CS Capstone Progress Report

April 16, 2019

Slide Sentinel

Prepared for

Oregon State University

Dr Chet Udell

Prepared by

Group 29

The Slide Sentinel

James Stallkamp
Lucas Campos-Davis
Kevin Koos

Abstract

This document provides an overview of the Slide Sentinel capstone teams work over the past term. This includes an overview of what problems we have encountered and how they have/will be solved. Images included show off alpha level functionality.

Contents

1 Purpose

The purpose of the Slide Sentinel project is to provide users a cost effective way to monitor remote areas that are at risk of landslides. Slide Sentinel will accomplish this with an array of sensors devices that communicate telemetry data to a central hub. Slide Sentinel will automatically upload the data that it collects to a remote repository. In addition to the hub system there exists a online client that allows users to access and visualize the data that Slide Sentinel collects. The data that Slide Sentinel collects will hopefully help to prevent or mediate future damages from landslides.

The senior capstone team will also be working on the ONE Hub, a generalized version of the Slide Sentinel hub designed for the LOOM sensor ecosystem. LOOM is a much larger sensor ecosystem that encapsulates many devices, eventually slide sentinel will be fully integrated into LOOM. This means the hub will be able to communicate with other LOOM sensors over nRF, LoRa, or WiFi. Data collected by the ONE hub will be sent to a remote repository of the user's choosing.

2 Goals

The senior capstone team for Slide Sentinel is focused on implementing software for the online client and to handle routing communication between the hub and nodes. Another major goal included for all aspects of the project is to eventually integrate the entire system into the LOOM ecosystem. Additionally the Slide Sentinel may have other small software issues related to being integrated with LOOM that the capstone team may work on.

3 What is Next

There are several tasks remaining to be completed. The script used to populate a spreadsheet used for data store is still being finalized. The online client needs some kind of functionality added to allow users to directly access data. Also the online client has some aesthetic and general formatting work that needs to be done before presenting. Last the client would like the entire slide sentinel system to be integrated into the greater LOOM system that the opens lab uses.

4 Problems and Solutions

4.1 Cellular

During the beginning of the project, it was made clear that we would be using a cellular network along with satellite communication as a low cost solution to uploading data. Work initially began using a Pycom Gpy since it could communicate with cellular networks out of the box. However, this proved to be a great challenge due to Pycom devices only recently getting software to properly communicate over our specific cellular network. For this reason, and that the target device we will be using has changed to the Adafruit feather M0, the decision was made to use an Adafruit breakout board called a FONA, which simplifies the process to communicate over a cellular network.

4.2 Rockblock+

The Rockblock+ proved to be a challenge for us for a while due to circumstances outside of our control. In order to use any Rockblock product, credits need to be bought to upload strings as well as a line rentals to activate the Rockblock to be able to communicate to the satellites. Due to complications with how these charges were going to be funded, we needed to wait until we have the necessary approval to move forward with testing the Rockblock. This approval took around a weeks time to go through due to having to use a wire transfer to transfer funds. Additionally, when it came time to work on the Rockblock+, we had not gotten the line rental yet. Fortunately, approval for the line rental came through quickly this time.

Unique to the Rockblock+ compared to the normal Rockblock, it uses a RS-232 interface to interface with the microcontroller. A custom printed circuit board was created by Kamron of OPEnS for interfacing a Adafruit feather M0 with both the Rockblock and Rockblock+. Before the custom PCB was finished and tested, he helped us with making a breadboard version of the PCB so we could begin testing early.

4.3 LOOM Pushingbox Depreciation and Rock7

With LOOM 2 coming in the near future and new additions being made to the whole LOOM project, previous features are being reevaluated if they should be included in LOOM. Pushingbox, due to security and limitations is being depreciated from LOOM in the near future. Since our project is supposed to integrate into LOOM later in the projects lifespan, we want to align with the direction LOOM is taking and want to remove the need for Pushingbox in our project. While this is simple for directly uploading to Google sheets from the microcontroller, uploading from iridium's Rock7 servers is not. The Iridium Rock7 server requires that the server it sends a POST request to responds with a 200 HTML only and nothing else. If it doesn't get a 200 response code back, the server will automatically put the message into a re-delivery queue where the message will be resent again after a certain amount of time. When a HTML request is sent to a Google URL, a one-time use URL is generated for the request and redirected there for security reasons. This redirect responds with an HTML response code of 302, not the 200 which Rock7 requires. Additionally, the request will go through on Google's side and work, but since Rock7 server think the message did not deliver, the message is sent repeatedly. Due to these reasons, an intermediary is needed for these two servers to talk to one another correctly. A small NodeJS app was created and hosted on Microsoft Azure Cloud Services to act as this intermediary. This program essentially acts in just the same way as Pushingbox and allows for the redirection of the endpoints remotely.

