Reconstruction of the VIP Stadium - IoPT Team's Video Capture System at Bobby Dodd Stadium

ECE4012 Senior Design Project – VP3

VP3, Video Project Team
Project Advisor, Dr. Randal Abler, Dr. Edward Coyle

Sangwon Song, ssong86@gatech.edu

Submitted

2017 May 4

Table of Contents

Executive Summary

In the Internet of Things (IoT) paradigm, many of the objects that surround us will have access to and be accessible from the Internet. It thus provides a platform for communication, discovery and cooperation between these IoT-enabled objects with their environment and potential users. The IoT can thus lead to a dramatic expansion of services and information that are available to any person via the Internet. We are focusing on new services and opportunities that the IoT offer to people, or what we call the Internet of People and Things (IoPT) in this project.

The Internet of People and Things (IoPT) encompasses internet-enabled personal electronics [9] that we carry, drive, that populate our homes and businesses, etc. In the Vertically Integrated Projects (VIP) Stadium IoPT team, we consider the effect of the IoPT on the people involved in and the venues for large events like football games.

The team's current goal is to create an IoPT within Bobby Dodd stadium to enhance the game-day experiences of 55,000 Georgia Tech football fans, as described in Figure 1. The application shows the real time, on-demand information about a football game at Bobby Dodd stadium. To date, this includes videos of plays, game stats, and drive trackers. Although a basic IoPT system is in use (http://estadium.gatech.edu) already, many improvements are needed to make it simpler and more reliable, make it easier for fans to use, allow fans access to more information, and make it easier to modify later.

We will focus in this senior design project on improving the way the current IoPT system gathers and shares video information from the football game. At the moment, as shown in Figure 2, the team uses a system of 4 interconnected devices to capture, encode, and store video of each play in the football game from the video stream that fees the large Panasonic display at the south end of Bobby Dodd stadium. The current system has significant shortcomings:

- It is now 7 years old and has become unreliable, sometime causing loss of important game videos.
- It does now allow the beginning and the end points of videos of plays to be adjusted if after recording the part of the play is missed during the manual recording process.
- It has limitations in the resolution of video inputs it can handle and encoding standards that can be supported at the output.

The expected outcomes of the project are new system hardware and software that overcomes the above limitations. The expected cost of the hardware plus a new enclosure will be $2000.

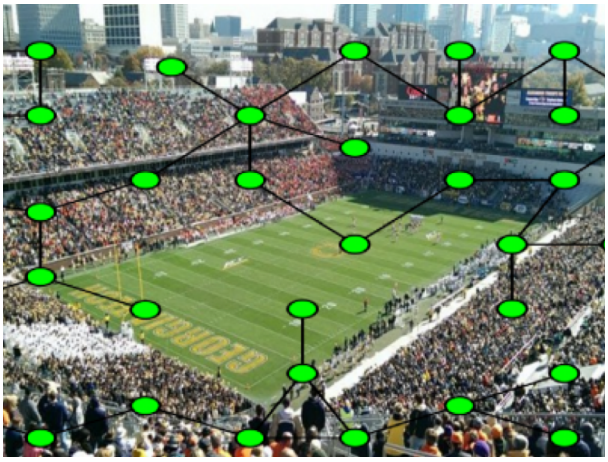


Figure 1. Visualization of the Stadium IoPT deployed in Bobby Dodd Stadium.

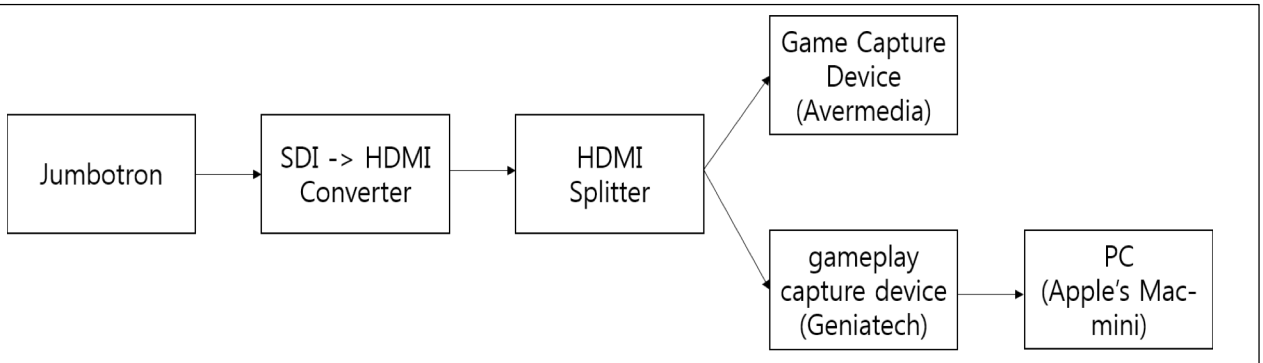


Figure 2. The current streaming system.

