AccessScore: A framework for generating individually tailored accessibility visualizations for people with mobility impairments

Anthony Li, CMSC499A

University of Maryland

Abstract

Prior work on Project Sidewalk has gathered a large collection of individual point labels identifying the locations of accessibility features and accessibility issues in the streets of Washington, DC. While this data is useful in its raw form for many applications, it is necessary to develop user-friendly visualizations to make it useful for people with mobility impairments. AccessScore presents a framework for developing interactive visualizations that are customizable for individual mobility needs. It consists of a tunable model that computes location-specific accessibility scores based on walking routes to nearby points of interest, paired with customizable visualizations that display model results at various scales.

---

1. Introduction

Through the use of volunteer and paid contributors to label Google Street View imagery, Project Sidewalk has been able to perform a virtual accessibility audit for the entirety of Washington, D.C [1]. The labels mark the locations of curb ramps, missing curb ramps, sidewalk obstructions, poor sidewalk surfaces, and other features that could impact an area's accessibility. Because one of the main goals of Project Sidewalk is to use technology to assist people with mobility impairments, it is necessary to find methods of presenting the collected data in ways that are directly useful to those who could most benefit from it. There is a wide range of possibilities for new accessibility tools that utilize Project Sidewalk data, but visualization tools represent some of the most direct applications. Thus, the focus of AccessScore was to develop user-friendly visualizations of Project Sidewalk data.

Prior Project Sidewalk development created an initial implementation of AccessScore, a tool designed to utilize Project Sidewalk data to show the accessibility of neighborhoods in a choropleth map. However, this implementation was found to have limited usefulness due to the simplicity of the model used, which this project aimed to address.

In developing new AccessScore visualizations, a few design goals were prioritized:

1. The visualizations should be directly useful to people with mobility impairments. While many visualizations could be useful to researchers, city planners, and other audiences, people with mobility impairments were considered the primary audience of these visualizations.
2. The visualizations should be able to adapt to individual mobility needs and preferences. Through past interviews of people with mobility impairments conducted by the Project Sidewalk team [2], it was clear that needs varied depending on the individual and the mobility device used (wheelchair, walker, etc.) if any. Thus, any static visualization would likely not be accurate for the entire target audience.
3. The visualizations should aim to predict actual navigation experience in the areas represented. For instance, an area with many accessibility issues may still present a good navigation experience if the accessibility issues are mostly located in secluded, rarely visited areas.

2. Methodology

2.1. Model development - Review of literature and existing work

A prerequisite to creating accurate and useful accessibility visualizations is developing algorithmic models to quantify accessibility using the raw data collected. Because of its importance and impact on the resulting visualizations, development of a reasonable and adaptable model was a major focus of the project.

Significant prior work has been done in the area of quantifying the importance of different accessibility features and accessibility obstacles. Meyers et. al. [3] ranked the impact of various outdoor features on accessibility using month-long surveys of people with mobility impairments, informing the weights used in an accessibility scoring model that was tested. Additionally, Sonenblum et. al. [4] measured typical daily travel distances for wheelchair users, which further informed model development.

Because scoring accessibility bears some resemblance to scoring walkability, some inspiration was also taken from Walk Score [5], a website that provides walkability scores for queried addresses.

2.2. Model algorithm

The proposed AccessScore scoring algorithm works as follows:

1. Given a location to be scored, find the nearest n points of interest (POIs) corresponding to each of p POI categories. For instance, if $n = 2$ and $p = 3$:[restaurants, libraries, schools], the algorithm would find the two closest restaurants, two closest libraries, and two closest schools to the queried location.
2. For each POI found in Step 1, compute a walking route to the POI.
3. Discard any routes that exceed a maximum distance m.
4. Initiate accessibility score to zero.
5. For each accessibility-improving feature found along the routes remaining after Step 3, add a constant c to the accessibility score. (The value of c can differ depending on obstacle type, severity, and user customizations.)
6. For each accessibility-harming feature found along the routes remaining after Step 3, subtract a constant d from the accessibility score. (The value of d can differ depending on obstacle type, severity, and user customizations.)
7. Divide the resulting accessibility score by the total length of the remaining routes, in miles, to produce the final accessibility score.