5 Interface Design and Feedback

5.1 Spreadsheet

The spreadsheet used for displaying the raw data is hosted on Google sheets and displays the necessary information to the users. As of the latest version of the spreadsheet, the columns are separated into all the fields that the hubs send out. Everything is displayed to the user and any abbreviations are expanded into their real meaning by the spreadsheet to reduce the number of bytes sent by the hub. Aesthetically, the spreadsheet looks like a normal spreadsheet but sports a green background. Below is a picture of the spreadsheet.

5.2 Map Client

The Map client has been created as a web page and has most of its basic functionality implemented. The Map client is deployed as a node js application and currently is hosted on a personal machine. This map client can load a specific sheet selected by the user from a pre-defined Google spreadsheet. The map client searches the retrieved data for longitude and latitude in the column names and visualizes the corresponding points onto Google maps. The Google API proved to be rather difficult to work with as the API itself is very particular. Security and app permissions proved to be difficult

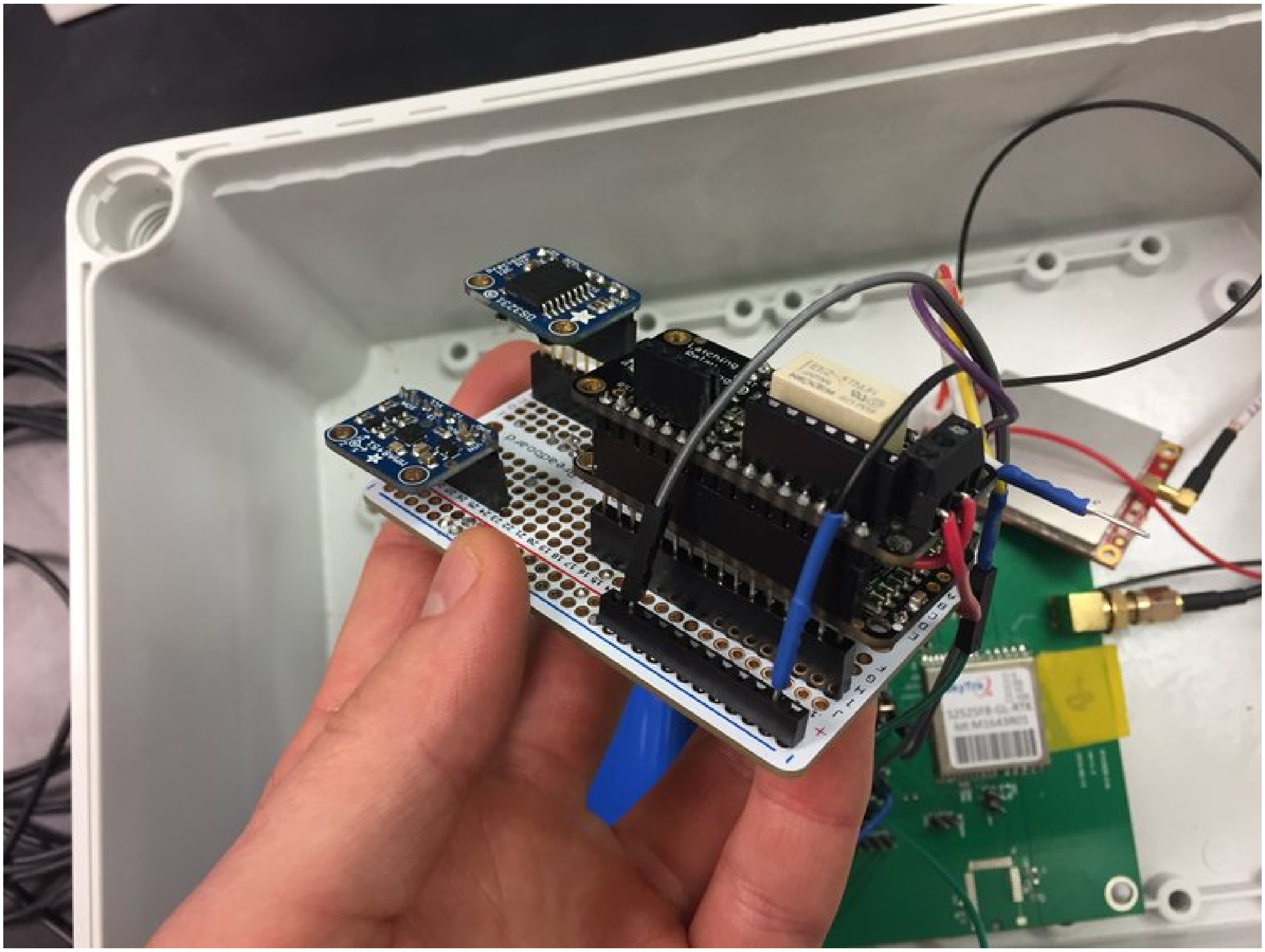
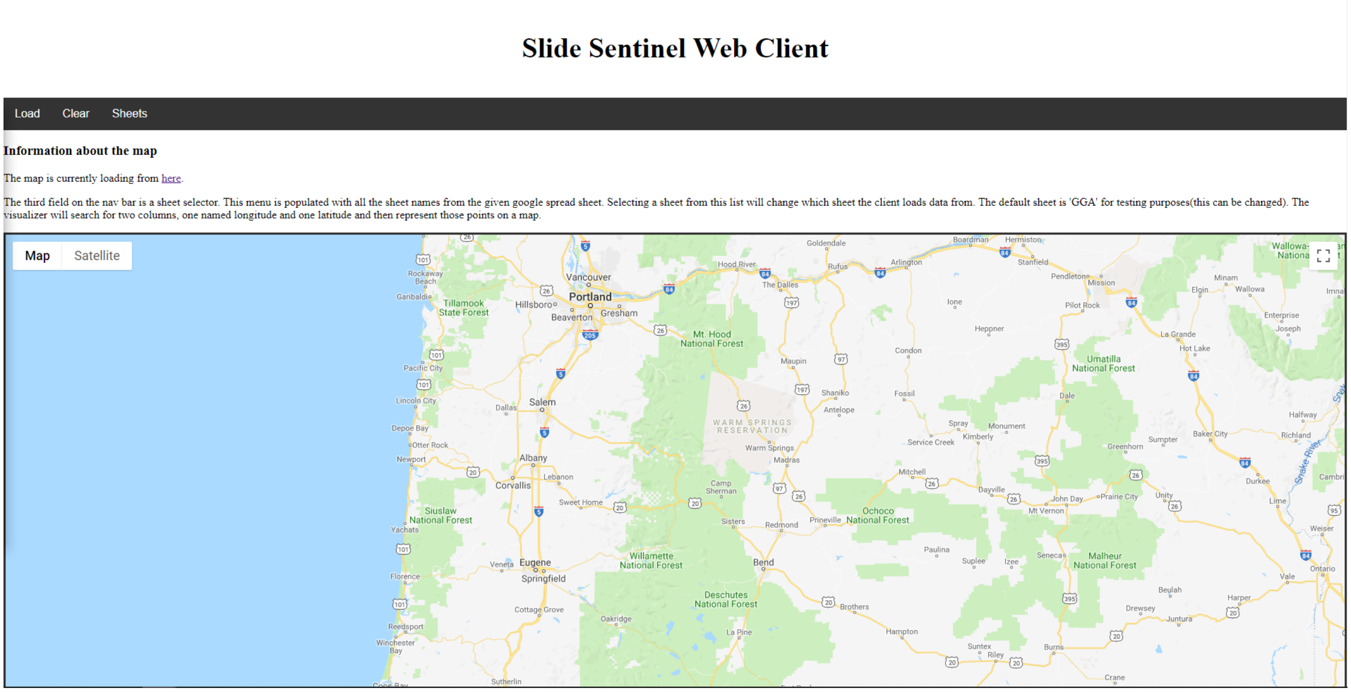


Fig. 1. OPEnS open house revision of the hub

as we had to work through Google's credential system and we had no prior experience. The web client still needs to be polished aesthetically and the client may want additional functionality added. These future additions should be small and can be implemented quickly.

