Independent Work Final Report, Fall 2018

**Sidewalk Gradient Mapping Project**

Samuel Arnesen ‘20

Adviser: Prof. Kyle Jamieson

**1. Introduction**

In the United States alone, 3.6 million people require a wheelchair to move around and an additional 26 million people have some other form of movement-related disability [1]. Unfortunately, the ease with which individuals can navigate unfamiliar spaces is highly correlated with one’s physical abilities. Features of urban spaces such steep or uneven sidewalks, high curbs, or obstructed walkways are at most trifling nuisances to the typically-abled but may be insurmountable barriers to others depending on the degree of their handicap [2]. Moreover, even when the area is strictly traversable, following a particular path could still cause an individual to feel immense discomfort if it requires an overly-onerous exertion of energy.

Generally, locals eventually discover routes that are in alignment with their particular physical abilities. Through conversation, trial-and-error, or guidebooks, people who are familiar with the area generally adapt to the local topography and although their new route may be longer or less convenient, it is nonetheless still acceptable as stop-gap solution to the traversability problem. However, for people who are unfamiliar with an area, this solution is insufficient [2]. Whether the individual is coming in from out of town or just visiting, say, a new restaurant, personal knowledge may not be enough for the person to make an informed choice regarding how they can get from their starting point to their destination.

Even when granulated personal knowledge is present, a need still exists for this information to be presented in a coherent and centralized manner. A business owner, for example, who is trying to explain to potential customers how to reach their shop may have detailed knowledge of the local conditions but still has to figure out how to communicate that to a customer base with heterogeneous mobility constraints. Even to the extent that this is doable using existing technology, there is little incentive for individual businesses to make the effort given how such a small percentage of potential customers would even consider using the resource.

The most obvious approach to solve this problem is to use mapping. A map that displays sidewalk steepness, smoothness, and obstacles could provide potential users with the means to identify areas that are inaccessible and generate routes that are traversable. The classic example of such a map is a contour map which has lines that mark areas of equal elevation [3]. Unfortunately, a lack of interactivity and granularity undermines the usefulness of these maps. Other maps, like one produced by the University of Washington called AccessMaps, suffer from no such problems but mostly lack the geographic breadth to be generically useful [4].

It is in this vacuum that I propose my Sidewalk Gradient Mapping Project (SGMP). In brief, SGMP is a web map that displays the surface gradients for a particular area and finds accessible routes between user-specified starting and end points. Built off of Mapbox, the project services the niche left unsatisfied by contour maps and AccessMaps: an interactive and granular map that can cover more than just a single metropolitan area. Ideally, the project could be useful to individuals in need of a route-finding application but whose movement might also be constrained by steep terrain.

The rest of this paper will be as follows. Section 2 describes related approaches to tackling this problem with an especial focus on the contour maps (Section 2.3) and AccessMaps (Section 2.4) briefly mentioned above. Section 2.5 will then cover an attempt by Google to tackle a similar problem within public transit systems as it is an interesting product with potential for future integration. Section 2 will begin with by discussing how this problem is situated within the broader medical and urban planning literature.

Section 3 describes the top level implementation of SGMP. Sections 3.1 and 3.2 explain the project’s various features, how they were implemented, and why they were implemented the way they were. Section 3.3 then describes the basic setup of the basemap off of which the project is built and justifies its various features and Section 3.4 briefly explains the project’s user interface and the design choices that went into building it. Finally, Section 3.5 describes the data sources used in the project.

Section 4 is the longest section and describes how exactly the project was built. Section 4.1 begins by describing the data processing that went into constructing the surface gradient map layer. Section 4.2 then describes how the route-finding features were built. Section 4.3 describes how the route-finding and surface gradient features were combined to form a singular map.

Section 5 evaluates the successfulness of the project. Although the section only uses the test areas of Princeton, NJ and Seattle, WA to evaluate overall accuracy, since the approach used is generalizable, a successful result in those two areas should indicate an overall successful approach. Using both qualitative and quantitative metrics, Section 5.1 describes the evaluation of the surface gradient features of the project while Section 5.2 describes the evaluation of the route-finding features.

Section 6 concludes the paper. After beginning with a summary of the findings presented so far, the section (and the paper in general) ends with a discussion of possible ways in which the project could be improved upon or developed further in the future.

**2. Related Works**

In this section lies an explanation of the different ways in which others have tried to tackle similar problems. In Section 2.1 we will begin by discussing the outside literature on movement-related disabilities from the perspective of the medical and urban design communities, before pivoting to discussing contour maps in Section 2.3, AccessMaps in Section 2.4, and the features of Google Maps in Section 2.5. To be certain, the differing approaches described here are by no means comprehensive; other attempts to tackle related problems do exist. Nonetheless, contour maps and AccessMaps were selected for special attention because we believe them to be broadly representative of the different types of approaches, with contour maps being the stand-in for the low-tech options and AccessMaps for their high-tech alternatives.

