Cross-Country Course Elevation Analysis

Final Project Report

SD May 2019 - Team 37

David Kirshenbaum
Andrew Mumm
Connor Smith
Thomas Chambers
Ryan Hilby
Jacob Feldman

Advisor/Client: Dr. Brian Hornbuckle

May 1st, 2019

Table of Contents

Reference Materials and Acknowledgements

List of Figures

List of Definitions

- **LIDAR (also LiDAR, Lidar, or LADAR):** Light detection and ranging. A method of measuring distance in which lasers are aimed at the target, and the return time and wavelength is measured in order to calculate distance to the target.
- **GIS:** Geographic Information Systems. A framework for gathering, analyzing, and viewing data related to Earth including topography, roads, terrain, etc.
- **USGS:** United States Geological Survey

- **XC:** Abbreviation for cross-country
- **Iowa DNR:** Iowa Department of Natural Resources

Acknowledgement

- **Dr. Amy Kaleita:** We would like to thank Dr. Amy Kaleita for lending us the differential GPS equipment that we initially experimented with using in the project.
- **Dr. Bradley Miller:** We would like to thank Dr. Bradley Miller for teaching us more about GPS and topographic data. What we learned from Dr. Miller is greatly beneficial for data analysis.
- **Dr. Yuyu Zhou:** We would like to thank Dr. Yuyu Zhou for sharing his expertise on how to best visualize the elevation profiles we aim to generate.
- **Dr. Brian Hornbuckle:** We would like to thank Dr. Brian Hornbuckle for creating the project and helping us find the resources we need to succeed.

0. Executive Summary

The sport of cross country (XC) has built its reputation on the rough terrain that has challenged its runners over its 100+ year history. Historically speaking, this "rough terrain" was defined by a heavy inclusion of hills in addition to other course elements like varied footings, hurdles, and water crossings. However, there are prominent figures in the cross-country community, notably former Iowa State XC coach Bill Bergan, that have expressed concern about the degradation of the sport's spirit via the loss of hills. Iowa State itself has recently fallen victim to this trend, as it hosted the 2018 Big XII XC Conference Championships on a significantly easier route of its nationally-renowned cross-country course as depicted in Figures 1 and 2.

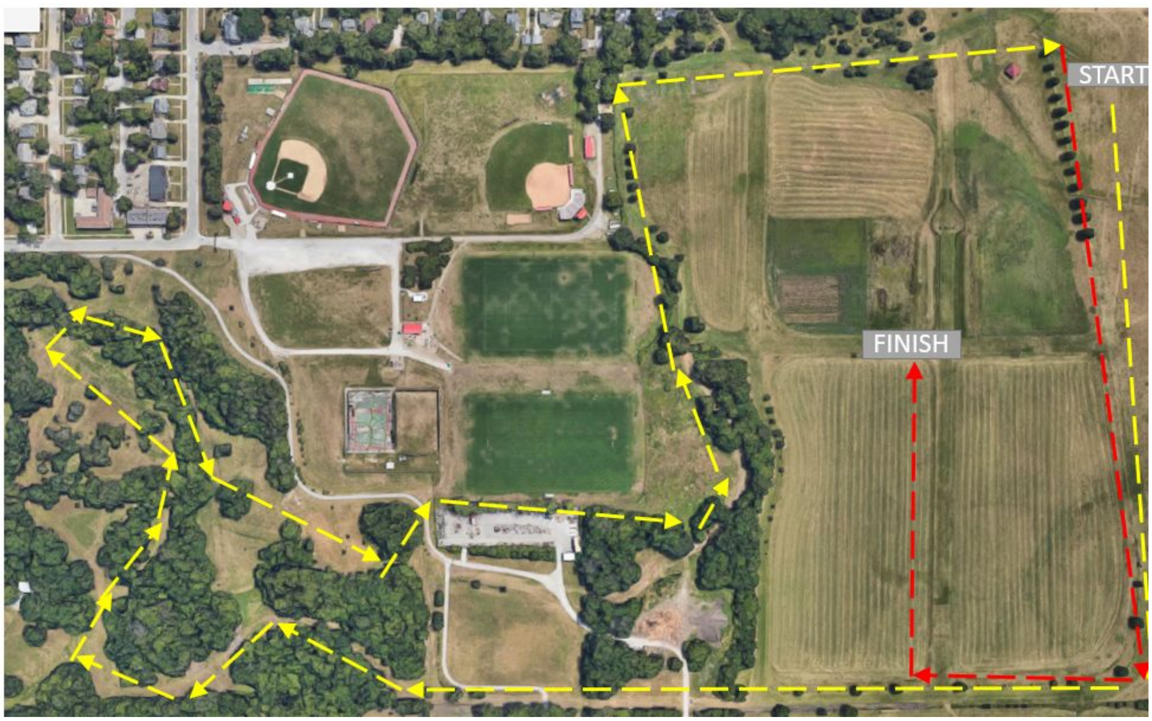


Figure 1- Past ISU XC Course Route - Note the forested, hilly section on the left

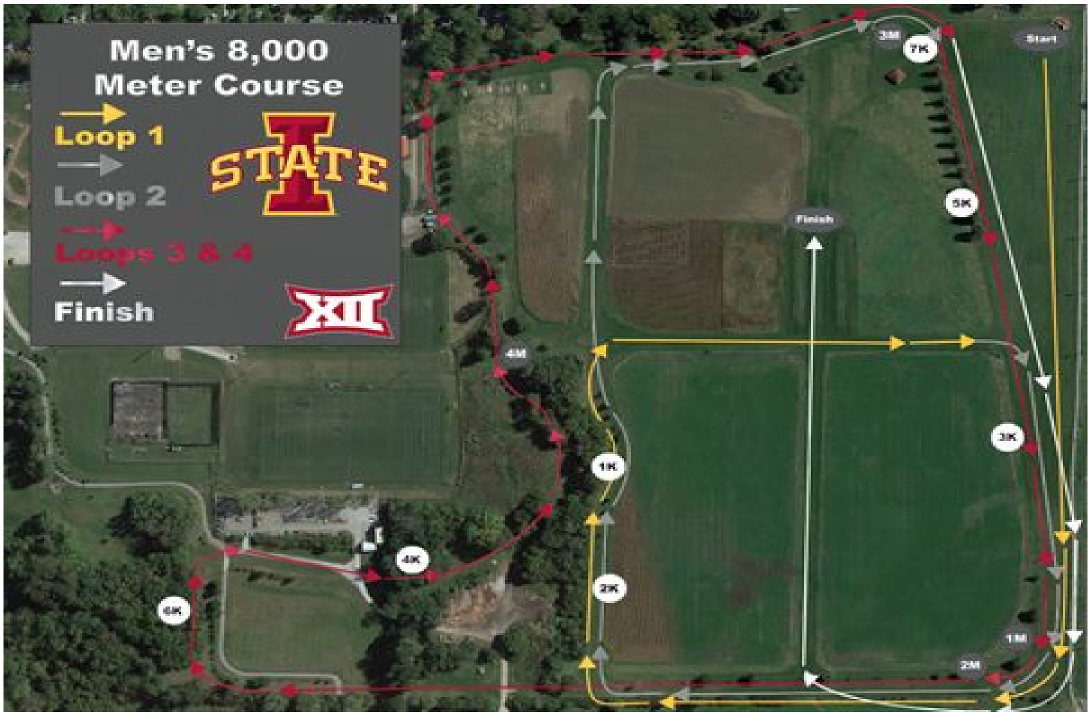


Figure 2 - 2018 Big XII Championship Course - Note how it completely avoids forested hills section featured in the original course in Figure 1 and loops on flat ground instead

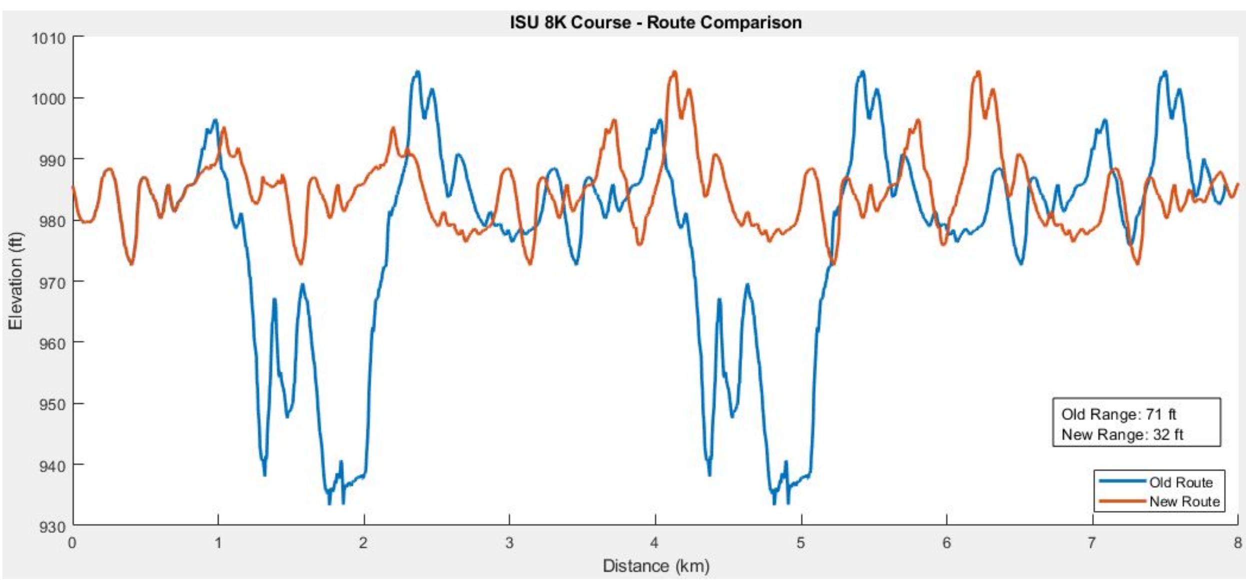


Figure 3 - Elevation profile comparison for ISU course before/after rerouting

It is our belief that we are now in a defining era for cross-country as a sport. Since the outset of our project, we have assumed the worst - that courses are indeed trending towards flatter and less interesting routes. This assumption was the motivating factor in developing our rating system and website with the hope that it will make it easier for course designers, spectators, coaches, and athletes to visualize and comprehend the true nature of their XC courses' difficulties. Armed with this new awareness of course difficulty, we hope that designers will be more inclined to add more difficult features to their course to increase its difficulty rating as point of pride for their XC meet.

Our project's work consisted of three distinct phases.

