Project Periscope

Project proposal

Team:
Aabhas Sharma (asharm36)
Michael Congdon (mcongdo2)
Siddharth Murali (smurali2)

TA:
Ankit Jain

ECE 445 – Senior Design

Fall 2015

University of Illinois at Urbana-Champaign

Appendix

1. Introduction
    a. Statement of Purpose
    b. Objectives
        i. Consumer Benefits
        ii. Product Features
2. Design
    a. Block Diagram
    b. Block Description
        i. Camera Unit
        ii. Computer – Oculus SDK
        iii. Oculus Rift unit
        iv. Swivel unit
3. Requirements and Verification
    a. Modular Functionality
    b. Tolerance Analysis
4. Cost and Schedule
    a. Cost Analysis
        i. Labor
        ii. Parts
        iii. Grand Total
    b. Schedule

1. Introduction:

Title: Project Periscope

- Statement:

  Currently there's a lot we can do with VR, but can we actually be in an entirely different place in a matter of seconds? We're not quite there yet. The idea behind Project Periscope is to be able to create a more engaging virtual reality experience - one where we can actually have eyes in a very different part of the world.

  Our plan, in its element, is quite simple – connect a camera to an Oculus Rift. We want to view a camera's live feed on the Oculus' display while being able to physically navigate the camera based on the Oculus wearer's head movements – give the whole experience a more realistic feel. What this essentially means is that the camera will follow the wearer's movements, allowing a user to not just see a live feed of someplace else, but also be able to freely navigate through the view with a broader field of vision – all this in real time.

Objective:

Consumer Benefits

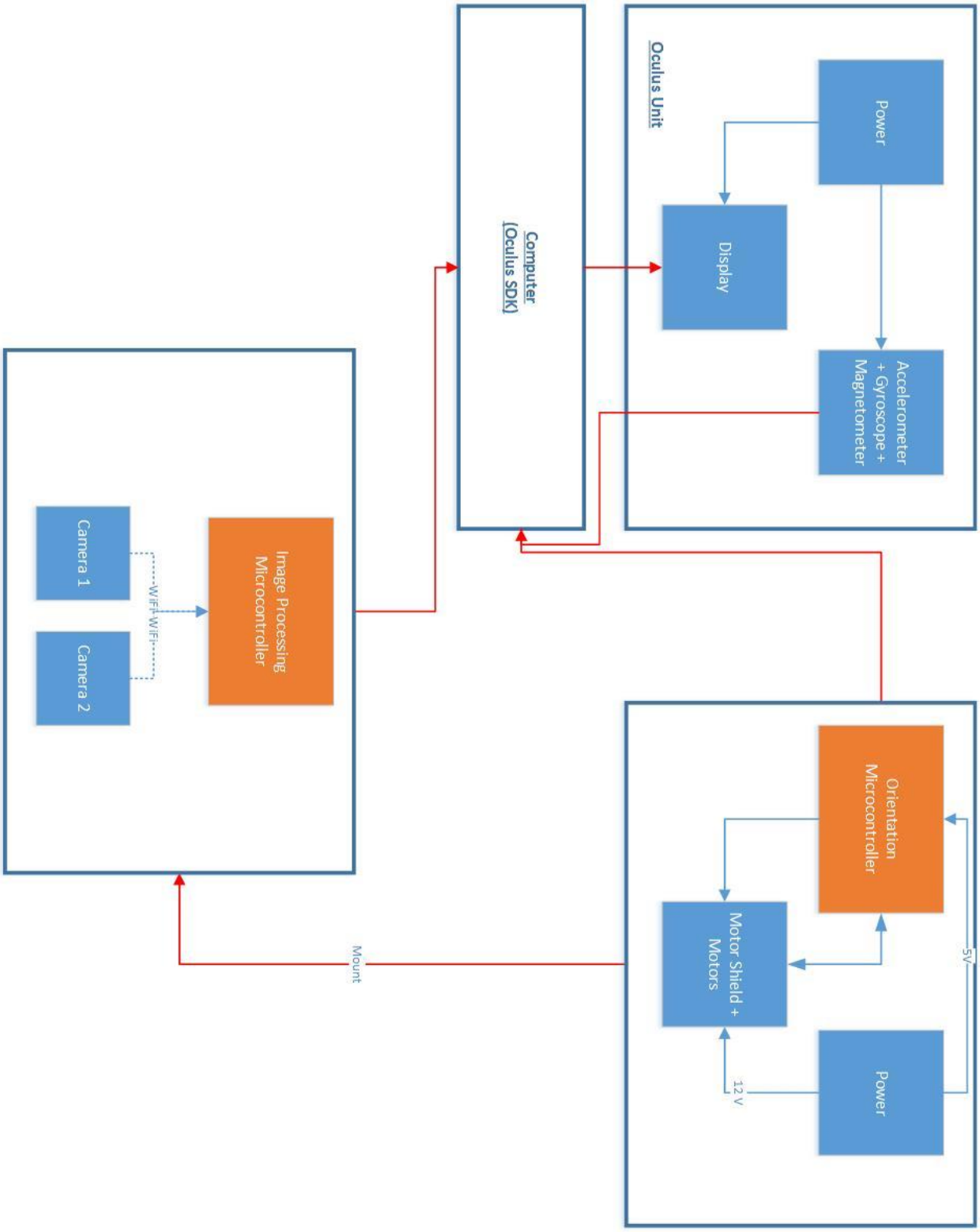
- Be able to view a very different part of the world real time
- Physically navigate through their field of vision - simulating human vision and free range of movement
- Multiple applications:
  - Can be used as a remote surveillance system (with mobile capabilities as well)
  - Can be used for remote imaging - aerial, terrestrial, and marine
  - Exploring the world - see places where you can't otherwise go to. For example, visit the Eiffel Tower! A VR substitute for View Master
  - Can bridge telepresence with media events - such as football games and conventions or conferences like ComicCon - where the user can explore the event with their own freedom - look wherever they find most interesting
  - Open up future possibilities for nano-imaging, especially for the medical field, allowing doctors to see inside the human body like never before