**2.1 Outside Discussion**

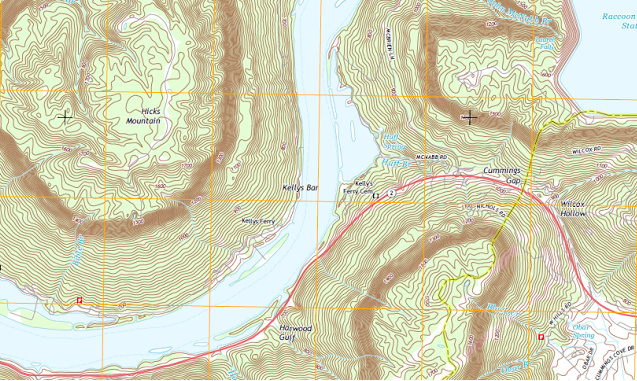
To get a rough sense of how even slight changes in gradient can cause substantial difficulties for individuals with limited mobility, it is useful to look at the existing accessibility guidelines on building design. In their guidelines to businesses and organizations that are building wheelchair ramps, the United Nations recommends that no ramp have a gradient of more than 5% meaning that it should not rise more than one meter over a horizontal distance of 20 meters [5]. In extreme cases, they do allow for gradients of up to 10%, although they also warn that such ramps can be a “hazard.” Granted, ramps are not directly analogous to an outside city environment. On the one hand, ramps are likely to be smoother than outside sidewalks and legal guidelines are likely to be overly cautious, but on the other hand, ramps are also generally short so the average person with limited mobility is more likely to be able to traverse it even if it is somewhat steep for them. These competing factor mean that these guidelines should still be instructive as to the point at which a pathway becomes too steep.

Besides gradient, researchers and limited mobility individuals have also documented other barriers to travel. The Guardian reports that “barriers can range from blocked wheelchair ramps, to buildings without lifts to inaccessible toilets, to shops without step-free access.” [2] Cities have employed a variety of strategies to combat these problems. Melbourne, for example, sends smartphone alerts to users when construction or outages make part of their subway system inaccessible. Kuala Lumpor passed strict accessibility rules in 2010 that committed the city to be 75% barrier-free by 2012 [6]. Washington, D.C. has certified all its subway stations, rail cars, busses as accessible with shortened gaps between rail cars and a large fleet MetroAccess busses that pick up limited mobility residents from near their homes [7].

Despite these efforts, there is still substantial demand for increased accessibility measures in cities. In 2014, the U.K. government calculated that lack of accessibility was dissuading nearly 20% of customers from patronizing city businesses [8].

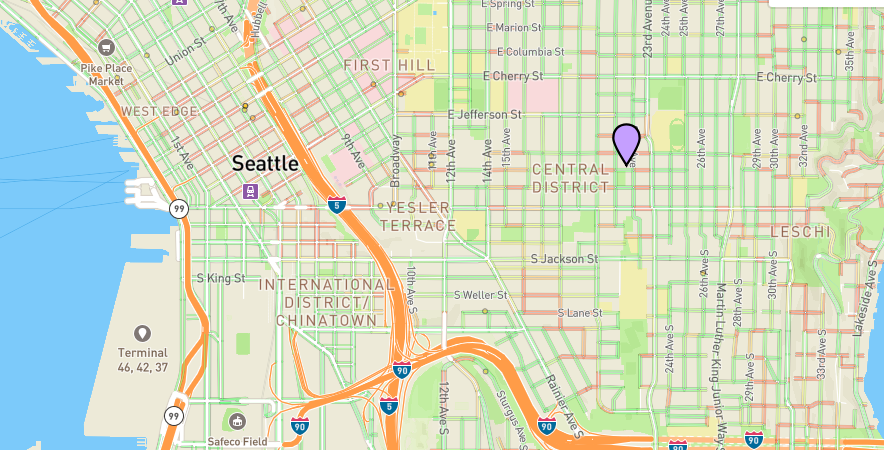
**2.2 Contour Maps**

One of the more common ways to communicate which areas are steeper than others is to use contour maps. Generally speaking, a contour map is a black-and-white map with minimal detail that has lines running across the map that mark areas of equal elevation. This type of map is most commonly found in hiking guides as it is useful for hikers wanting to determine which parts of a path are steepest. Reading a contour map is quite simple: close-together elevation lines indicate a steep path (since that means that little space exists between regions of differing elevation) while spread out elevation lines indicate a gradual path [3].