Reconstruction of the VIP – Stadium IoPT Team's

Video Capture System at Bobby Dodd Stadium

1. Introduction and Purpose

The VIP – Stadium IoPT team will reconstruct the video capture/streaming system used at the Bobby Dodd Stadium during Georgia Tech's home football games. This project will improve the gameday procedure more efficiently, which is transferring the game play video files to the VIP's main server and making associations with XML play description text files. The users of the "eStadium" web application will have the improved access to the football game's information.

1.1 Objective

The goal of this project is to redesign and improve the current eStadium (previous name of the Stadium IoPT VIP team) system that is deployed in Bobby Dodd stadium, so this will help the gameday procedure more efficiently on the engineer's side. A diagram of the desired system in Figure 3, the VIP – Stadium IoPT team will perform this project with 4 major tasks:

- The team will choose new hardware and develop new software to capture both the game plays and the entire game in a single device. As drawin in Figure 3 and compared with Figure 2, This will help the system reduce the number of the capture devices required in this project from 2 to 1.
- The team will create an interface using PHP and include an additional video monitor to help the team work more efficiently during football games.
- The team will develop new software to enable the start and end time stamps of each play video to be adjusted to optimize the video of the play.
- The team will automate the process of recreating the play videos that are served each time the time stamps are alerted.

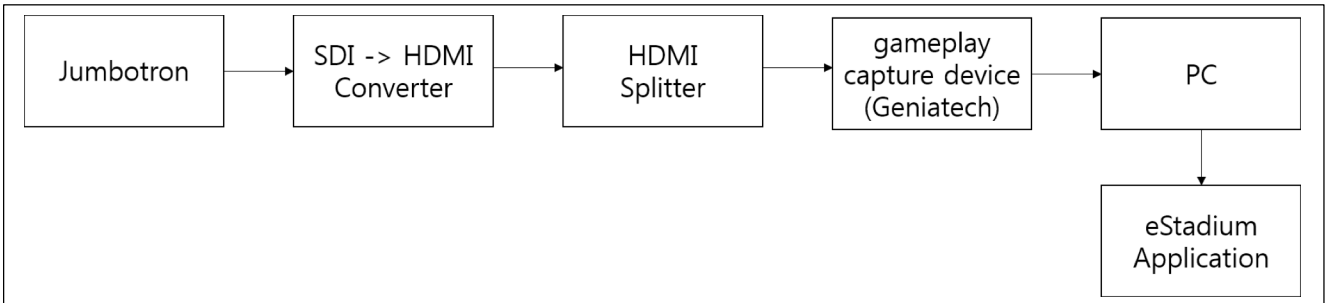


Figure 3. New Video Capture System at Bobby Dodd Stadium

1.2 Motivation

The motivation for this project is to make the gameday procedure simpler and more efficient than the current/old system while also improving the quality of the video files available on the web application, eStadium. As shown in Figure 2, the team currently uses four devices to capture both individual game plays and the entire game. The system has been in use for 6 years and all parts of it have developed technical problems. The team has been replacing the broken parts making plans to reconstruct the entire system. Also, the system is focused on the hardware side. In other words, if there are any technical issues on the hardware, the data will be damaged too. This happens sometimes at the stadium. For example, during the game with Miami University on Oct 1st, 2016, the entire system crashed and stopped the recording, so we could not record video clips for first 20 minutes.

In addition, after each game, the team has to wait in the stadium's operation booth until the backup is done. Thus, if the team can stream directly to the server, the overall system will be more stable, useful and effective. It will reduce the amount of time required on a game day and eliminate some unnecessary works.

1.3 Background

There are numerous streaming devices and related software in the market. For example, in our current system box, the team has been using eyeTV game capture system from Geniatech [6] and a video recording device from Avermedia [7].]. One possible replacement for the recording system is YouTube, which provides a Live Streaming API [1]. For the video capture device, one newer option costs only $200 [3], much less than the eyeTV.

New software that we would develop would include PHP applications that can be used for all of the above devices as long as the software users can access to the streaming video data. However, the software tends to be inaccessible to the software users because of safety. In other words, since we will use a commercial software, and if a user can modify the software data or interface, the company would stop him or her because it is their product. Thus, creating a PHP application accessing the video screen will be needed necessarily.

2. Project Description and Goals

The team will first design a new streaming and capture system for the Stadium – IoPT at Bobby Dodd football stadium. There are two main stages of the project; Research and Design and Implementation. On the Research and Design (R&D) part, a few team members will determine which commercial software and hardware combination is the best, including which criteria should be used to define what is best.

The team is requesting $2000 for the new video capture system. It includes $500 for Apple's Mac mini desktop PC, $300 for a new video capture device, $300 for SDI to HDMI converter, $15 for HDMI splitter, $200 for two monitors, and the rest of $685 for extra in case the parts are broken during the project operation and replaced. If any device from the current system is acceptable, it will be used in the project in order to reduce costs.

On the implementation part, a few team members will create a mechanism to save time stamps of video files to the server data base by using PHP programming.

Once the R&D and implementation phases are completed, the whole team will start working on the actual streaming system so that a team member can stream to the server and save the video files of plays at the same time. During the streaming, the system will automatically generate the full-length football game video on the server. This will require additional work for the team to deal with a dynamically growing file on a server.