2.3. Model to visualization conversion

Many techniques can be explored for utilizing the proposed model to create map-based visualizations, such as heat maps and choropleths. Neighborhood-level accessibility visualizations were a primary focus of the first iteration of visualizations. With inspiration taken from Walk Score, the following algorithm was used to generate area-based accessibility visualizations using the proposed model:

1. Overlay a grid of rectangular cells onto the region of interest (e.g. Washington D.C.), with each cell having a width of approximately w.
2. For each cell, compute an accessibility score for the cell's center. This score represents the entire cell's accessibility.

3. Scale the cell accessibility scores linearly between 0 and 1, with 0 representing the least accessible cell and 1 representing the most accessible cell.
4. Color the cells using a 10-level color scale, with darker colors representing less accessible areas.

A significant advantage of the proposed model and visualization algorithms is that they can be extended to produce street and neighborhood-based visualizations. For instance, a street-based visualization can be produced by computing accessibility scores at the endpoints of each street segment, averaging the two scores to produce an accessibility score for each street segment. Visualizations for larger, irregularly shaped neighborhoods could be created by averaging the scores of all cells contained in the neighborhood, perhaps with more weight given to cells with higher populations.

Additionally, by only considering features along walking routes to POIs, the algorithm is able to give more weight to street segments that are more likely to be traversed. This is important in enabling the visualizations to address the third design goal of accurately reflecting navigation experience.

2.4. Enabling customization and interactivity

As reflected in the second design goal, it is important to allow any visualizations generated to be customized according to the user's mobility needs. This was accomplished by making parameters c and d of the model algorithm customizable so that users can decide the relative importance of different feature types.

Allowing for customization of the visualization created some implementation challenges since it was important that the visualization remains fast and responsive even with customization options. A visualization that required a long wait to reload after a customization change would present a poor user experience.

Fortunately, since the customizable parameters are only factored in at the final steps of the scoring algorithm, many model steps can be precomputed. These include finding routes to POIs from cell centers and counting the features along routes. The precomputed results can be sent to the client which then performs the final steps of the scoring algorithm with the customized values of c and d. Using this architecture, only simple arithmetic calculations need to be performed on the client as opposed to more expensive spatial operations.

2.5. Technical implementation

Various libraries and tools were explored for the implementation of the model and visualization, including Mapbox, Leaflet, MapD, turf.js, and deck.gl. Ultimately, deck.gl was used to build the visualization largely for its ability to use OpenGL to accelerate client-side rendering. Python, node.js, and turf.js were used to perform the precomputable steps of the algorithm, and Leaflet and Mapbox Studio were used for initial prototyping of the model. Additionally, the Google Maps Directions API was used to find nearby POIs and compute walking routes. The visualization generation pipeline works as follows:

1. Use Python and the Google Maps Directions API to find POIs and compute walking routes to them from cell centers. Save the routes in geojson format.
2. Use node.js+turf.js to load the routes generated in Step 1, along with accessibility feature labels downloaded from the Project Sidewalk Admin API. For each cell, count the number of each feature type along the cell's routes using turf.js. Export the feature counts in geojson format.
3. Embed the feature counts created in Step 2 into a deck.gl-powered visualization webpage. Using client-side code, compute cell accessibility scores dynamically using the feature counts and user-customized values of c and d and render the visualization using deck.gl.

Code for each of the above steps along with documentation is available in the project's Github repository.

2.6. Parameter tuning and model evaluation

Finding ideal values for all of the parameters used in the model requires a robust framework for evaluating the quality of model output. Since the short timeframe of this project did not allow for the development of one, model results could only be evaluated subjectively. Generally, we expected downtown, tourist-heavy areas of Washington D.C. to be most accessible since investment in accessibility is likely to be significantly greater in these areas. The following "first-guess" values were used for the initial iteration of the visualization and produced results that fit this expectation, but it is very likely that better values can be found to produce more accurate results.

- $n = 3$: i.e., we find the nearest three POIs in each POI category

- $p = 7$: we use 7 POI categories, specifically: [”grocery”, ”restaurant”, ”school”, ”coffee”, ”park”, ”museum”, ”hospital”]

- $m = 1.5$ miles: This is the maximum distance of any route used for accessibility scoring. In theory, this should be close to the maximum distance a user with mobility impairments would reasonably travel to generate the most realistic routes. Sonenblum et. al. [4] informed reasonable values. However, if set too low, there may not be enough routes found to generate reliable results. It is possible to implement AccessScore visualizations in a way that allows users to select the value of m from multiple options. However, allowing m to be adjustable on a continuous scale may prove too computationally expensive as it would become difficult to precompute routes.