First, we worked to determine the viability of the different elevation data sources that we had available. This was accomplished by conducting several "ground truth" studies where both the accuracy and precision were compared for phone-based GPS, high-end Garmin handheld GPS, Google Maps, and the Iowa DNR's LIDAR database against the USGS' geodetic points that serve as defined reference points for elevation around the United States.

After making our selection of our datasource (Iowa DNR LIDAR), the software team designed and set up a MYSQL database using AWS RDS for hosting. Once the database was created, we wrote code to automate uploading elevation data to it. We then set up an AWS Lambda function written in python to connect to the database and perform cross country course analysis. After the database and back end were set up we built an Angular web app for users to interact with. The web app allows users to draw cross country courses with the use of the Google Maps API. The drawing tool collects coordinate data from the drawn course and sends it to the Lambda function for processing. The connection between Angular and Lambda is done using the AWS SDK. After Lambda finishes processing the data a response is sent back to the front end where users will be able to view to results that are displayed with Chart.js. While the software development team was working on developing the aforementioned server and website, the ground truth team constructed a rating system for XC courses after consulting XC coaches, biomechanics faculty, and medical journals. We also contacted dozens of Iowa high school XC coaches to get XC course maps that we could trace with our web tool and then use in the calibration of our relative difficulty scoring system. In these communications, we also specifically sought out course maps that showed a change in race routing over different terrain from one year to another that could be used to test our project's overarching hypothesis.

Finally, we used the web tool to trace the 6 sets of courses we were able to find that had changed their routes in recent history. We traced the routes before and after the change and recorded the differences in our tool's scoring of them. We then used statistical hypothesis testing to look for a significant trend of the sample altered course set getting less difficult over time. At a significance level of 0.1, we were unable to conclude that courses are growing less difficult. The process flow of our problem approach is illustrated below in Figure 4.

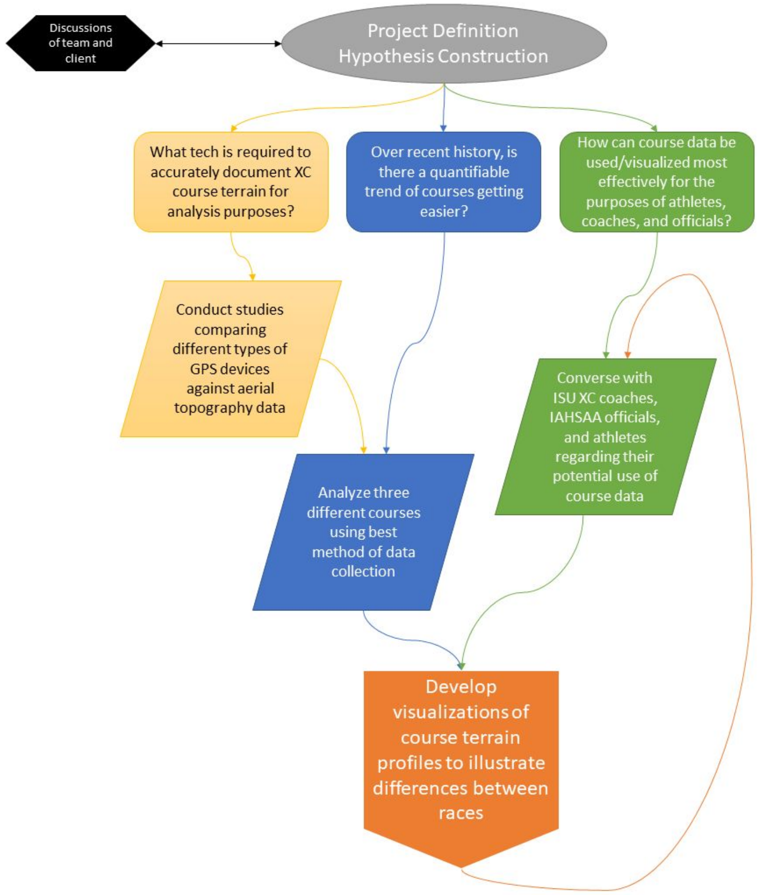


Figure 4 - Process Flow Diagram

1. Requirements

1.1 Functional Requirements

- The initial ground truth validation studies need to provide definitive information regarding the accuracies and, subsequently, the viability of using topographic data sources available that are also feasible and scalable to a wider deployment.
    - **Met -** LIDAR was experimentally shown to be an accurate source of topographic data. While it was proven to be feasible and scalable for the entire state of Iowa, it is inherently not scalable for use outside Iowa.
- The web app tool needs to be able to use LIDAR files as its data source.
    - **Met -** We currently use LIDAR as our data source to provide elevation to the users.
- The web app must easily allow users to provide the course XY data themselves.

- **Partially met** - Users can quickly draw courses from a satellite perspective using Google Maps API.
- The web app must be able to run classification algorithms on the elevation profiles and classify hill-like topography in to subclassifications like big hills and rolling hills.
    - **Met -** The app finds and classifies big climbs and rolling hills, then displays them in the report.
- The elevation profiles and their derived metrics must be presented in a visually appealing manner and in an easy-to-interpret format.
    - **Partially met -** User interface is up to our own personal standards, but we were unable to survey users' experiences as we had originally intended.

1.2 Use-Cases

We foresee our app being usable by cross country coaches to better design racing strategies for their athletes based on the information presented by our website. We also foresee athletes and spectators desiring to use our application out of sheer curiosity, as the analytics we are able to produce for courses go well beyond any historical precedent for high school XC meets.

The information presented on our website will be of particular interest to users who are encountering a course for the first time. Typically, athletes need to walk the race course before the race itself in order to understand what course elements are present like big hills or rolling hills sections as well as where they are located. Our website could be used to replace these walkthroughs, as we are able to report that same information to users.

1.3 Non-Functional Requirements

- Server will match x, y coordinates in a course to elevation within 10 seconds.
    - **Not Met -** Our original goal of 10 seconds turned out to be harder to achieve than we originally thought. A typical 8km (the standard for NCAA cross country courses) course takes 32 seconds to complete, 31 seconds of which are spent retrieving points from the database. As such, to meet this requirement, we would need to change our database design heavily; for instance, we could split up counties into multiple tables. This solution comes with a host of its own problems, and we didn't have time to modify our database that heavily.
- 90% of surveyed users must not report issues/confusion after using app
    - **Not Met -** We ran out of time to conduct a survey so we do not have the data to back this up.
- Elevation data source must be consistently within 3 m of the USGS official elevation
    - **Met -** LIDAR elevations were found to be within 3 m of the USGS reported elevation 95% of the time.

- 90% of users report that they comprehend the meaning of the various metrics produced by the classification
    - **Not Met -** We ran out of time to conduct a survey so we do not have the data to back this up.
- 90% of users report that the scorecards are presented in visually appealing and easily interpretable format
    - **Not Met -** We ran out of time to conduct a survey so we do not have the data to back this up.
- Quantitative ratings of 0-10 course score must be within ±1.5 points of average trial runners' qualitative rankings of courses.
    - **Not Met -** Due to the late snow and rain during the second semester we were not able to have runners run the course to test our difficulty rating.

2. System Design & Development

2.1 Design Plan

In order to determine whether or not cross country courses have indeed been trending towards flatter routes in recent years, we first had to face the challenge of selecting a source of elevation data that could be trusted to give us the most valid answer to our project's overarching question. This challenge was overcome by designing a series of four different physical site surveys where the different technologies available to us were compared and contrasted, pointing us in the direction of LIDAR data recently compiled by the Iowa DNR for the entire state of Iowa.

The next phase of the design plan was deciding how "hilliness" could be best quantified. We turned to faculty from Iowa State as well as from other universities to seek counsel on this question, but we ultimately took guidance from a series of medical journal articles that recommend using a metric of energy cost to capture the physical challenge presented to runners by inclines and declines.

At this point of the project, we decided that we wanted to go beyond our original goal of simply answering the question of courses possibly getting less hilly. Instead of doing these calculations manually for only a few courses, the idea of constructing a website that could be used not just to compare courses of changed routes but to compare *any* course against another using a single unified difficulty rating system was hatched. With the recommendations of ISU Coach Bill Bergan, we decided to design software that could also capture information like hardest/easiest miles of a course, how much of the course is rolling hills, and the amount of "big" hills contained in a course.

With this new goal in mind, the software engineers set out to determine how to interface with the LIDAR files available and how to structure the frontend and backend computing responsibilities for the most expeditious computation of all ratings. We used AWS lambda to run Python scripts that made all course calculation, a MySQL database storing billions of coordinates for elevation in the state of Iowa, and an Angular front-end running chart.js for our visuals and Google Maps for user input.

It was only once the LIDAR data retrieval portion of the software was integrated with the Python rating scripts that we were able to answer the title question of the project. We contacted over 50 XC coaches in the state of Iowa looking for maps they had of courses that had changed from one year to another. By analyzing routes with our rating software both before and after they had been rerouted, we would be able to look for trends of courses getting less hilly and, accordingly, less difficult.

2.2 Design Objectives

Our design plan was assembled in accordance with four key design objectives that we outlined for ourselves at the beginning of this project back in EE 491.

First, we needed to make an educated choice as to which source of elevation data would be most useful for our application of documenting XC course topography. This choice needed to be made with factors of ease of use, legitimacy, and availability in mind. After concluding that LIDAR was the most promising source of ground truth, we still needed to experimentally test the legitimacy of the Iowa DNR LIDAR database. The DNR's website makes no guarantees as to the accuracy or precision of their LIDAR database, so the responsibility of testing its accuracy and precision fell on our shoulders.

Second, we aimed to develop a course rating system that would be simplistic enough to make intuitive sense to a user of the web app but also rigorous enough to have a real physical meaning that represents a fair assessment of a course's difficulty associated with its elevation patterns. In order to give even more insights as to a course's features for a runner that may be new to a course,

Third, we wanted to create a straightforward website for users to view the elevation profile, route, rating, and other metrics of an inputted course. These services also needed to be supported by a robust backend computing framework that could quickly process the requests for elevation data from our massive LIDAR files.

Finally, we looked to conclusively answer the question of cross country courses becoming less hilly by using our software tool.

2.3 System Constraints

While the free access to LIDAR data for the state of Iowa has proven to be invaluable to the success of our project's goals, it also presents a constraint - our product currently cannot be used to trace any race courses outside Iowa. While all of our Python scripts for analysis will work perfectly fine on any elevation profile that is passed through them, the actual retrieval structure for the Iowa LIDAR data had to be specifically designed for the Iowa DNR's database. While it would be very easy to allow for the upload of GPS-based elevation data for other states, our primary choice of LIDAR makes our software much more valuable for use inside Iowa than outside Iowa.