The overall goals for the system thus include:

- Finding appropriate hardware/software combination
- Creating a feature that enables saving of recording time stamps of a video file
- Saving a dynamically growing file on a server
- Making and maintaining associations of video clips of plays with play annotations for the eStadium mobile application even when the time stamps are adjusted

3. Technical Specifications & Verification

The portable hardware devices should be small enough to fit in the box that can be easily transported to and from the stadium. The system should require fewer devices than the current system given hardware advances in the last 6 years.

Table 1. Specification of the system

System Components	Specification
Hardware (Game Capture Device)	Game Capture/Streaming Device, including its own software < $300
Software	Import/Export function and file management
Local machine	Apple's Mac mini, running scripts, performing implementation, and other testing
SDI to HDMI converter	Converting the SDI signal to HDMI signal to be recognizable on the local machine
HDMI splitter	Splitting the single HDMI signal to multiple signals to connect with the game capture device and local machine
Dual monitors	Monitoring the recording process and scripts running

4. Design Approach and Details

4.1 Design Approach

This project consists of two major parts; Research and Design (R&D) part, and Implementation. Since we had 4 members in the team, two of them worked in the R&D part to find appropriate hardware and software combinations. As a result, we chose Magewell's USB to HDMI game capture device and VLC media player for the software. They also analyzed ffmpeg video encoding software to save the start and stop record time stamps for the file [5].

The first approach was dealing with a fixed size file on a local machine, the Apple's Mac mini that is one of the main parts of the system. Once they find an appropriate hardware and software

combination, we tested the system to see if they enable us to visualize the feed from SDI signal and to capture video clips.

The second challenge was moving this concept to the server instead of a local machine and checking to see if it is the same process as on local machine or not. During this process, the team investigated the difference and similarities of mp4 files in between the local machine and the server [5]. Since the eStadium application is a web application, the video file format must be web-friendly. As a result, we found mp4 file format has the best web-friendly connection.

The third challenge was to access the growing streaming video file generated by the software the team chooses. The team has to find the way to associate time stamps from the recording process with it. VLC supports various types of video file formats. We found avi file format can be sliced while it is growing dynamically.

For the implementation part, a few team members will work on making the GUI, described in Figure 4 by using PHP. They will decide to use either the JumboTron feed directly from the stadium or the transferred signal feed from the local machine to the eStadium server. If the tasks for both parts are done, the whole team will work on the actual streaming system by implementing the full system in a new enclosure.

In the testing phase, the team will use a personal database, not the main database because this avoids damage to the main server, game data, and the service available to fans.

Lastly, the team will demonstrate the new system at the stadium on the main eStadium server and start streaming, and make associations with the application. Our goal is to test the new system in the Georgia Tech football team's Spring game.

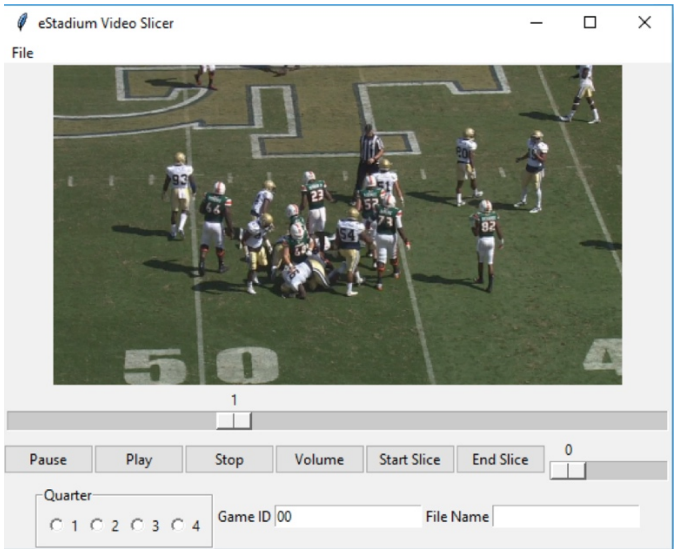


Figure 4. PHP GUI for streaming

4.2 Codes and Standards

The new streaming system uses mp4 and avi files for streaming, and the team will analyze these files to determine how to incorporate time stamps resulting from the play recording steps. ISO/ICE 14496 standard has MPEG-4 code for audiovisual objects [5]. The team will be transferring data and information between a server and a local system. IEC/TA1 can be used for audio, video, data services, and contents. Also, because the system is working based on .mp4, a video file, ITU-T H.264 will be used for advanced video coding for generic visual services.

4.3 Constraints, Alternatives, and Tradeoffs

Constraints

This project will be used next football season, so the time is a critical constraint. There is possibility that the team might not be able to finish the project on time. When the team chooses the hardware, the cost is another constraint. The hardware is a commercial device, so when the team chooses the hardware, they should determine as much as possible beforehand if it is able to work on the system, and if it has exporting and importing functions for video files. In addition, the project is based on the previous system, the hardware should be small enough to fit a portable box.

Also, since the eStadium application is web-based and we wanted to make video slices during recording, the video file format should be web-friendly and editable while it is growing dynamically.

