ABSTRACT

Title of Dissertation:	SCALABLE METHODS TO COLLECT AND VISUALIZE SIDEWALK ACCESSIBILITY DATA FOR PEOPLE WITH MOBILITY IMPAIRMENTS
	Kotaro Hara, Doctor of Philosophy, 2016
Dissertation directed by:	Professor Jon E. Froehlich Department of Computer Science

Poorly maintained sidewalks pose considerable accessibility challenges for people with mobility impairments. Despite comprehensive civil rights legislation of Americans with Disabilities Act, many city streets and sidewalks in the U.S. remain inaccessible. The problem is not just that sidewalk accessibility fundamentally affects where and how people travel in cities, but also that there are few, if any, mechanisms to determine accessible areas of a city a priori.

To address this problem, my Ph.D. dissertation introduces and evaluates new scalable methods for collecting data about street-level accessibility using a combination of crowdsourcing, automated methods, and Google Street View (GSV). My dissertation has four research threads. First, we conduct a formative interview study to establish a better understanding of how people with mobility impairments currently assess accessibility in the built environment and the role of emerging location-based

technologies therein. The study uncovers the existing methods for assessing accessibility of physical environment and identify useful features of future assistive technologies. Second, we develop and evaluate scalable crowdsourced accessibility data collection methods. We show that paid crowd workers recruited from an online labor marketplace can find and label accessibility attributes in GSV with accuracy of 81%. This accuracy improves to 93% with quality control mechanisms such as majority vote. Third, we design a system that combines crowdsourcing and automated methods to increase data collection efficiency. Our work shows that by combining crowdsourcing and automated methods, we can increase data collection efficiency by 13% without sacrificing accuracy. Fourth, we develop and deploy a web tool that lets volunteers to help us collect the street-level accessibility data from Washington, D.C. As of writing this dissertation, we have collected the accessibility data from 20% of the streets in D.C. We conduct a preliminary evaluation on how the said web tool is used. Finally, we implement proof-of-concept accessibility-aware applications with accessibility data collected with the help of volunteers.

My dissertation contributes to the accessibility, computer science, and HCI communities by: (i) extending the knowledge of how people with mobility impairments interact with technology to navigate in cities; (ii) introducing the first work that demonstrates that GSV is a viable source for learning about the accessibility of the physical world; (iii) introducing the first method that combines crowdsourcing and automated methods to remotely collect accessibility information; (iv) deploying interactive web tools that allow volunteers to help populate the largest dataset about

street-level accessibility of the world; and (v) demonstrating accessibility-aware applications that empower people with mobility impairments.

SCALABLE METHODS TO COLLECT AND VISUALIZE SIDEWALK ACCESSIBILITY DATA FOR PEOPLE WITH MOBILITY IMPAIRMENTS

by

Kotaro Hara

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2016

Advisory Committee:
Professor Jon E. Froehlich, Chair
Professor David Jacobs
Professor Benjamin B. Bederson
Professor Andrea Wiggins
Professor Niklas Elmqvist

To my family

Acknowledgements

I would like to thank my advisor, Jon E. Froehlich, for guiding and supporting me for many years to finish my Ph.D. Thank you for always setting high goals and pushing me to reach beyond what I could have done alone. I appreciate all the support and patience you have provided throughout this process. Thank you for making me a better researcher.

I would also like to thank my committee members, Ben B. Bederson, David Jacobs, Andrea Wiggins, and Niklas Elmqvist. Ben, thank you for challenging me with hard questions. David, you were a great source of insights for building a computer vision system. Andrea, thank you for your thoughtful comments on the crowdsourcing research. Niklas, thank you for investing time and being in my committee; you supported me when I really needed help.

I was fortunate to work with research collaborators Shiri Azenkot and Jin Sun. I also thank my mentor and collaborator at MSR, Shamsi T. Iqbal. Undergrad and high school advisees: Christine Chan, Jonah Chazan, Zachary Lawrence, Victoria Le, Anthony Li, Robert Moore, Sean Panella, Niles Rogoff, Daniil Zadorozhnyy, and Alex Zhang; I learned a lot through mentoring you. Thank you.

Ph.D. has been a (very) long journey and I wouldn't have survived without the support from my friends and colleagues. Alex Ecins, thank you for being a crazy house mate; I was never bored. Matt Mauriello, I wouldn't have finished Ph.D. without all the cups of coffee that we had together. Tak Yeon Lee, we protected the HCIL at night.

I also thank my friends and colleagues at the Makeability Lab, the HCIL, and the Department of Computer Science: Ioana Bercea, Cody Buntain, Tiffany Chao, Ruofei Du, Jorge Fayton, Alan Fong, Liang He, Yurong He, Jonggi Hong, Chang Hu, Angjoo Kanazawa, Seokbin Kang, Jessy Kate, Majeed Kazemitabaar, Tomas Lampo, Ran Liu, Sana Malik, Arunesh Mathur, Brenna McNally, Austin Myers, Varun Nagaraja, Ladan Najafizadeh, Leyla Norooz, Uran Oh, Alex Quinn, Manaswi Saha, Chrisoph Schulze, Lee Sterns, Aishwarya Thiruvengadam, Michael Whidby, Kristin Williams, Adil Yalçın, and Philip Yang. Thank you all for making my grad life fun!

Finally, thanks to my parents, Mitsuyo and Nobuaki, and my sister, Rumi, for their unconditional care and love. Thank you for always supporting me.

Table of Contents

List of Tables

List of Figures

Chapter 1 Introduction

Poorly maintained sidewalks pose considerable accessibility challenges for people with mobility impairments [127,129]. According to the most recent U.S. Census (2010), roughly 30.6 million adults have physical disabilities that affect their ambulatory activities [185]. Despite comprehensive civil rights legislation for Americans with Disabilities, many city streets, sidewalks, and businesses in the U.S. remain inaccessible (*e.g.,* [43,65,85]). For example, maintenance issues such as buckled or cracked sidewalks can pose significant accessibility challenges so too do larger, more permanent infrastructural issues such as utility poles or fire hydrants directly in sidewalk paths or the lack of curb ramps at intersections or sidewalks (Figure 1.1). These issues are significant. In a precedent-setting court case in 1993, the court ruled that the "lack of curb cuts is a primary obstacle to the smooth integration of those with disabilities into the commerce of daily life" and that "without curb cuts, people with ambulatory disabilities simply cannot navigate the city" [1].

(a) Object in Path (b) No Curb Ramp (c) Surface Problem (d) No Sidewalk

Figure 1.1. In this dissertation, we describe the methods that combines crowdsourcing, online map imagery, and automated methods to semi-automatically locate, identify, and assess accessibility problems in the built environment. The images above show crowd annotations from the experiments on Mechanical Turk where minimally trained crowd workers were asked to find, label, and rate the severity of sidewalk accessibility obstacles in Street View images.

The problem is not just that sidewalk accessibility fundamentally affects where and how people travel in cities, but also that there are few, if any, mechanisms to determine accessible areas of a city *a priori*. Indeed, in a recent report, the National Council on Disability noted that they could not find comprehensive information on the "degree to which sidewalks are accessible" across the U.S. [132].

Methods to identify (in)accessible areas of an unfamiliar places is important for people with mobility impairments. Knowing where and what barriers exist can help affected travelers mitigate, prevent, or better prepare for accessibility problems in the built environment [21,127,135,167]. While prior research has identified common strategies that people with mobility impairments use to evaluate the accessibility of routes and destinations *a priori* (*e.g.,* seeking trip advice from caregivers [135,167]), we have limited knowledge about the role of modern and future interactive technology in informing travel-related decisions. In our own formative interview study with people with mobility impairments (Chapter 3), we show that location-based technologies that specifically incorporate accessibility information about the physical environment—what we call *assistive location-based technologies* (*ALTs*)—could indeed be useful and desired. The critical challenge to enable such technologies is to collect comprehensive data about the accessibility of the physical environment—a key contribution of this dissertation.

Traditionally, sidewalk assessments have been conducted via in-person street audits by government or volunteers [171,172], which are labor intensive and costly [157], or via citizen call-in reports, which are done on a reactive basis [220]. And,

although some cities offer sidewalk information online (*e.g.*, through government 311 databases [178]), these solutions are not comprehensive, rely on *in situ* reporting, and are not focused on collecting and providing accessibility information. The lack of data collection methods and the consequent lack of readily available sidewalk accessibility information limit us from designing and developing technologies to inform people about the city's accessibility [195].

To address this problem, this dissertation research introduces new scalable methods for remotely collecting data about street-level accessibility using a combination of crowdsourcing, automated methods, and Google Street View (GSV). For example, we evaluate whether we can efficiently locate curb ramps by tagging Street View images by combining computer vision-based object detection algorithms with crowdsourcing-based manual image labeling. The collected accessibility information could enhance capability of location-based technologies. For example, developers could enhance and incorporate neighborhood accessibility information into GIS tools that are used for urban analysis and policy making (*e.g.*, AMELIA [119]). Accessibility-aware way-finding applications (*e.g.*, MAGUS [127]) that have been available only in areas where such data existed (often with *in situ* data collection by government or researchers) could be deployed at much larger scale. These tools could change the way people view how friendly their neighborhoods are to mobility impaired people, transform the way they choose where to live, how governments plan and execute constructions and alteration of urban accessibility features, and could even

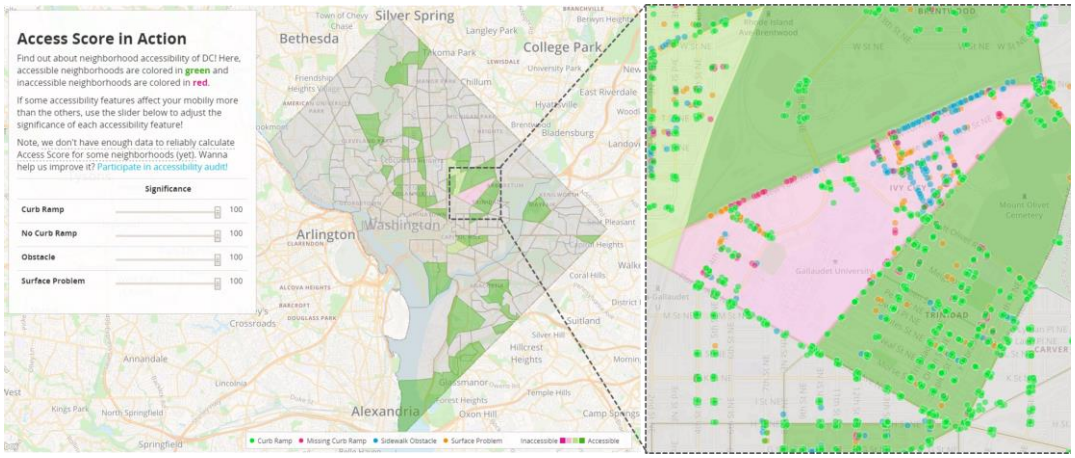


Figure 1.2. To demonstrate the utility of the street-level accessibility data collected by our methods, we create a proof-of-concept choropleth map, *Access Map*, that visualizes accessibility levels of neighborhoods in Washington, D.C. Mobility impaired travelers could use Access Map to quickly assess which neighborhoods are accessible and inaccessible.

influence the property values just like technologies to assess walkability of neighborhoods could influence real estate values [44,56]. Although the primary contribution of my dissertation is new scalable data collection methods for sidewalk accessibility, we also create proof-of-concept ALTs such as an online visualization tool that help demonstrate the value of the collected data (Figure 1.2).

1.1 Dissertation Research Approach and Overview

This dissertation describes four threads of research. First, we conduct an exploratory interview study with 20 people with mobility impairments. The interview study allows us to explore current strategies of mobility impaired individuals for assessing built environment accessibility as well as reveals a broad range of future designs and requirements of ALTs. We then introduce and study novel crowd-powered methods for collecting street-level accessibility data that enable the future ALTs. In the next two chapters, we design, develop, and evaluate systems that combine crowdsourcing,

automated methods, and GSV to scalably collect street-level accessibility data. In the last research thread, we demonstrate the value of the data collection methods and the collected accessibility data. We develop a volunteer-based data collection tool and deploy it, and we design and develop proof-of-concept ALTs using the collected street-level accessibility data. We describe each thread in detail below.

1.1.1 A Formative Interview Study with Mobility Impaired People

Previous work in urban design, public health, and assistive technologies have identified accessibility barriers such as lack of curb ramps, narrow and obstructed sidewalks, and poor travel surfaces [14,21,127,129,152]. However, little research has investigated how people with mobility impairments currently adapt to accessibility barriers. To build a better understanding of how people with mobility impairments assess the accessibility of the built environment and the roles of technologies therein, we conduct a formative interview study with 20 people with mobility impairments. The study involves three parts: a semi-structured interview (Part 1), a participatory design session (Part 2), and a design probe activity (Part 3).

The semi-structured interview in Part 1 was designed to investigate current methods and tools that people use to plan trips and assess the accessibility of the built environment. Findings from Part 1 reinforce and extend previous research in how people with mobility impairments assess accessibility [21,135,167]. We found that, while planning trips remains a challenge, modern location-based technologies support people with mobility impairments—even if not designed specifically for that purpose.

For example, participants found satellite and Street View imagery helpful to gauge the accessibility of their travel routes and destinations.

Part 2 and 3 of the study were designed to extract the key desired functionalities of future ALTs. In Part 2, we design and develop three scenarios where ALTs could potentially be used—exploration of neighborhood accessibility, accessibility-aware location search, and accessibility-aware navigation. The scenarios are used to guide the participants in ideating and sketching the designs of future ALTs. In Part 3, we elicited feedback on 12 paper mockups of ALTs that we prototyped. Part 2 and 3 elicited ten key design features (*e.g.*, accessibility-aware location search) for ALTs and six important data qualities for accessibility information (*e.g.*, credibility). These findings were important to guide the design directions of the remainder of this dissertation research. Our accessibility data collection methods use GSV to collect highly granular street-level accessibility information to enable assistive features, such as street-level accessibility visualization and accessibility-aware routing, that are desired by people with mobility impairments.

1.1.2 Crowdsourced Accessibility Data Collection Method

Figure 1.3. Our initial web-based Street View image labeling tool. Labeling images is a three step process consisting of outlining the location of the sidewalk problem in the image, categorizing the problem, and assessing the problem's severity.

We design, develop, and evaluated the novel crowdsourced accessibility data collection method that combines crowdsourcing and GSV. We develop a web labeling system that uses a manually curated database of Street View images (Figure 1.3). Using this tool, we investigate the feasibility of using minimally trained crowd workers from Amazon Mechanical Turk to find, label, and assess sidewalk accessibility problems in these images.

Chapter 4 reports on three studies. Exploratory Study presents a preliminary experiment examining benefits and limitations of three designs of labeling interfaces. Study 1 examines the feasibility of this labeling task with six dedicated labelers including three wheelchair users and three researchers. The study shows that motivated workers can indeed find and label accessibility features in Street View images. Finally, Study 2 investigates the comparative performance of turkers. In all, we collected 13,379 labels and 19,189 verification labels from a total of 402 turkers in Study 2. We show that turkers are capable of finding and labeling an accessibility problem correctly with 81% accuracy. With simple quality control methods, this number increases to 93%.

1.1.3 Semi-Automated Method to Collect Accessibility Data

The sole reliance on paid-human labor for collecting street-level accessibility data can be insufficiently scalable and it remains expensive for creating a large dataset [97]. Building on the work in Chapter 4, we present the first "smart" system, *Tohme*, that combines machine learning, computer vision, and custom crowd interfaces to find curb ramps remotely in GSV scenes. Our approach automatically evaluates the performance of computer vision algorithm and adaptively switch workflow of crowdsourcing tasks based on the predicted computer vision performance. Using 1,086 Street View scenes (street intersections) from four North American cities and data from 403 crowd workers, we show that Tohme performs similarly in detecting curb ramps compared to a manual labeling approach alone (F-measure: 84% *vs*. 86% baseline) but at a 13% reduction in time cost. Our work contributes the first computer vision-based curb ramp detection system, a custom machine-learning based workflow controller, a validation of GSV as a viable curb ramp data source, and a detailed examination of why curb ramp detection is a hard problem along with steps forward.

1.1.4 VGI Data Collection System and Proof-of-Concept ALTs

Finally, we (i) develop, deploy, and evaluate a volunteered geographical information (VGI) system for collecting the street-level accessibility data, and (ii) design and developed two proof-of-concept ALTs to demonstrate the value of the accessibility data collection methods and the street-level accessibility data collected with the VGI system. Informed by our four-year iterative design experience building GSV-based

accessibility data collection tools, we design and develop a VGI system to collect the street-level accessibility data. Between June and July, we invite volunteers via word-of-mouth and asked them to audit the accessibility of the streets in D.C. As of writing this dissertation, we collected street-level accessibility data from 20% of the streets in Washington, D.C. We conduct a preliminary evaluation of the accuracy of the collected accessibility data and show that the overall accuracy of the collected data is 77%.

We show the collected accessibility data's utility via embodiment of two technologies: an online map tool that visualizes Washington, D.C.'s street-level accessibility levels, and the spatial analysis that investigate relationship between neighborhoods' accessibility levels and other neighborhood characteristics like ethnicity and income levels.

1.2 Summary of Contributions

In summary, contributions of this dissertation are:

- Identification of methods that people with mobility impairments use to assess the built environment accessibility and the roles of technologies therein (Chapter 3).
- Identification of ten key design features and six data qualities of future assistive location-based technologies (Chapter 3).
- Design and development of crowdsourcing system to collect street-level accessibility data from GSV (Chapter 4).
- Evaluation of the crowdsourcing system that shows the feasibility of collecting

street-level accessibility data from GSV (Chapter 4).

- Design and development of a semi-automated system that combines crowdsourcing, computer vision-based object detection algorithm, and a "smart" workflow controller that semi-automatically and efficiently detect curb ramps in Street View images (Chapter 5).
- Evaluation of the semi-automated system for detecting curb ramps; we showed that we can improve the efficiency of data collection by 13% without sacrificing data collection accuracy (Chapter 5).
- Design, development, and preliminary deployment of volunteer-based accessibility data collection platform that explores the efficacy of volunteer-based accessibility data collection (Chapter 6).
- Demonstration of two proof-of-concept ALTs, Access Map and accessibility analytics that show the value of the accessibility data collection methods and the collected street-level accessibility data (Chapter 6).

1.3 Thesis Outline

Chapter 2 provides background around the built environment accessibility and situate this dissertation research in existing body of work in crowdsourcing and automated methods to collect data. Chapter 3 describes our formative interview study. Chapter 4 summarizes our work on designing and evaluating crowdsourced street-level accessibility data collection methods. Chapter 5 describes the design, development, and evaluation of Tohme—a semi-automated system that efficiently collect curb ramp data

from GSV. Chapter 6 describes the volunteered data collection system and the design and development of proof-of-concept ALTs. Chapter 7 reviews the contributions of the dissertation and puts forth remaining challenges in scalably collecting accessibility information of the built environment and potential future research directions.

Chapter 2 Background and Related Work

In this chapter, we discuss background and related work that are most relevant to this dissertation. First, we survey literature on street-level accessibility, including: guidelines for accessible sidewalk/streetscape design, people's coping strategies to overcome inaccessible areas, existing methods/tools to assess the built environment accessibility, and neighborhood (accessibility) audit methods. We then survey research on crowdsourcing, especially the topics related to crowdsourced image labeling and volunteered geographic data collection. Finally, we discuss automated methods for increasing data collection efficiency with a focus on technologies that our work builds upon (*i.e.,* computer vision and machine learning-based smart task allocation).

2.1 Sidewalk Accessibility

This section describes the following aspects of street-level accessibility. First, we review what sidewalk and street attributes impede or facilitate mobility for people with disabilities. This information helps inform what accessibility features should be identified in Google Street View (GSV) imagery with the accessibility data collection methods that we design. Second, we review literature on how people with mobility impairments currently assess accessibility of the built environment prior to their travel. Third, we explore what applications and services currently exist to serve accessibility information to people with mobility impairments to identify their advantages and limitations. Finally, we look into current practices around how the accessibility

information about the built environment is collected and compare with our accessibly data collection methods.

2.1.1 Sidewalk Accessibility Barriers and Facilitators

Poorly maintained sidewalks pose considerable accessibility challenges for people with mobility impairments [127,129]. Research in public health and urban planning has studied and identified problems such as missing curb ramps and poorly maintained sidewalk surfaces negatively affects sidewalk accessibility [14,21,71,127,129,152,164,184]. For example, Meyers *et al.* [129] provides a comprehensive list of what constitutes a barrier to navigation. Through telephone interviews and 28 daily telephone contacts, Meyers identified what wheelchair users perceived as accessibility barriers and facilitators in their daily lives such as presence and absence of curb ramps (*e.g.*, lack of curb cuts, obstructed travel paths)—See Table 2.1. This body of prior work informs *what* accessibility needs to be collected and thus inform the design of our system. Note that while Meyers identified significance of other environmental features (*e.g.*, indoor accessibility barriers like narrow corridors), this dissertation focuses on collecting information about the outdoor environment. In Chapter 7, we discuss potential methods for collecting accessibility information in indoor environments as future work.

In the U.S., the Americans with Disability Act (ADA) of 1990 [191] and its revised regulations, 2010 ADA Standards for Accessible Design (2010 Standards) [189], mandates that new construction and alterations of the built environment be free

Environmental	Indoor Built Environment	Outdoor Built Environment	Other
Bad weather or climate	Door handles or pressure	No curb cuts or blocked cuts	Wheelchair problems
No public transportation	No ramps or ramps too steep	No parking	Distance or time (too far to travel)
Traffic (e.g., no crossings)	Narrow aisles	Travel surfaces (grass, mud, ice)	No assistive technology
Landscape (e.g., hills or streams)	Inaccessible bathrooms	Obstructed travel	* Assistive technology
Pedestrian traffic	Broken elevators or lifts or none	High curbs	
Air quality	Counter heights (desks, restaurants)	* Adaptations (curb cuts, special parking)	
Unsafe neighborhoods	Door width	* Accessible parking	
* Accessible transportation	Fixed seating (No space for chairs)		
* Good weather	Floors, floor covering, thresholds		
* Level or graded terrain	* Adaptations (ramps, doors)		

Table 2.1. Meyers *et al.* surveyed a comprehensive list of accessibility barriers and facilitators [129]**. This dissertation is focused on collecting outdoor accessibility information from GSV. A discussion about methods of identifying other accessibility features will be discussed in the Future Work section. Asterisks ("*") indicate facilitators. The grouping is by the author.**

of the aforementioned accessibility barriers and readily accessible for everyone. Rules regarding sidewalk environment are described in Title II and Title III of the 2010 standards. Under the Title II, 28 CFR (Code of Federal Regulations) part 35.151 notes that:

> *Newly constructed or altered streets, roads, and highways must contain curb ramps or other sloped areas at any intersection having curbs or other barriers to entry from a street level pedestrian walkway, […and] newly constructed or altered street level pedestrian walkways must contain curb ramps or other sloped areas at intersections to streets, roads, or highways."*

2004 ADA Accessibility Guidelines (ADAAG) Chap. 4, Title III provides directions for the design of sidewalk attributes. For example: (i) accessible walking

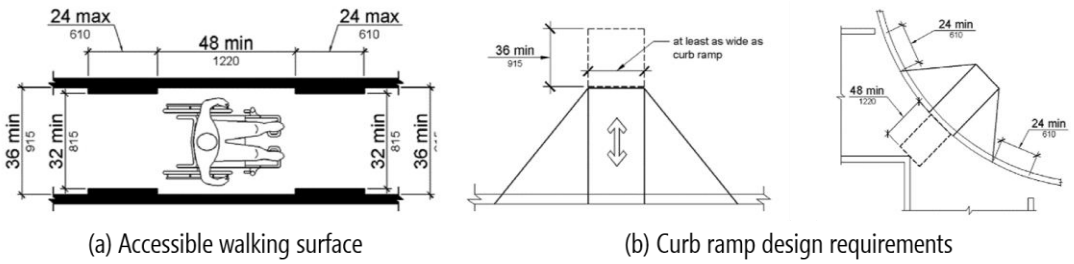


Figure 2.1. Examples of ADA regulations regarding sidewalk accessibility attributes. (a) The regulations ruled that accessible walking pass to have at least 36 inches wide. (b) Enough clearing spaces should be provided at both ends of curb ramps.

surfaces (*e.g.*, sidewalk segments) should be at least 36 inches (915 mm) wide so that wheelchair users can pass through (but the clear width shall be permitted to be reduced to 32 inches (815 mm) minimum for a length of 24 inches); (ii) at least 36 inches of landings at the tops of curb ramps should be provides (Figure 2.1).

To improve the public agencies' compliance with ADA, the regulation mandates the public agencies with more than 50 employees to make a transition plan to improve the accessibility of the built environment (28 CFR §35.150(d)). The plan should accomplish the following four tasks:

> *(i) Identify physical obstacles in the public entity's facilities that limit the accessibility of its programs or activities to individuals with disabilities; (ii) Describe in detail the methods that will be used to make the facilities accessible; (iii) Specify the schedule for taking the steps necessary to achieve compliance with this section and, if the time period of the transition plan is longer than one year, identify steps that will be taken during each year of the transition period; and (iv) Indicate the official responsible for implementation of the plan.*

Even then, many city streets and sidewalks do not meet requirements of inclusive design guidelines for the built environment (*e.g.*, [102,122,188]) and remain

inaccessible for people with mobility impairments after more than two decades of the enactment of ADA. Causes of unmet accessibility needs vary; the reasons include the lack of funding to update the infrastructures, political difficulties in obligating property owners to repair sidewalks, as well as its speed of update (*e.g.*, [43,65,85]).

As noted in the introduction, however, the problem is not just that there are inaccessible areas in the city, but it is also that there are few mechanisms to determine accessible areas of a city a priori [135,195]. In fact, in a recent report, the National Council on Disability noted that they could not find comprehensive information on the "degree to which sidewalks are accessible" across the US [132]. The goal of this dissertation research is precisely to address this issue. We design the methods to collect street-level accessibility information that enable technologies that inform city sidewalk accessibility and support people with mobility impairments.

2.1.2 Coping Strategies for Navigating Inaccessible Built Environment

The existing body of work has identified how people currently cope with the aforementioned accessibility barriers. Prior work suggests that people with mobility impairments rely on their own heuristics [21] or get advice from access consultants [135,167] to find accessible routes. For example, Sobek and Miller described that the Center for Disability Services in their university uses a combination of paper maps and expert knowledge to assist individuals in finding accessible routes between campus origins and destinations [167]. While the service is helpful, the strategy is not always available. Through interview and survey studies, Bromley *et al.* uncovered that people

with mobility impairments tend to employ *avoidance tactics* [21]. They choose to go to accessible areas and stores that they already know and adapt the timing of their travel to avoid crowds based on their heuristics [21]. However, the strategy is only effective when one is familiar with the accessibility of the built environment. Accessibility aware navigation tools could complement the strategy, but they are not widely available [176,186].

A limited body of work has investigated how people use location-based technologies to assess accessibility of the physical environment. One notable work is from Andrea Nuernbergeer's dissertation research in the mid-2000s [135]. Nuernberger studied then-current technological methods and explored desired technical solutions for finding and assessing accessible routes with 20 mobility impaired people [135]. She studied features desired in the future accessibility-aware navigation tools. While informative, Nuernberger focused specifically on navigation tools with less focus on other location-based technologies (*e.g.,* location search). Moreover, Nuernberger's work was conducted in 2005. Given the recent advent of widely available digital maps and GPS-equipped smartphones, it is appropriate to reinvestigate how people use technologies to support their trip planning, and extend the research by exploring designs of a wider variety of assistive location-based technologies. In chapter 3, our formative interview study extends this body of work by introducing how people with mobility impairments use modern location-based technologies to assess the built environment accessibility and what future technologies they desire.

2.2 Existing Accessibility-aware Map Tools

Accessibility-aware map tools such as navigation systems for people with mobility impairments have been available only in limited regions. This is because the street-level accessibility data that is necessary to implement these technologies could not be easily obtained. Our accessibility data collection methods could transform the way the data is collected and served, thereby enable these tools much more pervasively. In this section, we introduce the prior research and development of accessibility-aware navigation systems, point-of-interest (POI) search tools, and GIS analytic technologies. We then discuss how the collected accessibility data could transform these tools.

2.2.1 Accessibility-aware Navigation

Recently, prototype accessibility-aware navigation tools have been designed. Matthews *et al.* and Church *et al.* built map tools to compute accessible routes for wheelchair users in urban areas [37,127]. Later, similar systems were designed by various researchers (including successors to *Matthew et al.*'s tool; *e.g.,* [14,53,137,209,221]). More recently, there have been publicly available web applications like Handimap (www.handimap.org) that compute and show accessible routes for wheelchair users. These tools have been only available in the areas where the accessibility data can be readily obtained (*e.g.,* cities where government provides the accessibility data). Street-level accessibility data collected via our data collection methods could make these navigation tools available to every city where GSV is available.