***Figure 1: An example of a contour map [3]***

Despite their seeming intuitiveness, contour maps have severe limitations that make them an inadequate solution to the problem that SGMP seeks to address. Most notably, contour maps lack the granularity of detail necessary to be useful to someone who seeks to navigate an unfamiliar urban space. This problem is not just one that is descriptively true of most contour maps that currently exist but rather is an inherent problem in the very concept of a contour map. This is because there is a necessary and direct tradeoff between the granularity of detail of the contour map and the readability of said map. For example, if there was a new line for every five-meter change in elevation, the map of most major U.S. cities would be extremely cluttered and likely unreadable. And yet, even a five-meter increment is likely too sparse to be useful as even small changes in elevation can pose substantial problems for many people. This problem is compounded further in urban areas where artificial features like stairs can make short-distance elevation change quite extreme and where the lines would have to be superimposed on top of an already-busy layer of building and street labels.

**2.3 AccessMaps**

**Figure 2: A snapshot of an AccessMaps map of Seattle [4]**

A higher-tech alternative to contour maps is AccessMaps. Produced by the University of Washington, for each sidewalk in Seattle, WA, AccessMaps colors it green, orange, or red depending on its level of accessibility. AccessMaps also supports a route-finding feature where users can input a starting point and a destination and it will return a route that only requires traversing accessible sidewalks [4].

AccessMaps was the inspiration for SGMP and there are many areas in which AccessMaps might be superior. For example, unlike SGMP, when calculating the accessibility of a particular sidewalk, AccessMaps takes into consideration factors besides steepness such as unevenness, construction, and the presence of obstacles. However, that dedication to comprehensive data collection is also a weakness. Since information regarding obstacles and construction is not standardized across municipalities, AccessMaps is difficult to scale. As a result, at the moment, AccessMaps only covers Seattle proper. A leading motivator for this project is to provide a scalable alternative to AccessMaps that provides at least some guidance to people who live outside of Seattle.

**2.4 Google Maps**

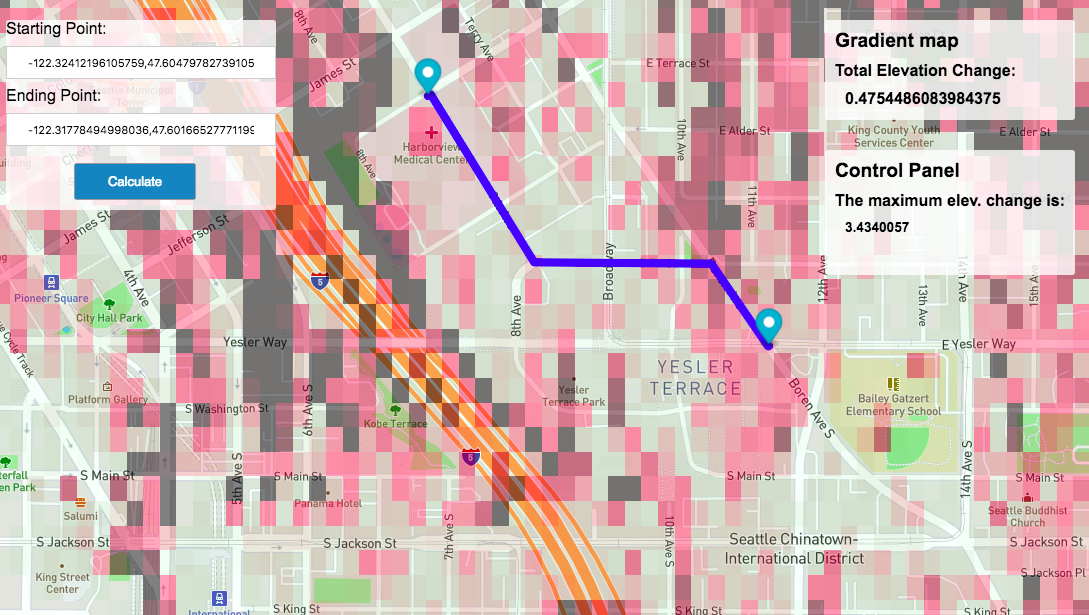
Although Google Maps at the moment does not offer any features specifically designed to highlight the steepness or accessibility of walking trails, since it is the most commonly used mapping product that is publicly available, it is nonetheless still worth it to give a cursory overview of the related features it does have.

In March of 2018, Google announced that it was adding a new feature to its popular mapping service that would provide information regarding the accessibility of transit platforms in select cities. So, for example, if a particular subway station does not offer elevator service, one could select not to have that subway stop included as an endpoint along one’s path [9]. Since this tackles a complementary problem to the central one tackled by SGMP, this feature could potentially be integrated into a future version of SGMP.

**3. Feature Overview**

This sections describes the general implementation of SGMP. In Section 3.1, we discuss the surface gradient features of the project and then in Section 3.2, we discuss the route-finding features. Section 3.3 then discusses the basemap itself, 3.4 discusses the user interface, and 3.5 discusses the data sources that were used.