Tradeoffs

The first tradeoff is the destination for the system's video stream. Like the current system, streaming on the hardware is more stable and less likely to lose data because the system stores the data in a physical device. In addition, it is possible that the team keeps this concept of system and make the procedure simpler by transferring data automatically from the beginning of a football game. However, streaming to the server will be more efficient because the whole team does not have to go to the stadium and the backup process is much faster at the end of the game. However, this is easy to lose the information if the server crashes or the network connection is lost because it is a virtual storage.

Another tradeoff is where the system receives the video signal from the stadium. If we can make a new system that streams the game to the server, it would be less redundant work for the team. However, receiving video signal directly from the feed to the stadium's jumbotron may be more stable and feasible.

5. Schedule, Tasks, and Milestones

Appendix A shows project roadblocks. Appendix B shows the project GANTT chart, and appendix C shows the project PERT chart. Table 2 shows specific tasks (building blocks) of the project and the time line (schedule) for these building blocks, with associated milestones.

Table 2. Tasks and Milestones

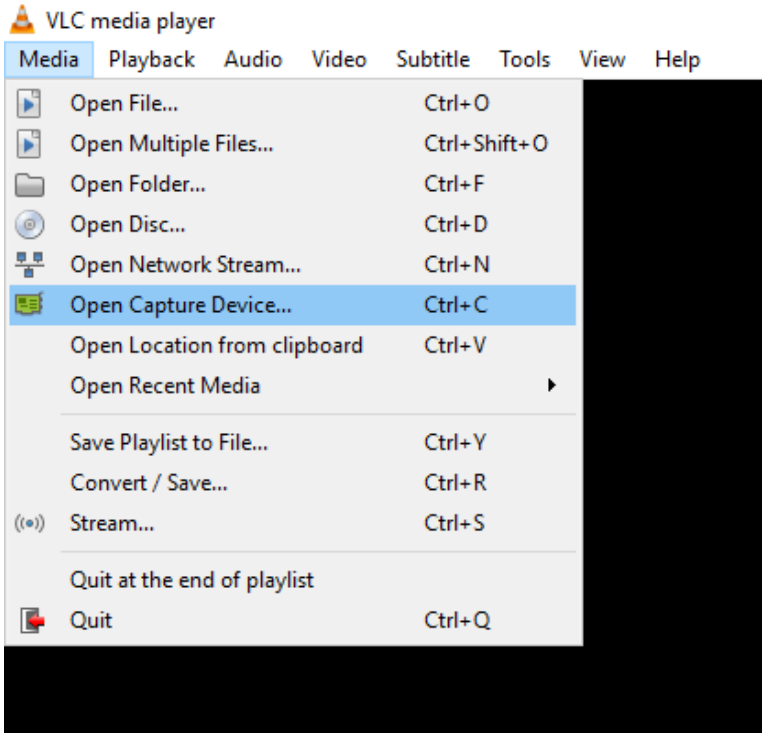
Task Name	Difficulty
Disassemble the current system's box & make a back-up of all current software and data	Low
Conduct research on and acquire a new hardware & software combination for video streaming and capture	Medium, Milestone
Analyze and understand the ffmpeg process and how time stamps can be incorporated in compressed video files	Low
Test our concept by inserting time stamps in a fixed size mp4 file and slice it into several mp4 files	Hard
Determine level of access to a streaming file (growing) for the software found	Hard
Create a mechanism to save time stamps for video clips of plays to database	Hard, Milestone

In real-time, splice the full length video on the server	Hard
Split the growing file on the server and save into a single mp4 file	Hard
Save the time stamps and make a list of them, and create new space for the full-length football game	Medium
Make a web-application using PHP which incorporates start/stop recording button for the Stadium – IoPT team members running the application on a game day	Low
Final Testing	Low
Demonstration period	Low

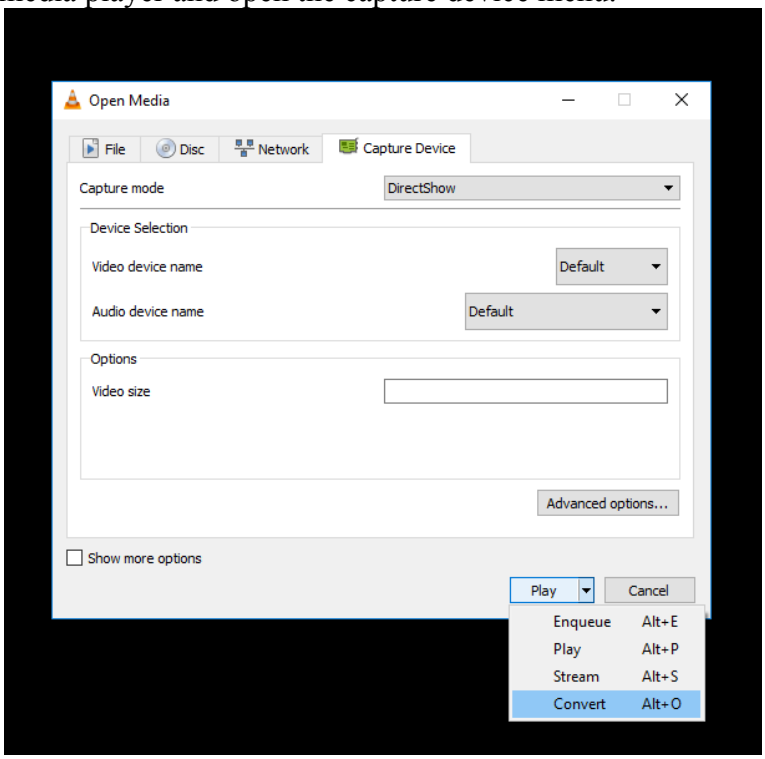
6. Final Project Demonstration

Step by step of the final project demonstration:

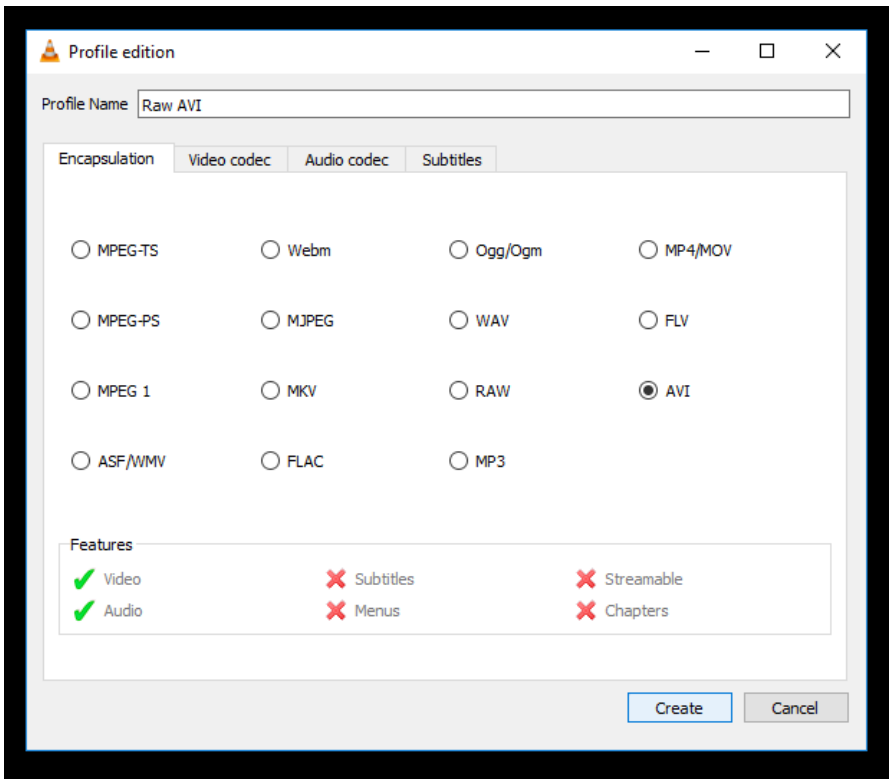
- Part 1: Setting up the local stream.



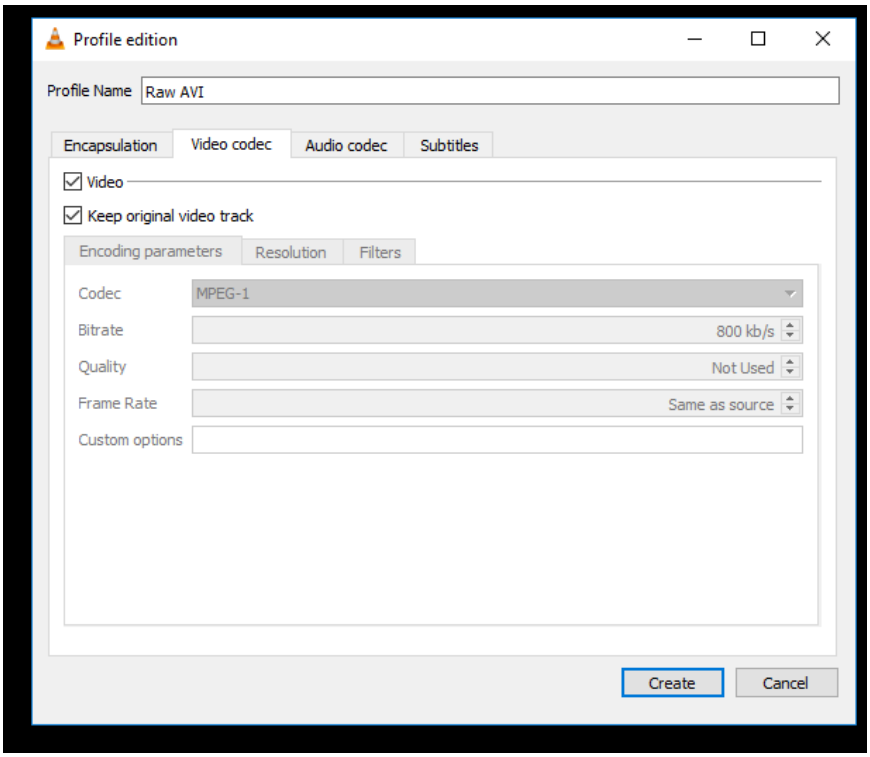
- Open VLC media player and open the capture device menu.