Possibly the hardest constraint we had to deal with is the raw size of the lidar files we used. Each county had somewhere around 225 million points, which, even after repeated attempts to compress that data, results in ~7-8Gb per county. Which leads to our next constraint: database size. The free tier of AWS RDS only allows for 20Gb of data which only allows for 2, maybe 3 counties worth of data. As a result of this constraint, we chose to not upload all counties in Iowa as we originally hoped; instead, we uploaded 7 counties that we had the most courses for. These counties are: Story, Linn, Polk, Johnson, Black Hawk, Dubuque, and Marion. We needed 65 Gb of AWS RDS storage to store these counties, which costs us $2.45 a month.

The technological savviness of the software's users is also a constraint to the project. It is key that the user interface is as simplistic as possible in order to lower the intimidation of the entry barrier for cross-country officials at every tier of the sport including older, small town athletic directors. Since there is currently no similar software being used by our target users today for this purpose, it is only reasonable to expect the users to be apprehensive about giving it a try.

2.4 Design Trade-Offs

We have decided to base our server off 3 meter resolution LIDAR files for our data set as opposed to the higher resolution 1 meter files that are also available. We are content with this choice because a 1 meter resolution would require 9 times as much file storage, and using a worse resolution can reduce the accuracy of our statistics.

We have chosen to only allow users to draw courses using our online web tool instead of allowing them to use their phones as XY sources from which we could pull the relevant elevation data from the LIDAR database. While our ground truth studies showed that the XY data is trustworthy enough, the convenience of not having to physically walk the entire course of interest outweighs the inconvenience of having to draw the course route from a satellite perspective.

We will have all calculations done server side because with large file sizes, pushing the data and analysis to a user's machine will not be feasible. The client side will be used for forming the graphs because we want graphs to be interactive, however it will require more work to be done on the user's machine for possible input lag.

Our app will have a weakness where it cannot tell terrain and obstacles on the traced course. This means if your course goes through sand, this would increase the courses difficulty, but our app does not take that into account. We currently only plan on giving statistics on elevation.

Instead of creating a web app to rate the hilliness of a cross country course, we could instead analyze several courses and report that analysis in a paper, along with detailed descriptions of how to apply the same analysis to other courses. However, the interactivity and reusability of a web app makes it the clearly more useful solution. Also, the intended user for this app, cross country coaches, would not be as interested in a research paper as they would an easy to understand and interactive visualization of the same data.

2.5 Architectural Diagram

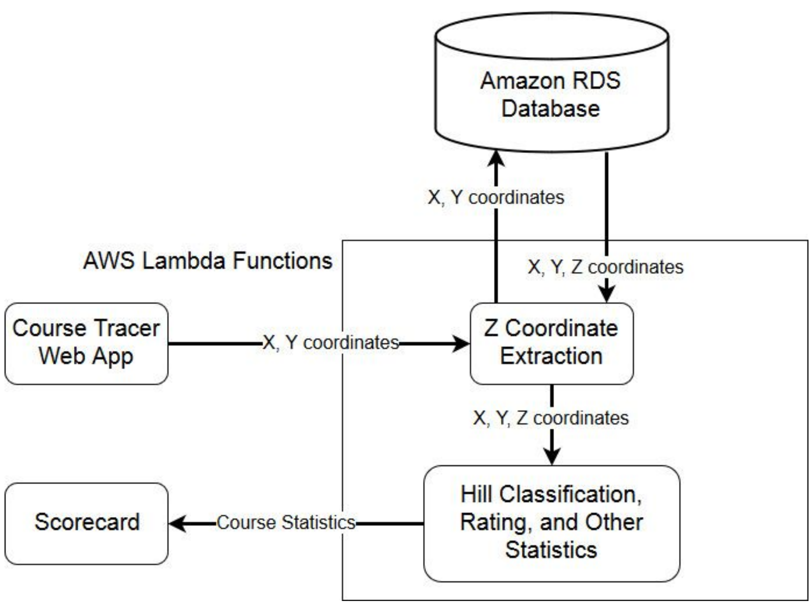


Figure 5 - Architectural System Diagram

2.6 Module, Constraint, and Interface Descriptions

2.6.1 Establishment of Ground Truth & Data Source Selection

Several factors were considered when deciding which devices would be evaluated as potential sources of elevation data for XC courses. The device/source needed to be relatively affordable and easy to use so that it could be deployed to high schools around the state for on-site surveys of course elevation. It would also need to be sufficiently accurate with its latitude and longitude (XY) bearings to properly document a surveyed route. It would also need to produce elevation data sufficiently precise to properly record all possible slopes on an XC courses from slight undulations of rolling hills to steep inclines and declines.

With these considerations in mind, we selected the following devices/sources for study:

- Google Pixel smartphone
- Motorola G4 smartphone
- Garmin Montana 680t
- Iowa DNR LIDAR database

All surveys were scheduled and conducted while monitoring the dilution of precision (DOP) conditions for the GPS/GLONASS constellations above Ames utilizing Trimble's GNSS Planning webtool (Trimble). Surveys were only conducted when DOP was at a daily minimum in order to yield the most accurate position computations.

The first survey was conducted at Lee Park in Ames. The three handheld GPS units were walked in an "L" shaped path to examine their XY accuracy when compared to the straight path. All three units were found to have an RMS deviation from the defined straight line of less than less than 5 feet, giving us confidence that we could trust the devices to produce adequately smooth XY paths. The National Collegiate Athletic Association (NCAA) has a set standard for the width of cross-country courses at 4 meters wide, so the 5 foot first standard deviation of these XY values would still be acceptable had we would have gone on to use them to trace course routes (Seewald). If the course was walked in the middle of the route, the traced line would reliably fall within the confines of the route's width boundaries.

The second and third surveys were conducted at the Iowa State Cross Country Course. First, all three handheld devices were walked around both the old 8 km course route that included the hilly forested section and the new route which omitted it. The LIDAR database was also used as an elevation source of comparison by tracing the route on Google Maps and then uploading those XYs to return the corresponding elevations in the LIDAR database. The elevations recorded at all these points were then plotted and compared as seen in Figure 6.

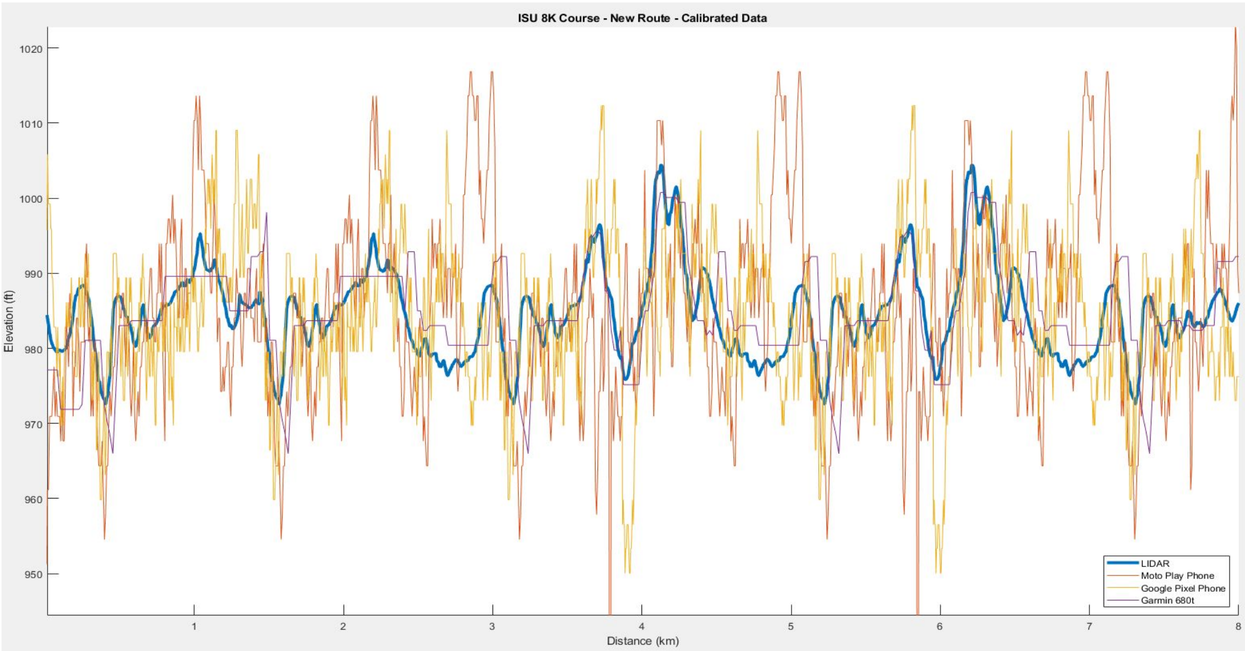


Figure 6 - Raw comparison of elevation datasources on Iowa State XC Course

Both of the phones (red and orange lines in Figure 6) were found to be very noisy sources of elevation, making the Garmin data and the LIDAR data the more attractive choices.

To better determine the precision of the four possible sources, all the devices were walked in a straight line for 100 meters both forward and backward beginning at an United States Geological Survey (USGS) geodetic point. Such points are physical markers scattered around the country where the USGS has verified the elevation. Hypothetically, a perfect elevation data source would produce an elevation profile from this survey that is mirrored due to the fact that the same terrain is being traversed twice. However, both of the phones once again failed to produce this mirrored quality we were looking for as shown in Figure 7.