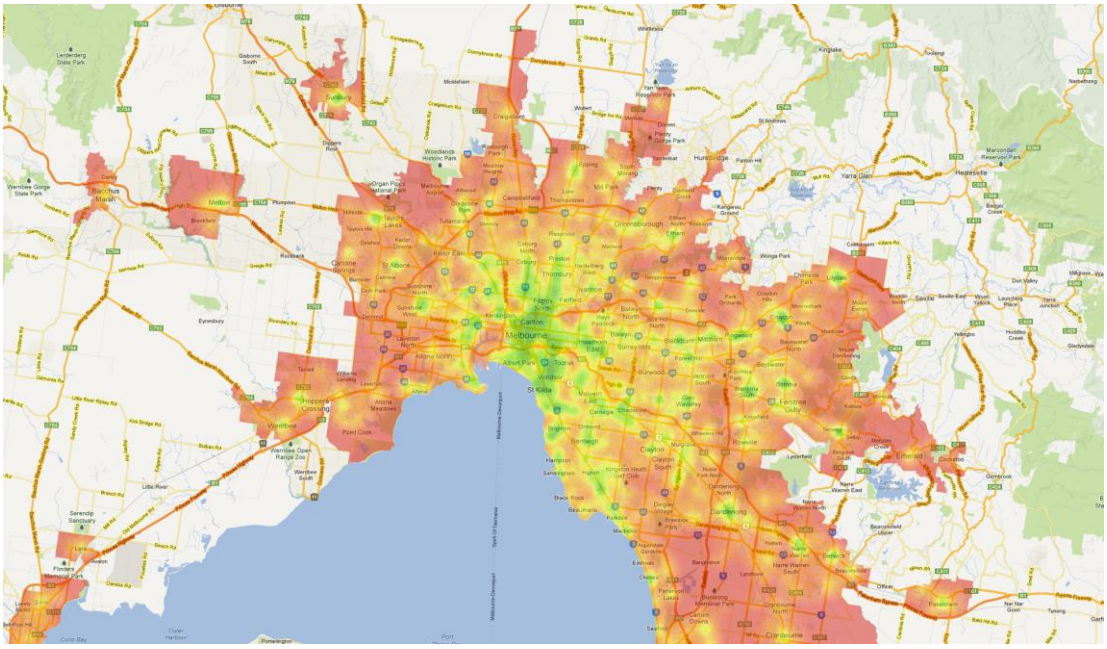


Figure 2.2. Walk Score visualization. Walk Score quantifies the city's walkability by assessing proximity to important amenities (*e.g.,* grocery stores). Green areas represent walkable regions and red areas indicate less walkable areas.

2.2.2 Accessibility-aware POI Search

Goh *et al.* suggested that services that allow users to assess accessibility of a given point-of-interest (*e.g.*, a store, restaurant) could be useful for people with disabilities [52,68,69]. People with mobility impairments could use the services to identify whether the places are navigable. Such services are emerging recently. Ding *et al.* surveyed web applications that allow users to search accessible points-of-interest such as Wheelmap (wheelmap.org) and Factual (factual.com) [52]. Another similar example is AXSMap (axsmap.com). These tools, however, focus on identifying accessibility of the building façade and/or indoor accessibility (*e.g.*, entrance accessibility) and do not take into account of sidewalk accessibility around the points-of-interest which impact people's access to reach the destinations. The street-level accessibility data collected from GSV

with our methods could complement these applications by presenting street-level accessibility around points-of-interest.

2.2.3 GIS-based Analysis Tools

Recent advancement in GIS tools reduced the barriers to conduct geographical analysis of neighborhood characteristics [28,56,98,143,156,222]. For example, Walk Score (Figure 2.2), an online tool that offers an easy-to-understand visualization of walkability of neighborhoods, has been used in public health research to gauge neighborhood quality (*e.g.*, presence of nearby parks) [28,56,98,222]. Walk Score evaluates walkability of neighborhoods by assessing the presence and proximity of 13 types of amenities (*e.g.,* grocery stores) using the data collected from Google Maps. These technologies, however, rely on publicly available GIS data—often selected based on priorities of either private entities or local administrative bodies [157]. This limits generalizability of the techniques—the data often does not include street-level accessibility information. This is exactly what this dissertation tries to solve by introducing new scalable methods of data collection using *remote* crowdsourcing, automated methods, and GSV. To show the value of the collected accessibility data, Chapter 6 of the dissertation introduces a Walk Score-like metrics and visualization that incorporate accessibility information.

2.3 Existing Sidewalk Assessment Methods

In this section, we compare and contrast our street-level accessibility data collection methods with existing neighborhood auditing methods.

2.3.1 Physical Accessibility Audit

Traditionally, in-person neighborhood audits are conducted by local government or volunteer organizations [171,172]. According to one of the audit guidelines [171], neighborhood audits involve inspection of various aspects of neighborhood qualities, which include sidewalk accessibility. While thorough assessments could elicit detailed data about neighborhood environment, in-person audit is time-consuming and its in-situ nature limits the areas where auditors can visit. It also requires a local organizing body to manage inspection personnel, which further limits scalability. In addition, the data collected via physical audit is not always accurate. For example, the information about curb ramp locations (Figure 2.3) collected and distributed by Washington, D.C. government contains some errors as shown in Figure 2.4. The street-level accessibility

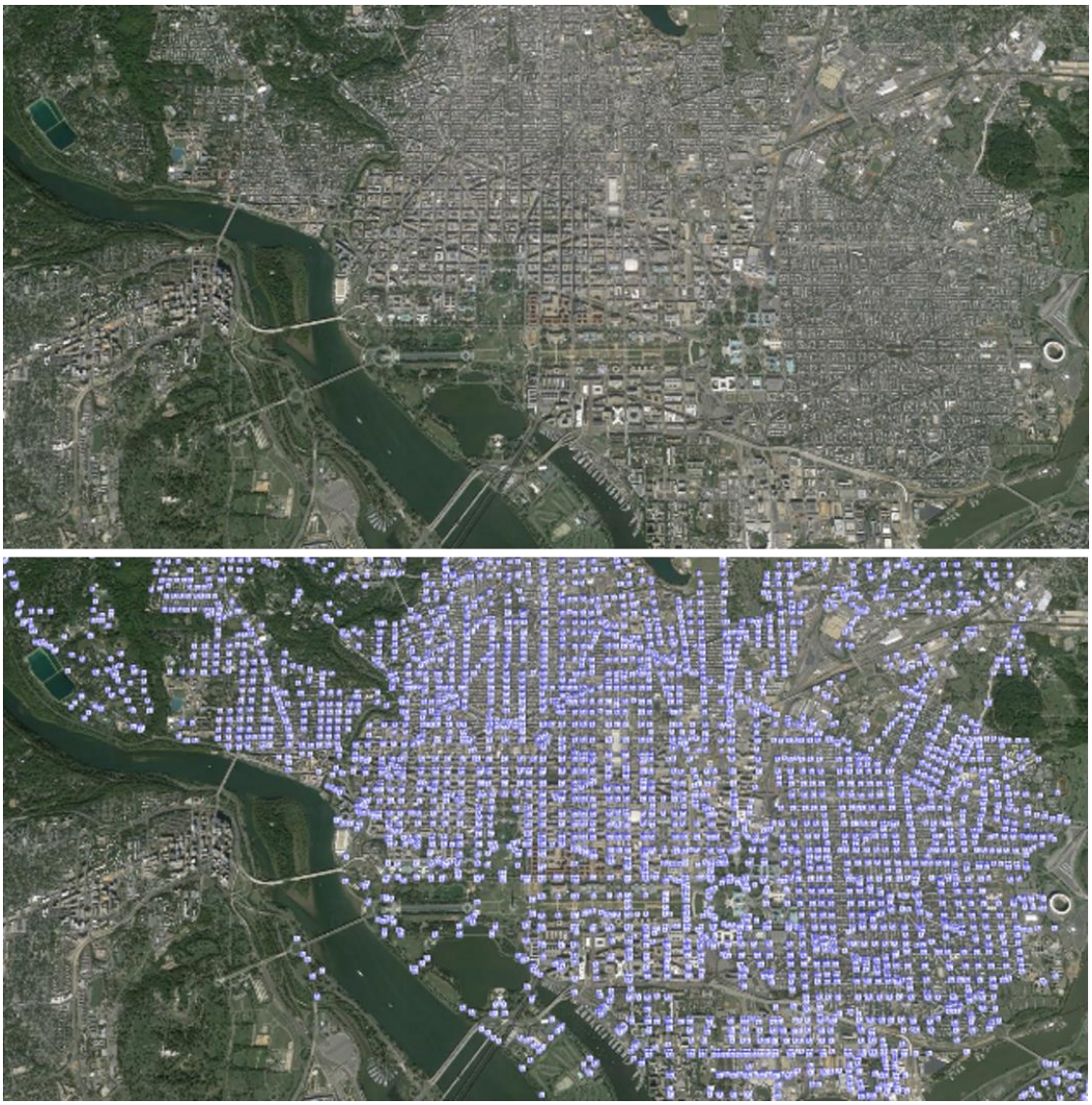


Figure 2.3. The curb ramp location data of Washington D.C. that has been collected and distributed by the D.C. government. The top image shows a raw aerial image of the D.C. area and icons in the bottom image shows placement of curb ramps.

data collected via our data collection methods could be used to cross-reference against the physical audit data to figure out potential locations where errors exist or be used as the primary data collection method (especially for city government that does not track and publish their street-level accessibility data). Furthermore, our data collection

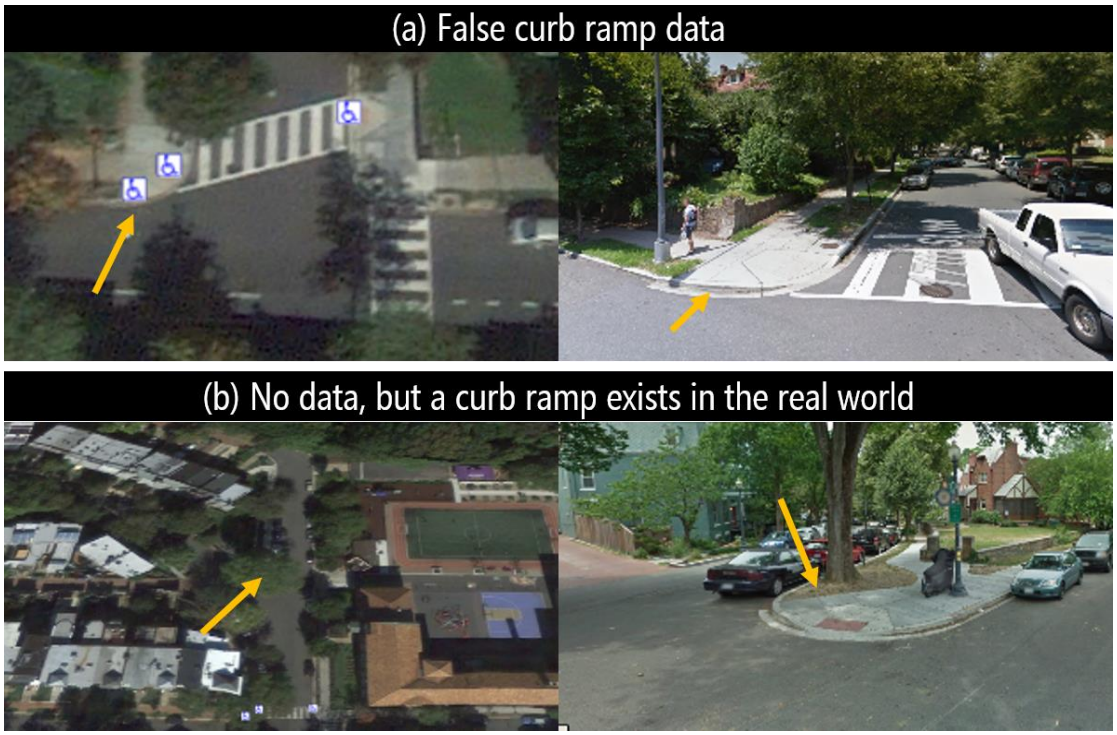


Figure 2.4. Examples of errors in the official curb ramp geographical data of Washington, D.C.: (a) although the official data indicates the presence of a curb ramp, there is no curb ramp in the real world; (b) the official data indicates that there is no curb ramp, but in fact there is a curb ramp at this intersection as we found through our own physical audits of these areas.

methods allow us to gather not only the curb ramp data, but also other accessibility features like missing curb ramps, sidewalk obstacles, and surface problems.

Distributed *in situ* crowdsourcing could alleviate the cost of organized neighborhood auditing. Participatory reporting of neighborhood issues has been accomplished through web and mobile applications (*e.g.*, [27,220]). For example, SeeClickFix allows citizens to report non-emergency neighborhood issues to local government agencies anytime [220]. The emergence of these tools has enabled unorganized neighborhood audits without a central organizing body; people can provide neighborhood information anytime. While SeeClickFix focuses on collecting general neighborhood information, applications such as Wheelmap.org [223],

Axsmap.com [224] focus on collecting information about accessibility of facilities (*e.g.*, presence of accessible entrances at restaurants) and other research prototypes focus on collecting sidewalk accessibility (*e.g.*, [88,128,148,196]). For example, a prototype system that Holone *et al.* developed allows people to collaboratively rate accessibility of outdoor locations, and the system also inform wheelchair users accessible routes using the collaborative collected data [88]. Even then, however, the *in situ* nature of physical audits tend to be time-consuming and labor intensive [39,143]. The coverage of audit data is also limited to the areas where users of the systems can travel. Unofficial audits may also be perceived by local residents as intrusive and can involve safety problems for auditors [29]. Finally, remote accessibility audits conducted by human workers can potentially be substituted by computer algorithms in the future as we discuss in Chapter 5. Therefore, one goal of this dissertation is to provide a *remote* auditing methods and tools that use GSV to complement the physical auditing techniques. See Table 2.2 for the summary of the benefits and limitations of remote and physical accessibility audits.

	Physical Auditing	Remote Auditing
Audit Efficiency	Physical audits are time-consuming and expensive to conduct largely because of the costs of travel [15,29,39,157].	Remote accessibility audits using Street View imagery are less time-consuming as there is no need for trave.
Audit Detail	Able to measure accessibility features' characteristics that are hard to observe solely from pictures [39].	Caution should be exercised when gathering more finely detailed observations (*e.g.,* width of sidewalks) that benefit from observation and measurement in the field [15,39]
Intrusiveness	Physical audits that involve surveying, taking photos, and/or videotaping may be perceived as intrusive by local residents [29,157].	Remote accessibility audits using existing GSV data is less intrusive.
Auditor Safety	Physical audits could involve safety problems for research stuff [157]. Data collection via citizen participation (*e.g.,* SeeClickFix) could be hampered in areas that are perceived dangerous as people get discouraged to walk [117].	Remote accessibility audits using GSV do not involve safety issues.
Computer Vision	Survey data collected from physical auditing cannot be used to train computer vision algorithms.	Accessibility features labeled in GSV could be used to train computer vision algorithms [83,84].

Table 2.2. Summary of benefits and limitations of physical and remote accessibility audits.

2.3.2 Inferring Sidewalk Accessibility from People's Movements

Alternative methods that incorporate people's contribution via natural activities could alleviate the labor-intensive nature of the in-situ techniques described above. For example, Kasemsuppakorn *et al.* [101] and Palazzi *et al.* [137] analyses GPS trajectory data of people's mobile phone to infer placement of sidewalks and potentially accessible routes. More recently, developers from Mapbox (www.mapbox.com) demonstrated the use of data from RunKeeper, a popular self-tracking mobile-application for runners, to map placement of sidewalks [130]. While these solutions may allow us to collect information of sidewalk connectivity, they are not readily scalable beyond the areas that people travel to (*e.g.*, some routes cannot be accessed because of street-level impediments). And, more importantly, analyzing GPS trajectories would only inform path connectivity and not accessibility. That is, people may travel on streets along with driving vehicles even there aren't accessible sidewalks.

Monitoring natural activities of wheelchair users could provide geographical data that is applicable to assistive location-based technologies. Researchers have augmented wheelchairs with motion sensors [93,94]. For example, Iwasawa *et al.* mounted iPod Touches with built-in accelerometers on wheelchairs of nine study participants with mobility impairment [93]. Using the tracked data from the motion sensors and an off-the-shelf machine learning algorithm, they showed that it is feasible to collect some sidewalk accessibility information such as surface conditions. Again, however, the method is only scalable up to the areas that people travel to. This could be problematic as wheelchair users would use their heuristic to avoid inaccessible neighborhoods, which reduces the opportunities to collect data from inaccessible neighborhoods.

2.4 Virtual Street Audit using Google Street View

We were not the first to think of using GSV as a virtual audit medium for cities—indeed, some early work in public health and urban studies began using GSV to assess the neighborhoods' built characteristics [11,15,39,157]. For example, Badland *et al.* investigated feasibility of collecting data related to walking function and cycling function (*e.g.,* walking/cycling surface condition) [11]. Ben-Joseph *et al.* assessed agreement between physical audit and virtual audit data about levelness and condition of sidewalks [15]. As an emerging area of research, work thus far has focused on assessing the validity of GSV as a data source (*e.g.,* assessing effect of data age). Importantly, high levels of concordance have been reported between audit data

collected using GSV versus more traditional means for measures including pedestrian safety, motorized traffic and parking, and pedestrian infrastructure [11,157]. GSV has been validated as a useful dataset for a range of foci within the built environment. Our work reinforced these finding by showing that GSV is a good data source for built accessibility features like presence and absence of curb ramps [77,78,84].

Although strongly related, this dissertation research is different in that we explore the use of minimally trained paid crowd workers and volunteer workers to perform street-level accessibility audit using custom data collection tools. Our work also investigates efficacy of incorporating automated methods to automatically detect sidewalk accessibility features and optimize crowd task workflow to efficiently collect the data.

2.5 Remote Physical Environment Data Collection

In this section, we discuss two areas from the crowdsourcing literature that are highly related to our data collection methods: crowdsourcing image labeling and volunteered geographic information (VGI).

2.5.1 Crowdsourced Image Labeling

The image labeling tasks in our crowdsourced accessibility data collection methods are analogous to that commonly performed in computer vision research for image segmentation, object detection and object recognition [159,210]. Since manually building a large dataset of annotated images for training computer vision algorithms is

expensive and time consuming [159], web-based image labeling tools have been developed to capitalize on the large user population accessible over the Internet (*e.g.*, [4,5,6,159]). For example, in von Ahn *et al*.'s work, textual labels are provided for images through a clever collaborative game-with-a-purpose, where users provide captions to describe objects in an image [4] or draw bounding boxes around specific items [5]. LabelMe [159] provides more granular segmentation by allowing users to draw polygonal-outlines around objects. While the previous tools relied on volunteers, Sorokin and Forsyth [168] experimented with "outsourcing" this task to Mechanical Turk, showing that a large number of high quality image annotations could be acquired relatively cheaply and quickly. Since then, image detection and annotation have become a common task in crowdsourcing platforms and produced datasets used in computer vision communities (*e.g.* ImageNet [50], Caltech-UCSD Birds 200 [201]).

For our data collection methods, image labeling efficiency is contingent on crowd workers' speed and accuracy in processing Street View images. Prior work exists in studying how to efficiently collect image labels (*e.g.*, [51,108,173]). Su *et al.* investigated cost-performance tradeoff between majority vote based labeling and verification based data collection [173], finding quality control via verification improves cost-effectiveness. Recent work by Deng [51] explored methods of efficiently collecting multiclass image annotations by incorporating heuristics such as correlation, hierarchy, and sparsity (*e.g.*, the presence of a keyboard in an image also suggests the presence of correlated objects such as mouse and monitor). Krishna *et al.* introduced methods that increase the efficiency of binary and multi-class image

labeling by an order of magnitude by (i) forcing the crowd workers to label images quickly, to the extent that they make mistakes, and (ii) automatically correcting labeling mistakes *post-hoc* by modeling the errors [108]. While crowd-powered image labeling research is relevant to our work, it is different not only in focus (*i.e.,* finding accessibility features) but also in the unique integration of GSV, crowdsourcing, and computer vision for scalably collecting sidewalk accessibility information.

The cost of data colleciton could be further reduced by getting contributions from volunteers. The last decade has seen significant developments in online citizen science applications such as the range of Zooniverse projects (www.zooniverse.org). In many of the projects (*e.g.,* Galaxy Zoo [114,204,218], Snapshot Serengeti [219]), images that are hard for computer algorithms to process are presented to volunteers to categorize and annotate. The image processing tasks that volunteers are asked to complete are simple enough that members of the public can engage meaningfully with minimal training [45]. These projects showed that it is feasible to motivate volunteers to contribute to data collection tasks for a scientific purpose. Although motivation of the projects are different from ours (*i.e.,* scientific exploration *vs.* accessibility) and our data collection methods involve arguably more complex interactions (*e.g.,* navigating in GSV environment), we follow the approach of above projects and develop a volunteer-based data collection system, then report on the small-scale deployment of the system in Chapter 6.

2.5.2 Volunteered Geographic Information

The goal of this dissertation work aligns with existing efforts of collaboratively mapping the world's geographical information using web tools, the research field known as *volunteered geographic information* (VGI) or *geographic volunteer work* [141]. OpenStreetMap (OSM), arguably the most successful VGI project, aims to create a set of map data that is free to use and editable for everyone [74,75,76]. In his dissertation work, Priedhorsky developed Cyclopath, a web-based mapping application serving the route finding needs of bicyclists in cities in Minnesota [140,141,142]. Similar to OSM, Cyclopath allows any bicyclists to collaboratively provide and edit bicycle route information online. OpenStreetMap had approximately 2,800,000 registered users as of July 2016 [230], and there were 2,184 registered users for Cyclopath as of 2011 [125]. These projects showed the viability and efficacy of eliciting contribution of many anonymous crowd workers to collect geographical information. Although similar in spirit, our work not only differs in the types of data collected (*i.e.,* street-level accessibility information), but it also uniquely shows GSV is a viable source of geographical information, and introduce novel methods to combines crowdsourcing and automated methods to collect data from GSV. We also note that the potential users of our volunteer-based system such as wheelchair users and their caregivers would have self-serving intrinsic motivations (*i.e.,* identifying accessibility own neighborhoods), which may improve their retention [134,153,160].

Though the quality of VGI data has been questioned, research has found that the data in VGI applications are comparable to more traditional proprietary/government

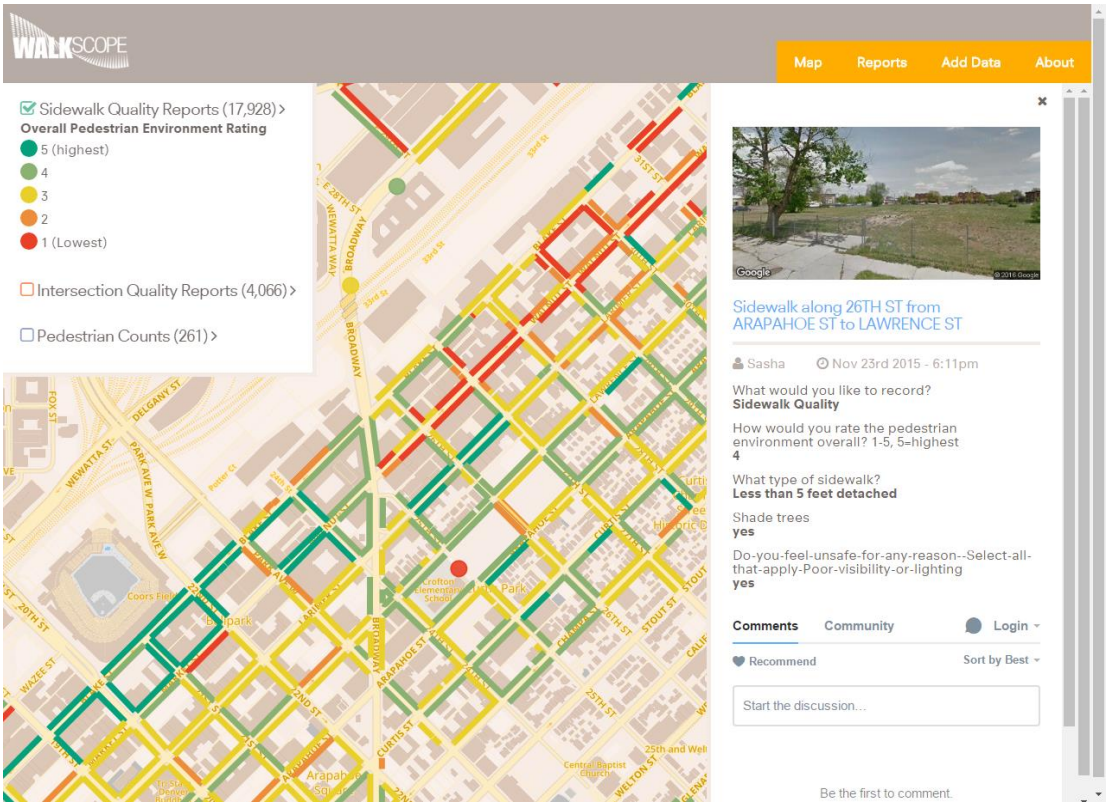


Figure 2.5. WALKscope. The web interface shows a vector layer of the sidewalks (segments) and intersections (dots) in the city of Denver. In the data exploration window, the application visualizes low quality sidewalks and intersections in red and high quality ones in green. In the data editing window, online users can provide information about the quality of the sidewalks and intersections.

data [75]. Research around OpenStreetMap found that the accuracy of collected data was comparable to traditional geographical datasets that are maintained by national mapping agencies [75]. Quality control mechanism similar to those employed in crowdsourcing (*i.e.*, majority consensus is used to maintain Wikimapia data [70]) as well as use of heuristics (*e.g.*, POI data entry of a café in the middle of a historic park tend to be erroneous [70]) should not be present is also used to assure its data accuracy However, its coverage still falls behind that of other official dataset, especially in rural areas and in countries where OpenStreetMap is less popular [133]. This geographic data coverage bias is common in VGI [74,124,145]. In chapter 6, we investigate the

quality and quantity of data collected via our remote data collection methods to see if our volunteer-based methods are viable for collecting geographical information about street-level accessibility.

One notable application that has been launched recently is WALKscope [199]. WALKscope, a VGI application developed by WalkDenver, invites volunteers to provide five-point Likert scale information about sidewalk and intersection quality through a web interface. Its interface visualizes sidewalk vector layers on top of satellite imagery; a user can click a sidewalk segment and rate its quality, as well as provide metadata like presence of obstacles, surface quality, and presence of landmarks (*e.g.,* benches). The added data is used to visualize sidewalk and intersection qualities—see Figure 2.5. While the focus of WALKscope is to provide useful data for pedestrians in general, the optional metadata about sidewalk obstacles and surface quality could be useful to inform the sidewalk accessibility for people with mobility impairments. The application, however, does not allow a user to see sidewalks from street-level (*e.g.,* via Google Street View), which makes it hard to observe the quality of sidewalks and limits remote contributors' ability to report sidewalk quality/accessibility.

2.6 Increasing Scalability with Automated Methods

Crowdsourcing accessibility data collection using GSV is labor intensive. Researchers in the crowdsourcing field believe that crowd-powered systems can be combined with automated methods to reduce the workload and increase productivity of the crowd work

[103]—an area of research that is often referred as *human in the loop*. Our work described in Chapter 5 relied on computer vision and machine learning-based automatic workflow controller to reduce cost of crowd work. Below, we discuss prior work in computer vision and automatic task allocation that our work builds upon.

2.6.1 Computer Vision

There is a growing body of research applying computer vision (CV) techniques to GSV [206,207,211,212,213]. For example, Xiao *et al.* introduced automatic approaches to model 3D structures of streetscape and building façades using GSV [206,207]. Zamir *et al.*[211,212,213] and Lin *et al.* [112] showed that large-scale image localization, tracking, and commercial entity identification are possible [113,211,212,213]. This work demonstrates the potential of combining computer vision with GSV. Varadharajan *et al.* developed computer vision system to track street condition from self-made image dataset similar to GSV to track cracked road surfaces [192]. However, research that focus on (semi-)automatically detecting accessibility features from online imagery has been limited. Notable exceptions are recent work by Ahmetovic *et al.* and Koester *et al.* [105] that introduced techniques that use computer vision algorithms to detect and localize crosswalks in satellite imagery in Google Maps and Street View imagery [3]. Ahmetovic *et al.* state in their paper that the collected crosswalk data could be used to populate geographical database to design assistive technologies to support people with visual impairments. This suggests increasing interests in using computer vision and online map imagery like GSV to populate geographical database.

Our work in semi-automatically detecting curb ramps builds on top of existing object detection algorithms from the CV community [48,62,194]. In our study, we used Deformable Part Models (DPMs) [62,63], one of the top performing approaches in the PASCAL Visual Object Classes (VOC) challenge, a major object detection and recognition competition [62]. Despite a decade-long effort, however, object detection remains an open problem [19,202]. For example, even the DPM, which won the "Lifetime Achievement" Prize at the aforementioned PASCAL VOC challenge, has reached 30% precision and 70% recall in 'car' detection [62].

2.6.2 Automatic Task Allocation

Our semi-automated system uses machine learning to control the image labeling workflow for efficiently collecting data from GSV (Chapter 5). Typical workflow adaptions include: varying the number of workers to recruit for a task [99,202], assigning stronger workers to harder versions of a task [18,47] and/or fundamentally changing the task an individual worker is given [95,111]. These workflow decisions are made automatically by workflow controllers often by analyzing worker performance history, inferring task difficulty, or estimating cost.

Most relevant to our work on semi-automated accessibility data collection is workflow adaptation research in crowdsourcing systems [99,111,202]. For example, Lin *et al.* and Welinder *et al.* rely on worker performance histories to either assign different tasks [111] or recruit different numbers of workers [202]. More similar to our work is [95,99] that infer task difficulty via automated methods and adapt work

accordingly. For example, Kamar *et al.* [99] analyzed image features with CV algorithms to predict worker behaviors a priori on image annotation tasks and used this to dynamically decide the number of workers to recruit. More recently, Gurari *et al.* introduced a framework that combines human-based and automated image segmentation [72]. In their framework, a prediction module predicts quality of automatic image segmentation; the module then decides to delegate image segmentation task to human or automatic segmentation.

Though similar, our work is different both in the problem domain (*i.e.*, finding accessibility attributes) as well as in approach. Rather than vary the number of workers per task, our workflow controller infers CV performance and decides whether to use crowd labor for verifications or labeling. In addition, we do not simply rely on image features or CV output to determine workflow but also contextual information such as intersection complexity and 3D-point cloud data.

2.7 Summary

This chapter has described background and related work of three areas of research that are most relevant to this dissertation. Our work complements and extends the existing sidewalk accessibility data collection methods by introducing novel ways to remotely collect street-level accessibility data from GSV. We surveyed the existing technologies that our data collection methods rely on, and how the techniques we designed differ from existing ones.

Chapter 3 Formative Interview Study

In this chapter we describe a formative interview study with 20 people with mobility impairments. Our goal is to investigate the current methods that people with mobility impairments use to assess the accessibility of the physical environment and to explore the future design of assistive location-based technologies. This chapter is based on our CHI2016 publication [79].