**3.1 Surface Gradient Features**

In order to offer even a semblance of utility, SGMP needs to indicate which areas have a steep gradient and which do not. The way SGMP ultimately does this is by superimposing another map of surface gradients on top the underlying basemap. Besides the particulars of how to build this superimposed map (which will be discussed in section 4), there are three additional elements that are worth discussing.

**Figure 3: A snapshot from SGMP - Seattle**

First, the superimposed map of gradients varies in opacity and color. That is, not only are steeper areas (i.e. gradients with high absolute values) colored a different color (red and black) than flatter areas, they are also much more opaque. This variance of opacity and coloring was instituted to balance the tradeoff between readability and granularity. Since most areas have a relatively flat gradient, making the opacity of the superimposed layer low allows most features of the map to be easily read while making the opacity of the steeper areas much higher allows them to stand out more as a warning. Although areas that are consistently quite steep might at times be difficult to read, that is quite rare in most urban areas, and so this design choice seems justified.

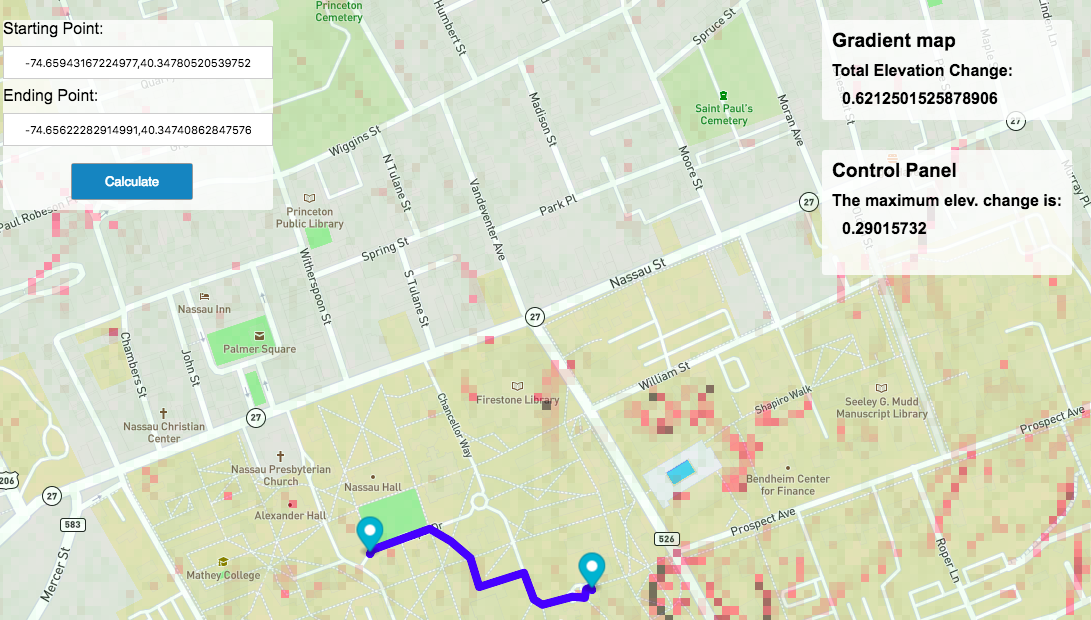
Second, each 10-meter x 10-meter increment is colored a single color that is determined by the difference between the highest and lowest point in that block. Although in theory higher levels of specificity are more useful and the data source used for the project does offer elevation data down to the meter for many areas, this pooling filter was nonetheless still applied for a few reasons. First, for readability. Extremely high levels of specificity are difficult to discern if the user zooms out of the map even slightly and so the user might miss important surface information unless it is pooled to show just the largest changes. Second, out of caution. Since going down an overly steep path could pose real safety risks to an individual, false negatives where an area is labelled as flat when it is not are much worse than false positives where an area is labelled as highly steep when it is in fact quite flat. As a result, over representing steep areas by having an entire 10-meter by 10-meter block’s color determined being only by its steepest part seems reasonable.

Lastly, unlike in AccessMaps, the surface gradient map is superimposed over the entire basemap and not just over roadways and sidewallks. The reasoning for this was that although it did add some additional degrees of clutter, it also better accounts for the fact that many pedestrians do not stay strictly on the sidewalk and many do cut across grassy stretches of park or other open areas.

**3.2 Route-Finding Features**

The way in which SGMP supports route-finding is quite simple. The user can either input a pair of coordinates or place two markers down on the map. SGMP then calls the Mapbox Directions API and retrieves a list of different ways to get from the starting point to the destination. If the first path requires traversing over a patch that has a gradient larger than 10% then that path is rejected and the next alternative is considered. The selected path is then displayed on the map along with its steepest gradient.