- $w = 2$ miles: This is the approximate width of each cell. It is probably desirable to make this similar to or slightly larger than the value of m so that most routes stay generally within their respective cells.

- c, d: These values are intended to be fully user-customizable on a continuous scale, though reasonable defaults should be provided. In the initial iteration, the default weight of each feature corresponded directly to its severity. For instance, an accessibility-harming feature (e.g. sidewalk obstruction) with severity 4 would reduce the accessibility score by 4 points at step 6 of the model algorithm. Accessibility-increasing features (i.e. curb ramps) behave the same way but on a reversed scale, so a curb ramp with severity 4 would increase the accessibility score by 2 points. (The scale reversal is necessary because severity 1 corresponds to a high-quality curb ramp, while severity 5 corresponds to a poor quality curb ramp.)

During model development, a model with the values of c and d set based on the results of the study by Meyers et. al. [3] was tested using Mapbox Studio. However, without a concrete framework for evaluating model accuracy, it was difficult to determine if using these adjusted values improved results, so the simpler severity-based weights were used in the first iteration of the visualizations.

2.7. UI Controls

The UI controls implemented for customizing the values of c and d are shown in Figure 2. UI design presents ample opportunity for exploration,

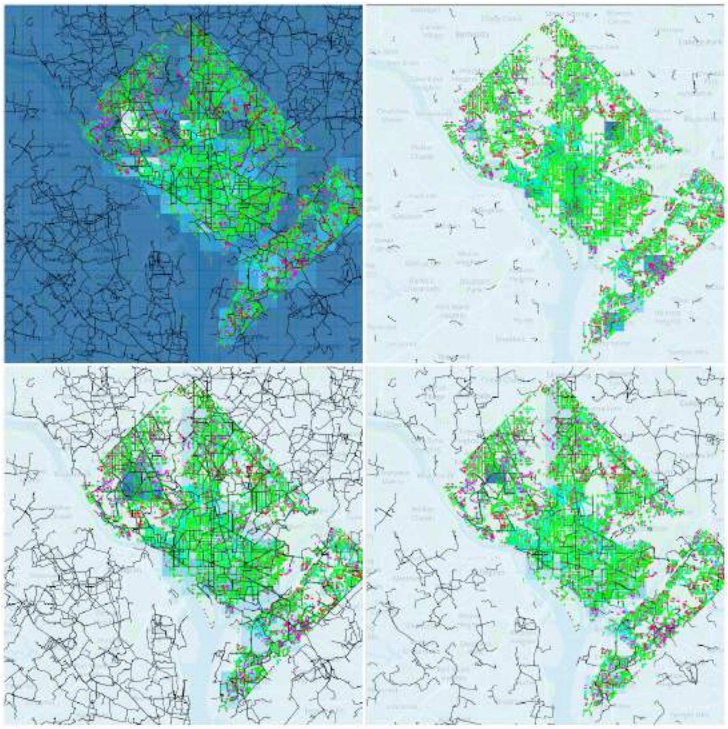


Figure 1: Four accessibility visualizations of Washington D.C. created by tuning model parameters, rendered using Mapbox Studio. In this figure, darker areas indicate more accessible locations. Note that these figures were generated without filtering out labels from the Project Sidewalk onboarding tutorial, creating a highly-accessible appearing region in the northwest corner of the city. The onboarding labels were removed for the first iteration of the interactive visualization.

and various potential designs are provided in the project Github repository. Optimal UI design should provide the user with the ability to customize the c and d parameters or remove some accessibility features from consideration without needing to understand the inner workings of the model.

3. Results

The result of this project is a flexible, extensible framework for modeling and visualizing accessibility using data collected by Project Sidewalk. It offers techniques to quantify the accessibility of any point location within Project Sidewalk's coverage area and create various types of personalizable interactive visualizations. A working example of an interactive visualization

Customize AccessScore

You can tailor the map to your specific accessibility needs by answering some questions.

What mobility device do you use?

○ Electric wheelchair
○ Manual wheelchair
○ Cane/walker
● None

How important are...