Product Features

- Wireless Streaming
- Wireless data-transfer between GoPro and the Oculus
- First-person remote viewing of virtually any environment
- Remotely controlled camera
- Head motion tracking

2. Design:

Block Diagram



Block Descriptions

- Camera
  - The camera is the “eyes” of our project. It has its own WiFi hotspot, which the laptop connects to in order to stream the video. The video feed is passed from the camera to an image processing program, which will stitch the two images together and then pass it to the Oculus SDK
  - Microcontroller 1
    - Will be using this to conduct the image processing required to modify and stitch together the camera feed to align with the Oculus display’s expected format.
- Computer - Oculus SDK
  - The Oculus SDK is what controls what you see as well as which way you turn. The video feed is transmitted from the image processing software into the SDK, and this transmits the feed into the Oculus display, showing you what the cameras see. The SDK also receives head movement data from the accelerometer/gyroscope and uses this to determine the direction in which the motor needs to move.
- Oculus Rift Unit
  - This consists of three parts – display, power supply, and motion tracking
  - The display gets the video from the SDK and shows it to the user as a 360 degree view.
  - The power supply powers the Oculus Rift Unit.
  - The motion tracking consists of an accelerometer, a gyroscope, and a magnetometer. These three combined relay the headset orientation data to the SDK.
- Swivel Unit
  - Swivel Unit Power Supply
    - Power supply will consist of a 12v rechargeable NiMh battery pack, and an array of AA batteries with a linear regulator, to supply 5v and 12v rails.
  - Orientation Control Unit
    - Arduino Uno running Simulink control algorithms. Takes in orientation data from the Oculus’s on-board gyroscope, and converts that to the positions required of 3 motors controlling the orientation of the cameras.
  - Motor Shield and Motors
    - The signals from the Arduino will first need to be stepped up through a motor shield powered by the 12v of the power supply. Unit will use three motors with rotary encoders to control roll, pitch, and yaw. Encoder signals will feed back to orientation control unit.