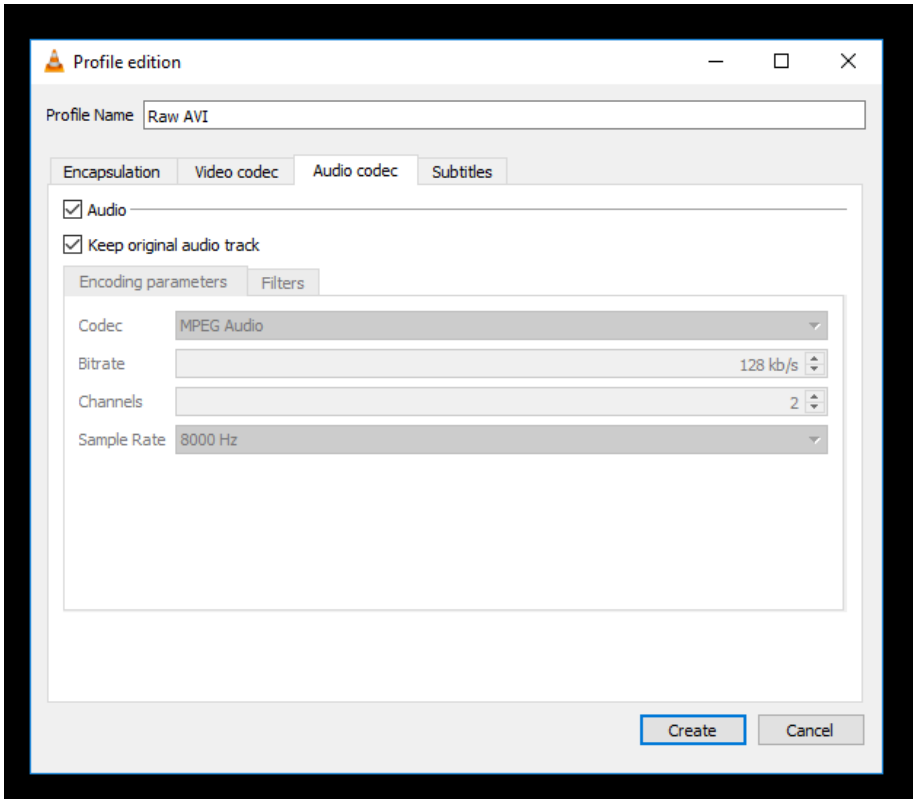


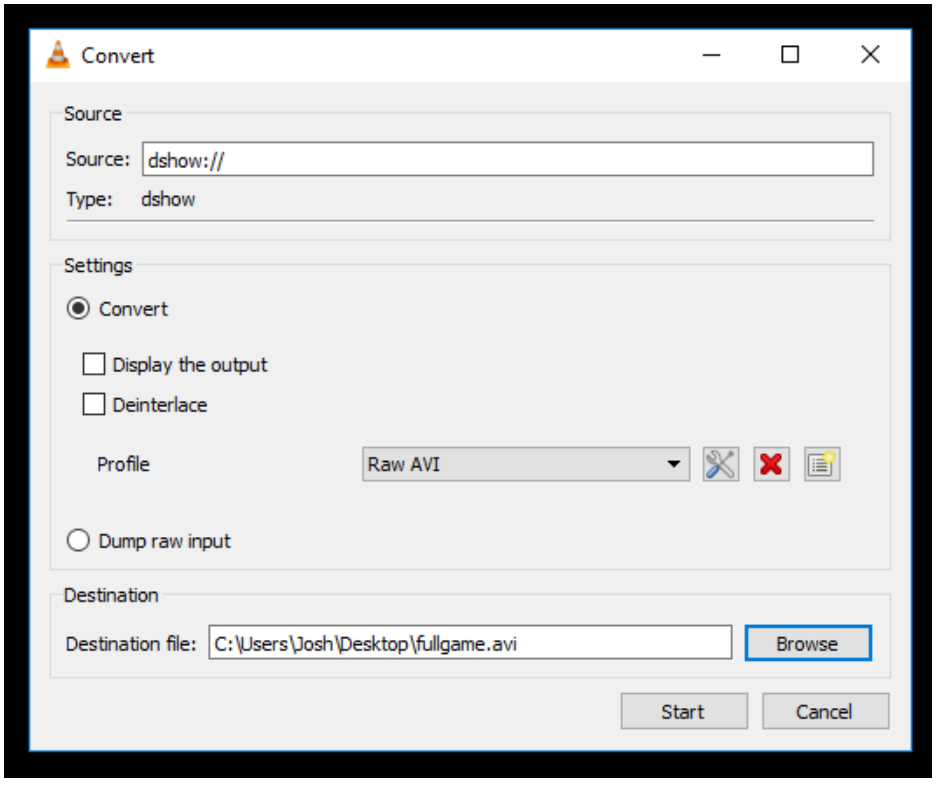
- Choose to convert the video coming in to your capture device into a field.



- Create a new video profile, for our purposes we need AVI encapsulation with raw video.