3.1 Introduction

Accessibility barriers in the built environment pose significant problems for people with ambulatory disabilities [21,26,65,71,91,127,129,152,177]. Knowing where and what barriers exist can help affected travelers mitigate, prevent, or better prepare for such problems [21,127,135,167]. Previous research has identified common strategies people with mobility impairments use to evaluate the accessibility of routes and destinations *a priori* (*e.g.*, seeking trip advice from caregivers [135,167]); however, this work either occurred before the modern era of location-based technologies like GPS-enabled smartphones or did not focus on the potential role of technology.

In this chapter, we investigate current methods and tools—both technological and non-technological—that people with mobility impairments use to evaluate the accessibility of the built environment (*e.g.*, streets, businesses) as well as to plan and execute travel. Through participatory design, we actively engage our participants in brainstorming and designing the future of what we call *assistive location-based technologies* (ALTs)—location-based technologies that specifically incorporate

accessibility features to help people with impairments explore, search, and navigate the physical world. As exploratory work, our research questions include: What modern technologies do people with mobility impairments use to evaluate the accessibility of the built environment? What role does technology have in making decisions about travel—both *a priori* (*e.g.*, when planning) and *in situ* (*e.g.*, when moving about)? How could future technologies be designed to further improve the way they navigate the physical world?

To address these questions, we conducted a three-part study with 20 mobility-impaired participants: a semi-structured interview (Part 1), a participatory design session (Part 2), and a design probe activity (Part 3). The semi-structured interview was designed to investigate current methods and tools that people use to plan trips and assess the accessibility of the built environment. In Part 2, we designed and developed three ALT usage scenarios, which were used to help guide the participants in ideating and sketching new ALT designs: interactive exploration of neighborhood accessibility, accessibility-aware location search, and accessibility-aware navigation. In Part 3, we presented 12 researcher-prepared paper mockups of ALTs and elicited feedback.

Findings from Part 1 reinforce and extend previous research in how people with mobility impairments assess accessibility [21,135,167]. We found that, while planning trips remains a challenge, modern location-based technologies support people with mobility impairments—even if not designed specifically for that purpose. For example, participants found satellite and Google Street View (GSV) imagery helpful to gauge the accessibility of their travel routes and destinations. Part 2 elicited ten key design

Figure 3.1. To explore how location-based technologies currently support users with mobility impairments as well as to examine desired future interfaces and uses, we conducted a three-part formative study with 20 mobility impaired participants. Above, photos from (a) a semi-structured interview, (b) a participatory design activity, and (c) a design probe.

features (*e.g.*, top-down maps of streets depicting accessibility information) for ALTs and five important data qualities for accessibility information (*e.g.*, credibility, frequently updated data). During our design probe in Part 3, participants reacted positively to our mockup, especially glanceable visualizations of indoor/outdoor accessibility and accessibility-aware routing interfaces, and provided design suggestions. Another data quality emerged in Part 3.

The contributions of this chapter include: (i) an examination of methods and modern tools that are used to assess the accessibility of the built environment; (ii) an analysis of ALT mockups designed by mobility impaired people; (iii) the first examination of the significance of data quality on ALTs; (iv) findings from mobility impaired people's reactions to 12 envisioned ALT interfaces. By enumerating key features and data qualities of ALTs, our findings should inform the design of future location-based tools—both general tools such as Google Maps or Yelp as well as specialized tools such as WheelMap [223] or AXSMap [224]) aimed at the accessibility community.

	Sex	Age	Phone	Technology	Disability
P1	F	48		Cane	Cerebral Palsy (affects fine motor control of both hands and feet)
P2	M	37		MW, EW	Cerebral Palsy (weak legs, restricted to using a wheelchair)
P3	M	48	SP	MW	Spinal Cord Injury (C5/6)
P4	F	22	SP	Scooter	FSH Muscular Dystrophy
P5	M	56	SP	MW	Spinal Cord Injury (L1/T12)
P6	F	77		Cane	Muscular weakening disease
P7	F	42	SP	Cane, Scooter	Juvenile Rheumatoid Arthritis
P8	F	72		Walker	Damaged patella tendon
P9	F	38	SP	EW	Muscular Dystrophy
P10	F	72		Walker, EW	Parkinson's disease
P11	F	24	SP	Scooter	Spinal Muscular Atrophy (Type 3)
P12	F	26		Walker, EW	Cerebral Palsy (poor balance and no depth perception)
P13	F	24	SP	Cane, walker	Diplegic Cerebral Palsy (affected muscle tightness in legs)
P14	F	56	SP	EW	Multiple Sclerosis
P15	M	52	SP	Walker	Cerebral Palsy, Knee injury
P16	F	37		MW	Cerebral Palsy
P17	F	31	SP	Walker, MW	Spinal Cord Injury (T-6)
P18	M	63	SP	MW	Spinal Cord Injury (T-11)
P19	F	19	SP	Cane, crutches	Hip replacement
P20	M	29	SP	MW	Spinal Cord injury (L-1)

Table 3.1. Participant demographics. Here, we use: MW=Manual wheelchair, EW=Electric wheelchair, and SP to indicate participants who have smartphones. P16 was excluded due to a cognitive impairment that prevented her from fully participating.

3.2 Method

We conducted a three-part study with mobility-impaired participants: (i) a semi-structured interview to inquire about current methods and tools that our participants use to support trip planning (Part 1**;** Figure 3.1a); (ii) a participatory design to elicit design and feature requirements of ALTs as well as their context of use (Part 2; Figure 3.1b); and (iii) a design probe activity to discuss designs and features of ALT paper prototypes designed by the researchers (Part 3; Figure 3.1c). Study sessions were audio and video recorded and transcribed by the members of the research team.

We recruited 20 participants (14 female) on a rolling basis through local accessibility organizations, word-of-mouth, and email listservs. Participants were on average 43.7 year old (*SD*=18.0; *range*=19-77; Table 3.1) and from the Washington,

D.C. area. To investigate potential differences in perspective and experience based on mobility level, we specifically recruited a range of participants [190]: 8 used electric wheelchairs/scooters, 7 used manual wheelchairs, and 10 used other manual assistive technologies (*e.g.*, cane, walker). The total number (25) exceeds 20 as some participants used more than one assistive device—see Table 3.1. All participants had experience with using laptop/desktop computers and 13 had smartphones. Prior to the study session, participants were asked to fill out an online background survey. Due to a cognitive impairment, which prevented full participation, P16 is excluded from our analysis. Participants were compensated $15/hour for their time and travel.

3.2.1 Part 1: Semi-structured Interview

Part 1 of our study was aimed at uncovering: (i) what accessibility challenges people with mobility impairments face in the built environment and the significance of these challenges; (ii) the tools and methods they use to assess accessibility; and (iii) how the problems impact their decision and ability to travel. The interviewer initiated inquiries with a fixed set of questions. As new topics emerged in accordance with participant's background, mobility level, and experience, participants were asked to elaborate on emerging topics.

3.2.2 Part 2: Participatory Design

We used participatory design with end-user sketching [182] to better understand what types of interactive designs, features, and uses people with mobility impairments desire for future ALTs. To help guide the design activity, we used scenario-based design

[34,155] with three scenarios; each scenario described a situation where ALTs could be helpful for evaluating the accessibility of the built environment. Our scenarios are based on GIS literature [13,58,147] that taxonomize location-based applications into three main areas: geographical exploration, search, and navigation. The scenarios were then adapted to an accessibility context. Before conducting the study, we refined our scenarios iteratively within our research team and later with a research partner who uses an electric wheelchair (*Age*=47; Male; SCI level C5). The three scenarios are:

- ***Scenario 1: Citywide Accessibility Exploration.*** *You are planning to rent a room in an unfamiliar city that you will move to in a few months. Imagine that there is a website that provides accessibility information about the city. What should that website look like?*
- ***Scenario 2: Accessibility-Aware Location Search.*** *Your friends are visiting you, and you want to take them to an Italian restaurant in Washington, D.C (your hometown). You would like to find a popular restaurant. You also want to make sure the business and its surrounding areas are accessible for you. What should the application look like?*
- ***Scenario 3: Accessibility-Aware Navigation.*** *You came to an unfamiliar city for your holiday. You remember there is a natural science museum that you want to visit. You open a navigation tool on your computer to find accessible routes from your hotel to the museum. What should the application look like?*

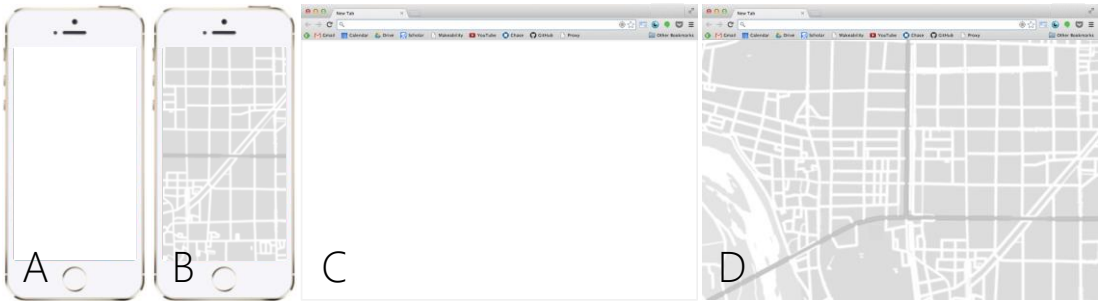


Figure 3.2. The four templates for sketching: (a) a blank mobile, (b) a map on a mobile, (c) a blank web browser, and (d) a map on a web browser.

To help our participants ideate and sketch design ideas, we prepared four paper templates, which they could use at their discretion: (i) a blank smartphone, (ii) a map on a smartphone (iii) a blank web browser, and (iv) a map on a web browser (with/without pins)—see Figure 3.2. While our templates are based on widely available technologies and familiar map interfaces, we did not restrict our participants from brainstorming ideas that use other user interfaces (*e.g.*, augmented-reality devices like Google Glass, smartwatches). Participants were asked to "think aloud" while sketching. Five participants were not comfortable sketching by themselves due to weak upper body strength. In these cases, the participants described their ideas and the interviewer sketched on their behalf.

3.2.3 Part 3: Design Probe

For Part 3 of our study, we designed 12 low-fidelity, paper-based prototype mockups of ALTs ranging from heat map visualization's of a city's accessibility to indoor navigation interfaces that provide accessible routes (Figure 3.3). Prior work suggests that using lower-fidelity interface representations in user studies elicits more honest

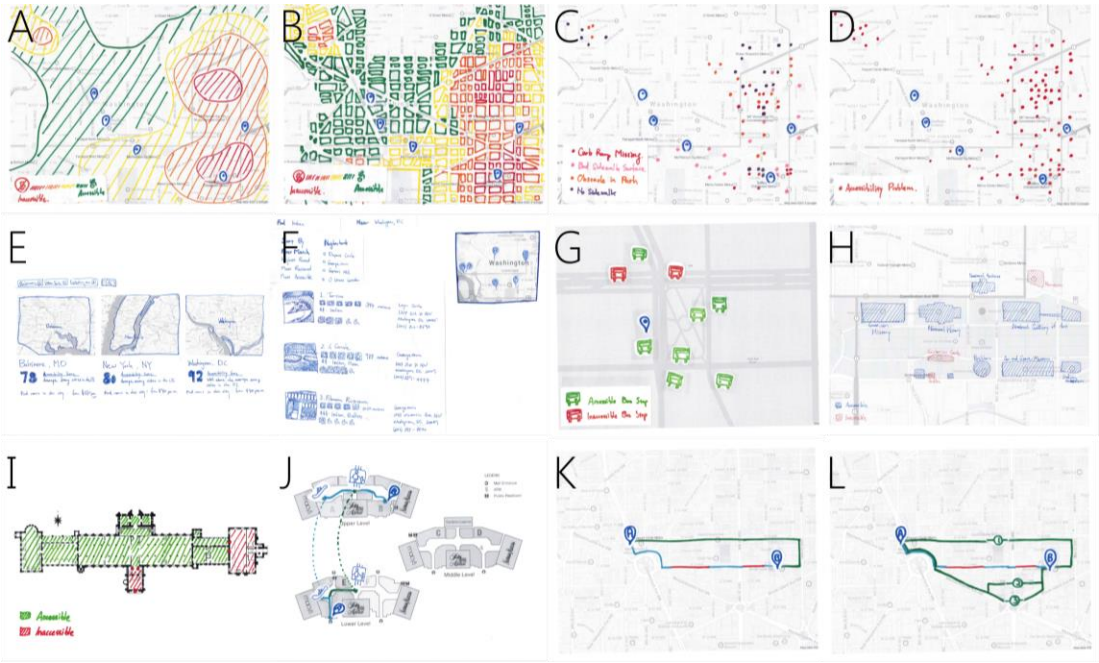


Figure 3.3. We demonstrated the twelve paper prototypes of ALTs to participants in Part 3 of the study. (a-d) street-level accessibility visualizations, (e) citywide accessibility score comparison, (f) accessibility-aware location search, (g) bus stop accessibility visualization, (h-j) building accessibility, and (k-l) outdoor wayfinding. The high resolution version of the prototypes are available as supplementary material.

feedback [109,151] and that the presentation of multiple, alternative design solutions reduces inflated praise and gives rise to stronger criticism, when appropriate [182].

Our mockups were used as *design probes* to elicit reactions, prompt critical feedback, and ground discussions. Similar to our scenarios in Part 2, probes were iterated upon within our research group and with our mobility impaired research partner before beginning our study. Some probes utilized fictitious '*Accessibility Scores*,' which were inspired by walkscore.com. *Walk Score* provides a number between 0-100 that represents the walkability of a given address; the score is based on proximity to destinations such as restaurants, libraries, and parks as well as population density and road metrics such as block length and intersection density [222]. While similar,

Accessibility Scores extend Walk Scores with notions of accessibility (*e.g.*, the presence of sidewalks and curb ramps, road grade, frequency of elevation changes, and sidewalk width). Some features of our probes, such as how accessibility scores are computed, are left intentionally vague to help provoke discussion. Below, we describe our 12 probes categorized into six groups. For readability, we refer to the specific probes in parentheses without the Figure prefix.

1. Accessibility Score Visualizations. We developed four top-down, map-based visualizations of accessibility scores to provide 'at-a-glance' information on a city's accessibility. Two probes used heat map representations with different granularities: neighborhood-level (3.3A) *vs.* sidewalk-level (3.3B). The two other probes used dots to represent specific accessibility barriers, both categorized (3.3C) and non-categorized (3.3D).

2. Citywide Accessibility Score Comparison. While the above visualizations are useful for exploring the general accessibility of a city or neighborhood, they do not easily support comparing the accessibility of different cities. This probe quantifies the accessibility of entire cities with a single accessibility score along with brief, textual rationale. Multiple cities can be entered/compared (as in 3.3E).

3. Accessibility-Aware Location Search. In this probe, we developed a point-of-interest search website similar to yelp.com but augmented with accessibility information. Users can search for a business or other point-of-interest with a keyword

and location. Each search result is accompanied by a 5-level accessibility score, which can be used for sorting and filtering (3.3F).

4. Finding Accessible Bus Stops. We developed one probe targeted at public transportation—in this case, finding accessible bus stops (3.3G). Users can enter a location and see proximal bus stops, which are color-coded based on accessibility (green for accessible, red for inaccessible).

5. Visualizing Building Accessibility. We developed three design probes for investigating the accessibility of buildings. The first design uses a top-down map visualization to indicate the accessibility of public buildings in a selected area (3.3H). Selecting a building zooms into its floor plans and highlights accessible and inaccessible features such as elevators and stairs (our second probe, 3.3I). The third design focuses on accessible routing interfaces for indoor environments (3.3J).

6. Outdoor Accessible Routing. Finally, our last category contains two probes related to accessibility-aware pedestrian routing algorithms and interfaces. Similar to Apple or Google Maps, both probes allow the user to enter a start and end location and view suggested routes. In our designs, however, the shortest path is visualized as well as the shortest *accessible* path. The probe in 3.3K shows one alternative accessible path while 3.3L shows multiple alternatives.

3.3 Data and Analysis

Each session lasted an average of 77.9 minutes (*SD*=16.3; range=53-119). Sessions were audio/video recorded and transcribed by the research team. We used iterative coding [20,89] to examine the transcripts, including the responses to semi-structured interview questions, verbal descriptions of participants' sketched prototypes, and feedback on our design probes. Our unit of analysis was a participant's response to the interviewer's question.

We iteratively refined the codes to ensure the code set was comprehensive and reliable. First, a member of the research team open coded the interview transcripts of the first three participants (P1-3), who used different types of assistive technologies. Similar and recurring ideas were grouped to create the initial codebook. Using this codebook, two researchers independently coded the same interview transcripts (P1-3). We used Cohen's kappa (κ) [42] to assess inter-coder agreement. The mean agreement was κ=0.40 (*SD*=0.14; *range*=0.04-0.6). Landis and Koch suggested that scores of $\kappa <$ 0.6 are at most moderate agreement [110]. In our case, 13 of 14 codes for Part 1 through 3 were < 0.6. The two researchers then met, resolved all disagreements, and updated the codebook accordingly. A second set of transcripts (P4-6) were selected and the coding process was repeated. This time there was much higher agreement: average κ=0.88 (*SD*=0.07; range=0.69-1.0). Again, disagreements were resolved through

Part 1. Semi-structured Interview
Accessibility barriers and enablers
Feelings about accessibility enablers and barriers
Methods or tools for overcoming accessibility problems
Impact of accessibility enablers and barriers
Methods or tools to assist with trip planning
Methods or tools to assist with evaluating accessibility
Part 2. Participatory Design and Part 3. Design Probe
Accessibility barriers and enablers
Context of use
Design of user interface
Accessibility data quality

Table 3.2. The final codebook. Though originally separate, Part 2 and Part 3 eventually shared the same codebook after iterations.

consensus and the codebook was updated a final time. One researcher coded the remaining transcripts using the final codebook (Table 3.2).

3.4 Findings

3.4.1 Part 1: Semi-Structured Interview

We discuss what and how accessibility barriers and facilitators affect mobility impaired people's lives and describe methods and tools they use to cope with problems. We use the phrase 'accessibility facilitators' to describe built environment or inter-personal features (*e.g.,* curb ramps, a helpful restaurant employee) that allow people to overcome the barriers [129,152].

Accessibility Barriers and Facilitators

Participants were asked about mobility challenges and anxieties for trips. All participants except for P15, who had strong mobility, mentioned at least one type of barrier. Overall, 17 barriers and facilitators emerged, which we categorized into

	Barriers and Facilitators	EW/S (N=8)	MW (N=6)	MAT (N=10)	All (N=19)
Outdoor	Leveled Ground	7 (88%)	5 (83%)	8 (80%)	15 (79%)
	Surface Type	6 (75%)	5 (83%)	7 (70%)	14 (74%)
	Curb Ramp	7 (88%)	5 (83%)	6 (60%)	14 (74%)
	Gradient	3 (38%)	5 (83%)	5 (53%)	11 (58%)
	Narrow/Obstructed Path	5 (63%)	6 (100%)	4 (40%)	11 (58%)
	Presence of Sidewalk	3 (38%)	6 (100%)	3 (30%)	9 (47%)
	Distance	0 (0%)	1 (17%)	5 (50%)	5 (26%)
Indoor	Elevator	4 (50%)	5 (83%)	8 (80%)	13 (68%)
	Entrance	6 (75%)	4 (67%)	4 (40%)	11 (58%)
	Restroom	4 (50%)	4 (67%)	4 (40%)	8 (42%)
	Accommodation	2 (25%)	3 (50%)	2 (20%)	5 (26%)
Other	Accessible Transportation	7 (88%)	5 (83%)	6 (60%)	14 (74%)
	Parking	3 (38%)	3 (50%)	6 (60%)	9 (47%)
	Stairs	2 (25%)	2 (33%)	5 (50%)	9 (47%)
	People's Attitude	3 (38%)	2 (33%)	3 (30%)	6 (32%)
	Crowded Area	1 (13%)	2 (33%)	2 (20%)	4 (21%)
	Weather	1 (13%)	2 (33%)	1 (10%)	4 (21%)

Table 3.3. The accessibility barriers and facilitators mentioned by the participants. Cells are shaded by response rate (darker shade=more frequent). *EW/S*=Electric wheelchair and scooter users, *MW*=Manual wheelchair users, *MAT*=Manual assistive technology users.

outdoor, *indoor*, and *other* in Table 3.3 The most prominent accessibility attribute for each category included: leveled ground (*e.g.*, steps, curbs) for outdoor, elevator for indoor, and accessible transportation for other (*e.g.*, paratransit, accessible buses). Note that leveled ground is different from a street or sidewalk's gradient (*i.e.*, steepness), which is its own distinct attribute.

The perceived severity of the identified accessibility barriers seemed to differ depending on the participant's mobility. For example, while six participants described distance as a barrier to navigation, five of these used manual assistive technology. For example, one cane user said: *"I can do grassy [surfaces]. But I need short distances and I need no stairs."* (P19), All manual wheelchair users mentioned the presence of sidewalks and unobstructed paths to be important facilitators of their movement. A participant who uses a manual wheelchair said: *"[At] some locations, […] sidewalks*

[are] narrow and for some reason have light poles right in the middle of a sidewalk, so I can't get through at all." (P3).

Impact of Accessibility Barriers

When asked about accessibility barriers and their impact, fifteen participants mentioned that barriers affected their travel decisions—both where to travel and whether to travel at all. For example, P17, who uses a manual wheelchair, said: *"I'll forgo going there if I can't confirm there is some sort of sidewalk for me to travel along."* In addition, nine participants discussed how accessibility affected their mode of travel: *"in New York City, the subway stations are not accessible […] so that was out of the question for us"* (P7). Seven reported that their decisions on where to stay/live depend on accessibility. For example: *"there's been a couple of hotels we've gone to where the actual door to the hotel wasn't accessible, so we've had to pick another hotel"* (P20). Finally, three participants mentioned how accessibility barriers socially excluded or separated them from others. P9 said: *"I wanted to go to a party and my friends are there and I can't go because I get there and [I find the place to be inaccessible]."*

Methods to Overcome Accessibility Barriers

Strategies to overcome the aforementioned accessibility problems organically emerged in the interview. Five strategies included: help from others, physical strength, detour, walk/roll onto the street, and setting expectations. Thirteen participants said they could rely on others: *"occasionally if it's not accessible, my husband can help me up steps"* (P7). Ten participants mentioned that they use physical strength to overcome barriers.

Among these ten, seven used manual assistive technologies. P8 said*: "my husband ran my walker down to the bottom and then I walked down holding on [railing]."* Seven noted they took detours when they encountered barriers. P17 said: *"the bus stop was on grassy hill. So I didn't get off there. I had to go up a few stops and of course it was past the actual shopping plaza."* Six mentioned they walk/roll on the street when sidewalks are not passable: *"they are digging up the sidewalks. And they force us to use either the sidewalk on the other side, or you're forced to be on the street."* (P1).

Methods for Accessibility Evaluation

Participants were asked how they plan their trips to unfamiliar locations and assess accessibility. "Low-tech" solutions included: talking to others, relying on heuristics, and performing an on-site accessibility audit. Participants used technologies to assess accessibility as well, including: websites and online forums, online imagery, and existing location-based technologies. We expand on each below.

Talking to Others: The most common method of assessing accessibility was talking to others (N=17). Our participants spoke with coworkers, friends and family members, employees who worked at their destinations, and accessibility consultants who knew about the accessibility of the facilities. P17 said: *"If a friend has been there, I'll ask 'do you remember if there was a little step' or 'do you remember what the access was [like].'"*

Heuristics: Our participants (N=11) relied on their experience and educated guesses to gauge the accessibility of places prior to or in lieu of travel. For example, P7 described: *"those towns that are historic, I just tend to stay away from them completely."*

On-site Accessibility Audit: When necessary, the participants (N=7) checked routes and neighborhood accessibility on-site. P12 said:

> *"If it's an important trip but I don't want to use [paratransit], what I will do is a dry run the day before: get lost and find my own landmarks and do it the next day where I will usually get lost again but not as badly."*

Websites and Online Forums: Fifteen participants noted they acquire accessibility information of the built environment from websites of hotels, restaurants, and other business facilities. Online forums were used to assess the areawise accessibility of neighborhoods and cities.

> *"I'm trying to find out if [places are] accessible, then I will usually use their website or Google. But if I'm trying to go to an entire area like Adams Morgan or checking out an entire area, I'll use forums. […] people can be like "oh this area and this street is cute but doesn't have the cobblestones." (p4)*

Online Imagery of the Built Environment: Eleven participants reported the use of online imagery of the built environment like GSV, satellite imagery, and building façade pictures found online. P20 said:

> *"I use Street View of Google. What that does is it gives me an idea. If there's any steps outside of the facility or outside of the place, I'm able to tell right away from Google Street View, or satellite or anything like that."*

Existing Location-based Technologies: Six participants reported that they use accessibility features of existing location-based technologies. P17 said: *"I used Yelp to find my restaurants, and I always go to the [indicator describing] wheelchair access. It's great that that's there, yes or no [for wheelchair access]."* As for emerging ALTs, while 2 participants knew about AXSMap, they did not use the application due to coverage area and data sparseness.

Combining Strategies. Finally, participants used not just a single method, but combined two or more strategies to crosscheck accessibility. For example, when asked about his preferred method, P20 said:

> *"I guess it would be a combination, there isn't an actual preference. Because then there's a flaw in each one. Street View is not always updated, and the perception of the person I'm talking to that's unfamiliar with my situation, they don't know exactly what I mean."*

Part 1 Summary. Our findings highlight common accessibility barriers and facilitators in the built environment, the impact of those barriers, and methods to mitigate or avoid accessibility problems, which reaffirm and extend prior work (*e.g.*, [129,135,152,167]). We also uncovered how modern technology is used to assess accessibility (*e.g.*, online imagery).

3.4.2 Part 2: Participatory Design

Participants were asked to sketch ideas and describe the design of future ALTs. We grouped recurring, emergent features of ALTs into 10 categories. We also describe five emergent data qualities [200] important to ALTs.

Common Solutions

Overall, participants sketched and described ten different features for envisioned future ALTs. For the first scenario (citywide accessibility exploration), the top four most frequent features included: Street-level Accessibility Visualization (N=12), Detailed Description (9), Routing (6), and Transportation (6). For the second scenario (accessibility-aware location search), participants wanted Detailed Description (13), Point-of-Interest Accessibility Rating (7), Remote Accessibility Inspection (7) and Floor Plan. Finally, for the third and final scenario (accessibility-aware navigation), participants wanted Routing (14), Transportation (8), Street-level Accessibility Visualization (7), and Remote Accessibility Inspection (4). We describe all ten emergent features below.

Features

Street-level Accessibility Visualization: Fourteen participants sketched or described top-down map tools that visualize accessibility barriers and facilitators in streets/sidewalks. These map-based visualizations were highly desired because, as our participants noted, they allow users to quickly explore the accessibility of a large area. P7's sketch in Figure 3.4, for example, shows the presence of curb ramps as blue pins in a mobile map interface. Color was often used to either represent types of accessibility attributes or the severity of an accessibility barrier.

Point-of-Interest Accessibility Rating: While the previous feature provides a way to browse street-level accessibility, eight participants wanted accessibility ratings of individual buildings (*e.g.,* Figure 3.4d). Our participants thought that these ratings

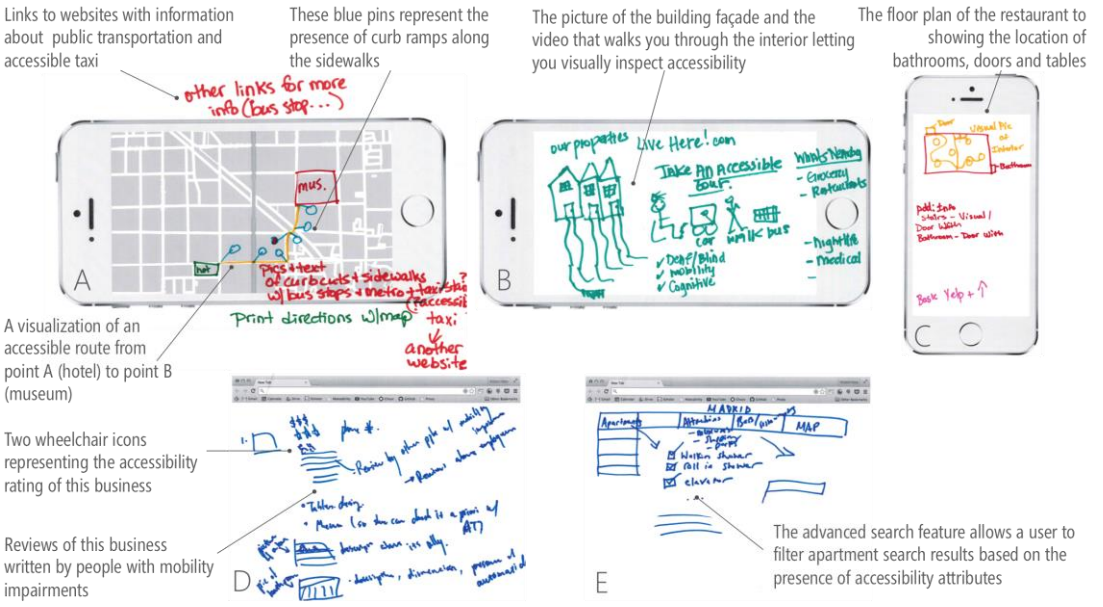


Figure 3.4. Examples of sketches from Part 2 of the study. (a) a mobile map that shows the accessible route and placement of curb ramps (sketched by P7); (b) a virtual video walk through feature to see within/around the housing (P9); (c) a floor map visualization to assess spaciousness of a restaurant floor (P20); (d) a search tool with accessibility rating of a place and reviews written by other mobility impaired (described by P12, sketched by a researcher), and (e) a location directory with advanced search feature to select accessibility attribute (P11).

should be either generated automatically with previously acquired accessibility metadata or provided by end-users (*i.e.*, crowdsourcing), which is the technique employed by sites like AXSmap [224]. In describing a Yelp-like tool, P3 said: *"that would be the crowdsourcing information with rankings from an individuals, 5 stars for accessibility, 2 stars for food."* As data gathering

Detailed Description: A large majority of our participants (*N*=17) sketched or described interfaces that provided detailed information about the accessibility of a place. Details were important not only because of the wide-range of needs amongst a diverse mobility-impaired population but also because, even for a single user, needs may change over time or situationally. In describing a location search tool, P17 said:

> *"if you click on this one up here you could have a box that comes up with accessible information and maybe this says 'no' and why: 'one step in front of entrance.' That lets someone decide 'well actually I could do that' or 'I'm going with a group, they can help me up that step' or 'I'm going by myself and I can't do this' […]"*

Floor Plan: Four participants mentioned that visualizations of a buildings floor plans annotated with information relevant to indoor accessibility would be useful (*e.g.*, stairs, elevators, narrow areas of traversal). In describing the sketch in Figure 3.4c, for example, P20 said that floor plans help reveal the general accessibility of a facility (a restaurant in this case). His tool visualizes the placement of tables and shows whether it is possible to reach a bathroom.

