloads/documents/Oculus_Developer_Guide_0.4.1.pdf#page=22

[2] http://static.oculusvr.com/sdk-downloads/documents/Oculus_Developer_Guide_0.4.1.pdf#page=23

[3] http://realitypod.com/wp-content/uploads/2013/08/walle-photo-1.jpg

[4] http://www.oculusvr.com/dk2/

[5] http://www.oculusvr.com/rift/

[6] http://static.oculus.com/sdk-downloads/documents/Oculus_Developer_Guide_0.5.0.pdf

[7] https://www.oculusvr.com/order/

[8] https://www.opengl.org/

[9] http://opencv.org/

[10] http://www.glfw.org/

[11] http://glew.sourceforge.net/