Profile edition
Profile Name
Raw AVI
Encapsulation
Video codec
Audio codec
Subtitles
Audio
Keep original audio track
Encoding parameters
Filters
Codec
MPEG Audio
Bitrate
128 kb/s
Channels
2
Sample Rate
8000 Hz
Create
Cancel


Convert
Source
Source:
dshow://
Type:
dshow
Settings
Convert
Display the output
Deinterlace
Profile
Raw AVI
Dump raw input
Destination
Destination file:
C:\Users\Josh\Desktop\fullgame.avi
Browse
Start
Cancel

- Save the video to the directory you want to transfer it from.

```
Josh@ALPHA1:/mnt/c/Users/Josh/Documents/vip$ ls
001.mp4  123.avi      __pycache__  slice.php  splice.sh      test.php  test.sh         videoproject  vlc.pyc
035.mp4  please.mp4   README       slice.sh   tempencode.py  test.py   transferAVI.sh  vlc.py
```

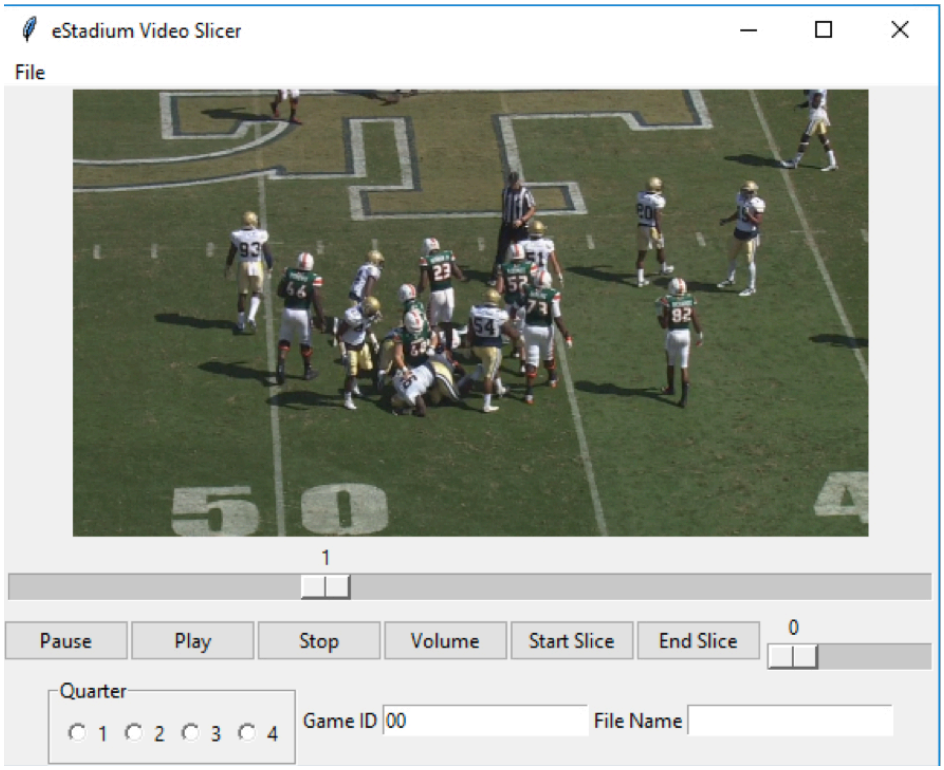
- Go to the directory that you just saved to. The file transferAVI.sh should be in that directory as well, with settings modified accordingly.

```
Josh@ALPHA1:/mnt/c/Users/Josh/Documents/vip$ sudo watch -n 1 "bash transferAVI.sh"
```

- Run this command to run the bash file every second. It will now stream to the server specified in its settings.

```
Josh@ALPHA1:/mnt/c/Users/Josh/Documents/vip/videoproject$ python3 gui.py
```

- Open the GUI application (Currently only works on Windows)



- Your server needs to have slice.php in the php file. Now when we press start slice and end slice the GUI will send commands to that file to slice the video on the server and place it into the correct folder.

7. Marketing and Cost Analysis

7.1 Marketing Analysis

The team needed to find appropriate hardware, preferably for under $300. We chose Magewell's USB to HDMI capture device. Luckily, VLC media player is open source software, so it is free to use. When it comes to selling this system as a product, the team will decide how much the current system will cost. Table 3 shows possible price of the system including all of devices in the box and GUI. The estimated cost of the system is $1755.

Table 3. Price of the New System

Component	Price ($)
Apple's Mac mini	$500
Capture Device – Magewell's USB to HDMI	$300
SDI to HDMI converter	$300
HDMI splitter	$15
LG monitor x 2	$200
PHP GUI	$150
Extra fee (cables, box, etc)	$300
Total	$1755

Figure 5. Magewell's USB to HDMI Game Capture Device

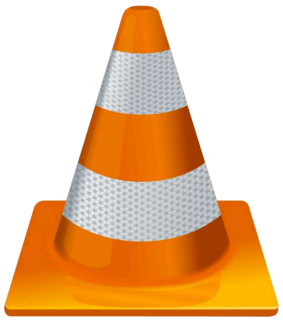


Figure 6. VLC Media Player Logo

7.2 Cost Analysis

The labor costs are estimated using the average hourly contractor of an IoT engineer's salary as $35 per hour [8]. Table 4 shows the summarized cost is $6,650.