6 Images

Fig. 2. WIP Version of the map client

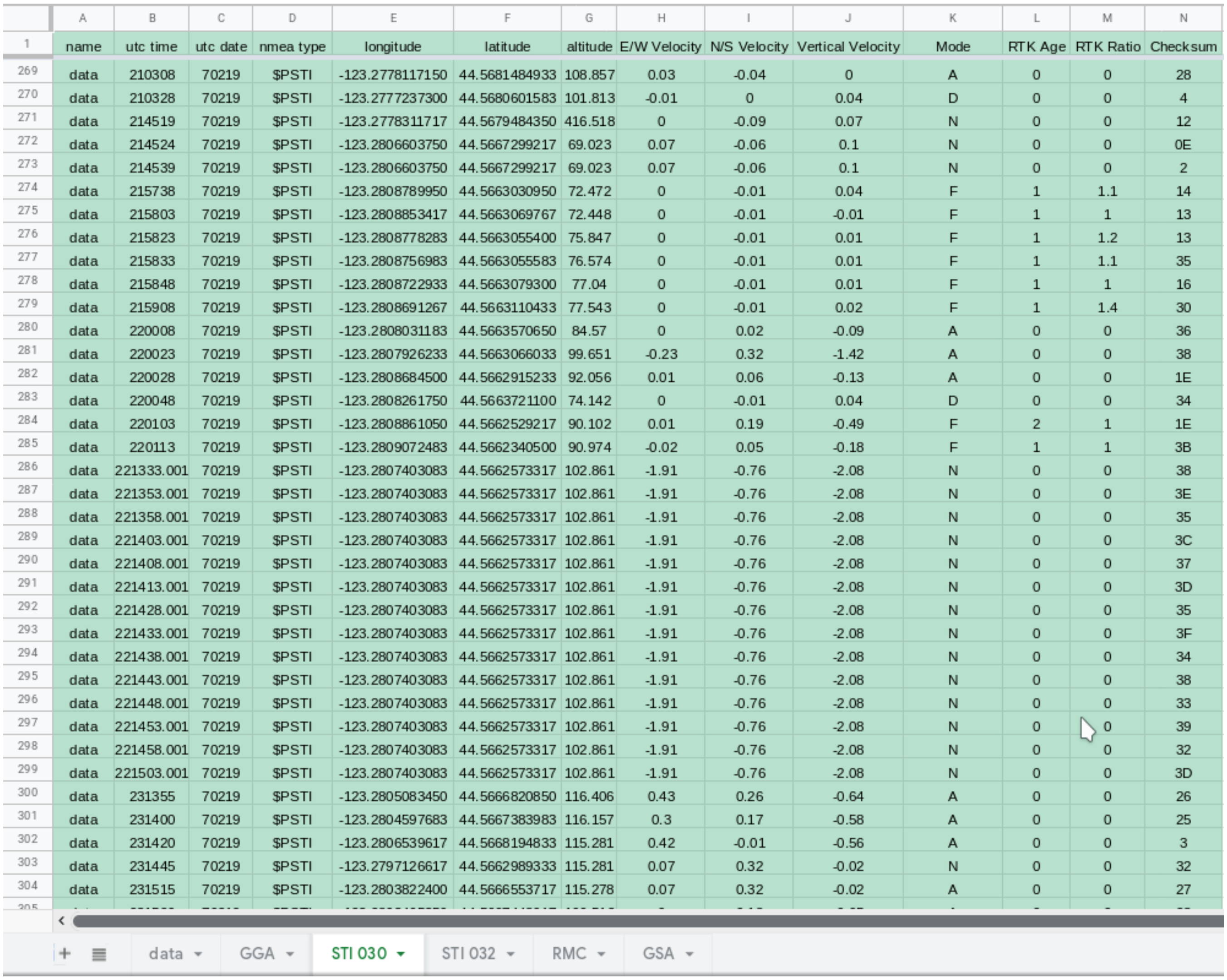


Slide Sentinel Web Client

Information about the map

The map is currently loading from here.