Visual Accessibility Inspection: Eight participants said that ALTs should provide *visual* methods to let users remotely inspect the accessibility of streets/sidewalks (*e.g.*, presence of curb ramps), building façades (*e.g.*, presence of stairs), and building interior (*e.g.*, maneuvering spaces) (Figure 3.4b). Desired inspection methods included pictures, videos, and interactive virtual reality of a room. With visual information, a user can inspect and confirm that the location is indeed accessible by themselves. P18 said:

> *"I guess it's more of a matter of confidence. If I look at the map and it says there's a curb cut here, I trust that's accurate. But I would be more confident if I could also see a picture of it and see 'yeah there's a curb cut there, and it looks like it's in pretty good shape.'"*

Discussion and Review: Five participants mentioned that user-generated reviews would be useful to assess accessibility and to help evaluate the credibility of provided accessibility information (Figure 3.4d). Some specifically said accessibility reviews should come from other people with mobility impairments to ensure that the reviewers share a common perspective of what constitute accessibility barriers: *"[a tool] needs to have reviews by other people with disabilities."* (P12)

Search and Filter: Two methods to query accessibility information emerged. First, five participants described tools to search and filter places based on accessibility attributes. For example, an advanced search option shown in Figure 3.4e allows its users to specify accessibility attributes for a hotel accommodation.

Routing: Second, fifteen participants mentioned ways for searching accessible paths between two locations—using either single modes or multiple modes of transportation (*e.g.*, a tool that automatically finds an accessible walking path to an accessible bus to a user's destination).

Transportation: Twelve participants wanted information about (accessible) transportation on their ALTs. Some described more advanced features like on-demand accessible cabs*:*

> *"If there was an app that showed where the cab was, kinda like in Uber, […] there's an accessible cab going down here in this direction, and you're here. It'll be to you in three minutes or whatever, so it can show like all the accessible cabs in your area." (P11)*

Universal Design: Finally, a request for universal design organically emerged. Three participants said the aforementioned features should be integrated into existing tools like Google Maps and Yelp rather than specialized, assistive-oriented tools that have smaller user bases and often fewer developer resources. *"I'm all for universal technology, so [an accessibility feature] would be integrated into an app that everyone uses rather than an accessibility app." (P11)*

Feature Summary. We grouped recurring and similar features in our participant-created ALTs into ten categories. Features ranged from getting a high-level overview of the accessibility of a neighborhood to more fine-grained information about the accessibility of a building. Some features specifically allowed users to upload and/or review content and assess credibility.

Data Quality

Prior work has shown that perceptions of data quality such as credibility and relevancy dramatically impact how the data is consumed [193,200]. Below, we describe five important data quality attributes from Part 2. Note that we did not specifically prepare questions about data quality, so these themes are emergent:

Granularity: Fourteen participants mentioned that the interface should present detailed accessibility information rather than just binary indicators. In designing her location-search tool, P7 said that ALTs should present:

> *"inside each room, dimensions, bathrooms and kitchen, specifics with heights of counters and turnaround space, having a floor plan,*

> *heights of light switches and whether or not there's carpet or hardwood—the type of floor."*

Relevance: Eight participants noted that not all accessibility information is relevant to their specific impairment, suggesting the need for drill-down interfaces that present well-categorized high-level accessibility information with detailed information available through interaction. In describing a location search tool, P4 said:

> *"for me, I just need a ramp and an elevator. But like I said, other people need other things, so they would have to probably come up with a list of all the different things that would be classified as accessible to different people."*

Credibility: Six participants mentioned that the data needs to be trustable: *"[...] can we even trust the website? I would have to know the person who reviewed it as accessible has either a similar disability to my own or understands the concerns of a person who my particular issues." (P12)*

Recency of Information: Six participants mentioned that up-to-date data is crucial, especially for accessibility barriers that change daily (*e.g.*, construction) or even hourly (*e.g.*, pedestrian traffic). P3 noted: *"Currency of information is always a key. […] Google Street View makes everything look accessible but does not include the construction that recently started."*

Coverage: Two participants described the issue of scarce data in emerging ALTs like AXSMap:

"AXS map? […] it doesn't get much traction, because […] they don't cover enough area, so it's like one neighborhood in NYC and it's like who's going to really look at that?" (P11)

Part 2 Summary. Through participatory design activities, we identified ten desired features and five essential data qualities for ALTs. The top three most desired features were providing detailed descriptions, accessibility-aware routing, and top-down map-based views of street-level accessibility. Data quality attributes often related to features (*e.g.*, high granularity of data corresponds to the detailed description feature).

3.4.3 Part 3: Design Probe

In the last part of our study, we used 12 probes (Figure 3.3) to explore designs and functionalities of future ALTs. We specifically conducted this part of the study after the participatory design to not bias the participants' ideation process while sketching in Part 2. Many features and probe designs ended up overlapping with participants' own ideas. Thus we focus on describing overall reactions to our probes here as well as specific feedback that differs from Part 2.

Overall Reactions

Accessibility Score Visualizations (3A-D): Eighteen participants reacted positively to the concept of visualizing street-level accessibility on a map (Figure 3.5). Of these four probes, participants were less supportive towards the neighborhood-level heat map probe (3A) because of low *location precision* and, instead, preferred the sidewalk-level

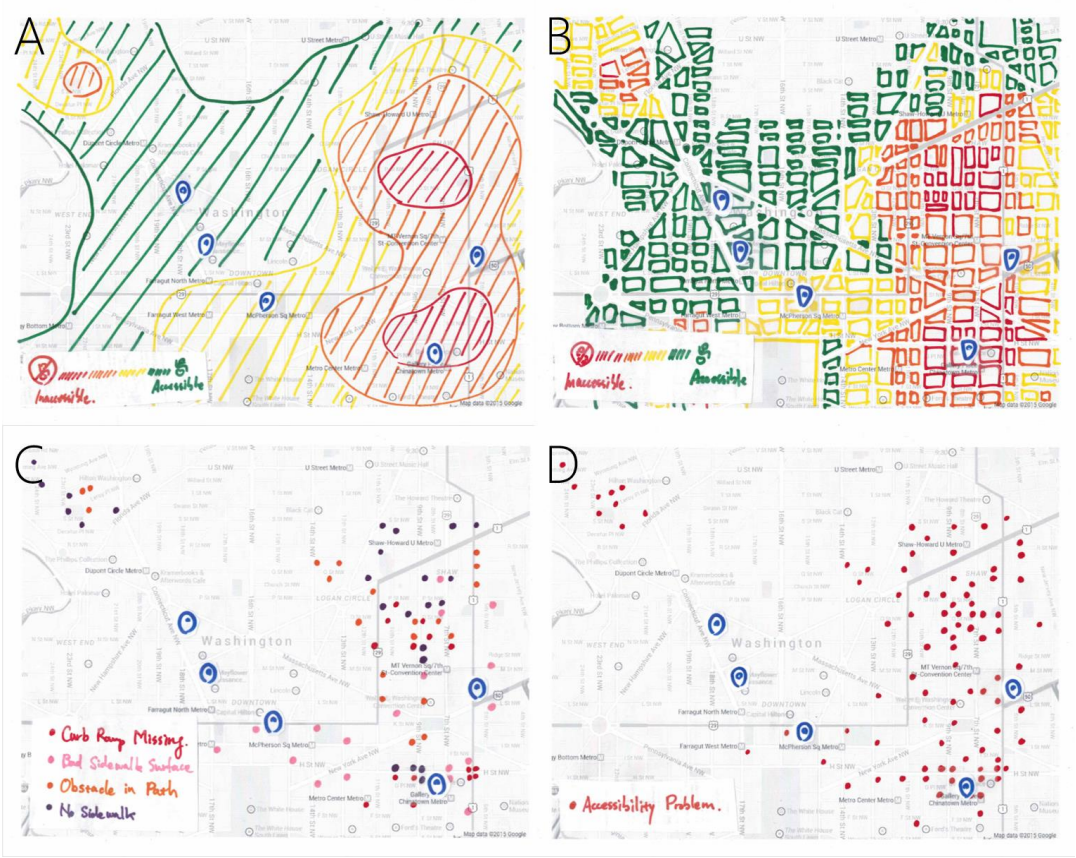


Figure 3.5. Design probe a-d that visualize street-level accessibility. (a & b) Neighborhood- and sidewalk-level accessibility visualizations that shows accessible areas in green and inaccessible areas in red. (c & d) Point-level visualization that show specific accessibility barriers in dots, both categorized (c) and non-categorized (d).

heat map probe (3B). Participants preferred the categorized dot probe over the uncategorized one due to a higher level of information granularity.

Citywide Accessibility Score Comparison (3E): Only six participants reacted positively towards the citywide accessibility score comparison (Figure 3.6). Participants expressed doubt about the utility of the application because they felt that a city, as a unit of accessibility evaluation, is too broad and coarse to provide any meaningful insights.

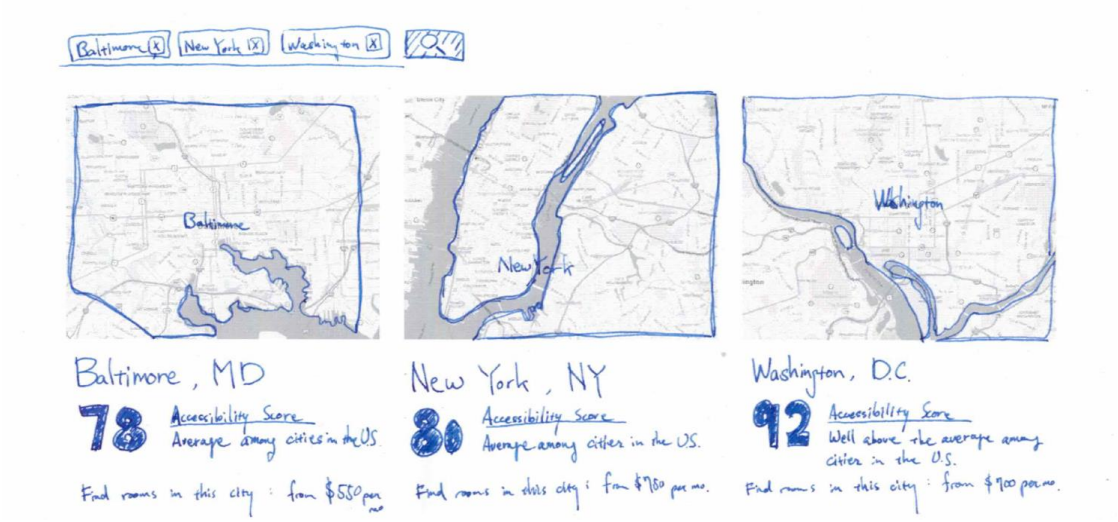


Figure 3.6. Citywide accessibility score comparison. This probe quantifies the accessibility of entire cities with a single accessibility score along with brief, textual rationale.

> *"I think the problem with it is, at least at the level you're displaying it here, is that it's too high level. It's not granular enough. Take for example New York, I might be interested in Manhattan, but not Brooklyn or Queens. But if you got this overall score that doesn't really tell me much." (P18)*

Accessibility-Aware Location Search (3F): Thirteen participants reacted positively towards the design of the location search tool (Figure 3.7). Participants suggested improving the design by allowing users to examine the rationale for the 5-level accessibility score (*e.g.*, presence of handicap parking). Other suggestions included provision of pictures of the building façade and accessibility reviews by others.

Accessible Bus Stops Visualization (3G): A majority of participants (N=15) favored the idea of visualizing bus stop accessibility (Figure 3.8). Design suggestions included providing rationale for why bus stops are (in)accessible, presenting general transit

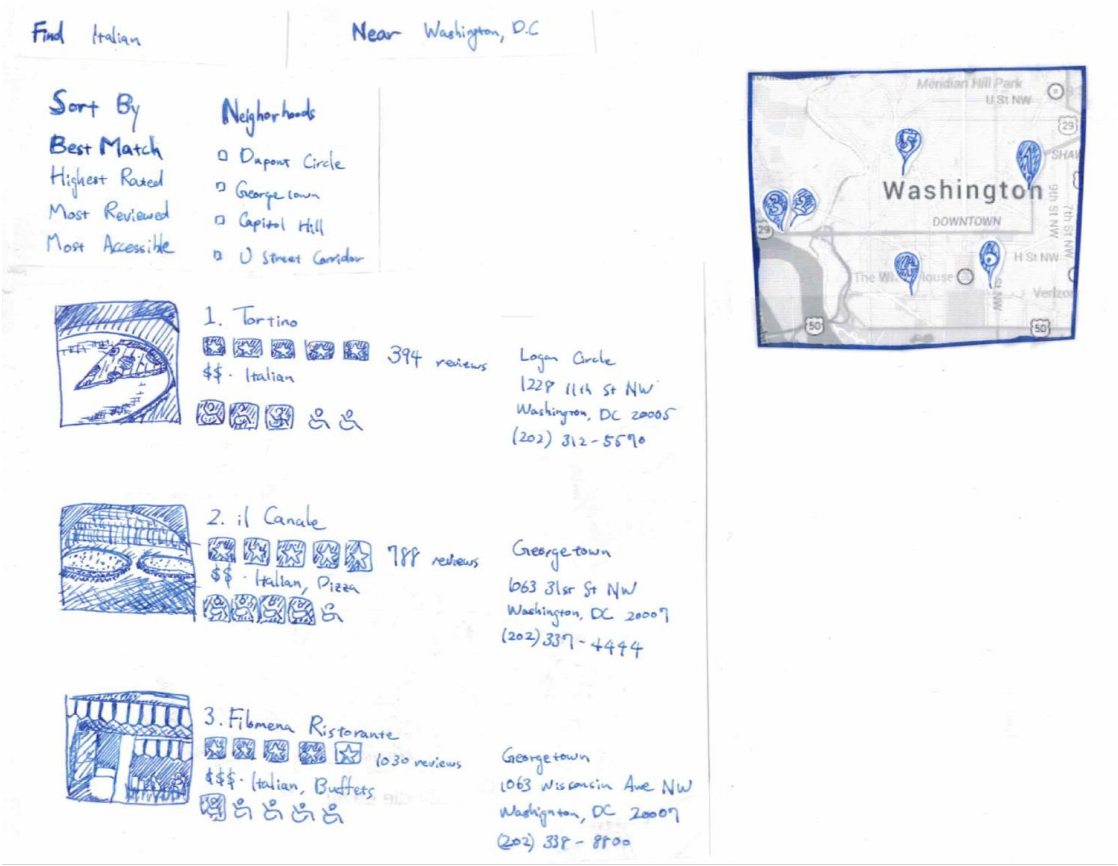


Figure 3.7. Accessibility-aware location search. A point-of-interest search website similar to yelp.com but augmented with accessibility information. Users can search for a business or other point-of-interest with a keyword and location. Each search result is accompanied by a 5-level accessibility score, which can be used for sorting and filtering

information, and offering similar information for different types of public transportation (*e.g.*, trains, subways).

Visualizing Building Accessibility (3H-J): Seventeen participants reacted positively to the idea of color coding the accessibility of buildings on a map and/or showing floor plan accessibility visualizations (Figure 3.9). For the floor plan visualization (3I), two participants suggested denoting what the areas are used for to improve understandability. In contrast, only 8 participants (42%) thought that the indoor routing

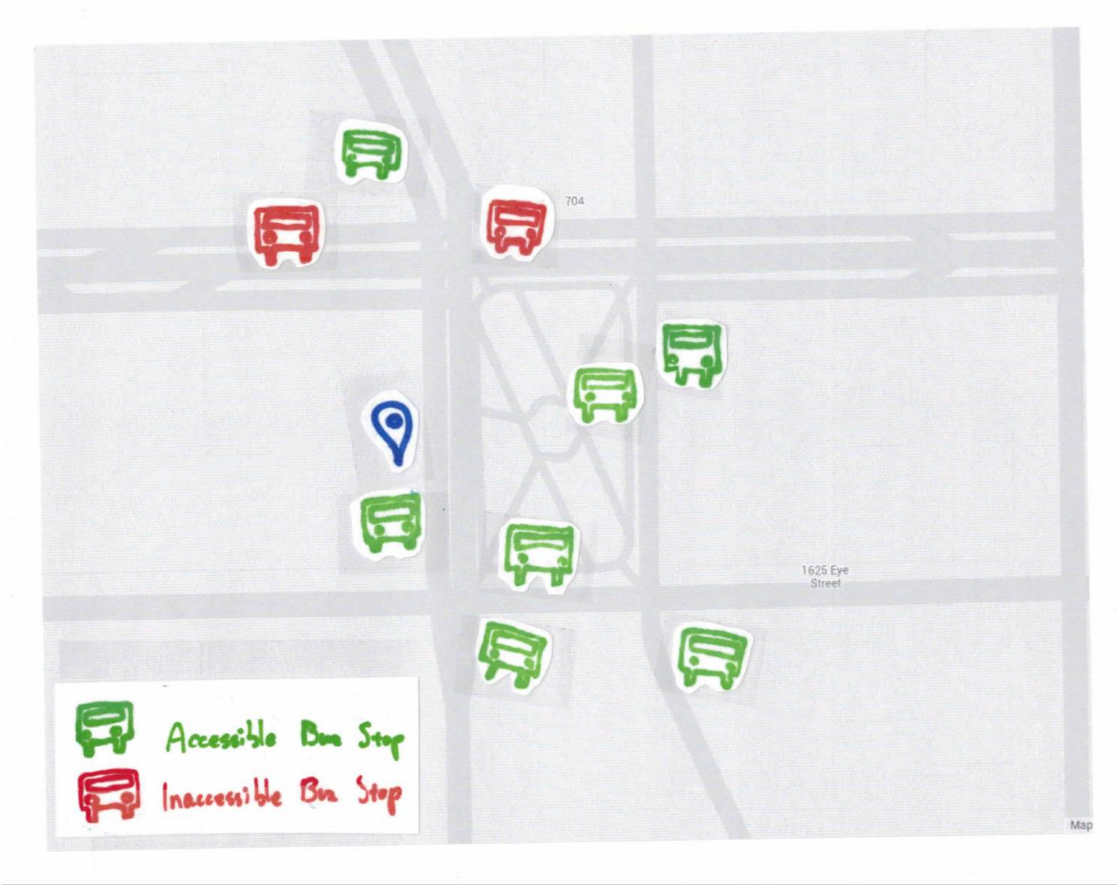


Figure 3.8 Accessible bus stop visualization. Users can enter a location and see proximal bus stops, which are color-coded based on accessibility (green for accessible, red for inaccessible).

tool (3J) would be useful. Most participants felt that more effective alternative methods are readily available (*e.g.*, talking to others, looking at mall directories).

Outdoor Accessible Routing (3K-L): Seventeen participants reacted positively towards accessibility-aware routing interfaces (Figure 3.10). Twelve preferred the interface with multiple routes (3L), while three preferred the simpler interface (3K). One participant suggested including audible turn-by-turn navigation, because moving her upper body to interact with her mobile tool was hard.

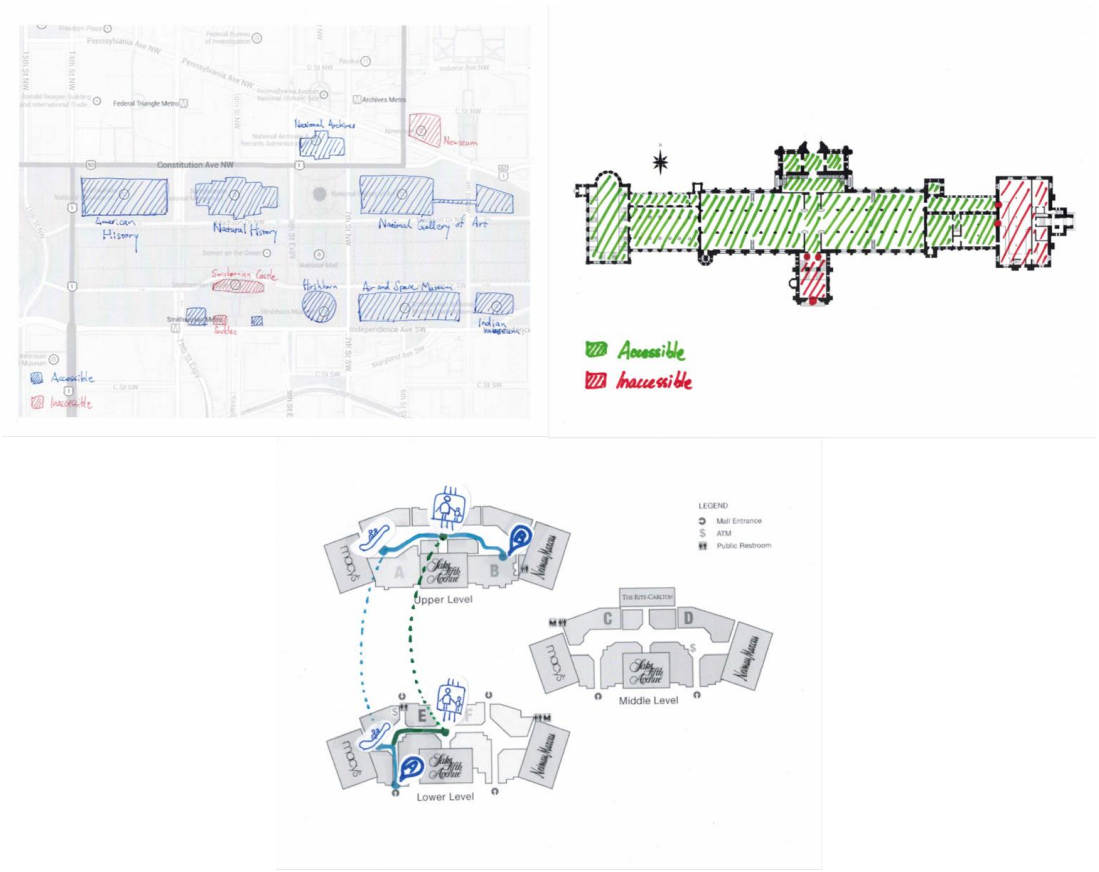


Figure 3.9. Visualizing building accessibility. (Top-left) The first design uses a top-down map visualization to indicate the accessibility of public buildings in a selected area. (Top-right) The floorplan visualization highlights accessible and inaccessible features such as elevators and stairs. (Bottom) The third design focuses on accessible routing interfaces for indoor environments.

Part 3 Summary. More than half of participants reacted positively towards all probes except for the citywide accessibility score comparison and indoor routing probes. In discussing mockups of Accessibility Score Visualizations, an additional data quality, *location precision*, emerged; which refers to geographical fidelity of accessibility data (*e.g.*, at sidewalk level or block level).

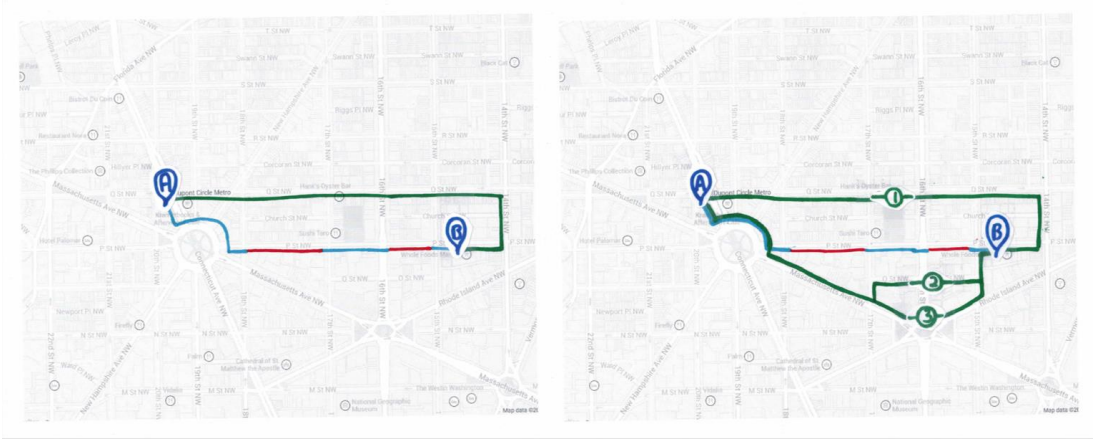


Figure 3.10. Accessibility-aware routing. Similar to Apple or Google Maps, these probes allow the user to enter a start and end location and view suggested routes. In our designs, however, the shortest path is visualized as well as the shortest accessible path. The probe on the left shows one alternative accessible path while the one on the right shows multiple alternatives.

3.5 Discussion

We reflect on the implications of our findings, describe study limitations, and offer suggestions for future work.

3.5.1 ALTs Design Considerations and Recommendations

The study in this chapter reaffirms the unmet needs of previously proposed/designed ALTs (*e.g.,* accessibility-aware POI search [68,176]) as well as presents desired assistive features that have not been described before (*e.g.*, visual accessibility inspection). Following the design practice described in [155], we formulate design recommendations based on findings from our three ALT scenarios:

Citywide Accessibility Exploration: Location precision and categorical granularity of accessibility barriers were valued in the design probe activities. We suggest providing two types of visualizations similar to Figure 3.3b and c. Information about accessible

routes to nearby locations (*e.g.,* cafes) is also recommended. Given a point on a map, provide a range of nearby amenities together with the detailed information about accessible routes to those destinations (*e.g.,* distance of the shortest accessible route).

Accessibility-Aware Location Search: For each location search result, provide an overall accessibility rating of the place. This will allow mobility impaired users to quickly browse through the list of results and find a few that are accessible and have high reputations. Providing rationale for the accessibility ratings is also strongly desired, including a list describing what barriers or facilitators make each location accessible or not.

Accessibility-Aware Navigation: Future ALTs that support routing should provide *multimodal* accessibility-aware navigation. The interfaces should provide routes with accessible transportation (*e.g.,* accessible taxis, buses) and accessible walking/rolling directions. To further improve the interface, provide geographical visualizations of neighborhood accessibility along the route. This will allow users to reason about why the routes are recommended.

Similar to prior research in data quality of online reviews for health care providers [193] and businesses [12], our findings support the need for ensuring and maintaining high data quality. The recommended designs above should allow users to verify accessibility information by incorporating features described in Part 2 (*e.g.,* Visual Inspection).

3.5.2 Future Work

One key challenge in designing and deploying ALTs is finding and maintaining up-to-date information about the accessibility of the built environment. Though prior work, including ours described in the following chapters, has explored crowdsourced or semi-automated methods to remotely collect outdoor accessibility information at scale [3,73,77,81,84], these methods rely on potentially out-of-date information, offer no way for users to update or comment, and do not yet work for indoor environments. Potential avenues of future research in this area include exploring potentially rich and scalable but untapped sources of accessibility information such as daily-updated satellite imagery (*e.g.,* Planet Labs [225]) or even surveillance video streams (*e.g.,* Placemeter [226], dashboard cameras on government vehicles [229]).

Our study elicited design features of future ALTs. How best to combine these for different scenarios is an open question. For example, P3 wanted accessibility-aware navigation tool on Google Glass so the application could help him navigate on-the-go. While feature requirements for the technology may be similar (*i.e.,* routing, transit information), more investigation is needed.

3.5.3 Accessibility Data in Sharing Economy

Should ALTs provide accessibility information of private properties? Although mentioned by only one participant, emergent sharing economies such as Airbnb raise important questions for ALTs. In the U.S., for example, there are no regulations that mandate residential housing to be accessible. Thus, it is not clear if Airbnb

accommodation owners need to comply with ADA [7]. P7 raises some important points about this complex issue:

> *"… most of the places in Airbnb are not accessible to rent out. But now there's is… [an advanced feature to search with] wheelchair accessibility of a home or something, but it's all dependent on the person who's renting the home and what their understanding of what accessible is."*

3.5.4 Limitations

We performed a qualitative study of 20 mobility impaired participants located in the eastern US. Future work should consider a larger, more diverse sample and compare perspectives. Our study focused solely on mobility-impaired users, future work should include people with other physical or sensory impairments. Finally, though useful in structuring the participatory design activity, our use of templates in Part 2 may have affected our results.

3.6 Conclusion

The study in this chapter provides the first work investigating modern and desired methods and technologies for evaluating built environment accessibility. We conducted a three-part study with 20 mobility impaired participants. Part 1 reinforced and extended findings in the literature regarding perspectives of accessibility barriers/facilitators. Through participatory design activities in Part 2 and 3, we

uncovered 10 key features of desired ALTs and six key data qualities, which have implications for the design of future ALTs.

Chapter 4 Collecting Sidewalk Accessibility Data with Crowdsourcing

This chapter describes our initial work on building and evaluating a system that collects street-level accessibility information by combining crowdsourcing and Google Street View (GSV). This chapter has adapted and rewritten content from papers at ASSETS 2012 and CHI 2013 [80,81].

4.1 Introduction

According to the most recent US Census (2010), roughly 30.6 million individuals have physical disabilities that affect their ambulatory activities [185]. Of these, nearly half report using an assistive aid such as a wheelchair (3.6 million) or a cane, crutches, or walker (11.6 million) [185]. Despite aggressive civil rights legislation for Americans with disabilities (*e.g.*, [1,191]), many city streets, sidewalks, and businesses in the US remain inaccessible [132].

