eyeris

Client

Virtual Reality Applications Center (VRAC)

Project Eyeris, May 13-20

TEAM

Scott Connell
Arjay Vander Velden
Tyler Burnham
Justin Derby
Will Bryan
Kris Scott

ADVISOR

Daji Qiao

CLIENT

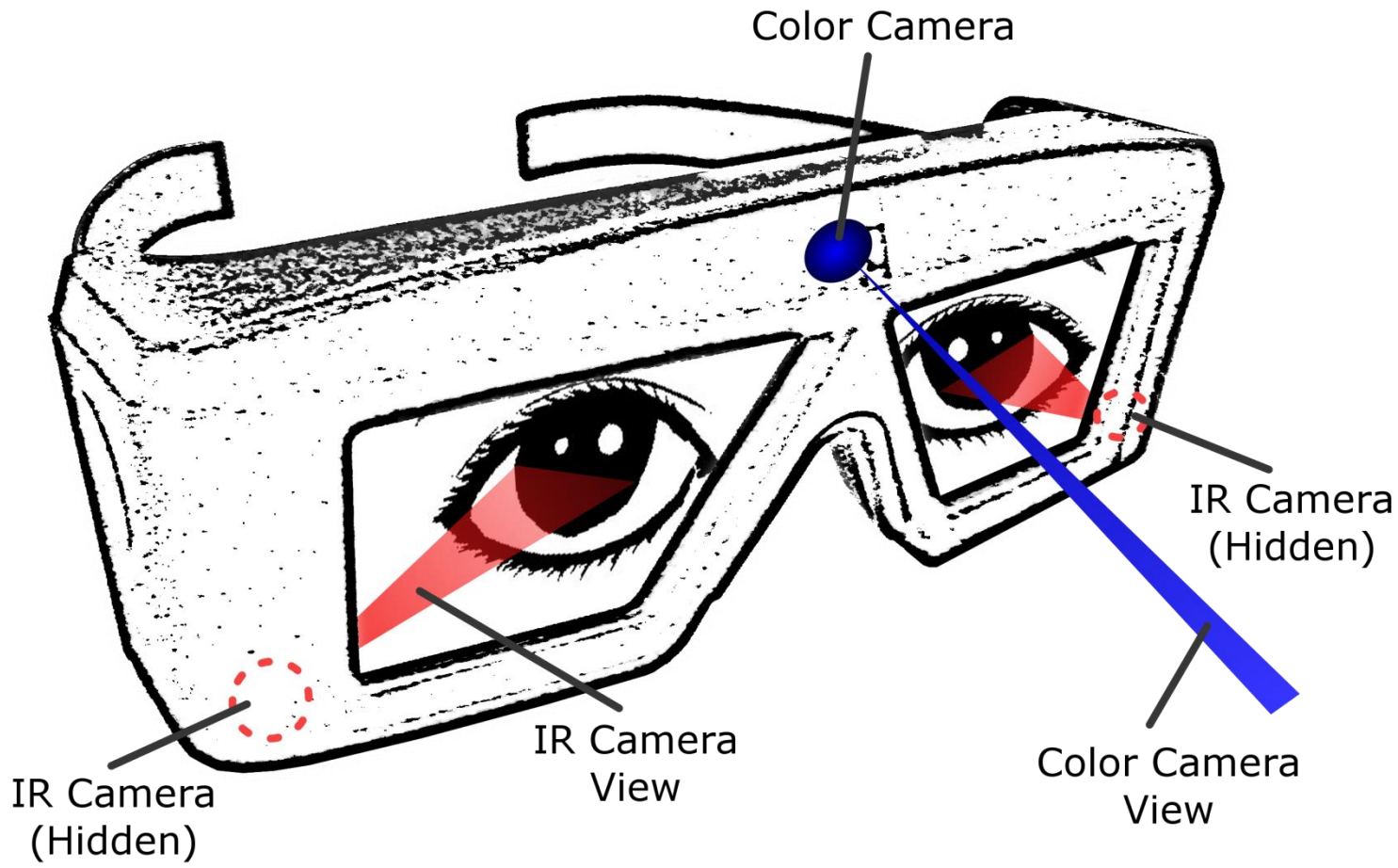
Stephen Gilbert, VRAC

System Description

Project Concept

- Embedded, real-time eye tracking system
- Stream outward video and eye tracking data wirelessly
- Will be used in the C6/MIRAGE for virtual reality applications