The downside of this approach is that there are corner cases in which all the options retrieved by Mapbox Directions API require going through a steep area. However, experimentally, these tended to be only for very short routes where Mapbox Directions had a strong preference for just going the direct route and not taking an unnecessarily circuitous route. Since requesting a route-finding calculation is fairly rare for very short distances, this tradeoff was one we were willing to make.

**3.3 Basemap**

**Figure 4: SGMP map of Princeton**

The map off of which all the data gathered in the project was superimposed comes from Mapbox’s Street-v7 map. The reason why Mapbox was chosen over reasonable alternatives like Google Maps was that it supports a greater degree of customization. More specifically, it was comparatively easier to add additional superimposed layers onto a Mapbox map than onto a Google Maps map. Mapbox also gets its mapping data from OpenStreetMap, an enormous open-source mapping project that ensures that most of its roadway data is likely to be accurate and up-to-date [10].

**3.4 User Interface**

The user interface for SGMP was designed to be self-evident and intuitive so little is needed in way of explanation. There are a few ways in which a user can interact with the map besides the standard web-mapping interaction features such as zooming in and out and moving from side-to-side. First, a user can input a starting point and a destination either by placing markers on a map or by inputting the coordinates directly. Second, a user can hover over an area to see its gradient. These two features borrow heavily from the way in which users typically interact with more popular mapping services like Google Maps and Apple Maps and so they should provide the user with a self-evident way of interacting with both the surface gradient and route-finding layers of the map.

**3.5 Data Sources**

The data source used in this project was the United States Geological Survey’s GIS data. The reason for this decision was quite simple: no other data source provides as granular and reliable information. For much of the United States, the USGS provides elevation information on each individual meter and even in their less-covered areas, they still have information down to the 1/9 arc-second (which is about 3 meters) [11]. That level of granularity vastly outstrips the specificity of other data sources such as the elevation data that Mapbox or OpenStreetMaps uses [12]. Although some cleaning of the data is necessary at points, this is easily done by throwing out unlikely outliers and given how many data points USGS data provides, this can also be done safely without leaving any area uncovered.

**4. Approach**

This section describes how exactly SGMP was built. In Section 4.1, we discuss how the surface gradient layer was built. In Section 4.2, we discuss how the route-finding features were built. Finally, Section 4.3 discusses how the two parts interacted.

**4.1 Surface Gradient Layer**

The first step in the process of creating the surface gradient layer was to download the necessary data from the USGS GIS website. Although the USGS supports multiple different file formats, for ease of use, we chose to download the data as an .img file. Since the processing program was written in Python2.7, we then open the file and save the data as an array using the python osgeo.gdal library, a library built specifically to handle large amounts of USGS GIS DEM raster data. In order to translate the array coordinates to actual geographic coordinates, we used the metadata on the dataset boundary conditions found in an .xml file that came with the original download. That information then gets passed into the processing program through a config file.

We then proceed to iterate through all the data points. For each 10x10 block, the elevations get sorted and the output for that particular block is the difference between the top elevation and the bottom elevation. It is in this step that the data cleaning also happens as outlier elevation values are discarded.

The output for each particular block then gets added to a temporary python dictionary that contains entries for the coordinates of the block and its calculated gradient. That in turn gets added to a larger python list of all the blocks in the entire dataset. In the end, this data gets dumped into a .geojson file that can be parsed by Mapbox.

In brief, Mapbox custom maps are just a large stack of “layers.” At the bottom are layers that contain all the road, terrain, and municipality data and at the top are layers that can be added by a developer. In order to create one’s own layer, one first needs to specify a data source that maps a particular set of values (e.g. gradient) to coordinates. In this case, the outputted .geojson file is the data source. Then, using the Mapbox JavaScript library, one can add a layer with the .geojson file as the data source in the HTML file that establishes what the main page looks like.

**4.2 Route-Finding Layer**

Although Mapbox actually provides a free route-finding plugin for general use, we decided not to take advantage of it as a tool because it limited our ability to customize the route-planning. That is to say, if we were to use the built-in route-finding tool, the user could input a starting point and a destination and the plug-in would return a path but one would not be able to process or filter that path according to gradient information because the client is never actually exposed to what the resulting path actually looks like.

So, instead, we had to build our own route-finding tool. The first step is to have the user input the starting and ending coordinates through an HTML form. For ease of use, we designed it such that the form entries also get auto-populated when one moves a marker on the map but that is not essential to the basic functionality of the SGMP.

That form data then gets passed to a python CGI script. To prevent the CGI script from opening up a new webpage, we defined the target for the CGI script to be an iframe of size 0. That CGI script opens the GIS data file using osgeo.gdal and then makes an HTTP GET request to the Mapbox Directions API which returns a geojson object. After parsing the returned object to find the steps one is supposed to take, we are left with a list of coordinates that correspond to the points at which one is supposed to make some move (generally a turn). We then take 1-meter “steps” along that path and sample the elevation at that point. The difference in elevation between consecutive points is used to find the gradient. The maximum gradient for the entire path is stored and if it is larger than the set threshold, we consider the next returned path to see if its maximum gradient is lower.

