Department of Computer Science & Engineering
The University of Texas at Arlington

Detailed Design Specification
CSE 4317: Senior Design II
Spring 2017

Team Telepresence
Rift Telepresence

Cameron Adams
John Green
Andy Le
Clement Olayiwola
Ty Simmel

Revision History

Revision	Date	Author(s)	Description
0.1	02.26.2017	CA, JG, AL, CO, TS	Document creation
0.2	03.10.2017	CA, JG, AL, CO, TS	Update document
0.3	05.08.2017	CA, JG, AL, CO, TS	Update document with new design

CONTENTS

List of Figures

List of Tables

1 INTRODUCTION

The project utilizes a stereo camera paired with an Oculus Rift virtual reality headset. The video is streamed into the Oculus as a stereo image allowing a better sense of depth than a traditional camera setup. The stereo camera is mounted to a custom 3-axis gimbal controlled by the head tracking in the Oculus headset. The gimbal stabilizes the stereo camera to match the angle of the Oculus headset. This stabilization on set angles allows the mounting of the gimbal on mobile or stationary platforms.

The Camera System, including the gimbal and motor controller, is detailed in section three. The encoding and rendering of the video is contained in the Processing System. The head tracking data signal translation falls into the realm of the Processing System and is covered in section four. Section five relates to the Virtual Reality System. The Virtual Reality System displays the rendered video and provides head tracking data.

2 SYSTEM OVERVIEW

The Camera System handles three primary tasks. The motor controller board receives signals from the Processing System. The control board then orients the gimbal to the angles specified by the signals. For the final task, the cameras capture stereo video and pass the video to the Processing System. The Processing System serves as an intermediary between the Camera System and Virtual Reality System. The two functionalities are processing the video and translating the head tracking data. The final system is the Virtual Reality System. This system is responsible for the stereo display of the video and the capture of the raw head tracking data.

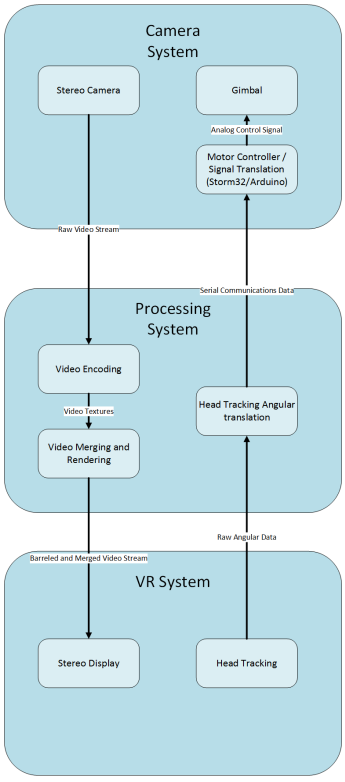

Figure 1: System architecture

3 Camera System Layer Subsystems

The Camera System Layer contains a camera subsystem, the gimbal subsystem, and signal controller subsystem. The camera subsystem streams the raw video feeds of the stereo cameras. The gimbal subsystem maintains the desired camera angles. The signal controller subsystem receives the angular data and controls the motor based on the data.

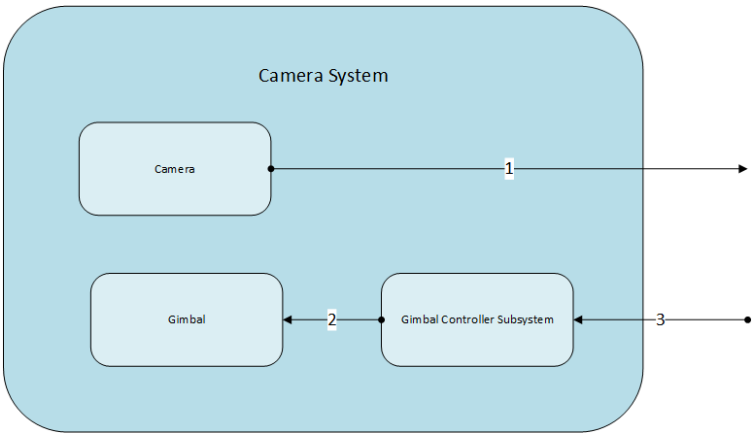

Figure 2: Camera System Layer subsystem description diagram

3.1 Layer Hardware

The Camera System has many hardware components. The camera subsystem contains two wide-angle HD USB camera modules, ELP-USBFHD01M-L21. The gimbal subsystem contains three brushless motors, Rctimer GBM2804, and a custom 3D printed housing. The signal controller subsystem contains a motor control board, Storm32-BGC V1.3.

3.2 Layer Operating System

Windows 7 or newer

3.3 Layer Software Dependencies

Requires the width and height of the video resolution to be set before the cameras begin to start sending video feed.

3.4 Camera Subsystem

The camera subsystem receives input from the cameras and sends the video signal to the Processing System.