As we described in this dissertation's Introduction, the problem is not just that sidewalk accessibility fundamentally affects *where* and *how* people travel in cities but also that there are few, if any, mechanisms to determine accessible areas of a city *a priori*. Indeed, in a recent report, the National Council on Disability noted that they could not find comprehensive information on the "degree to which sidewalks are accessible" across the US [132]. Traditionally, sidewalk assessment has been conducted via in-person street audits [171,172], which are labor intensive and costly

Figure 4.1. Using crowdsourcing and Google Street View images, we examined the efficacy of three different labeling interfaces on task performance to locate and assess sidewalk accessibility problems: (a) *Point,* (b) *Rectangle,* and (c) *Outline*. Actual labels from our study shown.

[157], or via citizen call-in reports, which are done on a reactive basis. As an alternative, we propose the use of crowdsourcing to locate and assess sidewalk accessibility problems *proactively* by labeling GSV imagery

We report on three studies in particular: design exploration for the image labeling interface (Exploratory Study), a feasibility study with motivated people (Study 1) and an online crowdsourcing study using Amazon Mechanical Turk (Study 2). In Exploratory Study, we conduct a preliminary study to explore the design of image labeling interface by asking crowd workers from Amazon Mechanical Turk to label accessibility issues found in a manually curated database of 100 GSV images. We examine the effect of three different interactive labeling interfaces (Figure 4.1) on task accuracy and duration. Because labeling sidewalk accessibility problems is a subjective and potentially ambiguous task, Study 1 investigates the viability of the labeling sidewalk problems amongst two groups of diligent and motivated labelers: three members of our research team and three "sidewalk accessibility experts"—in this case, wheelchair users. We use the results of this study to: (i) show that the labeling approach is reliable, with high intra- and inter-labeler agreement within and across the two groups; (ii) acquire an understanding of baseline performance—that is, what does good

Figure 4.2. We propose and investigate the use of crowdsourcing to find, label, and assess sidewalk accessibility problems in Google Streetview (GSV) imagery. The GSV images and annotations above are from our experiments with Mechanical Turk crowd workers.

labeling performance look like? (iii) provide validated ground truth labels that can be used to evaluate crowd worker performance.

For Study 2, we investigate the potential of using crowd workers on Mechanical Turk (turkers) to perform this labeling task. We evaluate performance at two levels of labeling accuracy: *image level*, which tests for the presence or absence of the correct label in an image, and *pixel level*, which examines the pixel-level accuracies of the labels provided (as in Figure 4.2). We show that, when compared to ground truth, turkers are capable of determining that an accessibility problem *exists* in an image with 80.6% accuracy (binary classification) and determining the correct *problem type* with 78.3% accuracy (multiclass classification). Using a simple majority voting scheme with three turkers, this accuracy jumps to 86.9% and 83.8% respectively. We also examine the effect of two quality control mechanisms on performance: statistical filtering and multilevel review (see [146]). Our findings suggest that crowdsourcing both the labeling task and the verification task leads to a better quality result. We also demonstrate the performance/cost tradeoffs therein.

Note that unlike labeling interfaces presented in other chapters (Chapter 5 & 6), the labeling interface in Chapter 4 used static, pre-cropped GSV images. Our labeling interfaces presented in Chapters 5 & 6 were built with Google's Street View API, so they allowed for panning to explore the entire GSV panorama and, in some cases, walking (Chapter 6).

The contributions of the work in this chapter are threefold: (i) the first step toward a scalable approach for combining crowdsourcing and existing online map imagery to identify perceived accessibility issues, (ii) measures for assessing turker performance in applying accessibility labels, and (iii) strategies for improving overall data quality. Our approach could be used as a lightweight method to bootstrap accessibility-aware urban navigation routing algorithms, to gather training labels for computer vision-based sidewalk assessment, and as a mechanism for city governments and citizens to report on and learn about the health of their community's sidewalks (*e.g.*, through accessibility scores similar to *walkscore.com*).

4.2 Evaluating Annotation Correctness

In this section, we provide an overview of the correctness measures. Because this is a new area of research, we introduce and explore a range of metrics—many of which have different levels of relevancy across application contexts (*e.g.,* calculating the accessibility score of a neighborhood *vs.* collecting training data for a computer vision algorithm). For the Exploratory Study, we assess *image level* accuracy measures and

we use both *image level* and *pixel level* measures for our main studies (Study 1 and Study 2).

4.4.1 Defining Levels of Annotation Correctness

Assessing annotation correctness in images is complex. To guide our analysis, we derive two spectra that vary according to the type and granularity of data extracted from each label: the localization spectrum and the specificity spectrum. The *localization* spectrum describes the positioning of the label in the image, which includes two discrete levels of granularity: *image level* and *pixel level*. For image level, we simply check for the absence or presence of a label anywhere within the image. Pixel level is more precise, examining individual pixels highlighted by the label outline. Our pixel-level analysis is analogous to image segmentation in computer vision and, indeed, our evaluation methods are informed from work in this space.

The *specificity* spectrum, in contrast, varies based on the amount of descriptive information evaluated for each label. At the finest level of granularity, we check for matches based on the five label categories as well as corresponding severity ratings: *Object in Path, Prematurely Ending Sidewalk, Surface Problem, Curb Ramp Missing*, and *No Problem* (indicating the user had clicked "no accessibility problems found"). Note that *Curb Ramp Missing* and *No Problem* were exempt from severity ratings. At the next level of granularity, we only examine problem types, ignoring severity ratings; we refer to this level as *multiclass*. Finally, at the coarsest level of granularity we group all problem categories into a *binary* classification of problem vs. no problem.

As the first work in the area, these dimensions of analysis are important for understanding crowd worker performance across various measures of correctness. Identifying an appropriate level of correctness may depend on the specific application context. For example, because of the focal length and camera angles used in GSV imagery, simply identifying that an accessibility problem exists in an image (*i.e.*, image-level, binary classification) localizes that problem to a fairly small geographic area: a specific street side and sidewalk within a city block. This level of geographic precision may be sufficient for calculating accessibility scores or even informing accessibility-aware routing algorithms. Binary classification—whether at the image level or the pixel level—also helps mitigate the subjectivity involved in selecting a label type for a problem (*e.g.*, some persons may perceive a problem as *Object in Path* while others may see it as a *Surface Problem*). In other cases, however, more specific correctness measures may be needed. Training computer vision algorithms to segment and, perhaps, automatically identify and recognize obstacles, would require pixel-level, multiclass specificity.

4.2.2 Image-Level Correctness Measures

For image-level analysis, we computed two different correctness measures: a straightforward accuracy measure and a more sophisticated measure involving precision and recall. For *accuracy*, we compare ground truth labels with turker labels for a given image and calculate the percentage correct. For example, if ground truth labels indicate that three problem types exist in an image: *No Curb Ramp, Object in*

Path, and a *Surface Problem*, but a turker only labels *No Curb Ramp*, then the resulting accuracy score would be 50% (1 out of 3 problems identified correctly and 1 correct for *not* providing *Sidewalk Ending*). Though easy to understand, this accuracy measure does not uncover more nuanced information about *why* an accuracy score is obtained (*e.g.*, because of false positives or false negatives).

As a result, we incorporated a second set of correctness measures, which extend from work in information retrieval: *precision*, *recall*, and an amalgamation of the two, *f-measure* that combines them into a single metric. All three measures return a value between 0 and 1, where 1 is better:

$$Precision = \frac{\#\ of\ True\ Pos\ Labels}{\#\ of\ True\ Pos\ Labels + \#\ of\ False\ Pos\ \ Labels} \quad \text{(Eq. 1)}$$

$$Recall = \frac{\#\ of\ True\ Pos\ Labels}{\#\ of\ True\ Pos\ Labels + \#\ of\ False\ Neg\ \ Labels} \quad \text{(Eq. 2)}$$

$$F\text{-}measure = 2 * \frac{Precision * Recall}{Precision + Recall} \quad \text{(Eq. 3)}$$

True positive here is defined as providing the correct label on an image, *false positive* is providing a label for a problem that does not actually exist in the image, and *false negative* is *not* providing a label for a problem that *does exist* in the image. In this way, precision measures the accuracy of the labels actually provided (*i.e.*, a fraction expressing the ratio of correct labels over *all labels provided*) while recall measures the comprehensiveness of the correct labels provided (*i.e.*, a fraction expressing the ratio of correct labels over all *possible correct labels*). For example, a precision score of 1.0 means that every label the turker added was correct but they could have missed labels. A recall score of 1.0 means that the turker's labels include all of the actual problems in the image but could also include non-problems. Given that algorithms can be tuned to

maximize precision while sacrificing recall and vice versa, the f-measure provides a single joint metric that encapsulates both. We use accuracy, precision, recall, and f-measure to describe our image level results.

4.2.3 Pixel-Level Correctness Measures

Pixel-level correctness relates to image segmentation work in computer vision. Zhang [216] provides a review of methods for evaluating image segmentation quality, two of which are relevant here: the *goodness method*, which examines segmentation based on human judgment and the *empirical discrepancy method*, which programmatically calculates the difference between test segmentations and "ground truth" segmentations for a given image. The goodness method can be advantageous in that it does not require ground truth; however, it is labor intensive because it relies on human judgment to perceive quality. Though judging the quality of segmentations can also be crowdsourced, partly mitigating the labor concern (*e.g.*, [16]), the quality of the judgment remains an issue.

We also explored two empirical discrepancy methods: *overlap* (or area of intersection) [6,181] and, again, *precision*/*recall* combined with *f-measure* [31,32], which is similar to that explained above though applied at the pixel level rather than the image level. For our first discrepancy method, overlap is defined as:

$$Overlap(A,B) = \frac{Area(A \cap B)}{Area(A \cup B)} \qquad \text{(Eq. 4)}$$

where *A* and *B* are the pixel outlines. Note that if the outline *A* is perfectly equal to the outline *B*, then *Overlap(A,B)=1*. If *A* and *B* are disjoint, then *Overlap(A,B)=0*.

Although this metric is easy to understand, similar to the straightforward *accuracy* measure for image-level analysis, it fails to capture nuances in correctness. Thus, for our second discrepancy metric we define precision, recall, and f-measure at the pixel level. From the image segmentation literature [228], *precision* is defined as the probability that a generated outline-fill pixel area correctly highlights the target object and *recall* is the probability that a true outline-fill pixel is detected. Thus, in order to calculate precision and recall at the pixel level, we need to compute three different pixel counts for each image:

1. **True positive pixels:** number of overlapping pixels between the ground truth segmentation and the test segmentation;
2. **False positive pixels**: number of pixels in the test segmentation *not* in the ground truth segmentation;
3. **False negative pixels:** number of pixels in the ground truth segmentation *not* in the test segmentation.

Precision and recall can then be computed by the following formulae (f-measure is the same as *eq. 3* above):

$$Precision = \frac{\#\ of\ True\ Pos\ Pixels}{\#\ of\ True\ Pos\ Pixels + \#\ of\ False\ Pos\ \ Pixels} \qquad \text{(Eq. 5)}$$

$$Recall = \frac{\#\ of\ True\ Pos\ Pixels}{\#\ of\ True\ Pos\ Pixels + \#\ of\ False\ Neg\ \ Pixels} \qquad \text{(Eq. 6)}$$

Before calculating pixel-level correctness for any of the measures, we flatten all labels with equivalent type into the same layer and treat them as a single set of

pixels. This allows us to more easily perform pixel-by-pixel comparison between ground truth labels and test labels marked with the same problem type.

4.3 Exploratory Study: Annotation Interface Design Study

To collect geo-labeled data on sidewalk accessibility problems in GSV images, we created an interactive online labeling tool in JavaScript, PHP and MySQL. Labeling GSV images is a three step process consisting of *marking* the location of the sidewalk problem, *categorizing* the problem into one of five types, and *assessing* the problem's severity. For the first step, we created three different marking interfaces to assess their label granularity *vs.* labeling speed trade-off: (i) *Point*: a point-and-click interface; (ii) *Rectangle*: a click-and-drag interface; and (iii) *Outline*: a path-drawing interface (Figure 4.1). We expected that the *Point* interface would be the quickest labeling technique but that the *Outline* interface would provide the finest pixel granularity of marking.

Once a problem has been marked, a pop-up menu appears with four specific problem categories: *Curb Ramp Missing, Object in Path, Prematurely Ending Sidewalk,* and *Surface Problem.* We also included a fifth label for *Other*. These categories are based on sidewalk design guidelines from the US Department of Transportation website [102] and the US Access Board [122]. Finally, after a category has been selected, a five-point Likert scale appears asking the user to rate the severity of the problem where 5 is most severe indicating "not passable" and a 1 is least severe indicating "passable." If more than one problem exists in the image, this process is

repeated. After all identified sidewalk problems have been labeled, the user can select "submit labels" and another image is loaded. Images with no apparent sidewalk problem can be marked as such by clicking on a button labeled "There are no accessibility problems in this image." Users can also choose to skip an image and record their reason (*e.g.,* image too blurry, sidewalk not visible).

4.3.1 Study Method

To investigate the feasibility of using crowd workers for this task, we posted our three labeling interfaces (*Point, Rectangle,* and *Outline*) to Amazon Mechanical Turk. Crowd workers ("turkers") could complete "hits" with all three interfaces but would see each image at most once. Before beginning the labeling task with a particular interface, turkers were required to watch the first half of a three-minute instructional video. Three videos were used, one for each condition, which differed only in the description and presentation of the corresponding labeling interface. After 50% of the video was shown, the labeling interface would automatically appear (thus, turkers were not forced to watch the entire video).

Each labeling interface pulled images from the same test dataset, which consisted of 100 static GSV images. These images were manually scraped by the research team using GSV of urban neighborhoods in Los Angeles, Baltimore, Washington DC, and New York City. We attempted to collect a balanced dataset. Of the 100 images, 81 (81%) contained one or more of the aforementioned problem

categories. The remaining 19 images had no visible sidewalk accessibility issues and were used, in part, to evaluate *false positive* labeling activity.

	No Curb Ramp	Object in Path	Sidewalk Ending	Surface Problem
Point	34	27	10	29
Rectangle	34	27	11	28
Outline	34	26	10	29

Table 4.1. Frequency of labels at the image level in our ground truth dataset based on a "majority vote" from three trained labelers.

To evaluate turker performance, we created baseline label data by having three researchers independently label all 100 images in each of the three interfaces. Inter-rater agreement was computed on these labels at the *image* level using Fleiss's kappa for each interface. More specifically, we tested for agreement based on the absence or presence of a label in an image and not on the label's particular pixel location or severity rating. We found moderate to substantial agreement [110] (ranging from 0.48 to 0.96). From these labels, we created a majority-vote "ground truth" dataset. Any image that received a label from two of the three authors was assigned that label as "ground truth" (Table 4.1).

4.3.2 Analysis and Results

We posted our task assignments to Mechanical Turk in batches of 20-30 over a one week period in June, 2012. In all, we hired 123 distinct workers who were paid three to five cents per labeled image. They worked on 2,235 assignments and provided a total of 4,309 labels (1.9 per image on average). As expected, the *Point* interface was the fastest with a median per-image labeling time of 32.9 seconds (*SD*=74.1) followed by

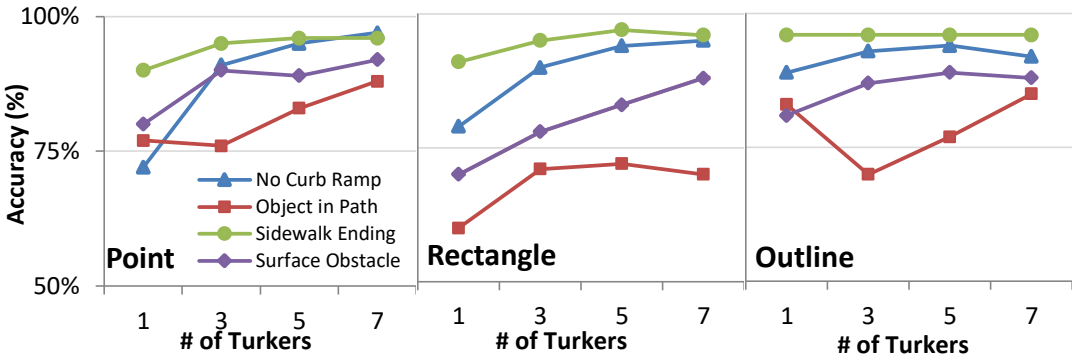


Figure 4.3. The number of turkers per image vs. accuracy for each of the three labeling interfaces. Note that the y-axis begins at 50%.

Outline (41.5s, *SD*=67.6) and *Rectangle* (43.3s, *SD*=90.9). When compared with our ground truth dataset, overall turker accuracies at the *image* level were: 83.0% for *Point*, 82.6% for *Outline*, and 79.2% for *Rectangle*.

We also explored accuracy as a function of the number of turkers per image and as a function of label type. To do this, we calculated four different turker-based majority vote datasets for each interface based on four different turker group sizes: 1, 3, 5, and 7. Group membership was determined based on the order of completion for each hit. The results are shown in Figure 4.3. Note that, again, we perform these comparisons at the *image* level rather than the individual label level and that we again ignore severity. These calculations are left for future work.

We did, however, employ an additional evaluation method by calculating the precision and recall rate of each interface, where:

$$Precision = \frac{True\ Pos}{True\ Pos + False\ Pos}, Recall = \frac{True\ Pos}{True\ Pos + False\ Neg}$$

		No Curb Ramp	Object in Path	Sidewalk Ending	Surface Problem	Overall
Point	Precision	0.90	0.53	0.80	0.76	0.71
	Recall	0.82	0.93	0.73	0.93	0.87
Rectangle	Precision	0.85	0.48	0.80	0.59	0.63
	Recall	0.85	1.00	0.73	0.71	0.84
Outline	Precision	0.89	0.47	0.89	0.71	0.67
	Recall	0.91	0.93	0.73	0.89	0.89

Table 4.2. Precision and recall results for the three labeling interfaces based on majority vote data with three turkers compared to ground truth. "Object in path" is consistently the worst performing label.

True positive here is defined as is providing the correct label on an image, *false positive* is providing a label for a problem that does not actually exist on the image, and *false negative* is *not* providing a label for a problem that *does exist* in the image. Our results are presented in Table 4.2. Both high precision and recall are preferred. The precision rate for *Object in Path* and *Surface Problems* are relatively low for all three interfaces. This indicates that turkers are making false positive decisions for those labels—that is, they tend to use these labels for things that are not actually problems.

4.3.3 Discussion for the Exploratory Study

In this section, we explored the feasibility of using crowd-sourced labor to label sidewalk accessibility problems from GSV images as well as investigated the trade-off in using three types of labeling interfaces. We showed that untrained crowd workers can locate and identify sidewalk accessibility problems with relatively high accuracy (~80% on average). However, there is a clear problem with turkers *overlabeling* images (*i.e.*, we had a high false positive rate). In addition, there is a non-trivial number of bad quality workers—11 out of 123 had an error rate greater than 50%. We investigate the turker performance in more details in Study 2. We explored the trade-off in using three

types of labeling interfaces. As expected, the *Point* interface was the quickest to label (32.9s) compared to *Outline* (41.5s) and *Rectangle* (43.3s) but not with a big margin. Therefore, we decided to use the *Outline* interface in the following study as it provides much more pixel-granularity compared to the *Point* interface.

4.4 Dataset (Study 1 & 2)

To collect geo-labeled data on sidewalk accessibility problems in GSV images, we used the web-based labeling tool (Figure 4.4)—the Outline tool described above (Figure 4.1) We also created a verification interface where users could accept or reject previously collected labels (Figure 4.5). Below, we describe the annotation interface and the primary dataset used in our studies. We return to the verification interface in the Study 2 section.

The test dataset used in the labeling interface consists of 229 images manually scraped by the research team using GSV of urban neighborhoods in Los Angeles, Baltimore, Washington, D.C., and New York City. We attempted to collect a balanced dataset. Of the 229 images, 179 (78%) contained one or more of the aforementioned problem categories; 50 (22%) had no visible sidewalk accessibility issues and were used, in part, to evaluate *false positive* labeling activity. Based on our majority-vote ground truth data (described later), we determined the following composition: 67 (29%) images with *Surface Problems*, 66 (29%) images with *Object in Path*, 50 (22%) with *Prematurely Ending Sidewalk*, and 47 (21%) with *Curb Ramp Missing*. This count is not mutually exclusive—48 (21%) images in total included more than one problem

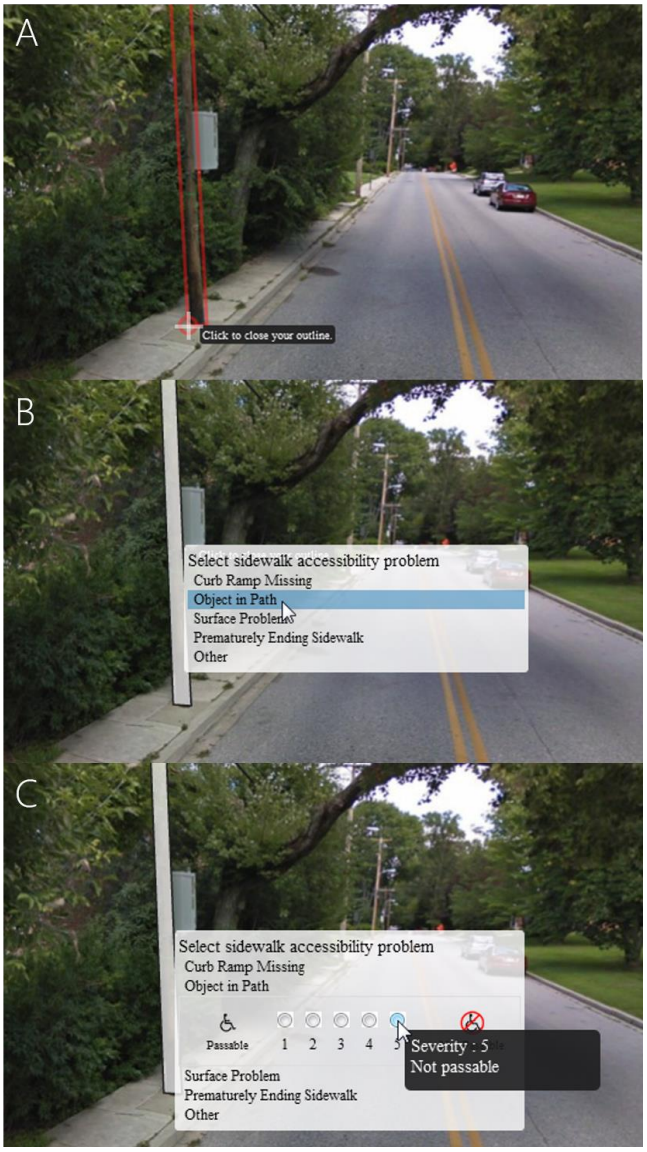


Figure 4.4. Labeling GSV images is a three step process consisting of (a) *marking* the location of the sidewalk problem in the image, (b) *categorizing* the problem into one of five types, and (c) *assessing* the problem's severity. Here, the utility pole is labeled *Object in Path* and rated 5 (*Not Passable*).

type. The label *Other* was used 0.5% of the time in Study 1 and 0.6% in Study 2 and is thus ignored on our analyses. As of September 2012, the average age of the images is 3.1 years old (*SD*=0.8 years). We return to the potential issue of image age in the discussion.

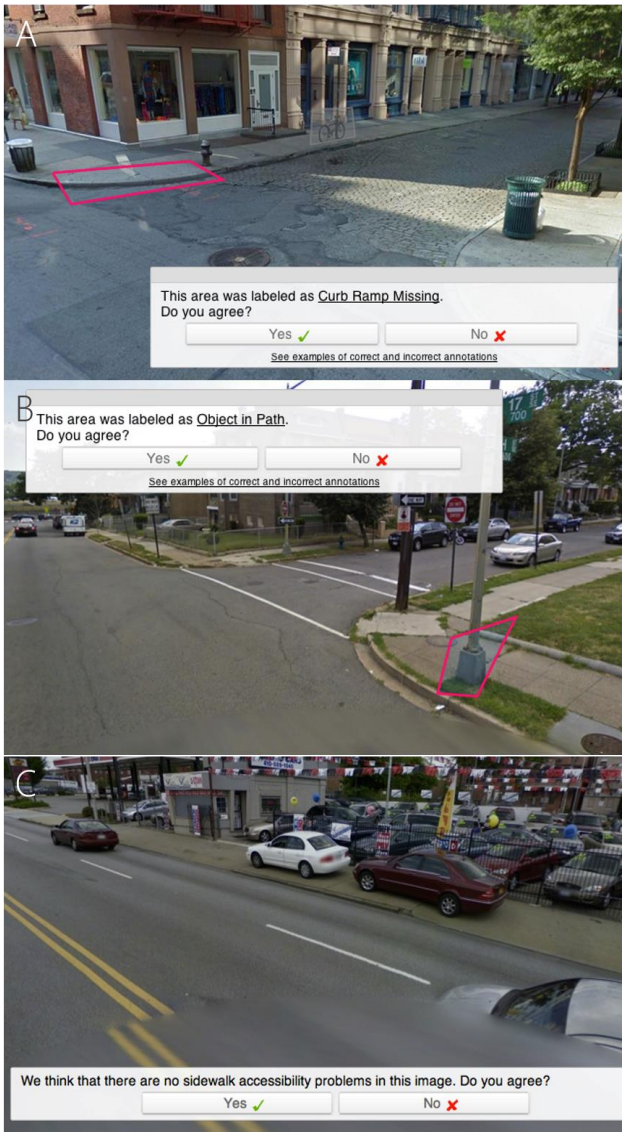


Figure 4.5. The verification interface used to experiment with crowdsourcing validation of turker labels—only one label is validated at a time in batches of 20. (a) A correctly labeled *No Curb Ramp* problem; (b) A false positive *Object in Path* label (the utility pole is located in the grass and not in the sidewalk); (c) A false negative example: The cars should have been marked as *Object in Path*.

4.5 Study 1: Assessing Feasibility

Labeling accessibility problems perceived in streetscape images is a subjective process.

As such, our first study focused on demonstrating that informed and well-motivated

labelers could complete the labeling task and produce consistent results. We had two additional goals: (i) to produce a vetted ground truth dataset that could be used to calculate turker performance in Study 2, and (ii) to help contextualize Study 2 results (*i.e.,* what does "good" performance look like for our labeling task?).

We collected independently-labeled data from two groups: three members of our research team and three wheelchair users (who served as "sidewalk accessibility experts"). We then computed intra- and inter-annotator agreement scores for within and between each group respectively. We explore agreement at both the image level and the pixel level across binary and multiclass classification.

4.5.1 Collecting Wheelchair User Ground Truth Data

Three wheelchair users were recruited via listservs and word-of-mouth: two males with spinal cord injury (tetraplegia) and one male with cerebral palsy. All three used motorized wheelchairs; one also used a manual wheelchair but rarely. Each wheelchair user took part in a single labeling session at our research lab. Participants were asked to label the images based on their own experiences and were instructed that not all images contained accessibility problems. They were also asked to "think-aloud" during labeling so that we could better understand the rationale behind their labeling decisions. The sessions lasted for 1.5-3 hours and included a short, post-labeling interview where we asked about the participant's personal experiences with sidewalk/street accessibility and about potential improvements to our labeling tool. All interviews were video recorded. In consideration of participant time and potential fatigue, only a subset of the

Image-Level Label Specificity	Label	Researchers (*N*=3, *I*=229)	Wheelchair Users (*N*=3, *I*=75)	Researchers vs. Wheelchair Users (*N*=2 groups, *I*=75)
Binary Classification	No Problem vs. Problem	0.81	0.68	0.79
Multiclass Classification	No Curb Ramp	0.81	0.82	0.83
	Object in Path	0.56	0.55	0.62
	Sidewalk Ending	0.86	0.71	0.78
	Surface Problem	0.62	0.40	0.74
	Overall	0.69	0.62	0.74

Table 4.3: Fleiss' kappa annotator agreement scores for image-level analysis between the researchers, the wheelchair users, and the researchers compared to the wheelchair users (this lattermost comparison is based on majority vote data within each group).

total 229 image dataset was labeled: 75 in total. These images were selected randomly from each of the four problem categories (*4 categories x 15 images = 60*) plus an additional 15 randomly selected "no problem" images. Participants were compensated $25-35 depending on session length. Below, we report on evaluating agreement between the researchers, the wheelchair users, and the researchers *compared* to the wheelchair users. For the latter calculation, we compare *majority vote* data from each group so *N*=2 rather than *N*=6.

4.5.2 Evaluating Image-Level Agreement and Performance

We computed inter-rater agreement on labels at the image level using Fleiss' kappa [110], which attempts to account for agreement expected by chance. As this was an image-level analysis, we tested for agreement based on the absence or presence of a label in an image and not on the label's particular pixel location or severity rating. Multiple labels of the *same type* were compressed into a single "binary presence" indicator for that label. For example, if three individual *Surface Problems* were labeled

in an image, for our analysis, we only considered the fact that a *Surface Problem* was detected and not how many occurrences there were exactly. This helped control for different annotator tendencies—some who would provide one large label to cover contiguous problem areas and others who would provide separate labels. Results are shown in Table 4.3 for both binary and multiclass classification (*N* represents the number of annotators and *I* the number of images, Table 4.4. uses the same notation).

Three key results emerge: first, both the researchers and the wheelchair users had moderate to substantial levels of agreement [110], which indicates that the labeling task, at least at the image-level, is feasible and that the labels are fairly consistent across labelers; second, and just as importantly, the third column in Table 4.4 shows high agreement *between* the majority vote data of the research team and the wheelchair users, which indicates that the accessibility problems identified by the research team are consistent with "experts"; and, finally, the multiclass agreement results show that *Object in Path* and *Surface Problem* have more disagreement than *No Curb Ramp* and *Sidewalk Ending*. This is because *Object in Path* and *Surface Problems* are often less salient in images and because they are occasionally substituted for one another (*e.g.*, some labelers perceive a problem as *Object in Path* while others as a *Surface Problem*).