Here a fairly difficult design choice had to be made regarding whether to prioritize a lower average gradient or a lower top gradient. We decided on the latter because a too-high top gradient could make a path truly inaccessible while a too-high average gradient is more likely to make someone “just” uncomfortable. Although in some cases the level of discomfort might be such that it makes the path functionally inaccessible, we still decided to prioritize the lower top gradient. In most cases anyways, the route with the lower top gradient also has a lower average gradient since outside of cliffs, most areas with a single instance of very high gradient also have other instances of steep ascent or descent.

Regardless, the next step is to convert the found path into a Mapbox layer. Here, we once again build a .geojson file but instead of each entry in the JSON object being a box of points, each entry is a line. In order to allow the main HTML page to build the new layer, the .geojson file is saved in the filesystem of the user who is running the CGI script and that file is set as the data source for the new layer. However, this method is not foolproof. Because most browsers cache files to improve rendering time, if one just saves the .geojson object to the same file every time, each time one tries to reload the layer to display a new path, one just runs into the cache hit and an old version of the layer will appear. To circumvent this problem, the timestamp (rounded to the nearest 10 seconds) of when the CGI form was submitted is used in the name of the .geojson file. Since the name changes each time, this prevents the caching problem from materializing and because both the CGI script and the home HTML page have access to the time, they both know what the file will be named. By using this workaround, we were able to get the route-finding layer (which displays not only lines but also text about the gradient along that route) to appear on the main page.

**4.3 Homepage**

As is common for web applications, the home page was built with a combination of HTML, CSS, and JavaScript. The <head> section declares some basic setup information and defines the script source to be from the Mapbox GL JS library which in turn allows the rest of the page to use functions from Mapbox to set up the map. The <style> section is written in CSS and defines how the text, buttons, and boxes all appear on the page relative to the underlying map.

In the <body> section, after setting up the HTML form, the first thing we do is declare a new instance of a MapboxGL map so that the entire page is taken up by the map. Then, we use a Mapbox library function to establish the rules for what should happen after the map loads. In particular, we specify that upon loading, two additional layers should be added to the map. The first is the surface gradient layer discussed earlier (here is also where we style this layer so that it varies in opacity and color). The second is the route-finding layer. However, since the map has just loaded and the user has yet to actually input a starting and ending point, the data source for the route-finding layer is set to a file called “blank.geojson” which defines a line of length 0 located at coordinates [0, 0] and the layer’s display properties are set to make it invisible. This way, although the layer is present, the viewer does not see any routes on the map until they submit the HTML form.

The remaining parts of the <body> section just define what happens when the user interacts with the map in particular ways. So, for example, when the user moves their mouse over a specific area, the gradient for that area shows up in a box in the upper right corner. Similarly, when they drag the markers around on the map, the starting and ending points of the HTML form become populated and the previous route (if there was one) disappears from the screen. The only really notable piece of code comes from where we define the rules for what happens when one clicks the “Calculate” button on the HTML form. Technically, the form itself only has two fields that are exposed to the user (“start” and “end”) and has no “Submit” button. A separate “Calculate” button was added in lieu of a traditional “Submit” button because we wanted multiple actions to happen when one submits the form. When clicked, three things occur. First, the current time is calculated as the file name to be used when sending back the calculated information. Second, the form is submitted. Third, after a three second wait, the source for the route-finding layer is updated to the new file, the layer’s display property is set to visible, and one of the boxes on the right-hand side becomes ready to display the gradient information about the route. The reason for the three second wait is that it takes some time for the CGI script to execute and if the homepage tries to retrieve the updated data before the script has finished executing then it will throw a 404 “File Not Found” error.

**5. Evaluation**

In this section we will evaluate whether or not the project was a success, with Section 5.1 focusing on the results of the surface gradient layer and Section 5.2 focusing on the results of the route-finding functions.

Two caveats are worth bearing in mind before we delve into the evaluation itself. First, although SGMP seeks to present a generalized approach to building an accessibility map that could in theory cover the entirety of the United States, for fairly self-evident reasons related to rendering time, computer space constraints, and hosting costs, we could not run tests on every area of the United States. Instead, we will run tests on just Princeton, NJ and Seattle, WA. We feel limiting our evaluation to these two areas is still quite reasonable because a) neither of these two regions have features that are likely to make our technique work dramatically better there than in other regions and b) none of the particular features of those two regions are hard-coded so to the extent to which our method might perform worse in other regions, that is more likely to reflect discrepancies in USGS GIS data than flaws in our approach (although if the USGS GIS data is unreliable, then arguably our project’s heavy reliance on it would be flaw in its approach). One limitation of restricting our evaluation to just Princeton, NJ and Seattle, WA is that both areas are fairly developed – Princeton is suburban and Seattle is urban – and so this will not capture how well our project works in rural areas. However, as we explained in Section 1, we believe the main use case of SGMP is the navigation of unfamiliar urban spaces so we believe this limitation to be acceptable.