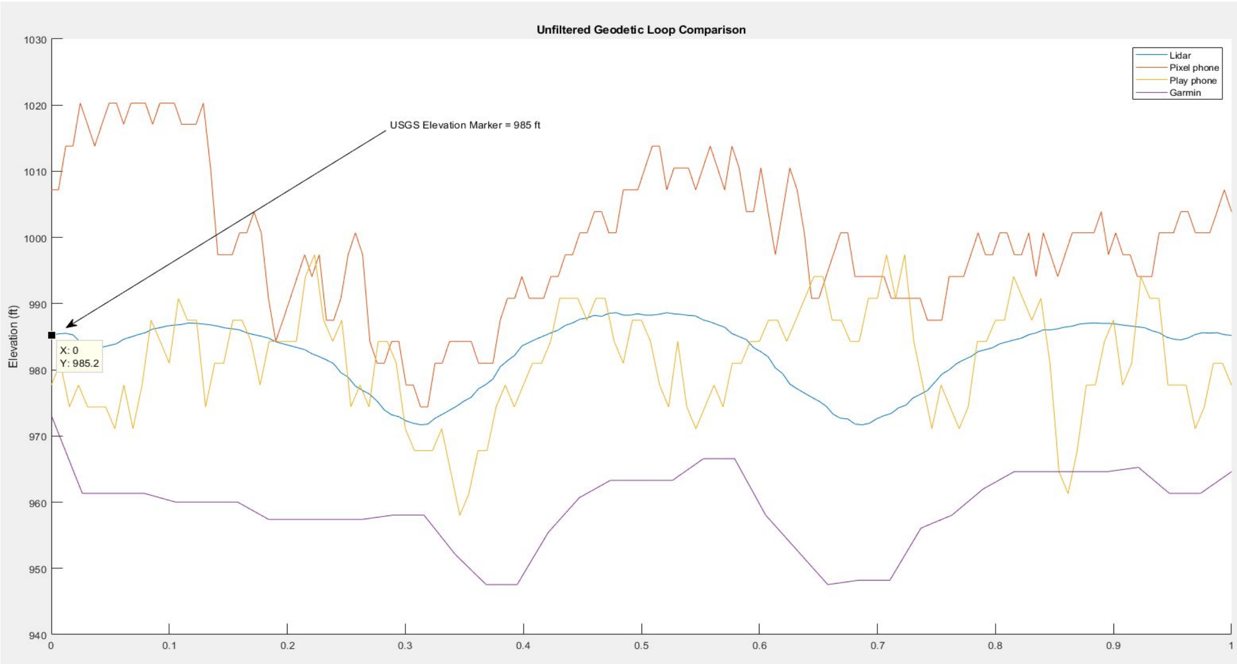


Figure 7 - Geodetic point precision survey results for each elevation source

As noted above in Figure 7, the USGS geodetic point's official elevation corresponded perfectly with the LIDAR elevation value at that point, while the Garmin demonstrated a systemic accuracy issue across the length of the survey. This inaccuracy, along with its weak precision, pushed us to select the LIDAR database as our datasource.

As a final test of the validity of the LIDAR data, we once again used the USGS geodetic point network and extracted elevations at 20 different points from all regions in Iowa where a geodetic point existed. We then compared the difference in elevations and found that the standard deviation of the variance between the two elevation sets was less than 2 meters, adding to our confidence that LIDAR was our best option as a source of "ground truth" for the project.

After the LIDAR selection had been made, the ground truth team continued to experiment with applying various filters to the phone-based elevation data with the hope of reducing its noise. The bulk of this experimentation was with MATLAB-implemented Savitzky-Golay filters of varying window lengths, the results of which are seen in Figure 8.

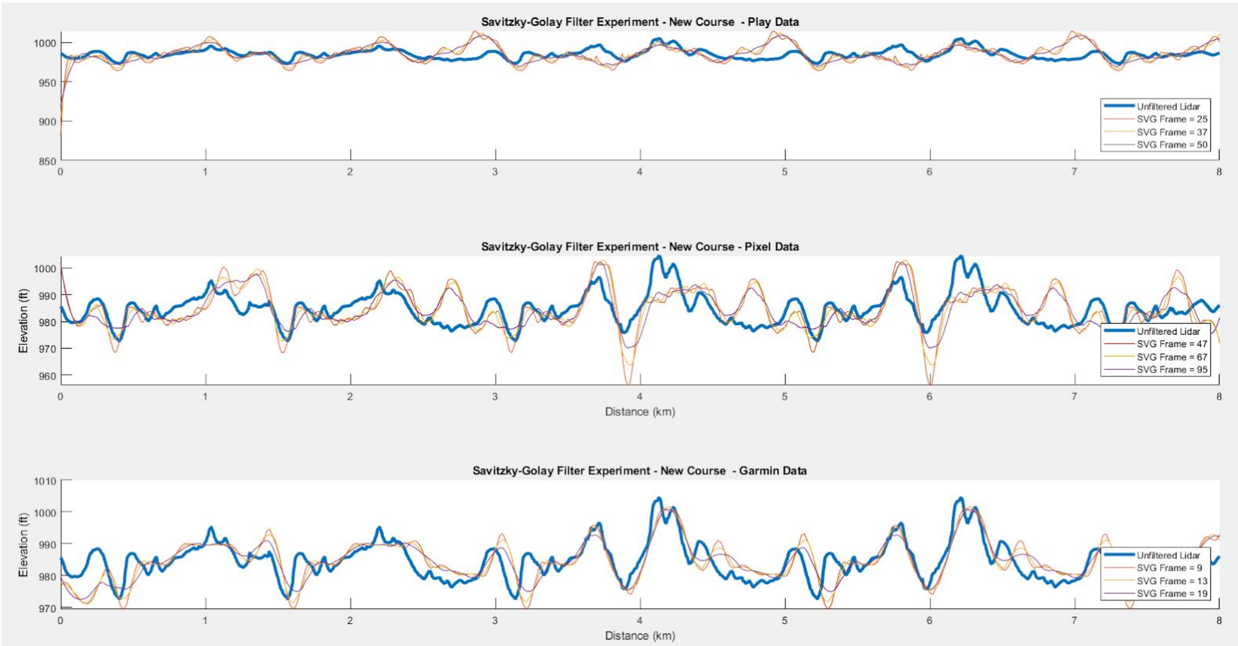


Figure 8 - SVG filtering frame width experimentation on ISU XC course elevation datasets

Each different datasource was found to have its own unique optimal frame width of elevation points. While the results of this filtering removed nearly all of the noise from the phone-based elevation data, in situ comparison of reported large hills did not correlate with what we observed while physically conducting the survey (see Figure 9), giving us little confidence that, even with filtering, the phone-based GPSs could ever be used to reliably capture the most subtle terrain features like rolling hills and smooth inclines/declines.

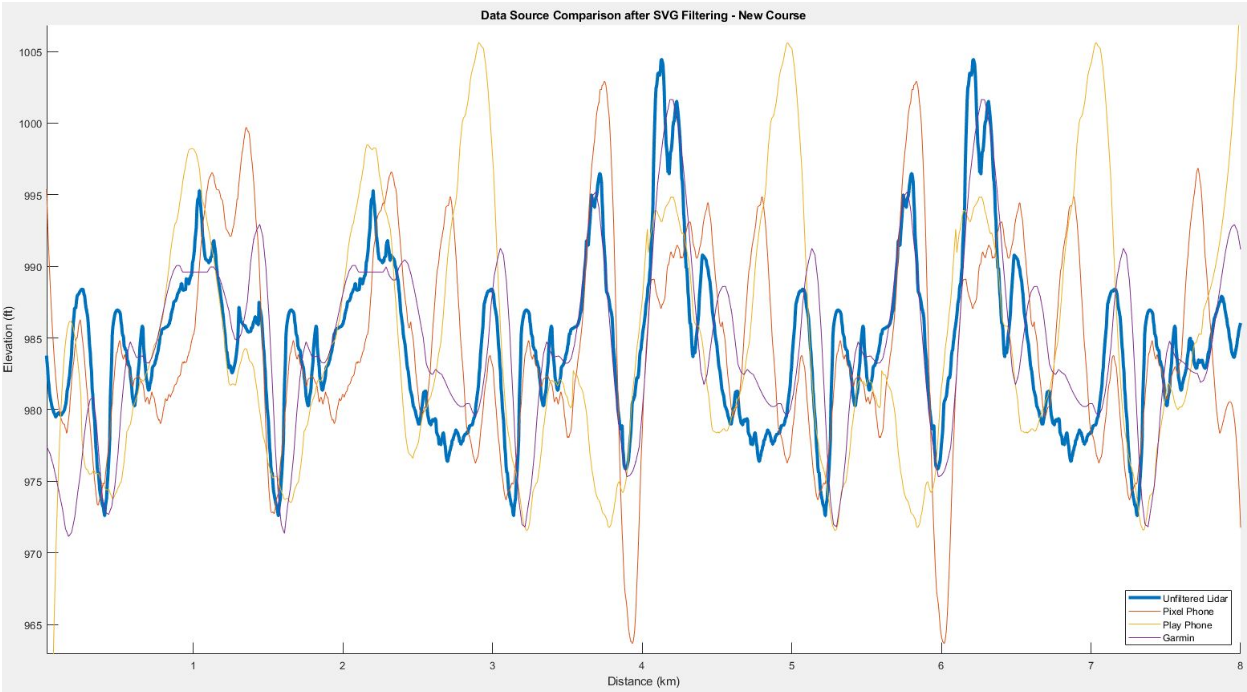


Figure 9 - Filtered handheld GPS data comparison vs. unfiltered LIDAR data

2.6.2 Rating System

Instead of of judging courses purely based on their amount of total elevation climb or average slope, we decided that we wanted to pursue a more rigorous analytical approach to what constitutes "difficulty" in an XC course. To answer this question, the team met with national champion former XC coach Bill Bergan to get his take on the matter, and we also reached out to Iowa High School Athletic Association and NCAA officials along with other faculty for their expertise. After much discussion, we decided to focus our efforts on the *energy cost* of running on various slopes. This energy cost model is the standard method for medical studies that analyze exertion when walking or running on inclines and declines. A University of Calgary-led study published in 2016 experimentally found that the energy cost (Cr) of running in terms of J·kg-1·m-1 has the following relationship with slope where $\boldsymbol{i}$ is the slope between two adjacent data points (Vernillo).

$$Cr = 155.4i^5 - 30.4i^4 - 43.3i^3 + 46.3i^2 + 19.5i + 3.6$$

Equation 1: Energy Cost as a function of slope

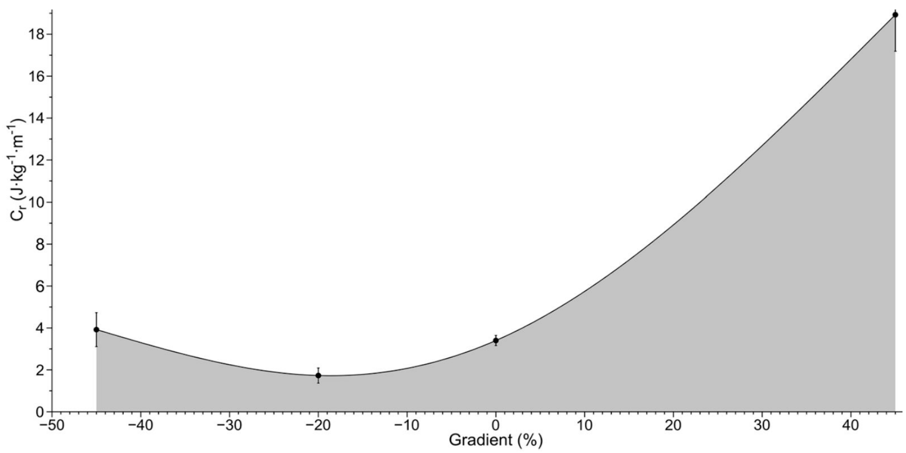


Figure 10 - Plot of Energy Cost vs. Slope Gradient

MATLAB was used to put this energy cost calculation in to practice during the prototyping phase of the project. Working with the LIDAR set of a few thousand elevation data points from a traced course, we calculated the slopes between each point. For each slope, the aforementioned energy cost is computed. The calculation is repeated for all consecutive points in the elevation dataset, yielding a summed cumulative energy cost of running that course. Additionally, a distance weighting is applied to individual costs to adjust for the added difficulty of a hill that is located farther in to a race than one that is right at the start (adjusting for fatigue factors). The justification for such distance weighting was presented in a 1991 article by J.C. Brueckner in the European Journal of Applied Physiology (Brueckner). These energy cost sums are then divided by their respective distances so that a 1 mile route's difficulty can be fairly compared to that of a 10 mile route.