Table 4. Cost of Labor

Task	Labor Hours	Labor Cost ($)
Maintaining and updating the database.	40	1400
Attending football games and recording the game	8	280
PHP GUI	40	1400
App Development	100	3500
Maintaining the box	20	70
Total	208	6650

8. Conclusion

As the result of this project, our team tried to run the Python GUI for the final presentation. However, opening the GUI did not work properly because of some libraries of Python had OS-specific limitations. We could not select the streaming buffer file for playing in the GUI. As we tried on multiple operating systems, only Windows could run the GUI and functionality correctly. However, our team successfully transferred the streaming buffer file to a server and made video slices on the server, and updated database with the information. We mostly succeeded the project's tasks and checked the new system can be deployed really soon.

All of the documents and codes that we worked are uploaded to wiki and GitHub's Repository:

- Wiki: https://vip.gatech.edu/wiki/index.php/Video_Project
- GitHub Repository: https://github.gatech.edu/Estadium/VideoProject

9. Leadership Roles

Since I was the only senior doing senior design, I was under so much pressure when I was doing this project. Fortunately, my other teammates helped me a lot with technical knowledges that I did not know well such as Python programming, and Shell scripting. As the team leader, I maintained the system box working properly and tested our codes with extra individual work. I was also leading the team presentations over the semester, so every team members had evenly distributed work every week. For technical work, I worked on finding the appropriate hardware, transferring the streaming buffer file, and updating information to my personal database.

10. References

[1] Google, “YouTube Live Streaming API Overview”[Online]
Available: https://developers.google.com/youtube/v3/live/getting-started

[2] Peter Harrop and Raghu Das, “Internet of People: Technology 2015-2025” [Online].
Available: http://www.idtechex.com/research/reports/internet-of-people-technology-2015-2025-000388.asp?viewopt=showall

[3] High Ground Gaming, “Best Capture Cards of 2016 for PC, Playstation, and Xbox” [Online].
Available: “http://www.highgroundgaming.com/streaming/best-capture-cards/

[4] Elgato, “Game Capture HD: High Definition Game Recorder” [Online].
Available: https://www.elgato.com/en/gaming/gamecapture-hd

[5] MPEG-4, “MPEG-4; Coding of audio-visual objects” [Online].
Available: http://mpeg.chiariglione.org/standards/mpeg-4

[6] Geniatech, “eyeTv 3; TV software for mac” [Online].
Available: https://www.geniatech.eu/eyetv/product/eyetv-3/

[7] Avermedia, “Game Capture HD II” [Online].
Available: http://www.avermedia.com/gaming/product/game_capture/game_capture_hd_ii

[8] Glassdoor, “IOT Engineer – Hourly contractor salary” [Online].
Available: https://www.glassdoor.com/Contractors-Salary/innoPath-Contractor-Salary-E35077.htm?filter.jobTitleFTS=IOT+Engineer

[9] Peter Harrop and Raghu Das, “Internet of People: Technology 2015-2025”[Online],
Available: http://www.idtechex.com/research/reports/internet-of-people-technology-2015-2025-000388.asp?viewopt=showall

Appendix A – Project Roadblocks

Task Name	Duration (week)	Difficulty
Disassemble the current system's box & make a back-up of all current software and data	1	Low
Conduct research on and acquire a new hardware & software combination for video streaming and capture	4	Medium, Milestone
Analyze and understand the ffmpeg process and how time stamps can be incorporated in compressed video files	2	Low
Test our concept by inserting time stamps in a fixed size mp4 file and slice it into several mp4 files	2	Hard
Determine level of access to a streaming file (growing) for the software found	1	Hard
Create a mechanism to save time stamps for video clips of plays to database	2	Hard, Milestone
In real-time, splice the full length video on the server	2	Hard
Split the growing file on the server and save into a single mp4 file	2	Hard

Save the time stamps and make a list of them, and create new space for the full-length football game	2	Medium
Make a web-application using PHP which incorporates start/stop recording button for the Stadium – IoPT team members running the application on a game day	2	Low
Final Testing	1	Low
Demonstration period	1	Low

Appendix B – GANTT Chart

		Project Timeline	Weeks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A		Disassemble the original system box & backup																
B	R&D	Get a new hardware & sotware combo																
C		Analyze ffmpeg file(fixed size)																
D		Put time stamps to a fixed size .mp4 file and slice it into each .mp4 file																
E		Access to a streaming file(growing file) of the software found																
F	Implemntation	Create a mechanism to save timestamps to DB																
G		Create a mechanism to in real time splice the full length video on the server																
H		Create a mechanism to splice the growing file on the server and save into a single .mp4 file																
I		Create a mechanism to edit those timestamps and create the new space for the full-length play																
J		Make a web-application using php which shows start/stop recording button																
K		Final Testing																
L		Demo & Presentation																

Appendix C – PERT chart