3.4.1 Subsystem Hardware

The camera modules mentioned above (model No: ELP-USBFHD01M-L21) will be used side-by-side, separated by about 65 mm to simulate the average distance between human eyes.

3.4.2 Subsystem Operating System

Supporting operating system: WinXP, Vista, Win7, Win8, Linux with UVC MAC-OS X 10.4.8 or later, Wince with UVC, Android 4.0 or above

3.4.3 Subsystem Software Dependencies

The video resolution of the cameras is required to be set before starting to send video feed to the video encoding subsystem in the Processing System.

3.4.4 Subsystem Programming Languages

None

3.4.5 Subsystem Data Structures

The video output of the camera modules is FMPEG with a user-specified frame rate and resolution per frame.

3.4.6 Subsystem Data Processing

None

3.5 Gimbal Subsystem

The gimbal subsystem is a custom built three axis gimbal with stereo camera mounts and external mounting plates.

3.5.1 Subsystem Hardware

There are three Rctimer GBM2804 motors and a custom 3D printed housing.

3.5.2 Subsystem Operating System

None

3.5.3 Subsystem Software Dependencies

The Rctimer motors rely on the PWM output of the Storm32-BGC board for angular instruction.

3.5.4 Subsystem Programming Languages

None

3.5.5 Subsystem Data Structures

None

3.5.6 Subsystem Data Processing

None

3.6 Signal Controller Subsystem

The signal controller subsystem receives a pulse width modulated signal representing the three angles that the camera should be aligned relative to the initialized position. This alignment is achieved through the use of two inertial measurement unit modules. The control board adjusts the motors to keep the gimbal in alignment.

3.6.1 Subsystem Hardware

There are three Rctimer GBM2804 motors, a custom 3D printed housing, a Storm32-BGC V1.3, and MPU6050 IMU.

3.6.2 Subsystem Operating System

The control board runs on configurable firmware. The version used was V0.90.

3.6.3 Subsystem Software Dependencies

The PWM signal must be a pulse between 1-2 ms with a 15ms period.

3.6.4 Subsystem Programming Languages

User specifications can be digitally sent to the Storm32-BGC board using OlliW's custom user interface which translates settings directly to assembly level programming of the onboard STM32F103RC MCU.

3.6.5 Subsystem Data Structures

The Storm32-BGC v1.3 utilizes a Futuba S-Bus for data communication across the board's components.

3.6.6 Subsystem Data Processing

None

4 Processing System Layer Subsystems

This system contains the video encoding subsystem, video rendering subsystem, and the head tracking translation subsystem. The subsystems communicate with the Camera System and the Virtual Reality System respectively. Video data will be received from the Camera System and rendered to the Virtual Reality System. The raw head tracking data is captured and translated to analog signals for the Camera System.

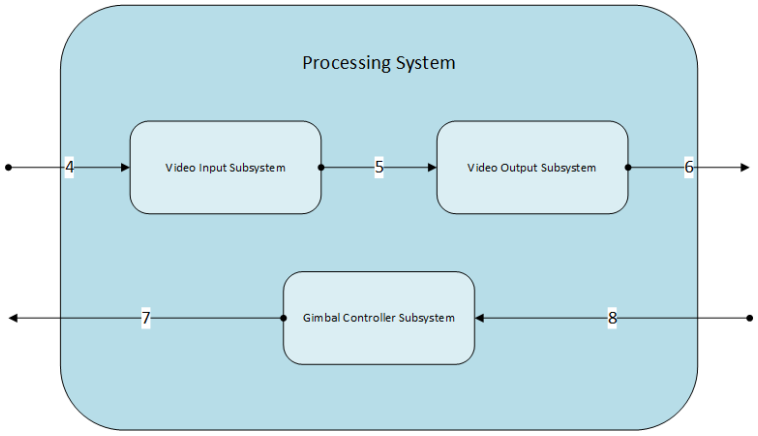

Figure 3: Processing System Layer subsystem description diagram

4.1 Layer Hardware

A computer powerful enough to encode and render two 1080p 30 frames per second video, an Intel i5 or an Nvidia GeForce GTX 970 are the suggested minimums, and an Arduino Uno is used as the serical interface for the Storm32-BGC gimbal controller.

4.2 Layer Operating System

The supported operating system is Windows 7 or newer.

4.3 Layer Software Dependencies

An Oculus Rift setup utility has been executed on the computer. Unity 5.5 if cameras need to be adjusted or changes are needed. Arduino IDE is recommended if serial interface changes must be made as well as for the initial setup of the Arduino Uno.

4.4 Video Encoding Subsystem

This subsystem receives the raw FMPEG video streams from the Camera System. The raw video is encoded into a Unity texture.

4.4.1 Subsystem Hardware

A minimum of Intel i5 or Nvidia GeForce GTX 970 is suggested.

4.4.2 Subsystem Operating System

Windows 7 or newer

4.4.3 Subsystem Software Dependencies

Unity 5.5

4.4.4 Subsystem Programming Languages

C# has been used as the default scripting language for its nativity in Unity.