As part of our desire to build an easy-to-understand rating system, we designed a 0 - 10 rating system that is associated with the energy cost sum for a given course. This scale was calibrated by equating the energy cost of running a perfectly flat course with no hills as a "0" on the scale. We then conducted a survey of 30 different cross country race courses across all regions of the state of Iowa and calculated their cost ratings. The average of these ratings was then computed and that value equated to a rating of "5" to denote a course of average difficulty. Accordingly, all courses below and above that specific difficulty energy cost value were mapped to indices on either side of "5" with equal spacing. All the course energy costs were then plotted using Microsoft Excel and a linear best fit line was generated to serve as the mapping function from energy costs to our goal of a 0 - 10 scale. This resulting trendline is the equation used by the webtool to compute 0-10 difficulty ratings for new courses evaluated on the web tool.

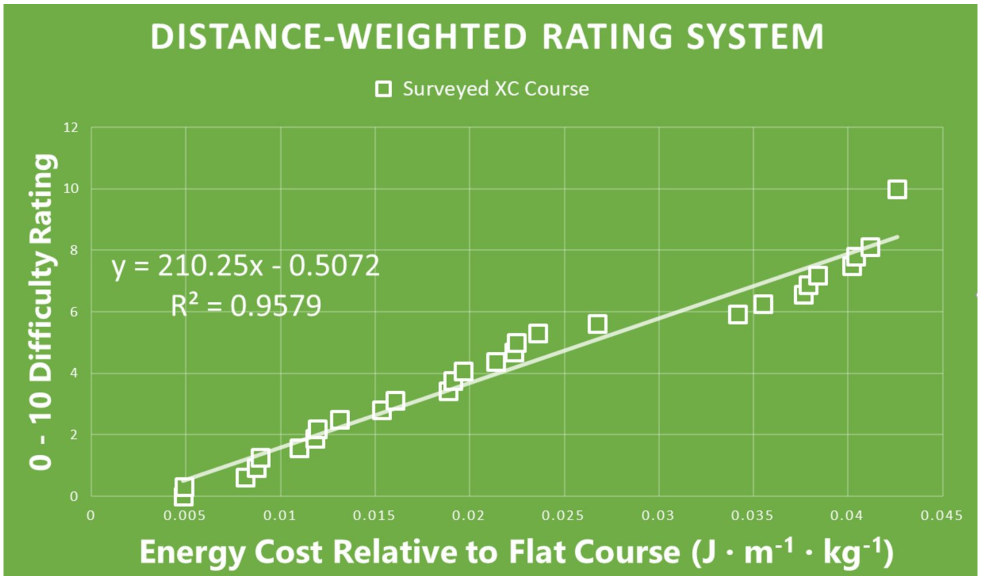


Figure 11 - 0-10 Rating System Calibration

We made the conscious design choice to not limit ratings to a maximum of 10. If a user were to draw the route up Mount Everest, the rating system would obviously return a value higher than 10. However, so long as a user draws a course that would reasonably be run with topography typical of Iowa, the difficulty rating would hover around a rating of 10. If this system were to be applied to more rugged topographies not found in Iowa, this survey-based calibration of the mapping equation would need to be repeated to more accurately sample the cross country courses of that region to determine a new average difficulty baseline.

While we believe this topography-based 0-10 rating system itself will be helpful for course designers to better understand the difficulty of their courses, we realized that we could detect other types of trends within the cross country courses processed by our tool that could be of particular interest to the XC runners and spectators.

First, we allowed for the segmentation of courses so individual miles (or kilometers) could be analyzed on an individual basis. This mile-by-mile difficulty breakdown is a common way for runners to describe a course's difficulty, so this feature will allow for runners and coaches to understand where the most difficult sections of the race are and plan their racing strategies accordingly.

The thresholds set for defining big hills was drawn from the team's personal experience of running. No research pointed to a conclusive set of thresholds to distinguish a normal and challenging hill. The thresholds were set to a minimum uphill length of 50 m and average grade of at least 8%. We decided on 50 m since we felt overcoming a challenging hill would be long enough. In a hypothetical situation, if a runner would run up this hill, it would be expected they would put in a reasonable effort to run up the hill. We also referenced the Iowa State University XC course to help establish this baseline since we knew there were at least 3 big hills from our own experiences.

Rolling hills can be understood as regions where the elevation oscillates to a certain degree. Like big hills, there wasn't a place to decisively define it numerically. The detection algorithm was calibrated on different areas/paths to suitably classify sections of courses. This process of calibration is pictured below, where a binary system was used when testing different parameters in the rolling hill detection to visualize the results. Where the square wave is high, the system is identifying rolling hills. Using this approach, the rolling hills script was tweaked and calibrated to measure the portion of the course composed of rolling hills.

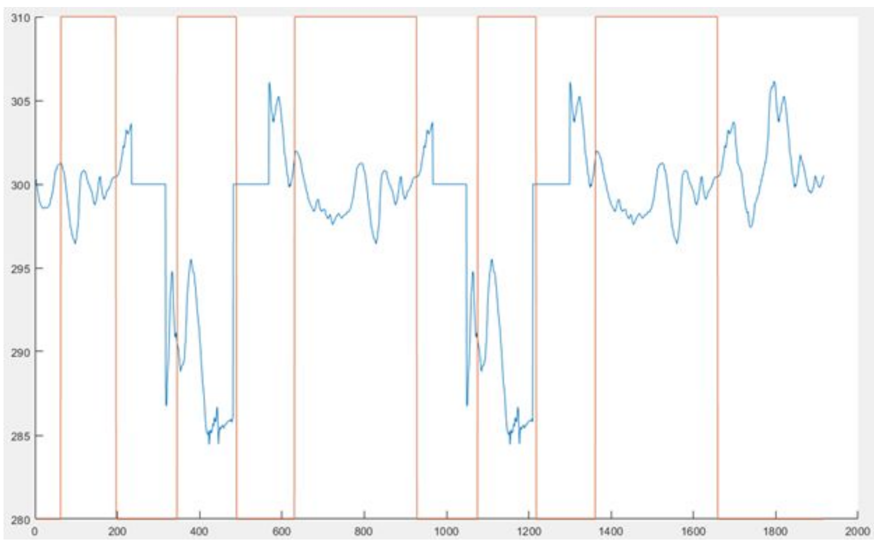


Figure 12 - Rolling Hills Testing and Calibration Example

All elements of the rating system were prototyped in MATLAB before their conversion to Python scripts for final implementation on our cloud server.

2.6.3 Web Tool

Following our analysis of different data sources' precision of elevation measurements, we have decided to design our application to work with the bare-earth model LIDAR data set from the Iowa Department of Natural Resources. This LIDAR model of the state of Iowa serves as the elevation ground truth by the application. The LIDAR data itself will be stored in a central database to be queried by the application on a county-by-county basis.

In order to input their course, users will map the course with the drawing tool which utilizes the Google Maps API (see Figure 13). The web app sends the longitude and latitude coordinates to the AWS Lambda function through the AWS SDK to find the elevation of each point by querying the database. The points are in a non-uniform distribution with varying distances between them. However, they are normally in the range of 2.5 to 4 meters. The Lambda function performs a series of analyses on the data it has collected and computes the interval between each point that it has elevation data for. After completing its analyses, the Lambda function sends the report back to the client app to be displayed to the user (see Figure 5). The report is displayed in a series of graphs using Chart.js. There is also metadata about the course along the right side of the user interface. The user interface displayed has 3 tabs: a tab for the elevation vs distance graph, one for the difficulty vs distance graph, and one for the course that they drew in the first step (see Figure 14).

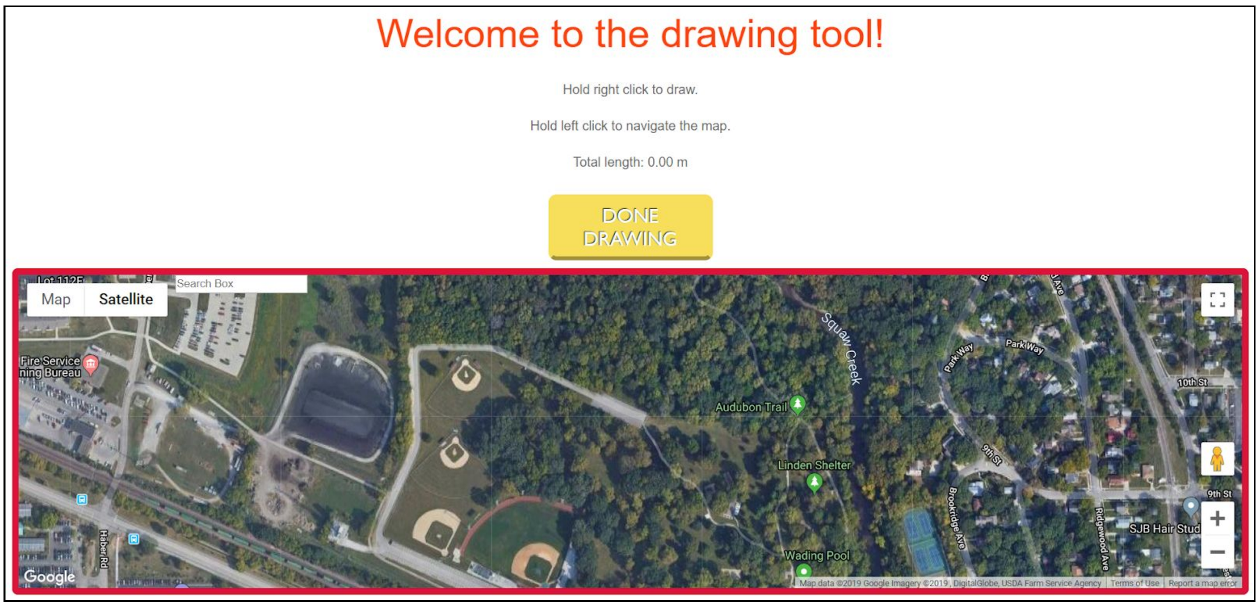
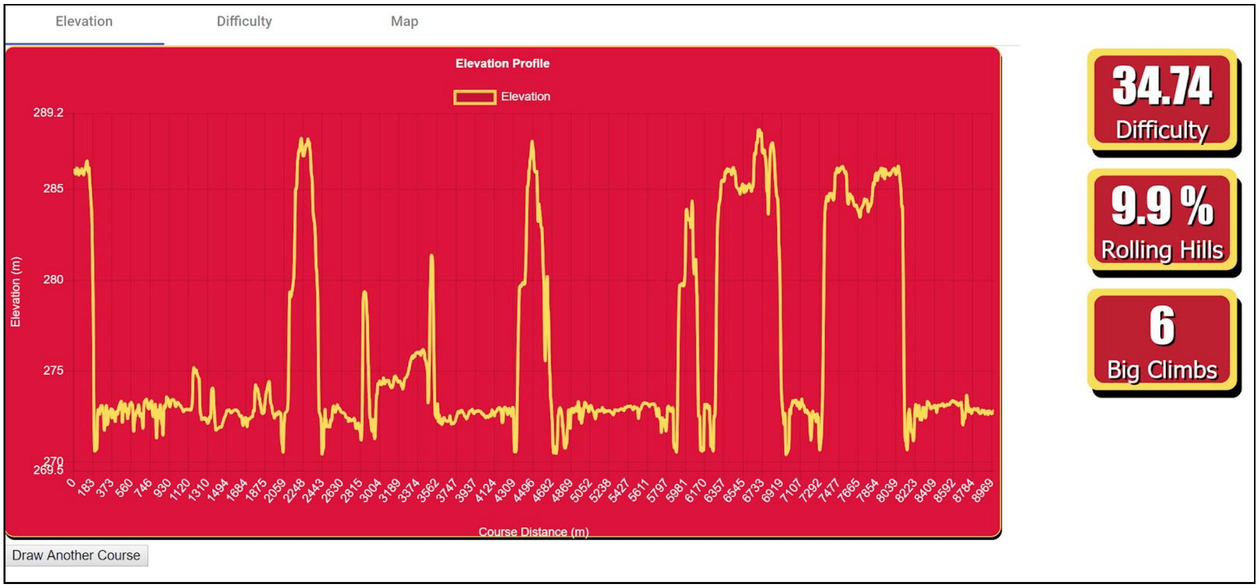