The third field on the nav bar is a sheet selector. This menu is populated with all the sheet names from the given google spread sheet. Selecting a sheet from this list will change which sheet the client loads data from. The default sheet is 'GGA' for testing purposes(this can be changed). The visualizer will search for two columns, one named longitude and one latitude and then represent those points on a map.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	name	utc time	utc date	nmea type	longitude	latitude	altitude	E/W Velocity	N/S Velocity	Vertical Velocity	Mode	RTK Age	RTK Ratio	Checksum
269	data	210308	70219	$PSTI	-123.2778117150	44.5681484933	108.857	0.03	-0.04	0	A	0	0	28
270	data	210328	70219	$PSTI	-123.2777237300	44.5680601583	101.813	-0.01	0	0.04	D	0	0	4
271	data	214519	70219	$PSTI	-123.2778311717	44.5679484350	416.518	0	-0.09	0.07	N	0	0	12
272	data	214524	70219	$PSTI	-123.2806603750	44.5667299217	69.023	0.07	-0.06	0.1	N	0	0	0E
273	data	214539	70219	$PSTI	-123.2806603750	44.5667299217	69.023	0.07	-0.06	0.1	N	0	0	2
274	data	215738	70219	$PSTI	-123.2808789950	44.5663030950	72.472	0	-0.01	0.04	F	1	1.1	14
275	data	215803	70219	$PSTI	-123.2808853417	44.5663069767	72.448	0	-0.01	-0.01	F	1	1	13
276	data	215823	70219	$PSTI	-123.2808778283	44.5663055400	75.847	0	-0.01	0.01	F	1	1.2	13
277	data	215833	70219	$PSTI	-123.2808756983	44.5663055583	76.574	0	-0.01	0.01	F	1	1.1	35
278	data	215848	70219	$PSTI	-123.2808722933	44.5663079300	77.04	0	-0.01	0.01	F	1	1	16
279	data	215908	70219	$PSTI	-123.2808691267	44.5663110433	77.543	0	-0.01	0.02	F	1	1.4	30
280	data	220008	70219	$PSTI	-123.2808031183	44.5663570650	84.57	0	0.02	-0.09	A	0	0	36
281	data	220023	70219	$PSTI	-123.2807926233	44.5663066033	99.651	-0.23	0.32	-1.42	A	0	0	38
282	data	220028	70219	$PSTI	-123.2808684500	44.5662915233	92.056	0.01	0.06	-0.13	A	0	0	1E
283	data	220048	70219	$PSTI	-123.2808261750	44.5663721100	74.142	0	-0.01	0.04	D	0	0	34
284	data	220103	70219	$PSTI	-123.2808861050	44.5662529217	90.102	0.01	0.19	-0.49	F	2	1	1E
285	data	220113	70219	$PSTI	-123.2809072483	44.5662340500	90.974	-0.02	0.05	-0.18	F	1	1	3B
286	data	221333.001	70219	$PSTI	-123.2807403083	44.5662573317	102.861	-1.91	-0.76	-2.08	N	0	0	38
287	data	221353.001	70219	$PSTI	-123.2807403083	44.5662573317	102.861	-1.91	-0.76	-2.08	N	0	0	3E
288	data	221358.001	70219	$PSTI	-123.2807403083	44.5662573317	102.861	-1.91	-0.76	-2.08	N	0	0	35
289	data	221403.001	70219	$PSTI	-123.2807403083	44.5662573317	102.861	-1.91	-0.76	-2.08	N	0	0	3C
290	data	221408.001	70219	$PSTI	-123.2807403083	44.5662573317	102.861	-1.91	-0.76	-2.08	N	0	0	37
291	data	221413.001	70219	$PSTI	-123.2807403083	44.5662573317	102.861	-1.91	-0.76	-2.08	N	0	0	3D
292	data	221428.001	70219	$PSTI	-123.2807403083	44.5662573317	102.861	-1.91	-0.76	-2.08	N	0	0	35
293	data	221433.001	70219	$PSTI	-123.2807403083	44.5662573317	102.861	-1.91	-0.76	-2.08	N	0	0	3F
294	data	221438.001	70219	$PSTI	-123.2807403083	44.5662573317	102.861	-1.91	-0.76	-2.08	N	0	0	34
295	data	221443.001	70219	$PSTI	-123.2807403083	44.5662573317	102.861	-1.91	-0.76	-2.08	N	0	0	38
296	data	221448.001	70219	$PSTI	-123.2807403083	44.5662573317	102.861	-1.91	-0.76	-2.08	N	0	0	33
297	data	221453.001	70219	$PSTI	-123.2807403083	44.5662573317	102.861	-1.91	-0.76	-2.08	N	0	0	39
298	data	221458.001	70219	$PSTI	-123.2807403083	44.5662573317	102.861	-1.91	-0.76	-2.08	N	0	0	32
299	data	221503.001	70219	$PSTI	-123.2807403083	44.5662573317	102.861	-1.91	-0.76	-2.08	N	0	0	3D
300	data	231355	70219	$PSTI	-123.2805083450	44.5666820850	116.406	0.43	0.26	-0.64	A	0	0	26
301	data	231400	70219	$PSTI	-123.2804597683	44.5667383983	116.157	0.3	0.17	-0.58	A	0	0	25
302	data	231420	70219	$PSTI	-123.2806539617	44.5668194833	115.281	0.42	-0.01	-0.56	A	0	0	3
303	data	231445	70219	$PSTI	-123.2797126617	44.5662989333	115.281	0.07	0.32	-0.02	N	0	0	32
304	data	231515	70219	$PSTI	-123.2803822400	44.5666553717	115.278	0.07	0.32	-0.02	A	0	0	27