4.5.3 Evaluating Pixel-Level Agreement and Performance

Calculating pixel-level agreement is more challenging. Because no widespread standards exist for evaluating pixel-level agreement for human labelers, we followed the process prescribed by Martin *et al.* [123]. We verify the labeling process by

Pixel-Level Label Specificity	Correctness Measure	Image Comparisons	Researchers (*N*=3, *I*=229)	Wheelchair Users (*N*=3, *I*=75)	Researchers vs. Wheelchair Users (*N*=2 groups, *I*=75)
Binary Classification	Area Overlap	Same	0.31 (0.21)	0.26 (0.22)	0.27 (0.21)
		Different	0.02 (0.05)	0.01 (0.04)	0.01 (0.04)
	F-Measure	Same	0.43 (0.25)	0.37 (0.26)	0.38 (0.26)
		Different	0.03 (0.08)	0.02 (0.06)	0.03 (0.07)
Multiclass Classification	Area Overlap	Same	0.27 (0.21)	0.22 (0.22)	0.23 (0.21)
		Different	0.01 (0.03)	0.00 (0.02)	0.00 (0.02)
	F-Measure	Same	0.38 (0.26)	0.32 (0.27)	0.33 (0.27)
		Different	0.01 (0.05)	0.01 (0.04)	0.01 (0.04)

Table 4.4: The results of our pixel level agreement analysis (based on [123]**) between the researchers, wheelchair users, and researchers compared to wheelchair users. Similar to Table 4.1, the rightmost column is majority vote data. Cell format: average (stdev).**

showing that pixel-level label overlap and f-measure scores are higher between labelers on the *same* image than across *different* images. These scores will later act as a baseline for defining good performance when evaluating turker labels. To compare between the *same* images, 678 comparisons are required (3 annotators x 229 images). For *different* images, 156,636 comparisons are required (3 annotators x (229 x 229) - 229). Because the wheelchair users only labeled 75 of the 229 images, their comparison count is correspondingly lower (225 for same, 16,650 for different). We ignore images for which all annotators labeled *No Problems Found* (as no pixel labels exist in these images). Our results are shown in Table 4.5.

From these results, we conclude that our pixel level annotations across labelers are reasonably consistent, although less so than for image level. Unsurprisingly, agreement is higher for binary classification than for multiclass, though not substantially. This indicates that a major source of disagreement is not the label type (*e.g.*, *Object in Path* vs. *Surface Problem*) but rather the pixels highlighted by the outline shape. We emphasize, however, that pixel outlines for even the same object across labelers will rarely agree perfectly; the key then, is to determine what level of

overlap and *f-measure* scores are acceptable and good. Our results suggest that *overlap* scores of 0.31 and 0.27 and *f-measure* scores of 0.43 and 0.38 for binary and multiclass classification respectively are indicative of what a motivated and diligent annotator can achieve. We emphasize that even 10-15% overlap agreement at the pixel level would be sufficient to confidently localize problems in images; however, this level of consistency may not be sufficient for training computer vision. We return to this point in the discussion.

4.5.4 Producing Ground Truth Datasets

Finally, now that we have shown the feasibility of the labeling task and found reasonably high consistency amongst labelers, we can use these Study 1 labels to produce a ground truth dataset for evaluating turker performance. We consolidate the labeling data from the three researchers into four unified ground truth datasets: binary and multiclass at both the image and the pixel level

Consolidating Image-Level Labels: To combine image-level labels across the three labelers, we simply create a majority-vote "ground truth" dataset. Any *image* that received a label from at least two of the three researchers was assigned that label as "ground truth."

Figure 4.6. Examples of ground truth labels. (a) All three researchers labeled the object blocking the path. One researcher labeled fallen leaf on the ground as a surface problem, but this label was filtered out by ground truth label consolidation process. (b) Labels of missing curb ramps by three researchers. (c) Three researchers labeled the end of the sidewalk as Prematurely Ending Sidewalk.

Consolidating Pixel-Level Labels: Combining labels from the three researchers at the pixel level is less straightforward. The consolidation algorithm will directly impact the results obtained from our correctness measures. For example, if we combine highlighted pixel areas across all three researchers (union), then turker precision is

likely to go up but recall is likely to go down. If, instead, we take the intersection across all three labelers, the ground truth pixel area will shrink substantially, which will likely increase turker recall but reduce precision. Consequently, we decided to, again, adopt a majority vote approach. To produce the majority vote pixel-level dataset, we look for labels from at least two of the three researchers that overlap by 15% of their unioned area. The value of 15% was chosen because it is the lower-quartile cutoff using researcher overlap data. For binary classification, the label type was ignored—thus, any labels that overlapped by 15% or more were combined. For multiclass, the labels had to be of the same type.

4.6 Study 2: Crowd Worker Performance

To investigate the potential of using untrained crowd workers to label accessibility problems, we posted our task to Mechanical Turk during the summer of 2012. Each "hit" required labeling 1-10 images for 1-5 cents (0.5 to 5 cents per image). Each turker new to the task was required to watch at least half of a 3-minute instructional video, after which the labeling interface automatically appeared. Note: one task encompasses labeling one image.

We first highlight high-level results before performing a more detailed analysis covering labeler count *vs.* accuracy, two quality control evaluations, and the best and worst performing images. For the analysis below, we do not consider severity ratings.

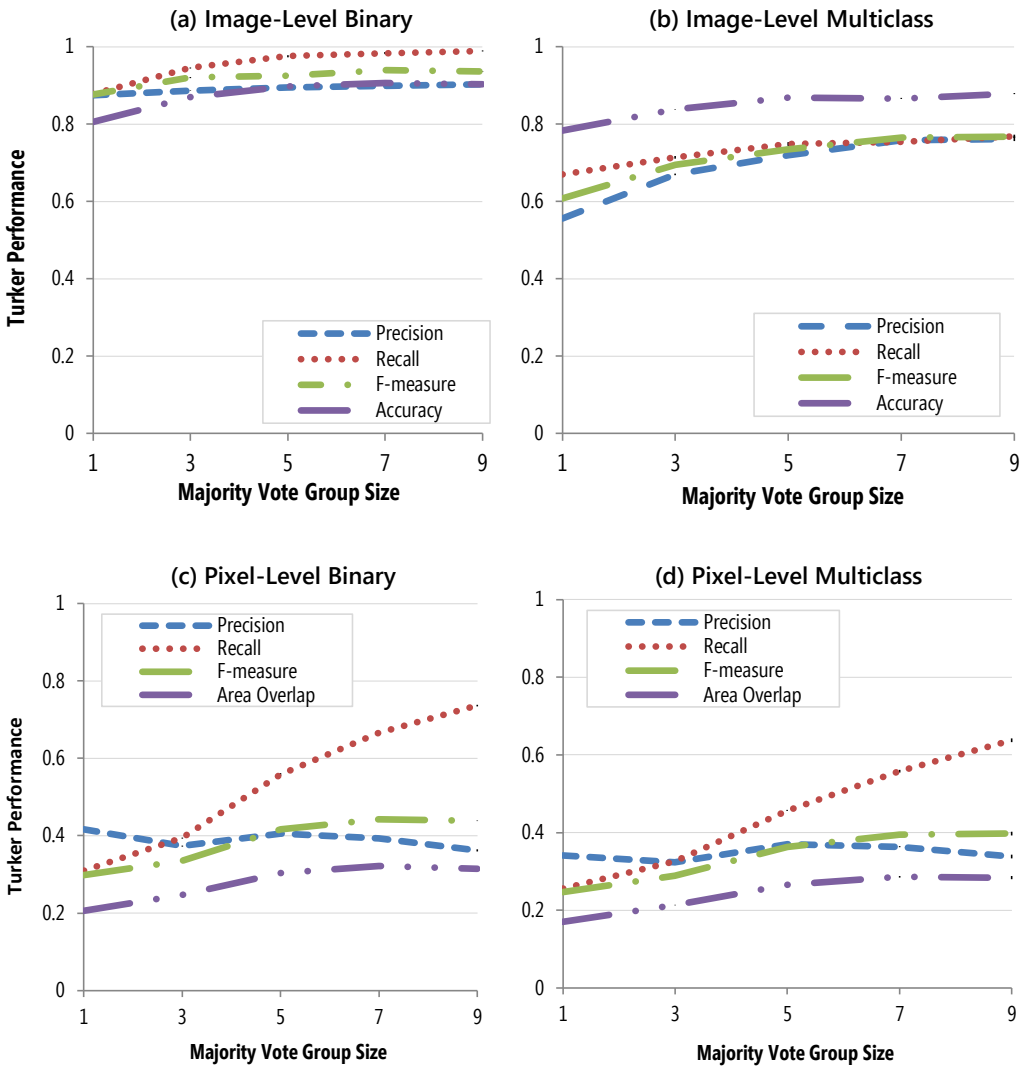


Figure 4.7: Binary and multiclass performance at the image- and pixel-levels with varying majority vote group sizes. Each graph point is based on multiple permutations of the majority vote group size across all 229 images. Standard error bars are in black (barely visible due to low variance).

Instead, we leave this for future work. However, given that we found a high rate of false positives amongst the turker data, we did examine the effect of removing labels that received a severity rating of a 1 (*Passable*) or a 2 (*Fairly Passable*). Our findings did not change significantly as a result.

4.6.1 High-Level Results

In all, we hired 185 distinct turkers who completed 7,534 image labeling tasks and provided a total of 13,379 labels. Turkers completed an average of 41.5 tasks (*SD*=61.4); 20 turkers labeled only 1 image and 10 turkers labeled all 229. The median image labeling time was 33.3s (*SD*=89.0s) and the average number of labels per image was 1.54 (*SD*=1.46). When compared with our ground truth dataset, overall turker accuracy at the *image* level was 80.6% for binary classification and 78.3% for multiclass classification. At the pixel level, average area overlap was 20.6% and 17.0% for binary and multiclass, respectively. These numbers are reasonably close to the values of 27% and 23% that we saw for wheelchair users *vs.* researchers.

4.6.2 Accuracy as a Function of Turkers per Image

Collecting multiple annotations per image helps account for the natural variability of human performance and reduces the influence of occasional errors; however, it also requires more workers [168]. Here, we explore accuracy as a function of turkers per image. We expect that accuracy should improve as the number of turkers increases, but the question then, is by how much? To evaluate the impact of the number of turkers on accuracy, we collected labels from 28 turkers for *each* of our 229 images. We compare our majority vote ground truth data with majority vote data across five turker group sizes: 1, 3, 5, 7, and 9. Because we have 28 turkers per image, we can run the analysis multiple times for each group size and average the results. For example, when we set the majority vote group size to three, we randomly permute nine groups of three turkers.

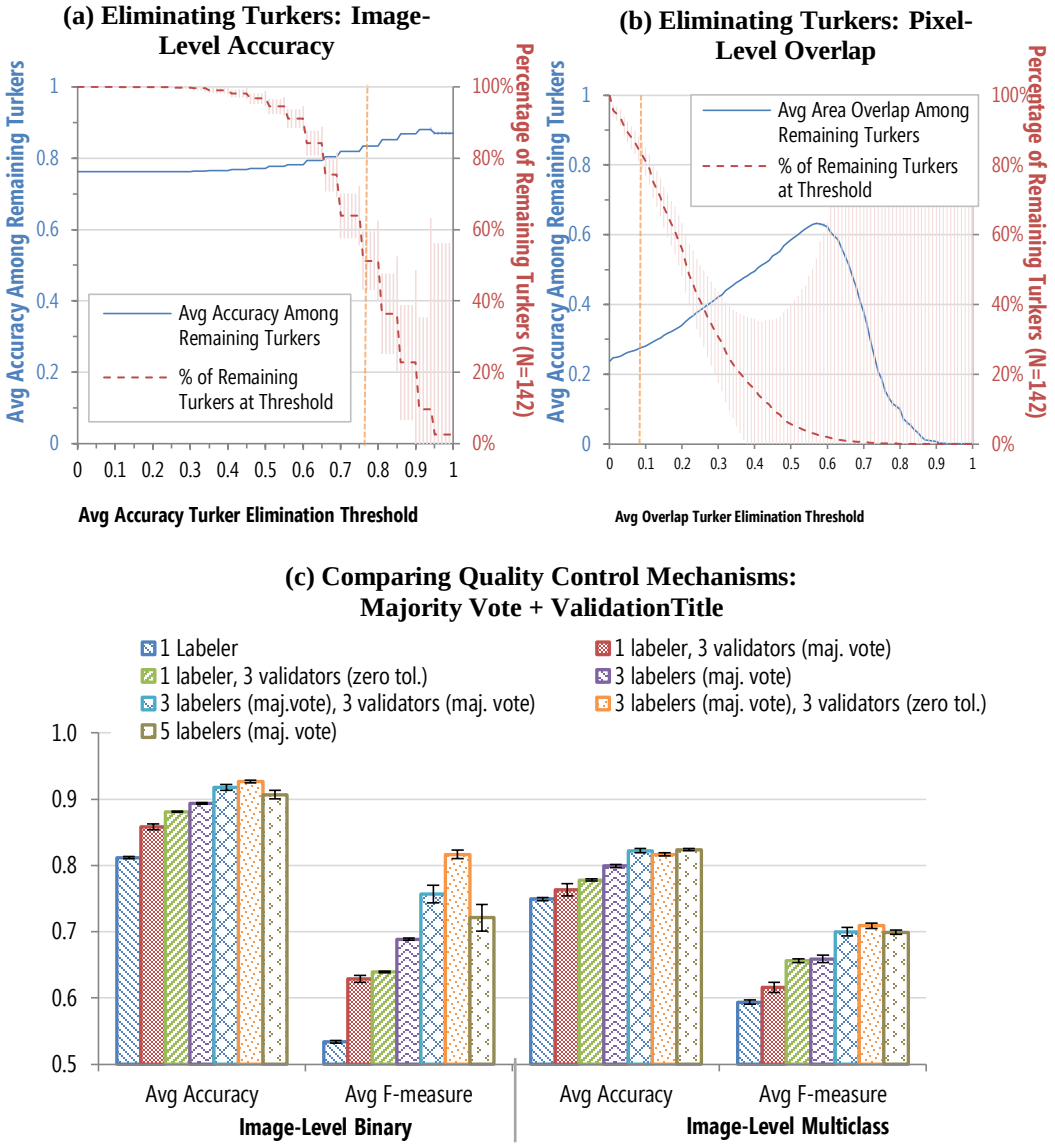


Figure 4.8: (a and b) Show the effect of increasingly aggressive turker elimination thresholds at the image- and pixel-levels based on average multiclass performance of 5 images. Error bars are standard deviation (for blue) and standard error (for red). As the threshold increases, fewer turkers remain and uncertainty increases. (c) Compares the effectiveness of various quality control mechanisms on performance at the image level.

In each group, we calculate the majority vote answer for a given image in the dataset and compare it with ground truth. This process is repeated across all images and the five group sizes, where (*X=majority vote group size*, *Y=number of groups*): (1,28), (3,

Image-Level Label Specificity	Label	Maj Vote Size: 1	Maj Vote Size: 3	Maj Vote Size: 5	Maj Vote Size: 7	Maj Vote Size: 9
Binary	No Prob vs. Prob	80.6 (0.1)	86.9 (0.3)	89.7 (0.2)	90.6 (0.2)	90.2 (0.2)
Multiclass	No Curb Ramp	78.6 (0.1)	86.0 (0.1)	90.2 (0.3)	91.6 (0.2)	93.7 (0.3)
	Object in Path	73.0 (0.1)	78.1 (0.2)	81.3 (0.3)	82.2 (0.1)	83.4 (0.2)
	Sidewalk Ending	84.7 (0.1)	88.3 (0.1)	88.5 (0.4)	89.5 (0.4)	89.8 (0.3)
	Surface Problem	77.0 (0.1)	82.1 (0.2)	84.9 (0.3)	85.9 (0.4)	88.4 (0.3)
	Overall	78.3 (0.0)	83.8 (0.1)	86.8 (0.2)	86.6 (0.2)	87.9 (0.1)

Table 4.5: Binary and multiclass label type accuracy at the image level across five majority vote group sizes. Cell format: avg% (stderr %).

9), (5,5), (7, 4), (9, 3). To compute the majority vote *answer* for each group size, we use the same label consolidation process as that used for the researcher majority vote labels.

We conducted this analysis at the image and pixel levels for binary and multiclass classification across our multiple correctness measures. Results are shown in Figure 4.7 (image and pixel level) and Table 4.5 (image level only). As expected, performance improves with turker count but these gains diminish in magnitude as group size grows. For example, at the image level, binary accuracy improves from 80.6% to 86.9% with 3 turkers and to 89.7% with 5 turkers but only to 90.2% with 9 turkers. For image-level multiclass, we see a similar trend. At the pixel level, the binary area overlap measure improves from 20.6% to 30.3% with 5 turkers but only to 31.4% with 9 turkers. Again, multiclass performance is similar (see Figure 4.7d). Even though group sizes beyond 5 continue to improve results at both the image and pixel level, this benefit may not be worth the additional cost.

Note that for the pixel level, the recall score rises dramatically in comparison to other metrics. This is because the consolidated majority vote pixel area tends to grow

with turker count (with more pixels labeled, recall will go up). Different consolidation processes will produce different results. Finally, similar to Study 1, *Sidewalk Ending* and *No Curb Ramp* labels performed better than *Object in Path* and *Surface Problem* (Table 4.5).

4.6.3 Quality Control Mechanisms

We explore two quality control approaches: filtering *turkers* based on a fixed threshold of acceptable performance and filtering *labels* based on crowdsourced validations collected through our verification interface. In both cases, we perform our analyses offline, which allows us to simulate performance with a range of quality control mechanisms.

For the first approach, we explored the effect of eliminating turkers based on their average multiclass performance at both the image and pixel level. The goal here is to uncover effective performance thresholds for eliminating poor quality turkers. We assign measure of errors to image-level and pixel-level correctness by using a Monte Carlo-based resampling approach called Bootstrap [57]. We first eliminate all turkers from our dataset who had completed fewer than five tasks. We then take samples of the remaining 142 turkers with replacement. For each sampled turker we randomly select five tasks that s/he completed to measure their average multiclass accuracy (for image level) or multiclass overlap (for pixel level). We shift our elimination threshold by increments of 0.01 and reject turkers if their average performance is lower than this threshold. At each increment, we also calculate overall performance across *all tasks*

among the remaining turkers. We repeat this process independently at the image and pixel levels N=1000 times to calculate error bars.

Results are shown in Figure 4.8 (a and b). In both figures, we see overall performance steadily increase as poor performing turkers get eliminated. However, the threshold where elimination takes effect differs between the two mechanisms due to differences in difficulty. For example, to achieve the same accuracy level as we would expect from majority vote with 3 turkers (0.84), the average performance elimination threshold needs to be 0.76 (marked in orange in the graph). At that threshold, image-level multiclass accuracy amongst the remaining turkers goes up to 0.84, but at a cost of eliminating 51.2% of our workforce. For pixel-level data, to achieve a score similar to the average area overlap between researcher labels (0.27), the elimination threshold needs to be set to 0.08, which increases the overlap score from 0.24 to 0.27 but reduces our workforce by 15% (again, orange line in graph). Thus, as expected, our results show accuracy gains with increasingly aggressive elimination thresholds; however, these accuracy gains come at a cost of reducing the effective worker pool. We expect that future systems can use these results to identify poor performing turkers *proactively* during data collection via ground truth seed images (*e.g.,* see [146]), and either offer additional training, or, in the extreme case, blacklist them. The threshold used depends on the accuracy needs of the application.

For the subjective validation approach, we use our verification interface (Figure 4.5). Here, turkers validate existing labels rather than provide new ones. We ensured that the same turker did not provide and validate the same image. As the validation task

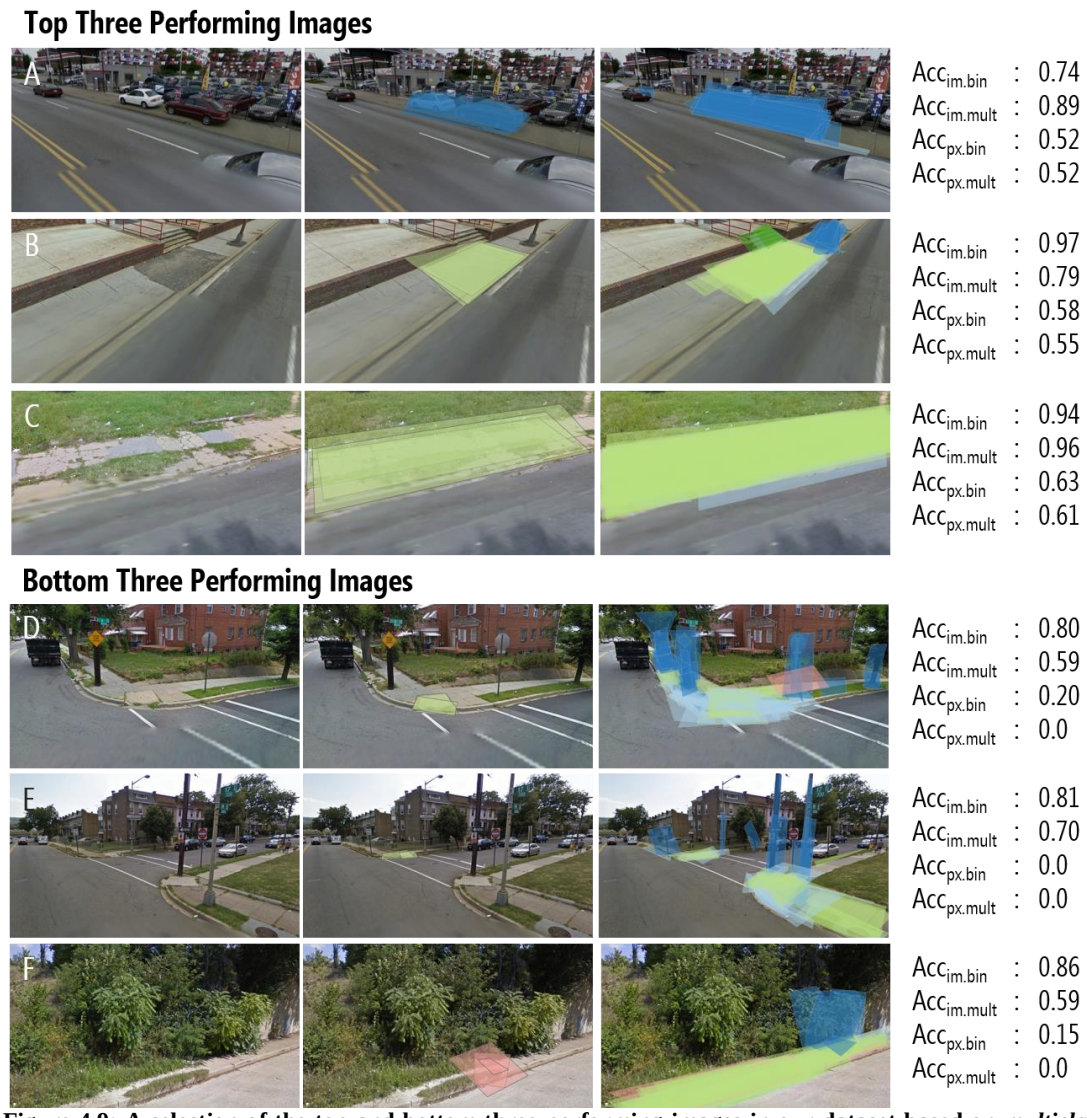


Figure 4.9: A selection of the top and bottom three performing images in our dataset based on *multiclass* pixel-level area overlap. Left column: original GSV image; center column: majority vote ground truth from researchers using 15% overlap; right column: turker labels. Numbers show turker performance results for that image, from top to bottom: image-level binary, image-level multiclass; pixel-level binary, pixel-level multiclass.

is simpler than the labeling task, we batched 20 validations into a single hit at a cost of 5 cents. We collected three or more validations per label across 75 images (the same subset used by the wheelchair users in Study 1). In all, we collected 19,189 validations from 273 turkers. Whereas the median time to label an image was 35.2s, the median

time to validate a label was 10.5s. Thus, collecting validations is quicker and cheaper than collecting new labels.

We performed a series of analyses with the validation data, using both majority vote validation and zero tolerance validation. For the latter, if any validator down-votes a label, that label is eliminated. We compare these results to no quality control (baseline), the use of majority vote labels, and a combination of majority vote labels plus subjective validation. Results are in Figure 4.8. As before, performance improves with additional turkers—either as labelers *or* as validators. The best performing quality control mechanism was 3 labelers (majority vote) plus 3 validators (zero tolerance) beating out 5 labelers (majority vote). This suggests that it is more cost effective to collect 3 labels *with* validation than 5 labels total per image, particularly given that validation requires less effort.

4.6.4 Best and Worst Performing Images

Figure 4.9 shows a selection of the top and the bottom performing images based on pixel-level multiclass overlap. For the worst performing images, there are many false positives: utility poles and stop signs are labeled as obstacles even though they are not in the sidewalk path. Figure 4.9f highlights two additional common problems: first, sometimes problem types have ambiguous categories—in this case, the ground truth label indicates *Sidewalk Ending* while many turker labels selected *Surface Problem*; second, it is unclear *how much* of the problem area should be highlighted. For *Sidewalk Ending*, the ground truth labels highlight only the sidewalk termination point—some

turkers, however, would label this section *and* any beyond it with no sidewalk (thereby greatly reducing their pixel-level scores). Future interfaces could detect these mistakes and provide active feedback to the turker on how to improve their labeling. In contrast, for the best performing images, the accessibility problems are, unsurprisingly, more salient and the camera angle provides a relatively close-angle shot.

4.6.5 Evaluation of Severity Scores

We have thus far focused on assessing the accuracy of crowd worker-provided accessibility labels, but largely ignored their associated severity scores. In this subsection, we conduct an exploratory analysis of the five-point scale severity scores. We then evaluate the severity scores' inter-rater agreements to discuss the utility of the data.

Explorative Analysis. We first conduct an exploratory analysis of the severity scores of the crowd worker-provided labels using the same data mentioned in Study 2. Note, this analysis focuses on the severity scores of the following label types: Object in Path, Surface Obstacle, and Sidewalk Ending. We omit Missing Curb Ramp labels from the analysis because we did not ask crowd workers to provide severity scores for this category (all missing curb ramps are considered severe accessibility problems). After filtering out the Missing Curb Ramp labels, we had 9,170 labels in total (Object in Path: 5,106, Surface Obstacle: 2,868, Sidewalk Ending: 1,283).

Because severity ratings are associated with corresponding polygonal labels, it is necessary to differentiate each label even they are on the same Street View image

and provided by the same crowd worker. Therefore, unlike the pixel-level accuracy analysis in which we "flattened" the polygonal labels into pixel level bitmap data, we retain each polygon and use it as a unit of analysis.

We use the "correct" crowdsourced labels for our analysis. In the accuracy analysis, we argued that 10-15% label area overlap is sufficient to judge if two labels are placed on the same accessibility feature on the given Street View image. Thus, we consider the crowd worker-provided labels that overlap with ground truth labels with more than 15% as correct and filter out the incorrect labels. We use one researcher's labels as ground truth for simplicity; the ground truth consisted of 71 Object in Path, 73 Surface Obstacle, and 48 Sidewalk Ending on 182 images. Overall, we had 2,513 correct labels (Object in Path: 884, Surface Obstacle: 1,046, Sidewalk Ending: 583). On average, each researcher label had 14.4 corresponding correct crowdsourced labels (min=0, max=31, SD=6.9).

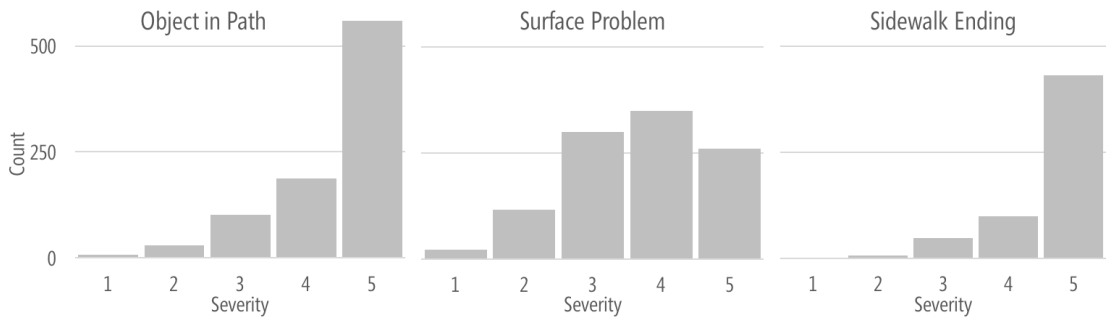


Figure 4.10. The histograms showing the distribution of severity scores associated with the correct labels provided by crowd workers. The raw counts are shown in Table 4.6.

	Severity				
Label Type	**1**	**2**	**3**	**4**	**5**
Object in Path	7	30	101	187	559
Surface Obstacle	22	116	300	349	259
Sidewalk Ending	0	6	47	99	431

Table 4.6. The number of severity scores associated with correct crowd worker-provided accessibility labels.

S**everity score distributions.**Table 4.6 and the histograms in Figure 4.10 depict the severity score distributions of each accessibility feature type. For all problem types, the histograms show the skew toward more severe ratings (3, 4, and 5). The severity scores for Object in Path and Sidewalk Ending labels are especially skewed toward 5 (*i.e.*, "not passable"), suggesting that crowd workers rate these accessibility features as significant mobility problems. Note, however, we may have categorized some labels as "incorrect" (and thus filtered out) even crowd workers appropriately labeled them and gave low severity scores. That is, there could be cases where the researcher did not label a less severe problem, but a crowd worker labeled and rated the problem as "passable" (*i.e.*, 1 or 2). Since these potentially valid labels are not separable from the labels that are actually inaccurate, we leave the assessment of these potentially valid labels as future work. Next, we evaluate the inter-rater agreement for severity scores.