3. Requirements and Verification:

- Modular Functionality

Requirement	Verification	Points
Oculus Rift Unit		
a) Connection with Computer SDK unit	a. Test using the Oculus Configuration Utility feature within the SDK	5
b) Display and outgoing orientation data	b. Test using the Oculus SDK pre-loaded demos	10
Camera unit		
c) GoPros power supply	c. Check individual GoPro displays for battery level	1
d) GoPros are recording	d. Individually use it to take pictures/videos	1
e) GoPros WiFi system connectivity	e. Connect PC to GoPros WiFi network. Check data packets coming over the network. Packets should be actively updating.	15
f) Microcontroller should connect to Oculus SDK	f. Test data sharing between SDK and Microcontroller using SDK's in built tools	10
g) Image processing program should give ideal output	g. Test software individually using sample GoPro videos	15
Swivel Arm Unit		
h) Sustain voltage rating under no-load and full-load conditions to within 5%	h. Measure voltage drop when driving motors simultaneously at maximum speed, should be within 5%	5
i) Position tracking causes less than 5% overshoot in worst case scenario	i. Provide (a rather large) step input to control unit and measure response, should maintain overshoot under 5% within a reasonably short settling time	10
j) Motor (more specifically the gear box) must be able implement orientation data realistically, quick head movements with slow tracking could be disorienting	j. Affix an estimated physical load to motor prior to construction, drive motor. Should be able to spin fast enough to track head movement realistically.	9
k) Microcontroller should connect to Oculus SDK for orientation data sharing	k. Test data sharing between SDK and Microcontroller using SDK's in built tools	5

l) Microcontroller should connect to motor	l. Test that by running elementary high/low commands to see motor's reaction	5
m) Microcontroller software should translate orientation data into motor accepted format	m. Test software individually using sample data	9
Total		100

- Tolerance Analysis
    - The most critical aspect that would have to be tested with extreme measures would be the latency of the entire system. With multiple bottlenecks in place, this will be the biggest challenge for us to overcome. The data from the camera needs to get across to the Oculus with a maximum delay of 0.5 seconds. At the same time, the orientation data from the Oculus needs to get translated and implemented by the swivel unit in under 0.5 seconds.

4. Cost and Schedule:

Item	Manufacturer	Quantity	Unit Price	Cost	Description
Hero3+ Silver	GoPro	2	$300.00	$600.00	Link
Oculus Rift	Oculus	1	$350.00	$350.00	Link
Microcontroller	Arduino	2	(pre-owned)	$0.00	Link
Motors	N/A	3	$40.00	$120.00	Link
NiMh battery pack	N/A	1	$30	$30.00	Link
5v regulator	N/A	1	$0.42	$0.42	Link
AA Batteries	N/A	4	$0.34	$1.36	Link
Battery Holder	N/A	1	$1.00	$1.00	Link
Motor Driver	N/A	3	$1.66	$4.98	Link
Total				$1,107.76	

Name	Hourly Rate	Total Time	Cost
Siddharth Murali	$40.00	200	$20,000
Aabhas Sharma	$40.00	200	$20,000
Michael Congdon	$40.00	200	$20,000
Total		600	$60,000

Section	Total
Parts	$1,107.76
Labor	$60,000
Grand Total	$61,107.76

Week	Task	Responsibility
21-Sep	Design power supplies and motor drivers	Michael
	Order parts	Siddharth
	Prepare for Mock Design Review	Aabhas
28-Sep	Begin constructing circuits and designing armature to hold camera	Michael
	Prepare for Design Review	Siddharth
	Get raw video feed from camera to computer	Aabhas
5-Oct	Talk with machine shop about constructing armature	Michael
	Program Microcontroller for image processing	Aabhas, Siddharth
12-Oct	Begin assembly of components onto armature	Michael
	Program Microcontroller for transmitting video to Oculus	Aabhas, Siddharth
19-Oct	Begin writing control logic for armature	Michael
	Program Oculus for viewing camera feed	Siddharth
	Program Oculus for getting head motion data	Aabhas
26-Oct	Test/validate motor function and basic control	Michael
	Test/validate Oculus functions (display, head tracking)	Siddharth
	Test/validate video transmission and image processing	Aabhas
2-Nov	Refine control logic	Michael
	Prepare Mock Demo	Siddharth
	Ensure complete functionality	Aabhas
9-Nov	Final assembly of components	Michael
	Prepare Mock presentation	Siddharth
	Fix any remaining issues	Aabhas
16-Nov	Final testing and debugging – Hardware	Michael
	Final testing and debugging – Software	Siddharth
	Prepare demonstration	Aabhas
23-Nov	Prepare for Final presentation and demonstration	All
30-Nov	Final Report	All
7-Dec	Lab checkout	All