Our second caveat is that there is no truly objective benchmark against which we can determine whether our project is a success. The traditional gold standard for elevation data is the USGS GIS data but since that is an input to SGMP, using it as a metric of comparison would be useless. For route-finding, the options are even bleaker as no reliable metric of comparison exists at all. So, in lieu of any truly ideal benchmark, we will instead use a composite of qualitative assessments of Princeton’s topography and quantitative comparisons to the results AccessMaps was able to get for Seattle. Since AccessMaps considers more than just sidewalk gradient when evaluating the degree of accessibility, we would expect some discrepancy between their product and our output even if our project worked perfectly well. However, with no better metric of evaluation presenting itself, this comparison will still lie at the heart of our evaluation.

**5.1 Surface Gradient Evaluation**

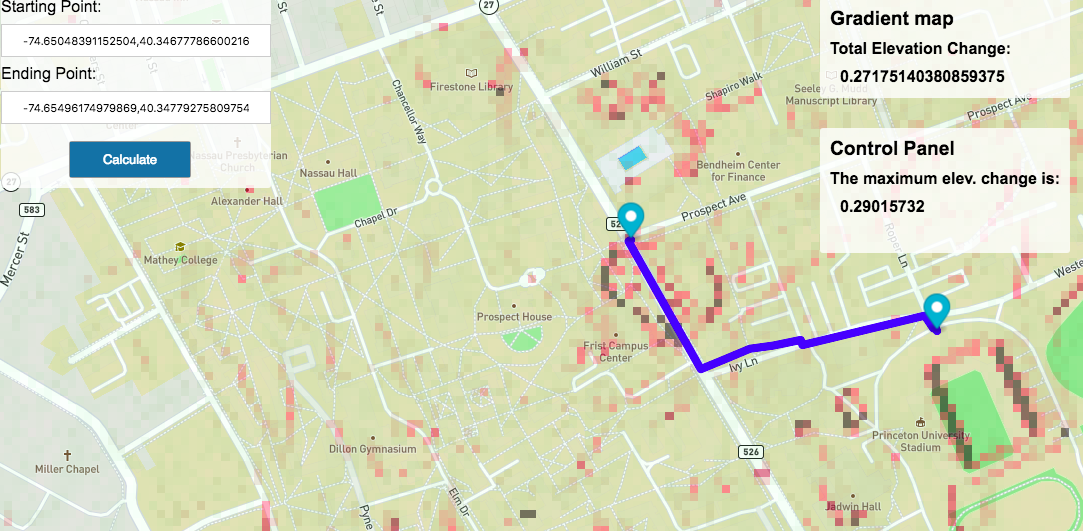
Qualitatively, the surface gradients displayed on the SGMP map seems to roughly track the underlying topography. For example, on the map of Princeton, NJ, the maximum steepness indicator is displayed exactly where the Princeton football stadium stands are, suggesting that the underlying USGS data was able to effectively capture the sharp change in elevation.

To be more precise in the evaluation of the accuracy of this layer, we sampled 20 intersections from AccessMap’s Seattle map and compared their results to the results in SGMP. Since SGMP tracks total elevation change over a 10-meter block and AccessMaps tracks the percent incline along a specified path, a 1:1 comparison was not able to be performed. Regardless, a correlation could still be drawn to determine whether or not the two methods roughly track. The resulting correlation was 0.52. (A full summary of the data can be found in Appendix 1). Although this seems low, upon further inspection, much of that discrepancy can be attributed to an inability on the part of the SGMP map to pick up on the presence of overpasses and bridges. When a bridge appears, the USGS GIS data picks up on the sudden increase in surface elevation. However, that increase in elevation is misleading because it reflects the presence of a bridge not a steep gradient. Future iterations of SGMP should better account for this problem by calling on OpenStreetMap or Mapbox APIs to determine whether a point is on a bridge or overpass.

**5.2 Route-Finding Evaluation**

In order to evaluate the success of our route-finding method, we compared the maximum elevation change of ten routes using our method to the maximum gradient found by traversing the same path within the AccessMaps map. Here, the correlation between the two numbers was only 0.35. (A full summary of the data can be found in Appendix 2). However, this was largely attributable to a single unexplained outlier; when removed from the data, the correlation rose to 0.8472. Although it is difficult to ascertain why exactly that particular data point had such a divergent response, it is possible that the AccessMaps map was picking up on a sidewalk unevenness parameter untracked by SGMP.