Figure 13 - Screenshot of the drawing tool



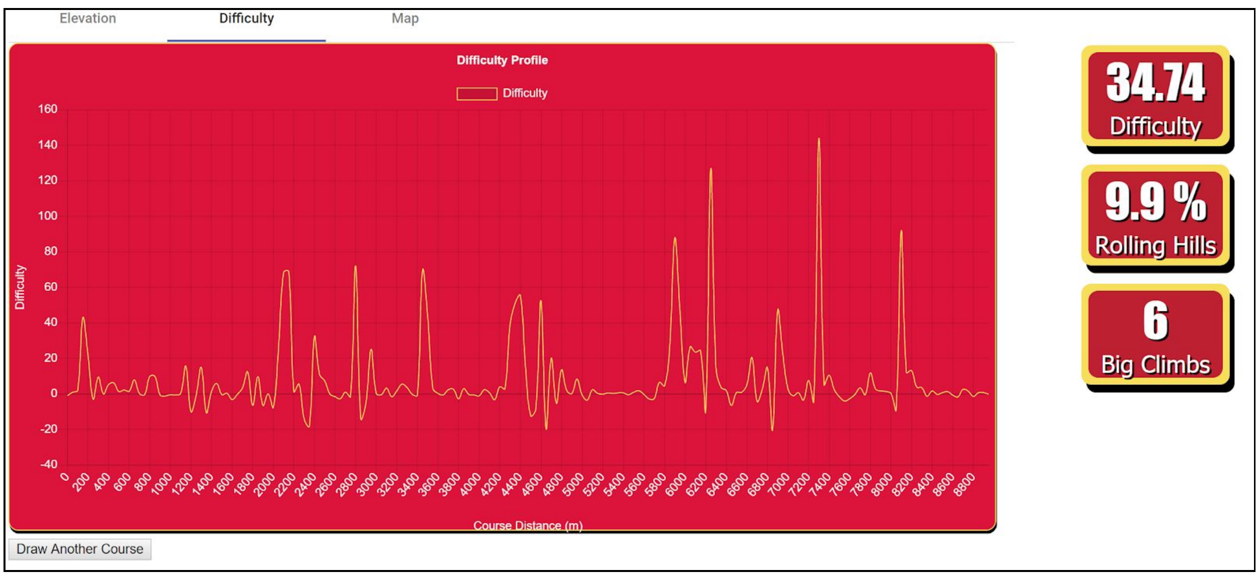


Figure 14 - Screenshots of the scorecard's interface

3. Implementation

3.1 Implementation Diagram

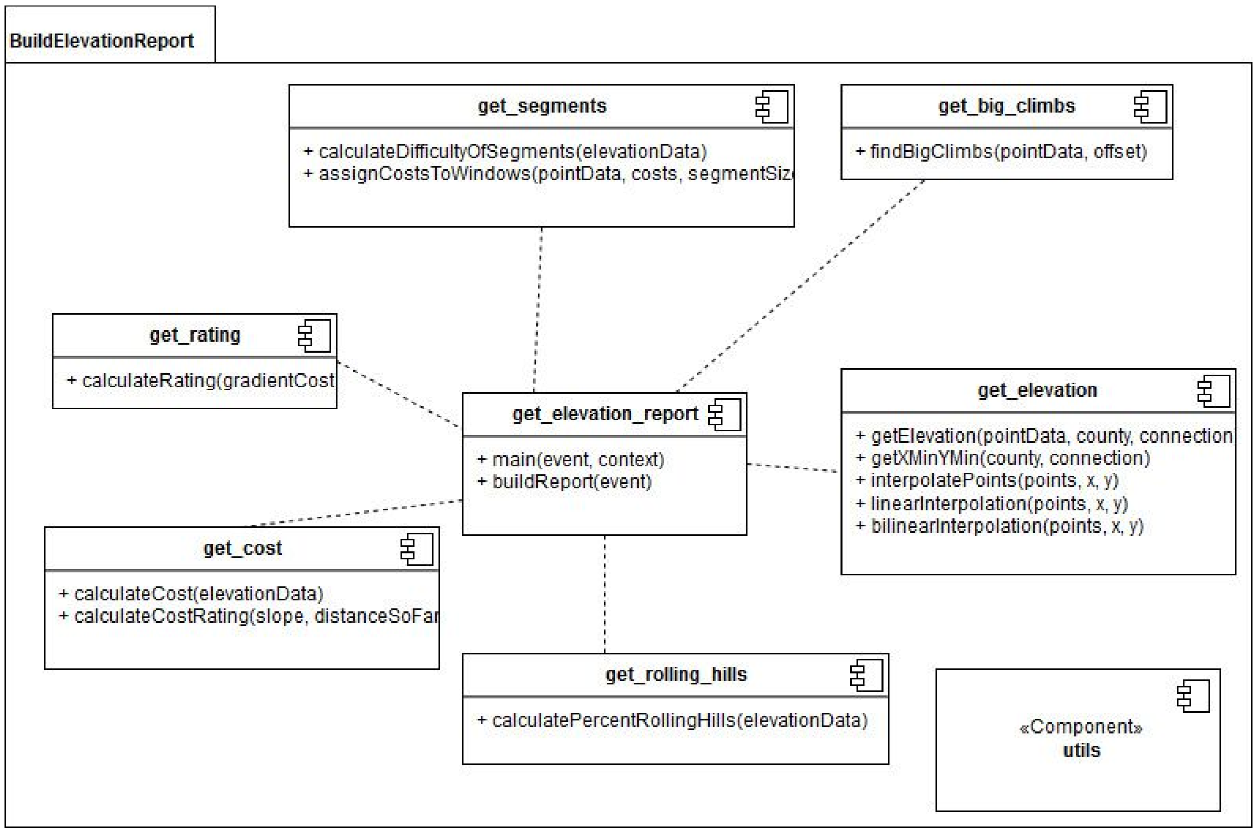


Figure 15 - Implementation Diagram

Technologies and libraries used for the front end include Angular 7, a typescript framework, Webpack, which compiles the Angular code into JS, Karma, an unit and integration testing framework, and the Google Maps API to allow the user to see a satellite map of the course they want to draw. The code was written in Typescript and JSON.

For the backend of the project, the project heavily relies on Amazon Web Services. AWS Lambda was used to perform on demand calculations when the user requested a course analysis. AWS Lambda contains the big hill classifier, rolling hill classifier, and energy cost functions which were written in Python. A MySQL relational database built in an Amazon RDS database was used to store the LIDAR elevation points and coordinates. To ensure communication between the functions and the database, the PyMySQL library was used.

3.2 Rationale for Technology & Software Choices

For the front end of our project represented by a XC course tracing app built for a web browser, we decided to write it in Typescript using the Angular 7 framework. Angular is a popular front end framework and combined with the Karma testing framework, the web app can be easily tested by unit and integration testing.

To reduce long waiting times during a course analysis, Amazon AWS was the best answer since our team could manage the data and improve the performance of our queries against the database.

3.3 Applicable Standards and Best Practices

In our evaluation of the Iowa DNR's LIDAR datasets, we abided by the American Society for Photogrammetry and Remote Sensing's vertical accuracy validation testing standards (Dharmapuri).

From our research, there is little previous academic work on the topic of cross-country course topography, so our project team was responsible for constructing many of our own standards for evaluating courses. We worked with former Iowa State coach Bill Bergan along with officials from the Iowa High School Athletic Association and NCAA and coaches from across the state of Iowa.

To ensure the web tool was working before conducting the analysis, we originally planned a month for testing after Spring Break in case our project ran behind schedule. With this target built into our plans, this ensured our project finished on time.

4. Testing, Validation, and Evaluation

4.1 Unit Testing

Beginning testing occurred with individual scripts written to identify and quantify features within elevations profiles. Big hills, rolling hills, and energy cost/difficulty were written and tested independently. The approach to detecting and quantifying hills was to first get a script capable of achieving expected functionality with a standard, generic number scale before calibrating it to the elevation data it was to be run using. By eliminating the scale factor, it allowed aspects of the code to be isolated and experimented with, and that let conceptual features be fleshed out that would be necessary to deliver consistent, expected results. Calibration was accomplished using real test cases, running collected elevation data with the scripts and observing if the results matched expectations. This was also an iterative process where during calibration it was discovered that additional code needed to be added to deal with unexpected situations.