Inter-rater Agreement Study Method. To assess the utility of the severity scores, we evaluate the inter-rater agreements to measure their reliability. The agreement of ordinal data such as our five point scale severity scores is often measured using Cohen's kappa or weighted kappa statistics [96]. Both agreement measures use a series of pairs of ordinal scores in their calculations, but a weighted kappa is a more relaxed measure compared to the original Cohen's kappa. For example, a pair of scores (5, 4) is considered 75% agreement in a weighted measure, whereas it is strictly counted as disagreement in a non-weighted counterpart—see [38,96] for more details. We use both of these measures to assess the reliability of the severity scores. In addition, we report a raw agreement score (# pairs with same severity score / # all pairs) for each label type. We report both raw agreement score and kappa statistics to illustrate the effect of agreements by chance.

As hinted in the previous paragraph, measuring agreement requires at least two labels that are associated with the same accessibility feature. Of 192 ground truth labels, 172 had at least two correct crowd worker labels (Object in Path: 59, Surface Obstacle: 65, Sidewalk Ending: 48). The following evaluation is done using these correct labels that overlap with ground truth.

Evaluating agreement statistics with only one series of severity score pairs does not reflect general characteristics of agreement. Therefore, we use Monte-Carlo approach to measure average agreements and their variability just like the accuracy analysis in the previous section. More specifically, for each accessibility problem, we

randomly sample a pair of two crowd worker labels that overlap with ground truth. Using a series of severity scores of the sampled pairs, we compute above mentioned agreement statistics. We repeat this sampling process for 1,000 times to measure their means and standard deviations (Table 4.7).

Label Type	Raw Agreement	Cohen's Kappa	Weighted Kappa
Object in Path (N=59)	0.48 (0.06)	0.05 (0.09)	0.11 (0.10)
Surface Problem (N=65)	0.31 (0.06)	0.06 (0.07)	0.13 (0.08)
Sidewalk Ending (N=48)	0.61 (0.06)	0.07 (0.11)	0.09 (0.11)

Table 4.7. Inter-rater agreement scores for each label type. The values show averaged scores over 1,000 Monte Carlo iterations with standard deviations in brackets.

Inter-rater Agreement Result and Discussion. Raw agreement varied from 0.31 to 0.61 (Table 4.7). The raw agreements on Object in Path and Sidewalk Ending were higher compared to the raw agreement on Surface Problem. However, both Cohen's kappa and weighted kappa suggest the agreements are weak (< 0.2) for all the label types [110]. The weighted kappas are slightly higher compared to corresponding unweighted measures. The fact that raw agreement scores are much higher than kappa statistics suggest that agreements are largely due to chance and the data distributions are skewed [139].

Our result shows that severity rating on accessibility problems labeled by crowd workers can agree from 31% to 61%. However, this is mostly due to chance as the kappa statistics suggest; as visualized in Figure 4.10, crowd workers rate problems as severe most of the time, so the scores could agree with high probability if they pick 4 or 5. This suggests that the scores of 4 or 5 themselves do not provide much information (*i.e.*, whatever is labeled tend to be a severe accessibility problem). Future work should

assess the value of rare data like scores 1 and 2, and investigate if rare severity scores could be useful for, for example, filtering out incorrect labels or characterizing each crowd workers (e.g., can a person who label less severe problems and rate as "passable" be considered more dedicated?).

4.7 Discussion and Conclusion

In this chapter, we showed that untrained crowd workers could find and label accessibility problems in GSV imagery. We also highlighted the effect of common quality-control techniques on overall performance accuracies. Here, we discuss limitations of our study and opportunities for future work.

We evaluated our approach with a manually curated database of images. Image quality was sometimes poor, either because of lighting conditions, which can often be auto-corrected, or blurriness. Camera angle was also fixed in our dataset. Providing multiple camera angles or even an interactive interface where users can control the camera angle should be explored; the GSV interface itself allows the user to control camera angle and zoom level. In part related to camera angle, future work should also explore how often sidewalks are obscured from view (*e.g.*, from parked cars) in GSV. Other data sources could be used to lessen this problem, such as high-resolution top-down satellite or fly-over imagery [174], volunteer-contributed geo-located pictures, or government 311 databases. GSV data can also be somewhat old (3.1 years in our dataset), a noted limitation in other virtual audit work as well [11]. Combining our GSV approach with other datasets should help mitigate this problem.

We did not take into account of the learning effect while creating the ground truth dataset. In Study 1, unlike experts (*i.e.,* wheelchair users) who labeled 75 Street View images, the researchers labeled all 229. This might have affected the agreement levels between researchers and experts (*e.g.*, the researchers were more accustomed to what accessibility problems are visible in Street View images). For the crowdsourcing study (Study 2), it would be an interesting future work to evaluate how fast crowd workers learn to correctly label accessibility features.

While we captured important accessibility characteristics of sidewalks, other problems may exist. For example, the wheelchair users in Study 2 indicated that sidewalk narrowness can also reduce accessibility. We did not have a way of measuring sidewalk width or providing a tool to assess narrowness. Future work should look at the ability to calculate widths, which could, perhaps, be reconstructed via the multiple camera angles offered by GSV or derived from the 3D-point cloud data that modern GSV cars collect (see [9]).

For quality control, future applications will obviously use a large majority of images where ground truth is *unknown*. Instead, "ground truth" seed images will need to be injected into the labeling dataset to actively measure turker performance (see [146]). Active monitoring will allow turkers to receive performance feedback, help assist them when they make common mistakes, and warn and, eventually, eliminate poor quality workers if they do not improve. Beyond turkers, we also build a volunteer-based participatory website to both visualize our results and highlight areas that need data collection. We discuss this in Chapter 6.

Our general approach of collecting useful, street-level information in a scalable manner from GSV images has application beyond sidewalks. We would like to expand our approach to assess the accessibility of building fronts, friction strips and stop lights at intersections, and non-accessibility related topics such as tracking and labeling bike lanes in roadways. Finally, accessible public rights-of-way do not just offer benefits to people with disabilities, they are also generally safer and more user-friendly for all pedestrians [122]. Our work effectively demonstrates a promising new, highly scalable method for acquiring knowledge about sidewalk accessibility.

Chapter 5 Detecting Curb Ramps in Google Street View Using Crowdsourcing, Computer Vision, and Machine Learning

This chapter describes our work on a system that combines crowdsourcing, computer vision, and machine learning to efficiently label curb ramps in Google Street View (GSV) images. The work was done in a collaboration with another graduate student Jin Sun and Prof. David Jacobs. Jin contributed in developing automated curb ramp detector. This chapter has adapted and rewritten content from a paper at UIST 2014 [84].

5.1 Introduction

Previous work has examined how to leverage massive online map datasets such as GSV along with crowdsourcing to collect information about the accessibility of the built environment [73,77,80,81,82]. Early results have been promising; for example, using a manually curated set of static GSV images, we found that minimally trained crowd workers in Amazon Mechanical Turk (turkers) could find four types of street-level accessibility problems with 81% accuracy [81]. However, the sole reliance on human labor limits scalability.

In this chapter, we present Tohme[1], a scalable system for remotely collecting geo-located curb ramp data using a combination of crowdsourcing, computer vision

[1] Tohme is a Japanese word that roughly translates to "remote eye."

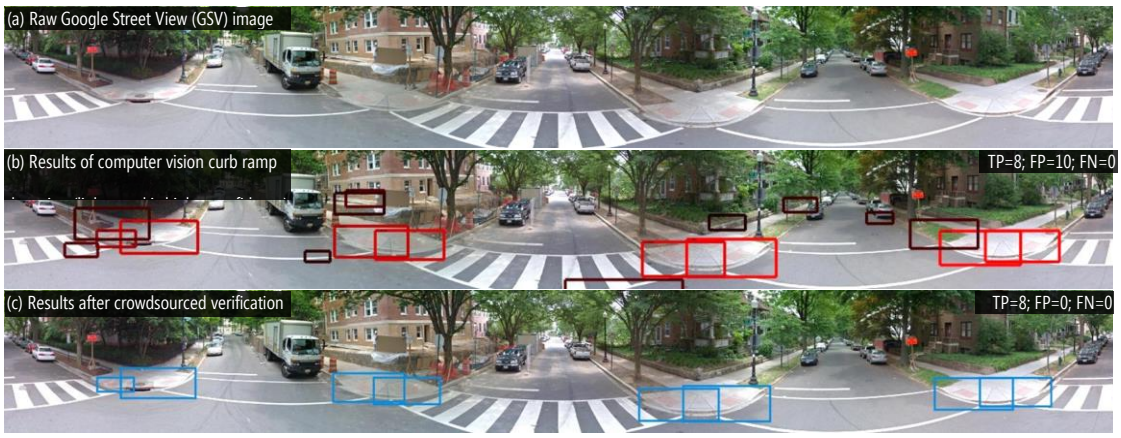


Figure 5.1: In this section, we present *Tohme*, a scalable system for semi-automatically finding curb ramps in Google Streetview (GSV) panoramic imagery using computer vision, machine learning, and crowdsourcing. The images above show an actual result from our evaluation.

(CV), machine learning, and online map data. Tohme lowers the overall human time cost of finding accessibility problems in GSV while maintaining result quality (Figure 5.1). As the first work in this area, we limit ourselves to sidewalk curb ramps, which we selected because of their visual salience, geospatial properties (*e.g.*, often located on corners), and significance to accessibility. For example, in a precedent-setting US court case in 1993, the court ruled that the "*lack of curb cuts is a primary obstacle to the smooth integration of those with disabilities into the commerce of daily life*" and that "*without curb cuts, people with ambulatory disabilities simply cannot navigate the city*" [1].

While some cities maintain a public database of curb ramp information (*e.g.*, [178,179]), this data can be outdated, erroneous, and expensive to collect. Moreover, it is not integrated into modern mapping tools. In a recent report, the National Council on Disability noted that they could not find comprehensive information on the "*degree to which sidewalks are accessible*" across the U.S. [132]. In addition, the quality of data available in government systems is contingent on the specific policies and technical

infrastructure of that particular local administration (*e.g.,* at the city and/or county level). While federal US legislation passed in 1990 mandates the use of ADA-compliant curb ramps in all new road construction and renovation [191], this is not the case across the globe. Our overarching goal is to design a scalable system that can remotely collect accessibility information for any city across the world that has streetscape imagery, which is now broadly available in GSV, Microsoft Bing Maps, and Nokia City Scene.

Tohme is comprised of four custom parts: (i) a web scraper for downloading street intersection data; (ii) two crowd worker interfaces for finding, labeling, and verifying the presence of curb ramps; (iii) state-of-the-art CV algorithms for automatic curb ramp detection; and (iv) a machine learning-based workflow controller, which predicts CV performance and dynamically allocates work to either a human labeling pipeline or a CV + human verification pipeline. While Tohme is purely a data collection system, we envision future work that integrates Tohme's output into accessibility-aware map tools (*e.g.,* a heatmap visualization of a city's accessibility or a smart navigation system that recommends accessible routes).

To evaluate Tohme, we conducted two studies using data we collected from 1,086 intersections across four North American cities. First, to validate the use of GSV imagery as a reliable source of curb ramp knowledge, we conducted physical audits in two of these cities and compared our results to GSV-based audit data. As with previous work exploring the concordance between GSV and the physical world

[11,39,73,77,157], we found high correspondence between the virtual and physical audit data. Second, we evaluated Tohme's performance in detecting curb ramps across our entire dataset with 403 turkers. Alone, the computer vision sub-system currently finds 67% of the curb ramps in the GSV scenes. However, by dynamically allocating work to the CV module or to the slower but more accurate human workers, Tohme performs similarly in detecting curb ramps compared to a manual labeling approach alone (F-measure: 84% *vs*. 86% baseline) but at a 13% reduction in human time cost.

In summary, the primary contribution of this paper is the design and evaluation of the Tohme system as a whole, with secondary contributions being: (i) the first design and evaluation of a computer vision system for automatically detecting curb ramps in images; (ii) the design and study of a "smart" workflow controller that dynamically allocates work based on predicted scene complexity from GIS data and CV output; (iii) a comparative physical *vs.* virtual curb ramp audit study (Study 1), which establishes that GSV is a viable data source for collecting curb ramp data; and (iv) a detailed examination of why curb ramp detection is a hard problem and opportunities for future work in this domain.

5.2 Dataset

Because sidewalk infrastructure can vary in quality, design, and appearance across geographic areas, our study sites include a range of neighborhoods from four North American cities: Washington, D.C., Baltimore, Los Angeles, and Saskatoon,

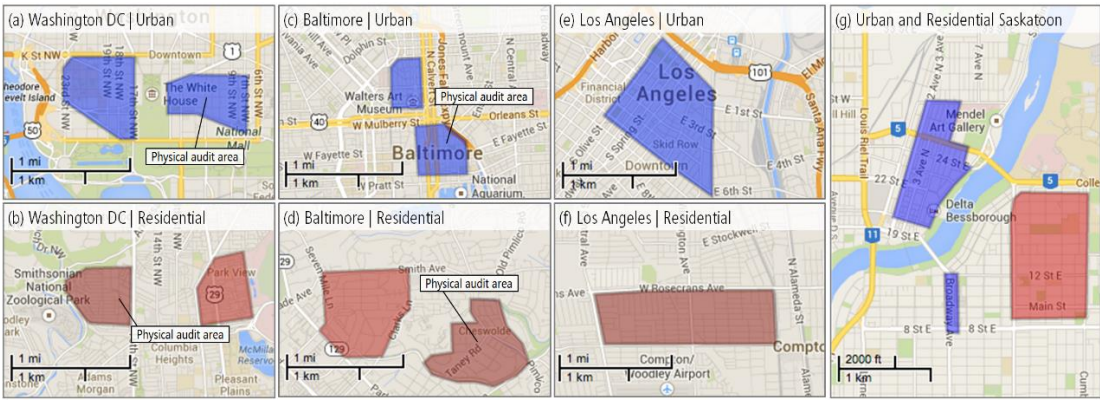


Figure 5.2: The eight urban (blue) and residential (red) audit areas used in our studies from Washington DC, Baltimore, LA, and Saskatoon. This includes 1,086 intersections across a total area of 11.3km^2. Among these areas, we physically surveyed 273 intersections (see annotations in a-d).

	Washington DC		Baltimore		Los Angeles		Saskatoon		Overall
Region Type	Downtown	Residential	Downtown	Residential	Downtown	Residential	Downtown	Residential	
Total Area (km^2)	1.52	1.13	0.73	2.24	1.91	1.89	0.74	1.13	11.28
# of Intersections	140	124	132	139	132	132	141	146	1,086
# of Curb Ramps*	818	352	476	229	358	186	321	137	2877
# of Missing Curb Ramps*	8	35	32	69	43	214	24	222	647
Avg. GSV Data Age (SD)	1.9 yrs (0.77)	1.6 yrs (0.63)	2.1 yrs (0.75)	0.4 yrs (0.65)	2.0 yrs (0.31)	0.9 yrs (0.24)	4.0 yrs (0.0)	4.0 yrs (0.0)	2.2 (1.3)

Table 5.1: A breakdown of our eight audit areas. Age calculated from summer 2013. *These counts are based on ground truth data.

Saskatchewan (Figure 5.2; Table 5.1). For each city, we collected data from dense urban cores (shown in blue) and semi-urban residential areas (shown in red). We emphasized neighborhoods with potential high demand for sidewalk accessibility (*e.g.*, areas with schools, shopping centers, libraries, and medical clinics).

We used two data collection approaches: (i) an automated web scraper tool that we developed called *svCrawl*, which downloads GIS-based intersection data, including GSV images, within a geographically defined region; and (ii) a physical survey of a subset of our study sites (four neighborhoods totaling 273 intersections), which was used to validate curb ramp infrastructure found in the GSV images. In all, we used *svCrawl* to download data from 1,086 intersections across 11.3km^2 (Table 5.1).

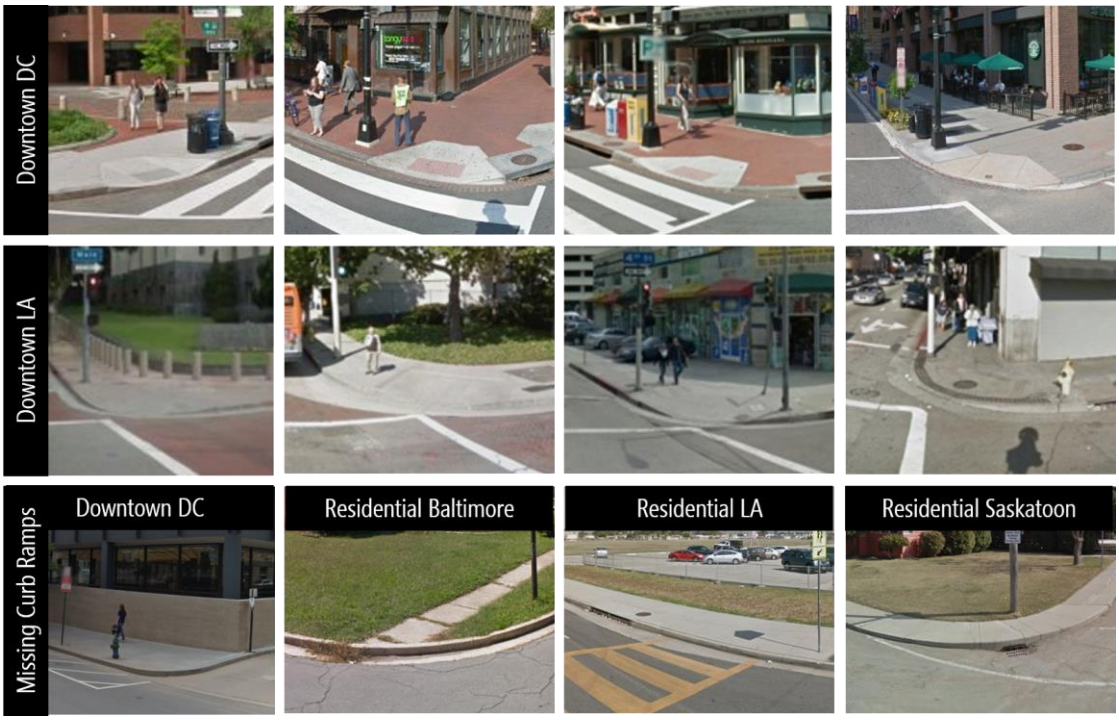


Figure 5.3. Example curb ramps (top two rows) and missing curb ramps (bottom row) from our GSV dataset

To create a ground truth dataset, two members of our research team independently labeled all 1,086 scenes using our custom labeling tool (svLabel). Label disagreements were resolved by consensus. From the ground truth data, we discovered 2,877 curb ramps and 647 missing curb ramps (Figure 5.3). Of the 1,086 scenes, 218 GSV scenes did *not* require marking a curb ramp or missing curb ramp because the location was not a traditional intersection (*e.g.*, an alleyway with no vertical drop from the sidewalk). These 218 scenes are useful for exploring false positive labeling behavior and were kept in our dataset. The remaining 868 intersections had on average 3.3 curb ramps (*SD*=2.3) and 0.75 missing curb ramps (*SD*=1.3) per intersection. A total of 603/868 intersections were marked as *not* missing any curb ramps. We use the

ground truth labels for training and testing our machine learning and CV algorithms and to evaluate crowd worker performance.

At download time (summer 2013), the average age of the GSV images was 2.2 years (*SD*=1.3). As image age is one potential limitation in our approach, it is necessary to first show that GSV is a reasonable data source for deriving curb ramp information, which we do next.

5.3 Study 1: Assessing GSV as a data source

To establish GSV as a viable curb ramp data source, we must show: (i) that it presents unoccluded views of curb ramps, (ii) that the curb ramps can be reliably found by humans and, potentially, machines, and (iii) that the curb ramps found in GSV adequately reflect the state of the physical world. This study addresses each of these points. Multiple studies have previously demonstrated high concordance between GSV-based audits and audits conducted in the physical world [11,39,73,77]; however, prior work has not examined curb ramps specifically. Though this audit study was labor intensive, it is important to establish GSV as a reliable data source for curb ramp information, as it is the crux of our system's approach.

We conducted physical audits in the summer of 2013 across a subset of our GSV dataset: 273 intersections spanning urban and residential areas in Washington, D.C. and Baltimore (Figure 5.2). We followed a physical audit process similar to Hara *et al.* [77]. Research team members physically visited each intersection, capturing geo-timestamped pictures (*Mean*=15 per intersection; *SD*=5). These images were analyzed

post hoc for the actual audit. Surveying the 273 intersections took approximately 25 hours as calculated by image capture timestamps.

5.3.1 Auditing Methodology.

For the auditing process itself, two additional research assistants (different from the above) independently counted the number of *curb ramps* and *missing curb ramps* at each intersection in both the physical and GSV image datasets. An initial visual codebook was composed based on US government standards for sidewalk accessibility [102,191]. Following the iterative coding method prescribed by Hruschka *et al.* [89], a small subset of the data was individually coded first (five intersections from each area). The coders then met, compared their count data, and updated the codebook appropriately to help reduce ambiguity in edge cases. Both datasets were then coded in entirety (including the original subset, which was recoded). This process was iterated until high agreement was reached.

	Physical Audit Image Dataset			GSV Audit Image Dataset		
	1st Pass (α)	2nd Pass (α)	3rd Pass (α)	1st Pass (α)	2nd Pass (α)	3rd Pass (α)
Curb Ramp	0.959	0.960	0.989	0.927	0.928	0.989
Missing C. Ramp	0.647	0.802	0.999	0.631	0.788	0.999
Overall	0.897	0.931	0.996	0.883	0.917	0.996

Table 5.2: Krippendorff's alpha inter-rater agreement scores between two researchers on both the physical audit and GSV audit image datasets. Following Hruschka *et al.'s* iterative coding methodology, a 3rd audit pass was conducted with an updated codebook to achieve high-agreement scores—in our case, α > 0.996.

5.3.2 Calculating Inter-Rater Reliability between Auditors

Before comparing the physical audit data to the GSV audit data, which is the primary goal of Study 1, we first calculated inter-rater reliability between the two coders for each dataset. We applied the Krippendorff's Alpha (α) statistical measure, which is

used for calculating inter-rater reliability of count data (see [107]). Results after each of the three coding passes using the iterative scheme from [89] are shown in Table 5.2. Agreement was consistently high, with the 3rd pass representing the reliability of codes in the final code set. There was initially greater inconsistency in coding *missing* curb ramps *vs.* coding existing curb ramps, perhaps because identifying a missing ramp requires a deeper under-standing of the intersection and proper ramp placement.

5.3.3 Comparing Physical vs. GSV Audit Data

With high agreement verified within each dataset, we can now compare the count scores *between* the datasets. Similar to [77,157], we calculate a Spearman rank correlation between the two count sets (physical and GSV). This was done for both the curb ramp and missing curb ramp counts. To enable this calculation, however, we first merged the two auditor's counts by taking the average of their counts for missing curb ramps and the average for present curb ramps at each intersection. Using these average counts, a Spearman rank correlation was computed, which shows high correspondence *between* datasets: ρ=0.996 for curb ramps and ρ=0.977 for missing curb ramps ($p < 0.001$). Overall, 1,008 curb ramps were identified in the virtual audit compared to 1,002 with the physical audit; differences were due to construction. The number of missing curb ramps was exactly the same for both datasets (89).

5.3.4 Study 1 Summary

Though the age of images in GSV remains a concern, Study 1 demonstrates that there is remarkably high concordance between curb ramp infrastructure in GSV and the

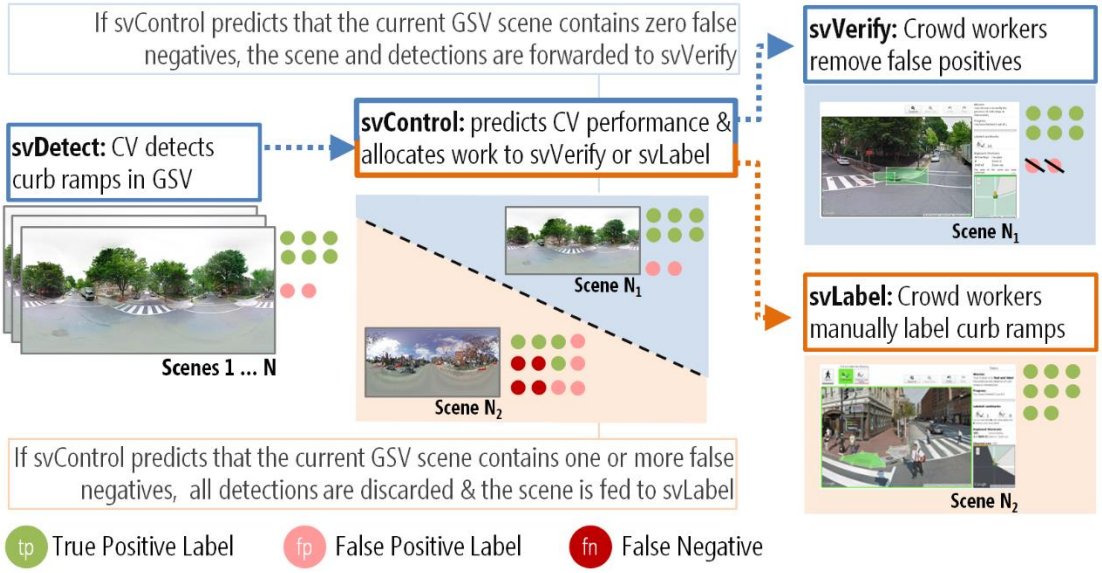


Figure 5.4. A workflow diagram depicting Tohme's four main sub-systems. In summary, *svDetect* processes every GSV scene producing curb ramp detections with confidence scores. *svControl* predicts whether the scene/detections contain a false negative. If so, the detections are discarded and the scene is fed to *svLabel* for manual labeling. If not, the scene/detections are forwarded to *svVerify* for verification. The workflow attempts to optimize accuracy and speed.

physical world, even though the average image age of our dataset was 2.2 years. With GSV established as a curb ramp dataset source, we now move on to describing Tohme.

5.4 A scalable system for Curb ramp detection

Tohme is a custom-designed tool for remotely collecting geo-located curb ramp information using a combination of crowdsourcing, CV, machine learning, and online map data. It is comprised of four parts depicted in Figure 5.4: (i) a web scraper, *Street View Crawl (svCrawl)*, for downloading street intersection data; (ii) two crowd worker interfaces for finding, labeling, and verifying the presence of curb ramps called *svLabel* and *svVerify*; (iii) state-of-the-art CV algorithms for automatically detecting curb ramps

(*svDetect*); and (iv) a machine learning-based workflow, called *svControl*, which predicts CV performance on a scenes and allocates work accordingly.

We designed Tohme iteratively with small, informal pilot studies in our laboratory to test early interface ideas. We also performed larger experiments on Amazon Mechanical Turk (MTurk) with a subset of our data to understand how different interfaces affected crowd performance and, more generally, how well crowds could perform our tasks. The CV sub-system, svDetect, also evolved across multiple iterations, and was trained and evaluated using the aforementioned ground truth labels. While our ultimate goal is to deploy Tohme publicly on the web, the current prototype and experiments were deployed on MTurk. Below, we describe each Tohme sub-system.

5.4.1 svCrawl: Automatic Intersection Scraping

svCrawl is a custom web scraper tool written in Python that downloads GIS-related intersection data over a predefined geographic region (Figure 5.2). It uses the Google Maps API (GMaps API) to enumerate and extract street intersection points within selected boundaries. For each intersection, svCrawl downloads four types of data:

1. **A GSV panoramic image** at its source resolution (13,312 x 6,656px). This is our primary data element (*e.g.*, Figure 5.1).
2. **A 3D-point cloud**, which is captured by the GSV car using LiDAR [9]. The depth data overlays the GSV panorama but at a coarser resolution (512 x 256px; Figure

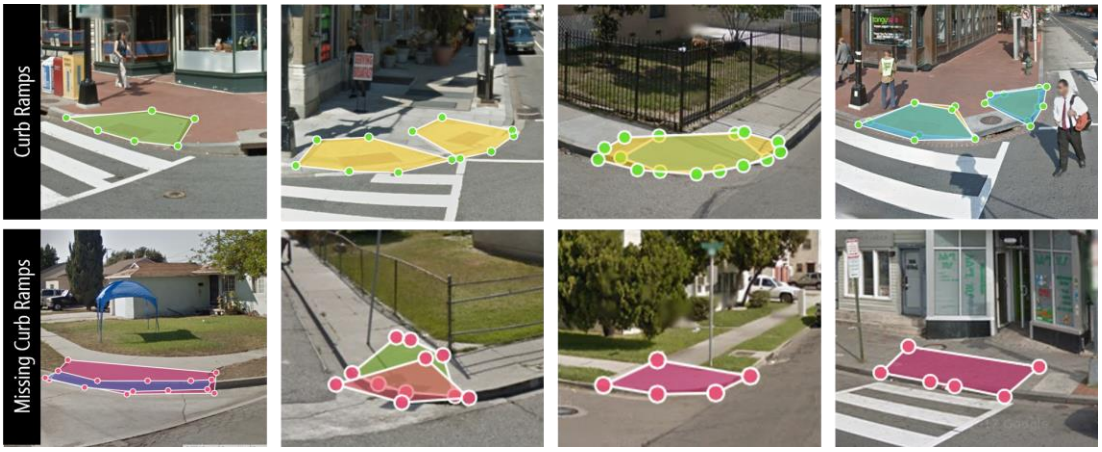


Figure 5.5. A workflow diagram depicting Tohme's four main sub-systems. In summary, svDetect processes every GSV scene producing curb ramp detections with confidence scores. svControl predicts whether the scene/detections contain a false negative. If so, the detections are discarded and the scene is fed to svLabel for manual labeling. If not, the scene/detections are forwarded to svVerify for verification. The workflow attempts to optimize accuracy and speed.

5.10). This is used by svDetect to automatically cull the visual search space and by svControl as an intersection complexity input feature.

3. **A top-down abstract map image** of the intersection obtained from the Google Maps API (Figure 5.13), which is used as a training feature in our work scheduler, svControl, to infer intersection complexity (like the depth data).

4. **Associated intersection GIS metadata**, also provided by the GMaps API, such as latitude/longitude, GSV image age, street and city names, and intersection topology.