Concept Sketch



Functional Requirements

- Track both eyes for 3D depth
- Two-hour, onboard cache of world view video and eye data
- Real-time, wireless transmission of world view video
- Real-time, wireless transmission of eye tracking data
- Video and eye tracking data must be in-sync

Non-Functional Requirements

General Requirements

- Glasses will be active, stereoshutter glasses
- System must be unobstructive and nonintrusive
- Battery must last at least three hours

Non-Functional Requirements

Video Requirements

- Outward-facing camera must be high definition (720p or 1080p)
- The inward-facing cameras must capture at 30fps

Non-Functional Requirements

Physical Requirements

- Weight constraints
  - Glasses – 2.3 lbs.
  - Backpack – 5 lbs.
- Dimensional constraints
  - Glasses – no wider than 10 inches

Existing Products

Existing Market

Name	Cost	Eyes Tracked	Mobile	Real-time
Tobii	$45,000	One eye	Yes	No (Recorded)
SMI	$30,000	One eye	Yes	No (Recorded)
MIT Graduate Project	$85.92	One eye	No	Yes
Google Glass	$1,500	One eye*	Yes	Yes

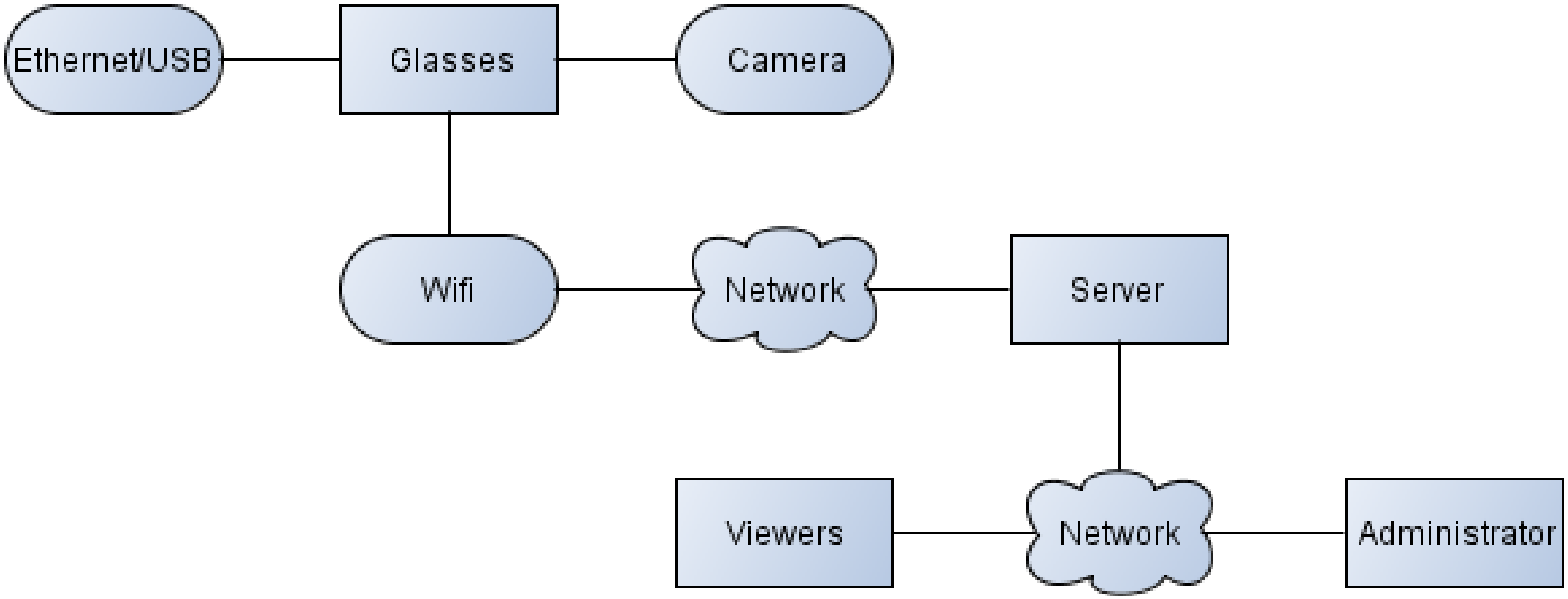
Project Justification

What can we improve upon?