Curb ramps? 3

Smooth sidewalks? 3

Unobstructed sidewalks? 3

Figure 2: Control UI implemented in an AccessScore visualization

has been made public and is linked in the project's Github repository. A screenshot is depicted in Figure 3.

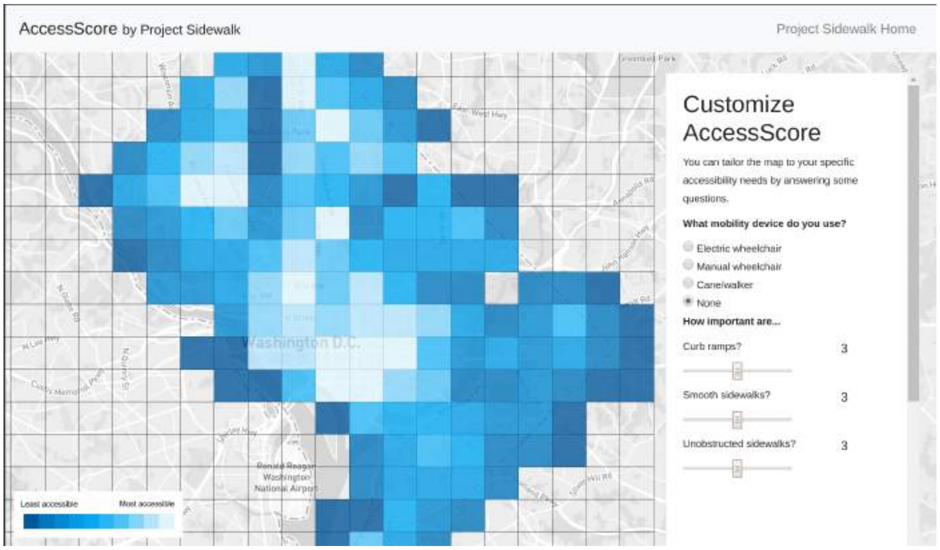
4. Github

Code and other referenced resources can be found at `https://github.com/tongning/access-score`.

5. Future Work

This project has significant potential for further work and development. The most immediate next step would likely be evaluating and improving the accuracy of the model, which would likely require the development of a robust and preferably quantifiable framework for evaluation. This may involve collecting data on the actual perceived accessibility of various regions of Washington, D.C.

Once such a framework is established, it would become possible to tune model parameters for more accurate results and make adjustments to the



Customize AccessScore

You can tailor the map to your specific accessibility needs by answering some questions.

What mobility device do you use?

○ Electric wheelchair
○ Manual wheelchair
○ Cane/walker
● None

How important are...

Curb ramps? 3
Smooth sidewalks? 3
Unobstructed sidewalks? 3

Figure 3: Implementation of area-based accessibility visualization

model to take additional factors into consideration. As examples, other factors that may be considered include distance to public transit options, elevation changes, and each label's distance from the starting point of a route.

Additionally, the current visualization implementation utilizes raw label data from Project Sidewalk. This may cause it to become increasingly inaccurate as a single accessibility feature can be labeled multiple times by separate Project Sidewalk contributors. The implementation should be adapted and tested to work on unique features rather than raw labels.

Finally, different visualization designs can be iterated and refined, preferably with feedback from user studies.

6. References

[1] M. Saha, K. Hara, S. Behnezhad, A. Li, M. Saugstad, H. Maddali, S. Chen, J. E. Froehlich, A Pilot Deployment of an Online Tool for Large-Scale Virtual Auditing of Urban Accessibility, Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility - ASSETS '17 (2017) 305–306.

[2] K. Hara, C. Chan, J. E. Froehlich, The Design of Assistive Location-based Technologies for People with Ambulatory Disabilities, in: Proceedings of

the 2016 CHI Conference on Human Factors in Computing Systems - CHI '16, pp. 1757–1768.

[3] A. R. Meyers, J. J. Anderson, D. R. Miller, K. Shipp, H. Hoenig, Barriers, facilitators, and access for wheelchair users: sbstantive and methodologic lessons from a pilot study of environmental effects, Social Science & Medicine 55 (2002) 1435–1446.

[4] S. E. Sonenblum, S. Sprigle, R. A. Lopez, Manual Wheelchair Use: Bouts of Mobility in Everyday Life, Rehabilitation Research and Practice 2012 (2012) 1–7.

[5] Walk Score, Walk score homepage, 2018.