Fig. 3. OPEnS Open house demonstration spreadsheet

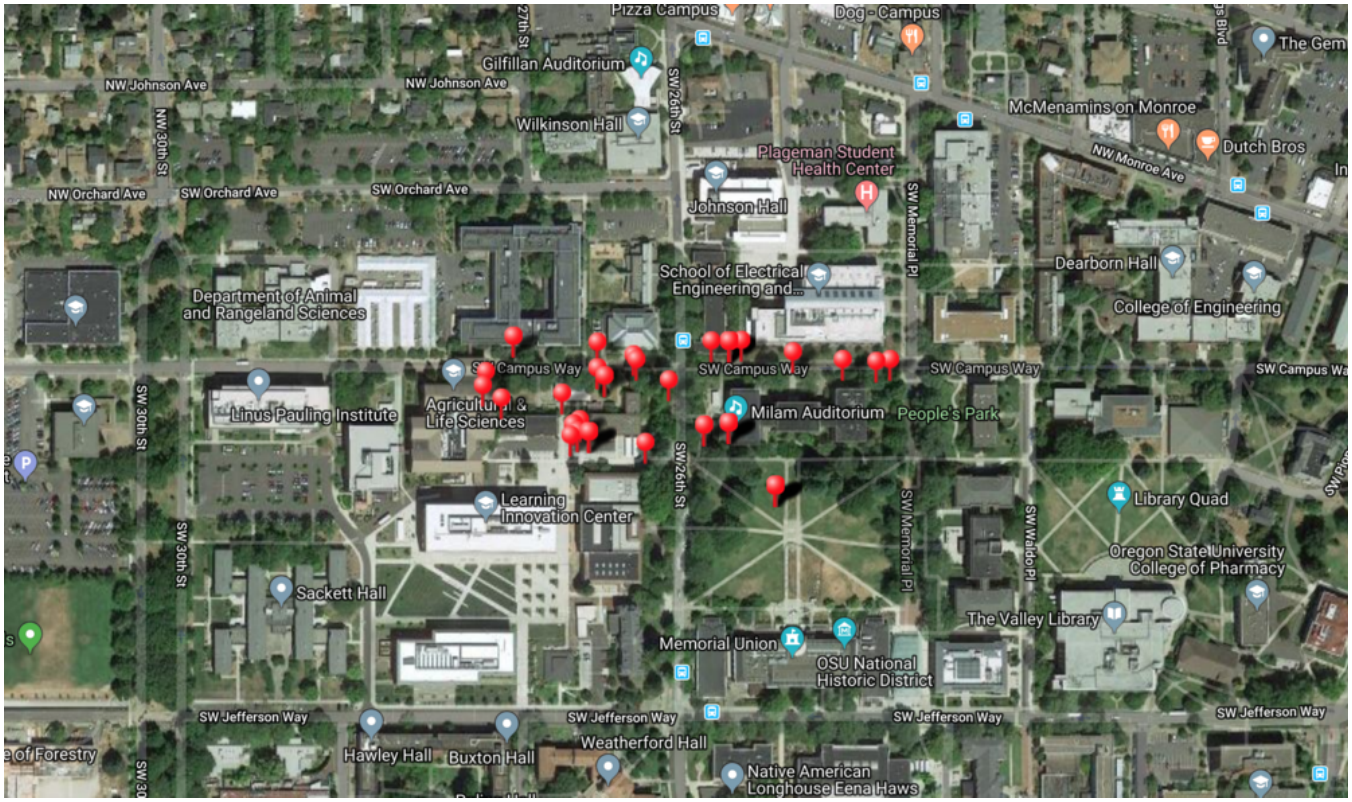


Fig. 4. Early version of the map client using GPSVisualizer. Used for OPEnS open house demonstration