4.4.5 Subsystem Data Structures

Array of textures and video frame buffers.

4.4.6 Subsystem Data Processing

FMPEG to Unity texture

4.5 Video Rendering Subsystem

This subsystem takes the two video textures and overlays them onto a user interface menu element. Then, the images are barreled and merged. The rendered image is streamed to the Virtual Reality System.

4.5.1 Subsystem Hardware

A minimum of Intel i5 or Nvidia GeForce GTX 970 is suggested.

4.5.2 Subsystem Operating System

Windows 7 or newer

4.5.3 Subsystem Software Dependencies

Unity 5.5

4.5.4 Subsystem Programming Languages

C#

4.5.5 Subsystem Data Structures

None

4.5.6 Subsystem Data Processing

For each frame of the videos, barreled distortion is applied to match the perspective of the Oculus Rift. Each camera input is applied to its respective eye's texture to create the illusion of depth.

4.6 Angular Translation Subsystem

This subsystem takes the raw head tracking data and converts it into angular data. The angular data will then be sent to an Arduino over a serial interface. The Arduino converts the data to a PWM readable signal and sends that signal to the gimbal subsystem of the Camera System.

4.6.1 Subsystem Hardware

Arduino Uno

4.6.2 Subsystem Operating System

Arduino IDE

4.6.3 Subsystem Software Dependencies

Unity 5.5

4.6.4 Subsystem Programming Languages

C

4.6.5 Subsystem Data Structures

The 'Servo.h' library has been utilized to easily translate integer values into PWM readable signals to be written out from the Arduino.

4.6.6 Subsystem Data Processing

At sixty times a second, the system reads the head tracking data and converts it to angular data. The conversion uses the x-y-z position values to generate euler angles. The angles are modified to conform to the system operational ranges.

5 Virtual Reality System Layer Subsystems

The Virtual Reality System is focused on delivering real-time video and tracking of the head movement on the virtual reality headset. This system contains the stereo display subsystem and the head tracking subsystem. The stereo display subsystem will interact with the video rendering subsystem, while the head tracking subsystem will interact with the angular translation subsystem.

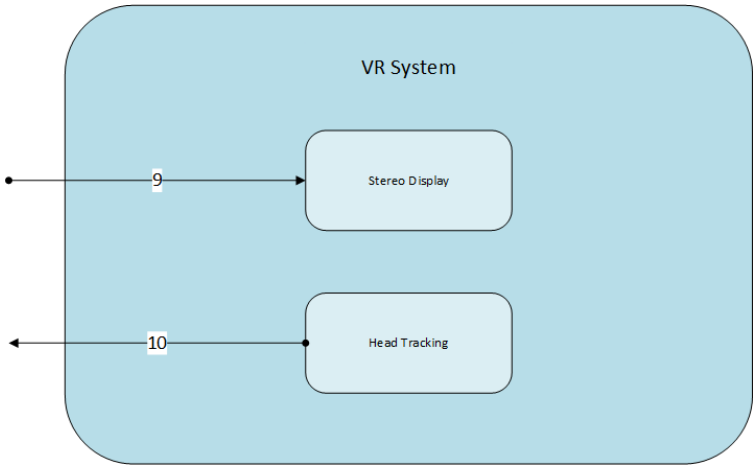

Figure 4: Example subsystem description diagram

5.1 Layer Hardware

The Oculus Rift headset and the Oculus positional sensor are used in this system.

5.2 Layer Operating System

Windows 7 or newer

5.3 Layer Software Dependencies

Oculus Rift drivers and utilities

5.4 Stereo Display Subsystem

This subsystem interacts with the video rendering subsystem of the Processing System by receiving video data. This subsystem decodes and displays the video feed from the video rendering subsystem.

5.4.1 Subsystem Hardware

Oculus Rift headset

5.4.2 Subsystem Operating System

None

5.4.3 Subsystem Software Dependencies

Oculus Rift drivers and utilities are installed on the computer.

5.4.4 Subsystem Programming Languages

None

5.4.5 Subsystem Data Structures

None

5.4.6 Subsystem Data Processing

None

5.5 Head Tracking Subsystem

This subsystem interacts with the angular translation subsystem of the Processing System by sending positional data gathered from the gyros and accelerometers in the Oculus headset combined with the Oculus positional sensor.

5.5.1 Subsystem Hardware

Oculus Rift headset and Oculus positional sensor

5.5.2 Subsystem Operating System

None

5.5.3 Subsystem Software Dependencies

Unity 5.5, Oculus Rift drivers, and Oculus Rift utilities are installed on the computer.

5.5.4 Subsystem Programming Languages

None

5.5.5 Subsystem Data Structures

None

5.5.6 Subsystem Data Processing

None

6 APPENDIX A

Include any additional documents (CAD design, circuit schematics, etc) as an appendix as necessary.