Overall then, the evaluation for the route-finding method was more positive than the result for the surface gradient layer. This is encouraging since we suspect the more likely use-case of SGMP would involve users interacting with the route-finding tool rather than just visually inspecting the map. One possible explanation for why this method was more successful was that the output of the route-finding gradient calculation (elevation change along the roadway over a distance of 1/9 arc-second) is much closer to how AccessMap calculates percent gradient than the output of the surface gradient layer (total elevation change over a 10m x 10m block).



**Figure 5: SGMP of Princeton Campus**

**6. Conclusion**

Overall, the Sidewalk Gradient Mapping Project seeks to provide individuals with movement-related disabilities with the means to find and evaluate routes that are accessible to them. To do that, we superimpose gradients calculated from USGS GIS data onto a normal web map and build a route-finding tool that returns the path with the lowest maximum gradient.

In the process of evaluation, some limitations of this approach appeared. Most notably, in urban areas, elevation could appear to change rapidly even though the area might still be traversable. The best example of this involves bridges and overpasses: the elevation appears to jump rapidly when one goes under a bridge because the USGS GIS data tracks the top of the bridge as the actual elevation.

However, the reason why we feel that none of the limitations exposed so far are truly dispositive is because future work could in theory overcome these problems. One possible avenue for future development would be to collate the GIS data with information from OpenStreetMaps or Google Maps on the presence of bridges and overpasses so that spurious areas of high elevation change could be ruled out. A more ambitious future development would be to add data from municipal departments of transportation regarding construction and other obstacles to pedestrian crossing. A more detailed route-finding application that offers step-by-step instructions could also be implemented in the future.

Ultimately, the goal of this project was to articulate a generalizable approach to constructing a sidewalk accessibility map of the United States. Although the current implementation still has some limitations, little in the evaluation suggests that our method failed to accomplish that aim. Resultantly, moving forward, we feel that the Sidewalk Gradient Mapping Project could be the basis of a future project that could meaningfully improve the lives of those in need of accurate and detailed mapping of the accessibility of urban spaces.

[1] Brault, Matthew W. “Americans with Disabilities: 2010, Household Economic Studies.” *United States Census* (July 2012): pg. 78. Web. <https://www.census.gov/prod/2012pubs/p70-131.pdf>

[2] Salman, Saba. “What would a truly disabled-accessible city look like?” *The Guardian* (14 Feburary 2018). Web. <https://www.theguardian.com/cities/2018/feb/14/what-disability-accessible-city-look-like>

[3] “Topographic Maps.” *United States Geological Survey* (2018). Web. <https://www.usgs.gov/products/maps/topo-maps>

[4] Langston, Jennifer. “New route-finding map lets Seattle pedestrians avoid hills, construction, accessibility barriers.” *UW News* (1 February 2017). Web. <https://www.washington.edu/news/2017/02/01/new-route-finding-map-lets-seattle-pedestrians-avoid-hills-construction-accessibility-barriers/>

[5] “Architectural Design Considerations: Ramps.” *United Nations enable* (n.d.). Web. <https://static.un.org/esa/socdev/enable/designm/AD2-01.htm>

[6] Ito, Akiko et al. “Good Practices of Accessible Urban Development.” *United Nations* (2015). <http://www.un.org/disabilities/documents/desa/good_practices_in_accessible_urban_development_october2016.pdf>

[7] “Accessibility.” *Washington Metropolitan Area Transit Authority* (n.d.). Web. <https://www.wmata.com/service/accessibility/>

[8] Office of Mark Harper. “High street could be boosted by £212 billion ‘purple pound’ by attracting disabled people and their families.” *Department for Work and Pensions, Office for Disability Issues* (27 August 2014). Web. <https://www.gov.uk/government/news/high-street-could-be-boosted-by-212-billion-purple-pound-by-attracting-disabled-people-and-their-families>

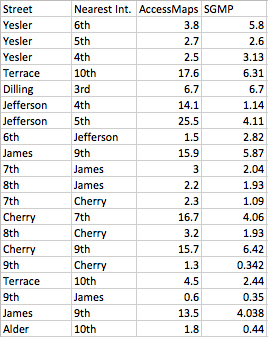
[9] Akasaka, Rio. “Introducing ‘wheelchair accessible’ routes in transit navigation.” *Google Blog* (15 March 2018). Web. <https://www.blog.google/products/maps/introducing-wheelchair-accessible-routes-transit-navigation/>

[10] Mapbox – About Maps. *Mapbox* (n.d.). Web. <https://www.mapbox.com/about/maps/>

[11] “Maps.” *United States Geological Survey* (n.d.). Web. <https://www.usgs.gov/products/maps/gis-data>

[12] “Access Elevation Data.” *Mapbox* (n.d.). Web. <https://www.mapbox.com/help/access-elevation-data/>

**Appendix 1: Surface Gradient Evaluation Data**



**Appendix 2: Route-Finding Evaluation Data**