4.2 Interface Testing

As the project progressed we found that all of our time was devoted to making sure that our software tool worked end to end manually. We needed to build up the infrastructure of our project so that we could be sure we had a working implementation in the end. This left us without anytime to implement the testing frameworks and utilities that we had originally planned on. If we had more time for the project, at this point in the project, then we would have begun to set up a scaffold for testing all of the components of the project. The project is currently in a working state so it would be the perfect time to add testing to make sure the work we have done stays functioning, and we would be able to have the confidence that the project would remain stable as it scales.

We had planned on a couple of methods for testing our project. For the front end we were going to use Karma to test our angular app. Karma is a testing suite built for Angular. With Karma we could write unit and integration tests with ease to be able to ensure the front end of the project performs as expected. To test the Lambda back end we would need to write unit tests for python and supply the scripts with dummy data. We had already began perform manual testing of the back end with dummy data to ensure that the scripts ran without errors, but we would still want to incorporate unit tests to verify that the exact output is what we expect it to be. Lastly, the database upload scripts are written in Java, which would be testable with J-unit tests. We have reached a steady enough state in the project where the architecture is not changing in as much or as often as it was in the early stages. While there was not previously a good opportunity to implement these testing methods before, going forward it would now be appropriate.

4.3 System Integration Testing

The original scripts for elevation analysis were written and tested in MATLAB. In the MATLAB environment, these scripts were integrated together so that the entire classification process could be called at once with a set of elevation data. During this integration, the code was also made more efficient by incorporating the results of the scripts in to one another. For example, the script for detecting big climbs could have that data excluded from the rolling hills detection and prevent overlaps.

When the scripts were needed to be implemented in the backend to be used within the web app, Mumm worked with the Data Analysis team to complete this process. The code was converted from MATLAB to Python by the respective coders and then reviewed side-by-side with Mumm. The code’s competency was confirmed, tweaked for readability, errors were fixed, and then everything was compiled to verify the code could be ran without problems.

4.4 User-Level Testing

One important area for testing was user interface. Our service depended on allowing a person to enter a course that is multiple miles long, in an accurate and convenient way. There are multiple features included in the system to facilitate this process. The Google Maps API includes functions to search and zoom to quickly move to regions and find locations. An included feature of our own was the ability to backstep through the traced course and erase mistakes.

4.5 Validation and Verification

Elevation data from our different sources were methodically compared to determine the best source to be paired with the software's output. The Lidar data was already processed by Iowa's DNR and made available in a format that our team could use. The GPS data to compare it with was collected in the field by two team members. Multiple factors were taken in consideration when doing the GPS surveys. The surveys were exclusively conducted on sunny days, at optimal times when the alignment of the GNSS constellation allows for the lowest dilution of precision (DOP) value. (Trimble) All sources were normalized to one distance index, allowing the data sources to be compared on a waypoint-by-waypoint basis. At each point in this index, the root mean square (RMS) disparity between each datasource reported distances between consecutive points was calculated. Our target for the RMS disparity between values was less than 5%.

In the developed web app, the course rating score system was calibrated by using a range of cross country courses provided to the team in different regions of the state. By using different portions of the state, the different types of geography could be taken in to consideration and give a clear view of the range. During this process, various components of the courses were validated.

In verifying the results of courses traced in the developed tool, several components were validated. The courses provided all were classified by lengths, which could be compared to the length listed by the web app. If the length the app listed was within 2% of its stated length, the course provided and its measurement were considered accurate. Additionally, the elevation profiles were verified and checked for anomalies - to ensure that the courses drawn by the webapp user didn't drift off-course to a radically different elevation.

4.6 Evaluation

A collection of courses were used and processed using our web app, with the scoring algorithms that were written being used to collect information on the courses. The web app printed out the information for the courses, of which we used the rolling hills, big climbs, and energy costs in our analysis. This data was collected while testing the app and calibrating the difficulty rating displayed.

A subset of the courses tested were those that had been replaced or rerouted by a school, allowing those courses to be compared directly. Unfortunately, there were only six courses that could be directly compared. In conducting our analysis we used the t-statistic hypothesis testing, which is designed to be used with a small sample size. With that approach, we could not conclude that there was a trend with the courses provided to us. Even if we removed one outlier where a course got substantially harder, there wasn't a clear trend from the information we had available. That said, with a small sample size of 6, our study wasn't conclusive either way.

5. Project and Risk Management

5.1 Tasks Decomposition and Roles/Responsibilities

Our team of 6 was composed of 2 Electrical Engineers, 1 Computer Engineer and 3 Software Engineers, with the software bias being appropriate for the focus of this project.The team split into 2 main groups for the majority of this project, as detailed in the below table. One team focused on building the core software systems that were used in the course drawing service, while the other focused on the process of obtaining and processing data to make that software system useful. The three members building the software system were naturally Software Engineers, with the engineers in other disciplines using their skills in signal analysis and troubleshooting to handle data processing.

Software Development	Data Analysis
Andrew Mumm David Kirshenbaum Jacob Feldman	Connor Smith Ryan Hilby Thomas Chambers

5.2 Project Schedule

The first half of our project schedule was dedicated to collecting elevation data through various sources to understand the limitations and usefulness of each source. The two main methods that were researched were LIDAR and GPS, which the team split focus between. With our growing understanding of elevation data sources, we began exploring options for conducting elevation analysis. The first semester concluded with a thorough comparison of elevation sources, and a roadmap for the web app that would be created in the second semester.

The second semester was full of work on the software development and collecting courses to do analysis on. Key components of the project were split up and worked on concurrently. Database management, frontend development, the backend framework, and scripts for processing elevation data had work begin in October. We had a working prototype at the end of the first semester, but decided to change our back-end so at the beginning of the second semester we began work on the back-end in AWS. Mid-march we had the website fully working again with some bugs. In April, full integration and testing took place.

The original plan for our work and how our schedule actually played out are pictured below. While we accomplished our major tasks, some portions experienced delays and others ended up cut. We were satisfied with the results, however, as the items cut from our suggested schedule didn't have a significant impact on our project and were part of brainstorming of things to do to keep the team occupied depending on the speed of progress.