- High resolution real-time streaming over Wi-Fi
- 3D vector tracking (adds depth to tracking location)
- Cost of the system: ~$550

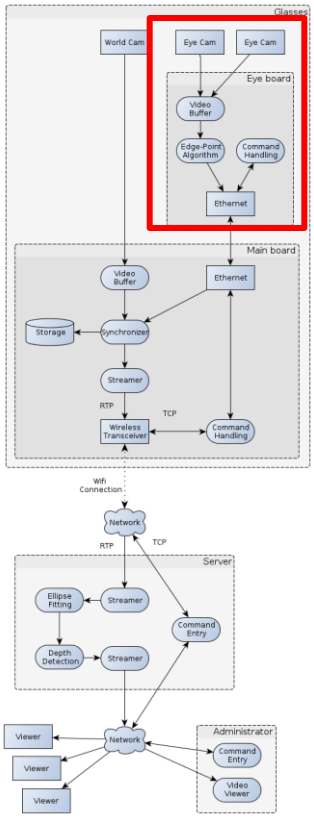
System Overview

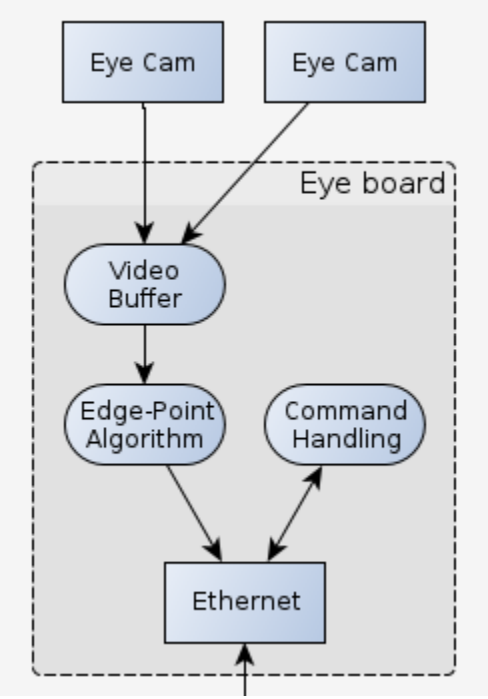
High-Level Decomposition

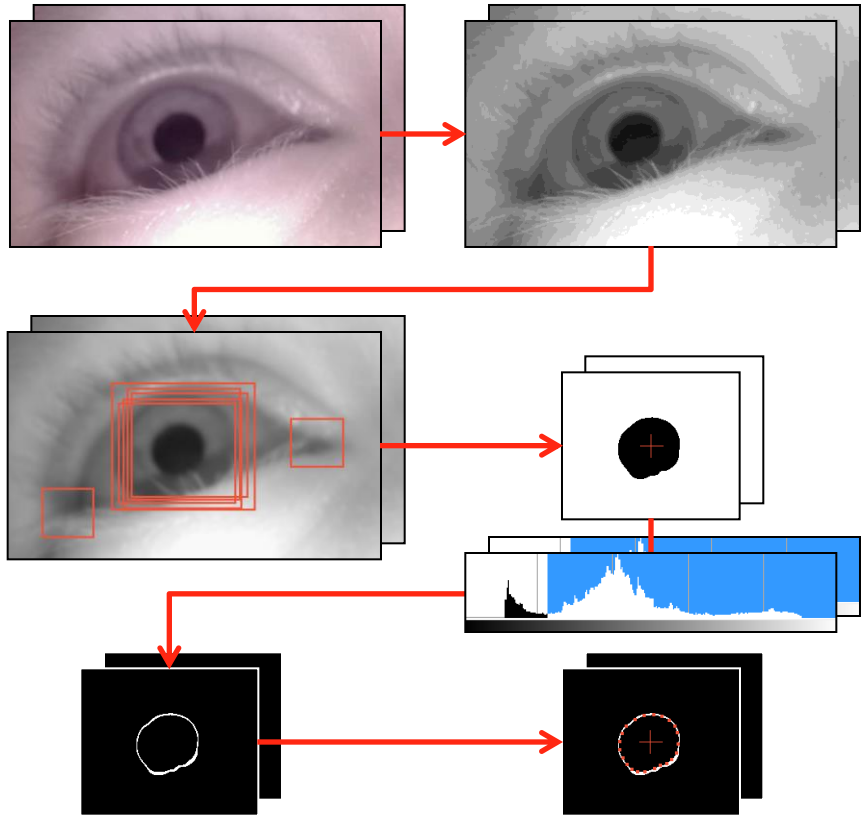


Functional Decomposition

Glasses

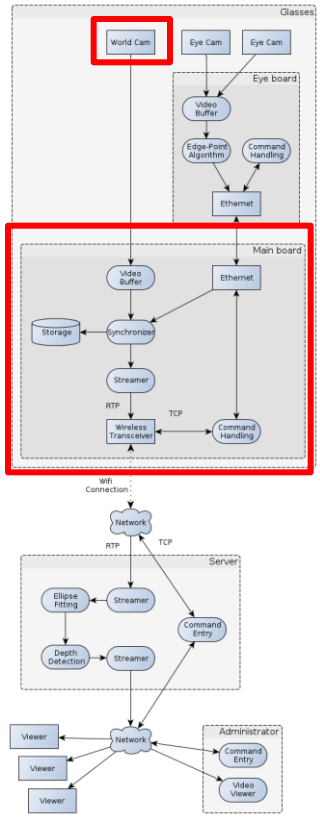


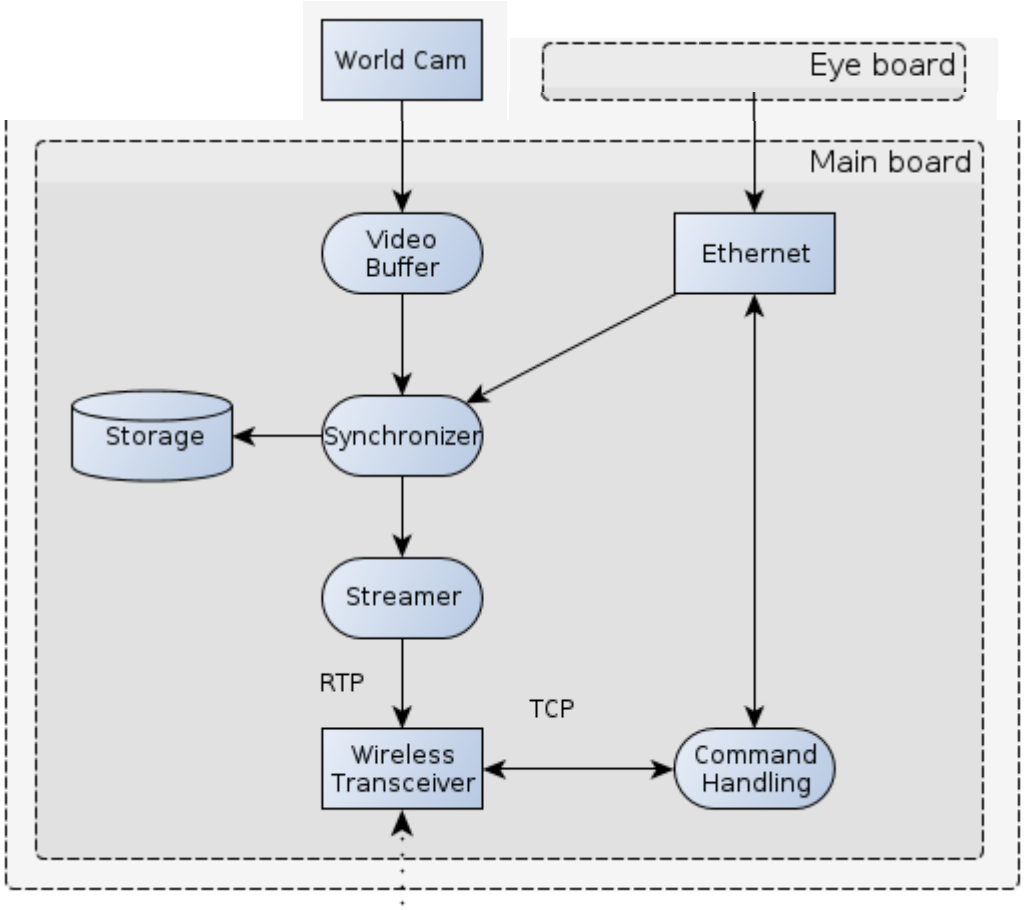


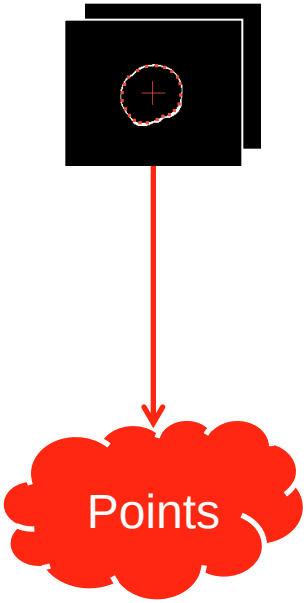


Functional Decomposition

Mainboard

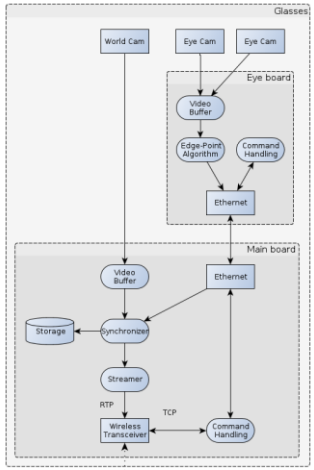


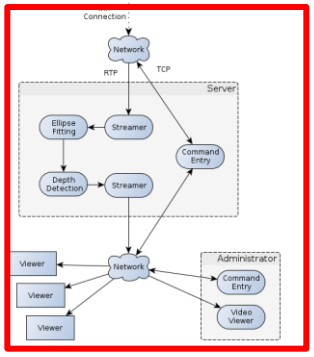


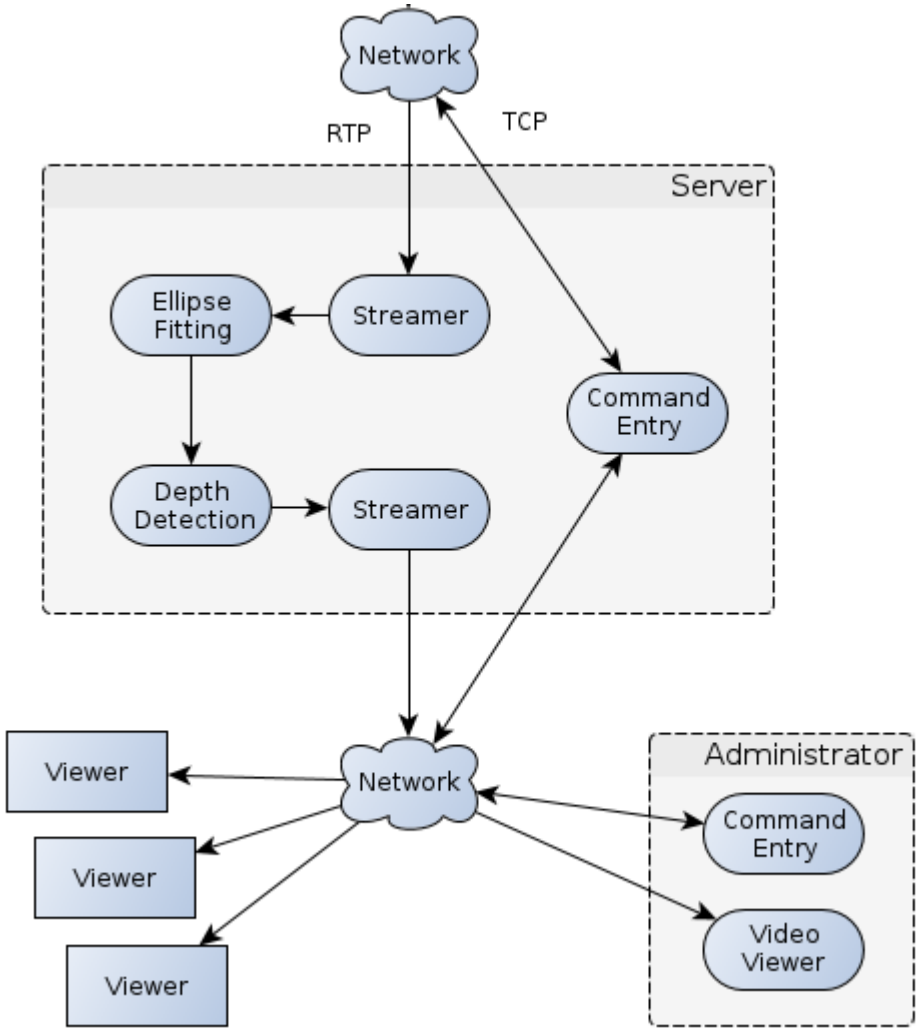


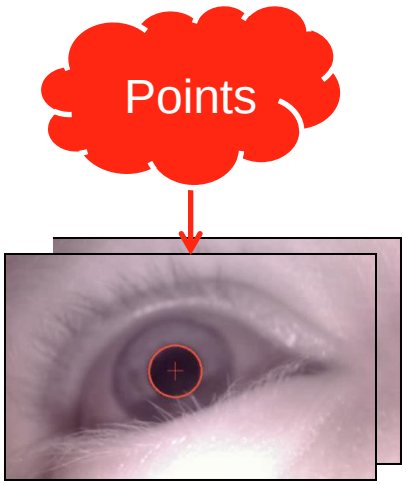
Functional Decomposition

Network









Design Rationale

- Eye tracking on a mobile platform uses a lot of processing power
  - Off-load to multiple embedded devices as well as to a dedicated server
- Provide a proxy off of the mobile platform to conserve battery
  - 1-to-1 connection vs. 1-to-many
- Point processing on embedded devices