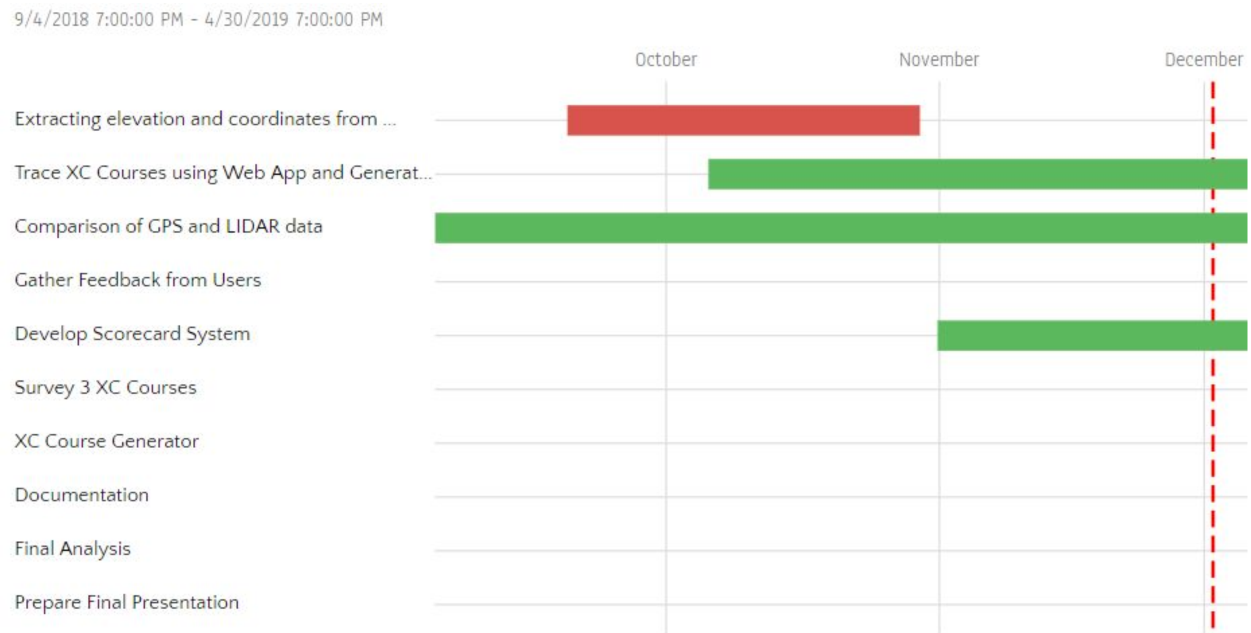


Figure 16: Gantt chart of project timeline in first semester

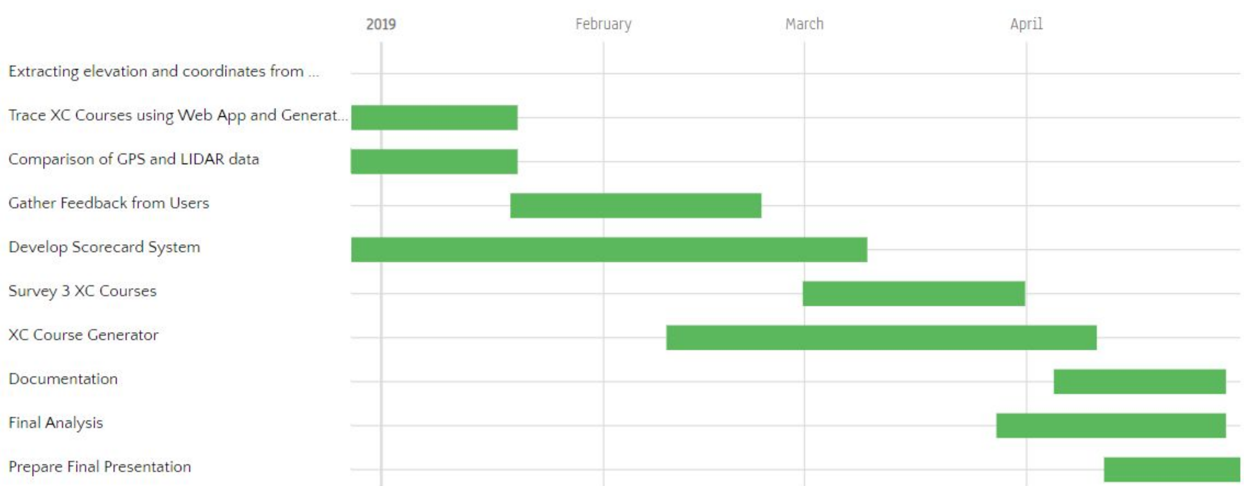


Figure 17: Gantt chart of proposed project timeline in second semester

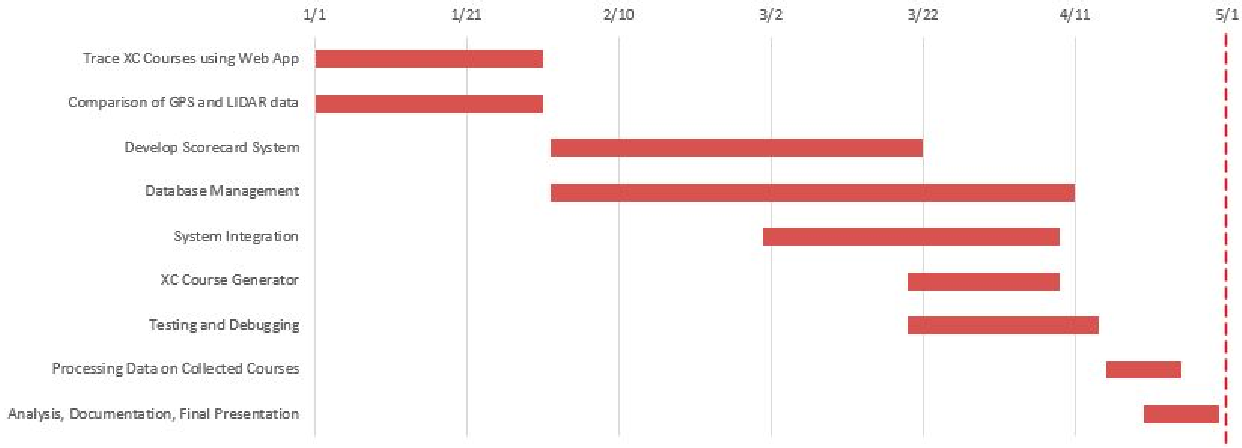


Figure 18: Gantt chart of actual final project timeline in second semester

5.3 Risks and Mitigation

The project had few security or safety considerations to take in to account during development. We initially considered it a safety concern that we used expensive GPS equipment loaned to us by ISU faculty. From the beginning of the project, when we were performing GPS surveys the work was done with a minimum of two people so that all equipment could be observed and protected.

Another consideration we knew would have to be approached was database costs. Our database was initially free but had size constraints that would begin to incur costs if it was exceeded. We discussed different approaches to the problem but ultimately settled on limiting our analysis to a few initial counties, which would be sufficient to prove the functionality of our project. Throughout this time Jacob Feldman was in charge of managing the database, monitoring its size and alerting the team of problems.

5.4 Lessons Learned

The development of our project was a great learning experience for everyone on the team. As the team got split between many important tasks that were being worked on concurrently, it made communication and a unified plan important. While we had difficulties in areas, it was helpful for each component of the project to have someone who took ownership of the work and could detail what the problems were so we could understand how our schedule would be impacted and how to proceed forward.

Additionally, our team learned a lot in the field of troubleshooting and developing solutions. Each portion of the project ran in to its own problems, and integration introduced further problems. Communication during these times was key, as well as flexibility in our schedules so that members could help work on solutions or shift their focus to different work if their current assignment was impeded by technical problems.

Our team learned a lot from a technical standpoint as well. As discussed previously, the biggest technical challenge we faced was the raw size of our dataset, and all the challenges that comes with that. As such, we learned a lot about optimizing upload, access, and structure for a large dataset. This included, but was not limited to, modifying database structure to optimize our main query and optimizing queries, data processing, and insert requests.

6. Conclusions

6.1 Concluding Remarks

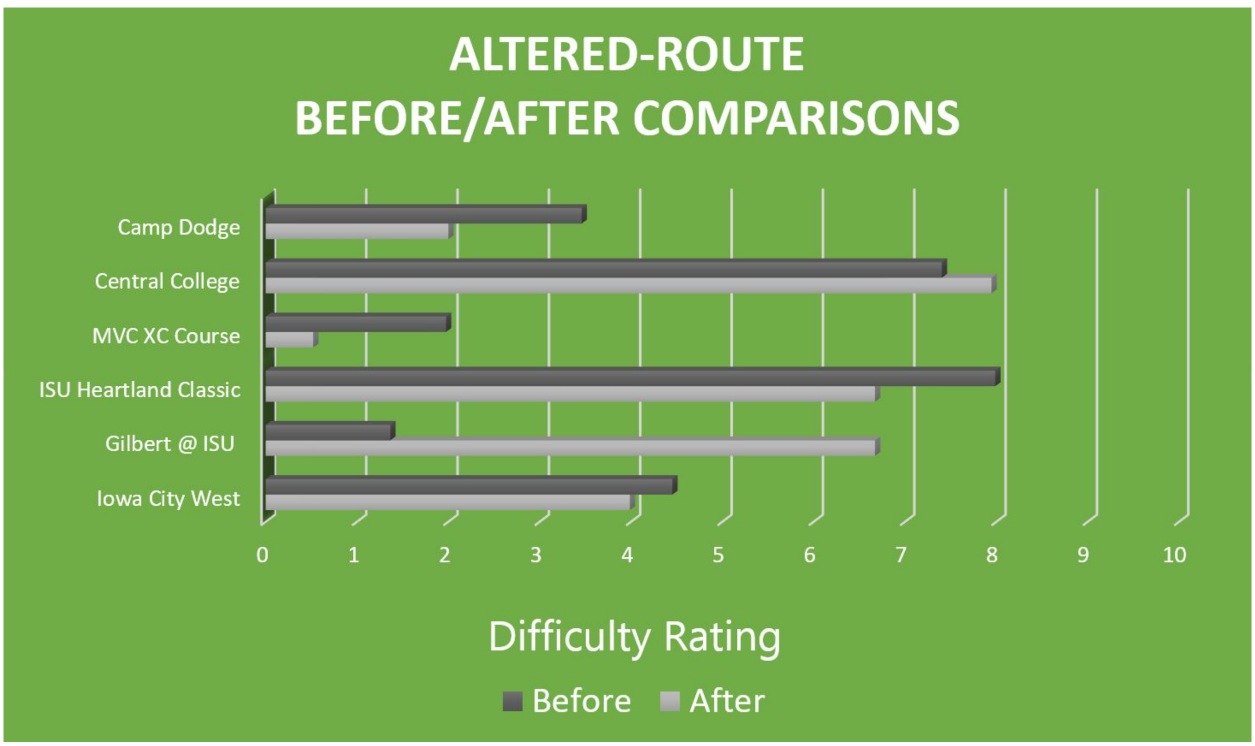


Figure 19 - Before/After route change difficulty comparisons

Figure 19 showcases the disparities in difficulty before and after course rerouting that we found in the six different cases of courses that had changed. While some courses grew dramatically easier after their reroutes (MVC XC Course), some (Gilbert) grew dramatically more difficult. Due to the small sample size (N < 30) of course maps we obtained that showed a route change, we elected to test conduct hypothesis testing via Student's t-test.

At a significance level of 0.1, we were unable to reject our null hypothesis (that courses are not becoming less hilly) in either case where all courses were taken in to account as well as when the Gilbert course change was neglected as an outlier in the dataset. Therefore, our team cannot conclude that XC courses are indeed trending towards becoming less hilly.

However, the small sample size (N = 6) of this evaluation should be viewed with caution. In order to give a more statistically rigorous conclusion, more altered courses should be examined. While our team struggled to find such altered courses, the vast majority of our inquiries to coaches were ignored. We believe that if a request for additional evidence of altered courses were to be made by a more authoritative body like the Iowa High School Athletic Association, more courses could be identified for analysis and the hypothesis testing could be repeated.

While the evaluation of our hypothesis was somewhat anticlimactic, the end result of our work (server + web tool + rating system) holds great potential. According to our market research, there is currently no running app or service available that utilizes a LIDAR database as accurate as ours. We were also unable to find any other topography-based rating system for running routes; all other existing rating systems rely solely upon the variation trends of runners' average times from one race to another. In contrast, our rating system is able to identify the root causes of these time and pace variations, not just variations themselves.

Based on these realities, we believe that our project is the best available assessment tool for running routes in existence today due to its unparalleled legitimacy of our LIDAR database and our use of actual elevation changes to evaluate courses on metrics of energy cost, big hills, rolling hills, and hardest/easiest miles.

6.2 Future Work

The most exciting aspect of our project is the great room there is for future expansion. We've only scratched the surface of what insights can be generated from the highly accurate elevation data we now have the ability to interface with, and we hope that future senior design groups will be able to use the foundation we've assembled to make an even more robust assessment tool to better support the future of cross country running and defend the sport's true spirit with challenging terrain.

As previously mentioned, our software only currently works within the borders of Iowa. However, the website could be modified to accept elevation data sets from more traditional sources like Garmin units or smartphones. While we would not be able to make the same guarantee of legitimacy that we can make with our LIDAR analyses, the use of less accurate data sources is still beneficial when considering that there is no other way for courses in other states to be analyzed than by what we've constructed. Our experimentation with Savitzky-Golay filtering may also prove useful if this future work was pursued.

While we are able to identify rolling hills and big hills, additional work could be done to classify hills more specifically. While elevation changes were the only cause of difficulty that we examined in this project, it can also be argued that sharp turns of a route also present an element of difficulty due to the forces they exert on a runner's body at high speeds. Work could be done to capture the sharpness of a route's turns and then include that information in the construction of the 0-10 course rating.

References

Brueckner, J.C. (1991). The energy cost of running increases with the distance covered. *European Journal of Applied Physiology*, 62(385-389).

Dharmapuri, S. (2014). Vertical Accuracy of Validation of LiDAR Data. *LiDAR Magazine*, 4(2).

“Iowa Geodata.” *Geodata.iowa.gov*, Iowa DNR, 30 Jan. 2018, geodata.iowa.gov/dataset/three-meter-digital-elevation-model-iowa-derived-lidar.

Seewald, Rachel, editor. *Cross Country / Track and Field* . NCAA, 2016.

“Trimble GNSS Planning.” *Trimble GNSS Planning*, www.gnssplanning.com/.

Vernillo, Gianluca (2016). Biomechanics and Physiology of Uphill and Downhill Running. *Sports Medicine*.

Team Information

Thomas Chambers - Ground Truth Engineer
thomasc@iastate.edu

Ryan Hilby - Ground Truth Engineer
rmhilby@iastate.edu

Connor Smith - Ground Truth Engineer
ctsmith@iastate.edu

Jacob Feldman - Data Handling Engineer
feldman@iastate.edu

David Kirshenbaum - UX Engineer
davidk@iastate.edu

Andrew Mumm - Software Engineer
amumm@iastate.edu