  - Don’t have to stream as much data

System Specifications

Software Used

Embedded

- OpenCV
- TBB (Threading Building Blocks)
- Boost (C++)
- Eigen
- Gstreamer

Client/Server

- C++ using QT Framework

Software: Technical Challenges

- Library dependencies and getting to know how to use them
- Finding a balance between accuracy vs. processing power
- Streaming custom serialized data over a network
- Making the server/embedded devices memory safe (stable) for long-term use

Hardware Used

Gumstix DuoVero

- Omap4430 Dual Core 1.0 GHz

PandaBoard ES

- Omap4460 Dual Core 1.2Ghz
- 802.11 b/g/n (wifi)

Technical Challenges

Gumstix DuoVero

- Selecting & Configuring OS
- Trying to use the OTG USB port

PandaBoard ES

- CPU Frequency Scaling Governor
- Buggy Kernel

Hardware Used

Microsoft LifeCam HD-6000

- Eye cameras
- Heavily modified

Logitech C270

- Outward facing camera
- Captures at 640x480

Belkin Power Pack 2000

- Can be connected in parallel

Hardware : Technical Challenges

Cameras

- Fitting the system into the stereoscopic glasses
- Finding a camera that works for this application

Power

- Getting correct current and voltage for each device
- Getting the correct battery life

Outstanding Issues

Requirements & Implementation Differences

	Requirement	Implementation
1	iMotions Interface	Left code to be able to output what ever format they need
2	Inward eye cameras record at 640x480	Inward eye cameras record at 160x120 for faster processing and bandwidth restrictions
3	Outward eye camera record at 720p	Outward eye camera default resolution records at 640x480 (480p)
4	Depth tracking in final solution	Time constraints; didn't have system running in time to correctly test and implement
5	Solution lasts 3 hours with battery	Time constraints; didn't have system running in time to do power characterization
6	2 hour onboard video cache	Server does processing, so the Client holds the video cache

eyeris

Cost Rundown

Part	Cost	Quantity	Totals
Gumstix DuoVero	$169.00	x1	$169.00
PandaBoard ES	$182.00	x1	$182.00
Microsoft LifeCam HD-6000	$29.20	x2	$58.40
Logitech C270	$32.00	x1	$32.00
USB Hub	$19.60	x1	$19.60
Belkin Power Pack 2000	$38.60	x2	$77.20
			$538.20

Task Responsibility

Justin Derby (SE)

- Eye tracking algorithm
- Develop Server codebase

Tyler Burnham (SE)

- Eye tracking algorithm
- Develop Client codebase

Arjay Vander Velden (CPR E)

- Middle layer software development

Scott Connell (CPR E)

- Communication framework
- Develop Mobile codebase

Will Bryan (CPR E)

- Embedded Programming / Interfaces

Kris Scott (EE)

- Embedded Programming / Interfaces
- Camera hardware

Existing Market

Tobii ($45,000)

- No hardware adjustments necessary
- System guided calibration
- Lightweight & discrete
- Parallax compensation
- Millisecond data sync

Existing Market

SMI ($30,000)

- Real time & recording
- Audio

Existing Market

MIT Graduate Project ($85.92)

- Tracks one eye
- Not mobile
- Boom

Existing Market

Google Glass ($1,500)

- No implemented eye-tracking
  - Next version might have it for unlocking device
- Not intended for eye-tracking