5.4.2 svLabel: Human-Powered GSV Image Labeling

In Tohme, intersections are labeled either manually, via svLabel, or automatically via svDetect. svLabel is a fully interactive online tool written in JavaScript and PHP for finding and labeling curb ramps and missing curb ramps in GSV images (Figure 5.5-

Figure 5.6. The svLabel interface. Crowd workers use the Explorer Mode to interactively explore the intersection (via pan and zoom) and switch to the Labeling Mode to label curb ramps and missing curb ramps. Clicking the Submit button uploads the target labels. The turker is then transported to a new location unless the HIT is complete.

(a) After labeling one corner, the user pans to the right. (b) The user then zooms to get a closer look (c) The user begins labeling the new corner in the zoomed view

Figure 5.7. svLabel automatically tracks the camera angle and repositions any applied labels in their correct location as the view changes. When the turker pans the scene, the overlay on the map view is updated and the green "explored" area increases (bottom right of interface). Turkers can zoom in up to two levels to inspect distant corners. Labels can be applied at any zoom level and are scaled appropriately.

5.7). Unlike much previous crowd-sourcing GSV work, which uses static imagery to collect labels [73,80,81], our labeling tool builds on *Bus Stop CSI* [77] to provide a fully interactive 360 degree view of the GSV panoramic image. While this freedom increases user-interaction complexity, it allows the user to more naturally explore the intersection and maintain spatial context while searching for curb ramps. For example,

the user can pan around the virtual 3D-space from one corner to the next within an intersection.

Using svLabel. When a turker accepts our HIT, they are immediately greeted by a three-stage interactive tutorial. The stages progressively teach the turker about the interface (*e.g.,* the location of buttons and other widgets), user interactions (*e.g.,* how to label, zoom, and pan), and task concepts (*e.g.,* the definition of a curb ramp). If mistakes are made, our tutorial tool automatically provides corrective guidance. Turkers must successfully complete one tutorial stage before moving on to the next.

Once the tutorials are completed, we automatically position the turker in one of the audit area intersections and the labeling task begins in earnest. Similar to Bus Stop CSI [77], svLabel has two primary modes of interaction: *Explorer Mode* and *Labeling Mode* (Figure 5.6). When the user first drops into a scene, s/he defaults into Explorer Mode, which allows for exploration using Street View's native controls. Users are instructed to pan around to explore the 360 degree view of the intersection and visual feedback is provided to track their progress (bottom-right corner of Figure 5.6). Note: users' movement is restricted to the drop location.

When the user clicks on either the *Curb Ramp* or *Missing Curb Ramp* buttons, the interface switches automatically to Labeling Mode. Here, mouse interactions no longer control the camera view. Instead, the cursor changes to a pen, allowing the user to draw an outline around the visual target—a curb ramp or lack thereof (Figure 5.5). We chose to have users outline the area rather than simply clicking or drawing a bounding box because the detailed outlines provide a higher degree of granularity for

developing and experimenting with our CV algorithms. Once an outline is drawn, the user continues to search the intersection. Our tool automatically tracks the camera angle and repositions any applied labels in their correct location as the intersection view changes. In this way, the labels appear to "stick" to their associated targets. Once the user has surveyed the entire intersection by panning 360 degrees, s/he can submit the task and move on to the next task in the HIT, until all tasks are complete.

Ground Truth Seeding. A single HIT is comprised of either five or six intersections depending on whether it contains a ground truth scene (a scene is just an intersection). This "ground truth seeding" [146] approach is commonly used to dynamically examine, provide feedback about, and improve worker performance. In our case, if a user makes a mistake at a ground truth scene, after hitting the submit button, we provide visual feedback about the error and show the proper corrective action. The user must correct all mistakes before submitting a ground truth task. If no mistakes are detected, the user is congratulated for their good performance. In our current system, there is a 50% chance that a HIT will contain one ground truth scene. The user is not able to tell whether they are working on a ground truth scene until after they submit their work.

5.4.3 svVerify: Human-Powered GSV Label Verification

In addition to providing "curb ramp" and "missing curb ramp" labels, we rely on crowd workers to examine and verify the correctness of previously entered labels. This

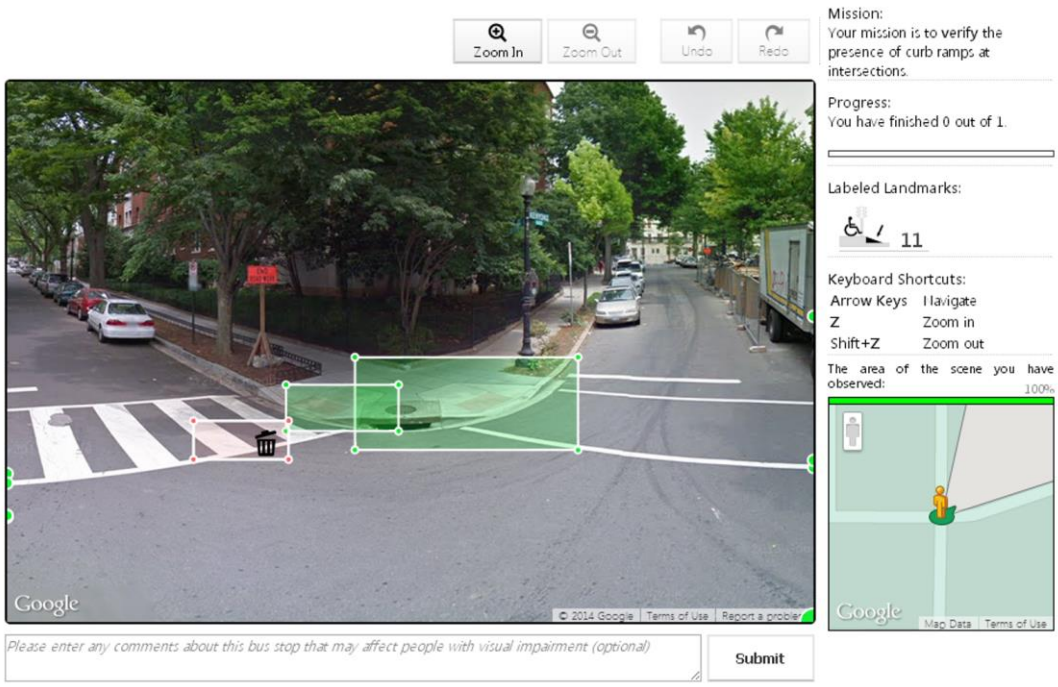


Figure 5.8. The svVerify interface is similar to svLabel but is designed for verifying rather than labeling. When the mouse hovers over a label, the cursor changes to a garbage can and a click removes the label. The user must pan 360 degrees before submitting the task.

verification step is common in crowdsourcing systems to increase result quality (*e.g.*, [81,173]). svVerify (Figure 5.8) is similar to svLabel in appearance and general workflow but has a simplified interaction (clicking and panning only) and is for an easier task (clicking on incorrect labels).

While we designed both svLabel and svVerify to maximize worker efficiency and accuracy, our expectation was that the verification task would be significantly faster than initially providing manual labels [173]. For verification, users need not perform a time-consuming visual search looking for curb ramps to label but rather can quickly scan for incorrect labels (false positives) to delete. And, unlike labeling, which requires drawing polygonal outlines, the delete interaction is a single click over the

offending label (similar to [194]). This enables users to rapidly eliminate *false positive* labels in a scene.

To maintain verification efficiency, however, we did not allow the user to spatially locate *false negatives*. This would essentially turn the verification task into a labeling task, by asking users to apply new "curb ramp" or "curb ramp missing" labels when they noticed a valid location that had not been labeled. Instead, svVerify gathers information on false negatives at a coarser-grained level by asking the user if the current scene was missing any labels after s/he clicks the submit button. Thus, svVerify can detect the presence of false negatives in an intersection but not their specific location or quantity.

Similar to svLabel, svVerify requires turkers to complete an interactive tutorial before beginning a HIT, which includes instructions about the task, the interface itself, and successfully verifying one intersection. Because verifications are faster than providing labels, we included 10 scenes in each HIT (*vs.* the 5 or 6 in svLabel). In addition, we inserted one ground truth scene into *every* svVerify HIT rather than with 50% probability as was done with svLabel. Note that not all scenes are sent to svVerify for verification, as discussed in the svControl section below. We move now to describing the two more technical parts of Tohme: svDetect and svControl.

5.4.5 svDetect: Detecting Curb Ramps Automatically

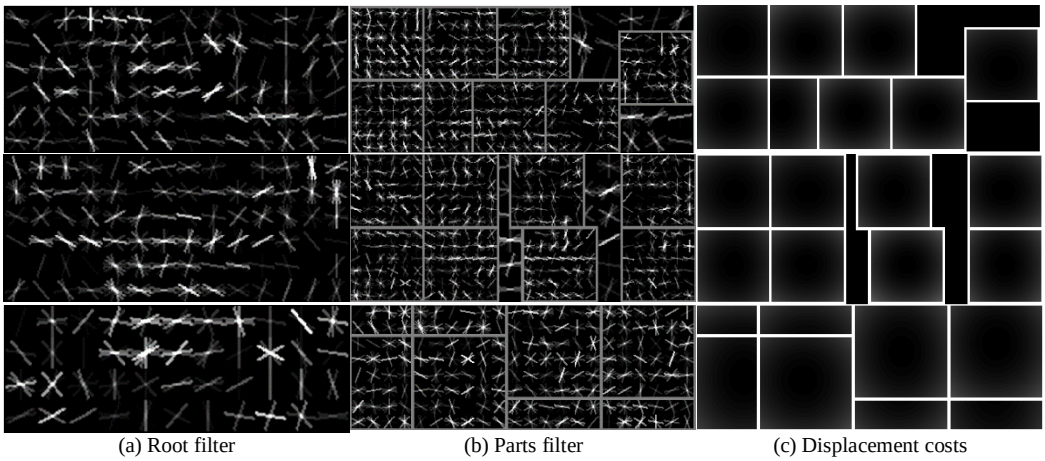
While svLabel relies on manual labeling for finding curb ramps, svDetect attempts to do this automatically using CV. Because CV-based object detection is still an open

problem—even for well-studied targets such as cars [63] and people [48]—our goal is to create a system that functions well enough to reduce the cost of curb ramp detection *vs.* a manual approach alone.

svDetect uses a three-stage detection process. First, we train a *Deformable Part Model (DPM)* [63], one of the most successful recent approaches in object detection (*e.g.*, [60]), as a first-pass curb ramp detector. Second, we post-process the resulting bounding boxes using non-maximum suppression [120] and 3D-point cloud data to eliminate detector redundancies and false positives. Finally, the remaining bounding boxes are classified using a Support Vector Machine (SVM) [23], which uses features not leveraged by the DPM, further eliminating false positives.

svDetect was designed and tested iteratively. We attempted multiple algorithmic approaches and used preliminary experiments to guide and refine our approach. For example, we previously used a linear SVM with a Histograms of Oriented Gradients (HOG) feature descriptor [83] but found that the DPM was able to recognize curb ramps with larger variations. In addition, we found that though the raw GSV image size is 13,312 x 6,656 pixels, there were no detection performance benefits beyond 4,096 x 2,048px (the resolution used throughout this paper). Because it helps explain our design rationale for Tohme, we include our evaluation experiments for svDetect in this section rather than later in the paper.

First Stage: The Curb Ramp Deformable Part Model (DPM). DPMs are comprised of two parts: a coarse-grained model, called a root filter, and a higher resolution parts



(a) Root filter (b) Parts filter (c) Displacement costs

Figure 5.9. The trained curb ramp DPM model. Each row represents an automatically learned viewpoint variation. The root and parts filter visualize learned weights for the gradient features. The displacement costs for parts are shown in (c).

model, called a parts filter. DPMs are commonly applied to human detection in images, which provides a useful example. For human detection, the root filter captures the whole human body while part filters are for individual body parts such as the head, hand, and legs (see [62]). The individual parts are learned automatically by the DPM—that is, they are not explicitly defined *a priori*. In addition, how these parts can be positioned around the body (the root filter) is also learned and modeled via displacement costs. This allows a DPM to recognize different configurations of the human body (*e.g.*, sitting *vs.* standing).

In our case, the root filter describes the general appearance of a curb ramp while part filters account for individual components (*e.g.*, edges of the ramp and transitions to the road). DPM creates multiple *components* for a single model (Figure 5.9) based on bounding box aspect ratios. We suspect that each component implicitly captures different viewpoints of a curb ramp. For our DPM, we used code provided by [67].

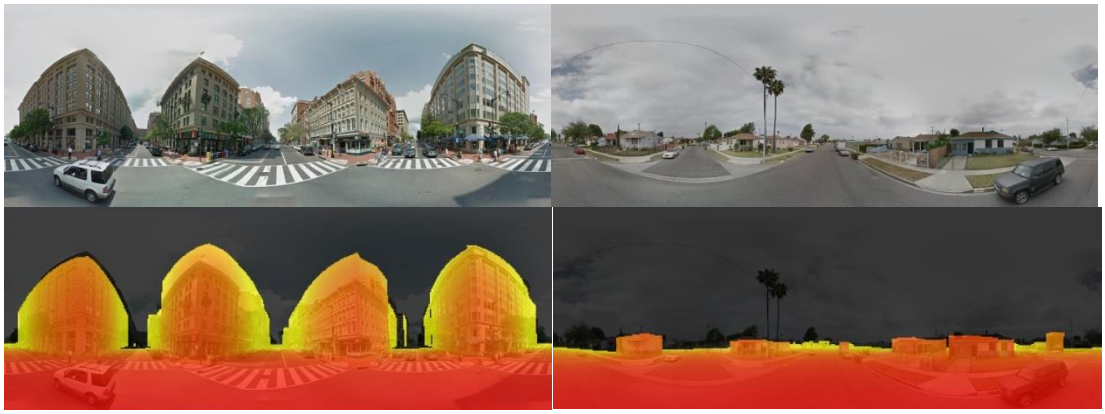


Figure 5.10. Using code from [205]**, we download GSV's 3D-point cloud data and use this to create a ground plane mask to post-process DPM output. The 3D depth data is coarse: 512 x 256px.**

Second Stage: Post-Processing DPM Output. In the second stage, we post-process the DPM output in two ways. First, similar to [120], we use non-maximum suppression (NMS) to eliminate redundant bounding boxes. NMS is common in CV and works by greedily selecting bounding boxes with high confidence values and removing overlapping boxes with lower scores. Overlap is defined as the ratio of intersection of the two bounding boxes over the union of those boxes. Based on the criteria established by the PASCAL Visual Object Classes challenge [61], we set our NMS overlap threshold to 50%.

Our second post-processing step uses the 3D-point cloud data to eliminate curb ramp detections that occur above the ground plane (*e.g.*, bounding boxes in the sky are removed). To do so, the 512 x 256px depth image is resized to the GSV image size (4096 x 2048px) using bilinear interpolation. For each pixel, we calculate a normal vector and generate a mask for those pixels with a strong vertical component. These pixels correspond to the ground plane. Bounding boxes outside of this pixel mask are eliminated (Figure 5.10 and 5.11).

Third Stage: SVM-Based Classification. Finally, in the third stage, the remaining bounding boxes are fed into an additional classifier: an SVM. Because the DPM relies solely on gradient features in an image, it does not utilize other important discriminable information such as color or position of the bounding box. Given that street intersections have highly constrained geometrical configurations, curb ramps tend to occur in similar locations—so detection position is important. Thus, for each bounding box, we create a feature vector that includes: RGB color histograms, the top-left and bottom-right corner coordinates of the bounding box in the GSV image along with its width and height, and the detection confidence score from the DPM detector. We use the SVM as a binary classifier to keep or discard detection results from the second stage.

svDetect Training and Results. Two of the three svDetect stages require training: the DPM in Stage 1 and the SVM in Stage 3. For training and testing, we used two-fold cross validation across the 1,086 GSV scenes and 2,877 ground truth curb ramp labels. The GSV scenes were randomly split in half (543 scenes per fold) with one fold initially assigned for training and the other for testing. This process was then repeated with the training and testing folds switched.

To train the DPM (Stage 1), we transform the polygonal ground truth labels into rectangular bounding boxes, which are used as positive training examples. DPM uses a sliding window approach, so the rest of the GSV scene is treated as negative examples (*i.e.,* comprised of negative windows). For each image in the training set, the DPM produces a set of bounding boxes with associated confidence scores. The number of

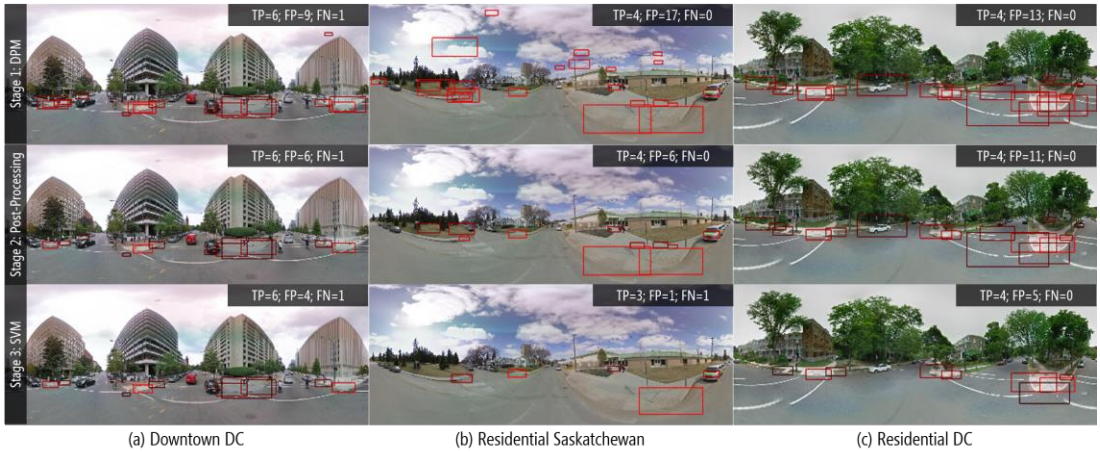


Figure 5.11. Example results from svDetect's three-stage curb ramp detection framework. Bounding boxes are colored by confidence score (lighter is higher confidence). As this figure illustrates, setting the detection threshold to -0.99 results in a relatively low false negative rate at a cost of a high false positive rate (false negatives are more expensive to correct). Many false positives are eliminated in Stages 2 and 3. The effect of Stage 2's ground plane mask is evident in (b). Acronyms: TP=true positive; FP=false positive; FN=false negative

bounding boxes produced per scene is contingent on a minimum score threshold. This threshold is often learned empirically (*e.g.*, [1]). A high threshold would produce a small number of bounding boxes, which would likely result in high precision and low recall; a low threshold would likely lead to low precision and high recall.

To train the SVM (Stage 3), we use the post-processed DPM bounding boxes from Stage 2. The bounding boxes are partitioned into positive and negative samples by calculating area overlap with the ground truth labels. Though there is no universal standard for evaluating "good area overlap" in object detection research, we use 20% overlap (from [64]). Prior work suggests that even 10-15% overlap agreement at the pixel level would be sufficient to confidently localize accessibility problems in images [81]. Thus, positive samples are boxes that overlap with ground truth by more than 20%; negative samples are all other boxes. We extract the aforementioned training features from both the positive and negative bounding boxes. Note that SVM

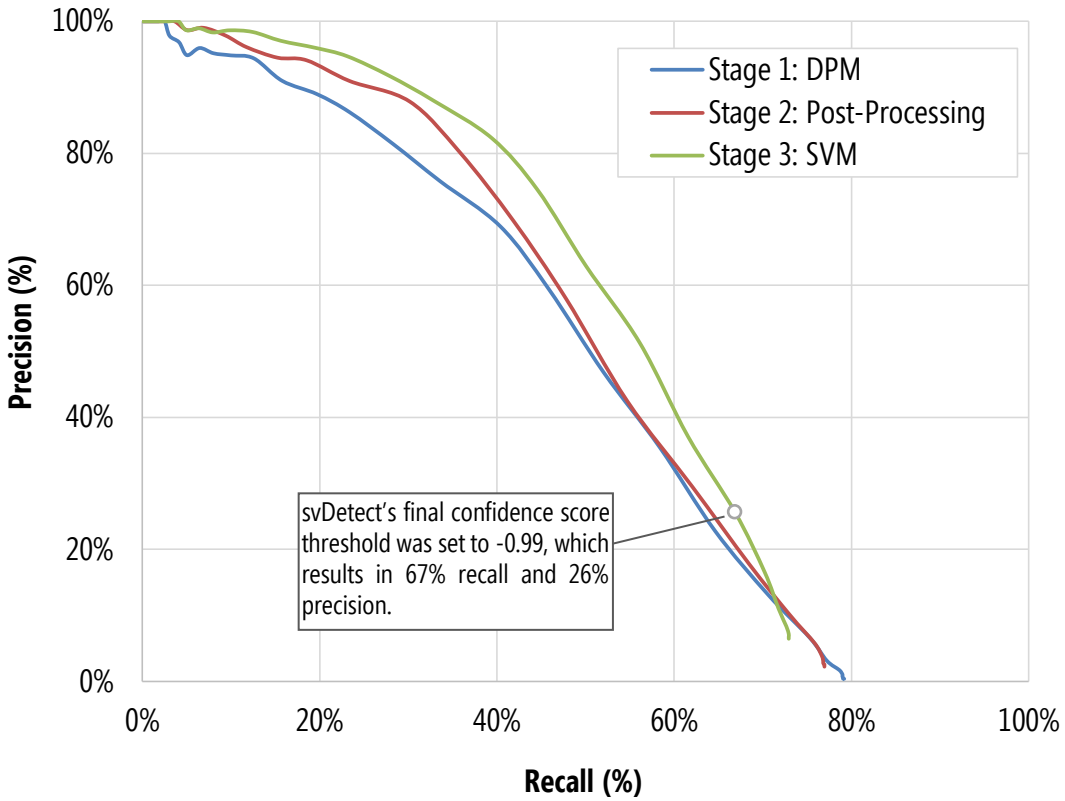


Figure 5.12. The precision-recall curve of the three-stage curb ramp detection process constructed by stepping through various DPM detection thresholds (from -3-to-3 with a 0.01 step). For the final svDetect module, we selected a DPM detection threshold of -0.99, which balances true positive detections with false positives.

parameters (*e.g.*, coefficient for slack variables) are automatically selected by grid search during training.

Results. To analyze svDetect's overall performance and to determine an appropriate confidence score cutoff for svDetect, we stepped through various DPM detection thresholds (from -3-to-3 with a 0.01 step) and measured the results. For each threshold, we calculated true positive, false positive, and false negative detections for each scene. True positives were assessed as bounding boxes that had 20% overlap with ground truth labels and that had a detection score higher than the currently set threshold. The results are graphed on a precision-recall curve in Figure 5.12. To balance the number of true

positive detections and false positives in our system, we selected a DPM detection threshold of -0.99. At this threshold, svDetect generates an average of 7.0 bounding boxes per intersection (*SD*=3.7); see Figure 5.11 for examples. Note: svDetect failed to generate a bounding box for 15 of the 1,086 intersections. These are still included in our performance comparison.

In the ideal, our three-stage detection framework would have both high precision and high recall. As can be observed in Figure 5.12, this is obviously not the case as ~20% of the curb ramps are never detected (*i.e.*, the recall metric never breaches 80%). With that said, automatically finding curb ramps using CV is a hard problem due to viewpoint variation, illumination, and within/between class variation. This is why Tohme combines automation with manual labor using svControl.

5.4.6 svControl: Scheduling Work via Performance Prediction

svControl is a machine-learning module for predicting CV performance and assigning work to either a manual labor pipeline (svLabel) or an automated pipeline with human verification (svDetect + svVerify)—see Figure 5.4. We designed svControl based on three principles: first, that human-based verifications are fast and relatively low-cost compared to human-based labeling; second, CV is fast and inexpensive but error prone both in producing high false positives and false negatives; third, false negatives are more expensive to correct than false positives.

From these principles, we derived two overarching design questions: first, given the high cost of human labeling and relative low-cost of human verification, could we

optimize CV performance with a bias towards a low false negative rate (even if it meant an increase in false positives)? Second, given that false negatives cannot be eliminated completely from svDetect, can we predict their occurrence based on features of an intersection and use this to divert work to svLabel instead for human labeling?

Towards the first question, biasing CV performance towards a certain rate of false negatives is trivial. It is simply a matter of selecting the appropriate threshold on the precision/recall curve (recall that the threshold that we selected was -0.99). The second question is more complex. We iterated over a number of prediction techniques and intersection features before settling on a linear SVM and Lasso regression model [180] with the following three types of input features:

- **svDetect results (16 features):** For each GSV image, we include the raw number of bounding boxes output from svDetect, the average, median, standard deviation, and range of confidence scores of all bounding boxes in the image, and descriptive statistics for their XY-coordinates. Importantly, we did not use the *correctness* of the bounding box as a feature since this would be unknown during testing.
- **Intersection complexity (2 features):** We calculate intersection complexity via two measures: *cardinality* (*i.e.*, how many streets are connected to the target intersection) and an *indirect measure* of complexity, for which we count the number of street pixels in a stylized top-down Google Map. We found that high pixel counts correlate to high intersection complexity (Figure 5.13).

Figure 5.13. We use top-down stylized Google Maps (bottom row) to infer intersection complexity by counting black pixels (streets) in each scene. A higher count correlates to higher complexity

	Turkers	GSV Scenes	HITs	Tasks	Avg. Turkers / Intersection	Label Stats	Avg. Task Time
svLabel	242	1,046	1,270	6,350	6.1 (0.6)	20,789 labels (17,327CRs, 3,462MCRs)	94.1s (144.4s)
svVerify	161	1,046	582	5,820	5.6 (0.6)	42,226 verified labels (28,801RLs, 13,425KLs)	43.2 (48.7s)

Table 5.3: An overview of the MTurk svLabel and svVerify HITs. While Tohme's svControl system would, in practice, split work between the svLabel and svDetect+svVerify pipelines, we fed every GSV scene to both to perform our analyses. Acronyms above include CRs=Curb Ramps; MCRs=Missing Curb Ramps; RLs=Removed Labels; KLs=Kept Labels. svVerify was 2.2x faster than svLabel.

- **3D-point cloud data (5 features)**: svDetect struggles to detect curb ramps that are distant in a scene—*e.g.*, because the intersection is large or because the GSV car is in a sub-optimal position to photograph the intersection. Thus, we include descriptive statistics of depth information of each scene (*e.g.*, average, median, variance).

We combine the above features into a single 23-dimensional feature vector for training and classification.

svControl Training and Test Results. We train and test svControl with two-fold cross validation using the same train and test data as used for svDetect. Given that the goal of svControl is to predict svDetect performance, namely the occurrence of false negatives, we define a svDetect *failure* as a GSV scene with at least one false negative curb ramp detection. The SVM model is trained to make a binary failure prediction with the aforementioned features. Similarly, the Lasso regression model is trained to predict the *raw number* of false negatives of svDetect (regression value > 0.5 is failure).

To help better understand the important features in our models, we present the top three correlation coefficients for both. For the SVM, the top coefficients were the label's x-coordinate variance (0.91), the mean confidence score of automatically detected labels (0.69), and the minimum scene depth (0.67). For the Lasso model, the top three were mean scene depth (0.69), median scene depth (-0.28), and, similar to the SVM, the mean confidence score of the automatically detected labels (0.21). If either the SVM or the Lasso model predicts failure on a particular GSV scene, svControl routes that scene to svLabel instead of svVerify.

svControl Results. We assessed svControl's prediction performance across the 1,086 scenes. While not perfect, our results show that svControl is capable of identifying svDetect failures with high probability—we correctly predicted 397 of the 439 svDetect failures (86.3%); however, this high recall comes at a cost of precision: 404 of the total 801 scenes (50.4%) marked as failures were false positives. Given that we designed svControl to be conservative (*i.e.,* pass more work to svLabel if in doubt about

svDetect), this accuracy balance is reasonable. Below, we examine whether this is sufficient to provide performance benefits for Tohme.

5.5 Study 2: Evaluating Tohme

To examine the effectiveness of Tohme for finding curb ramps in GSV images and to compare its performance to a baseline approach, we performed an online study with MTurk in spring 2014. Our goal here is threefold: first, and most importantly, to investigate whether Tohme provides performance benefits over manual labeling alone (baseline); second, to understand the effectiveness of each of Tohme's sub-systems (svLabel, svVerify, svDetect, and svControl); and third, to uncover directions for future work in preparation for a public deployment.

5.5.1 Tohme Study Method

Similar to Hara *et al.* [81], we collected more data than necessary in practice so that we could simulate performance with different workflow configurations *post hoc*. To allow us to compare Tohme *vs.* feeding all scenes to either workflow on their own (svLabel and svDetect+svVerify), we ran *all* GSV scenes through both. To avoid interaction effects, turkers hired for one workflow (labeling) could not work on the other (verifying) and vice versa.

Second, to more rigorously assess Tohme and to reduce the influence of any one turker on our results, we hired at least three turkers per scene for each workflow and used this data to perform Monte Carlo simulations. More specifically, for both

workflows, we randomly sampled one turker from each scene, calculated performance statistics (*e.g.,* precision), and repeated this process 1,000 times. Admittedly, this is a more complex evaluation than simply hiring one turker per scene and computing the results; however, the Monte Carlo simulation allows us to derive a more robust indicator of Tohme's expected future performance.

Of the 1,086 GSV scenes (street intersections) in our dataset, we reserved 40 for ground truth seeding, which were randomly selected from the eight geographic areas (5 scenes from each). We calculated HIT payment rates based on MTurk pilot studies: $0.80 for svLabel HITs (five intersections; $0.16 per intersection) and $0.80 for svVerify (ten intersections; $0.08 per intersection). As noted in our system description, turkers had to successfully complete interactive tutorials before beginning the tasks.

5.5.2 Analysis Metrics

To assess Tohme, we used the following measures:

- **Label overlap compared to ground truth:** as described in the svDetect section, we use 20% overlap as our correctness threshold (from [81]).
- **We calculate standard object detection performance metrics** including precision, recall, and F-measure based on this 20% area overlap—the same overlap used by svDetect.

- **Human time cost**: cost is calculated by measuring completion times for each intersection in svLabel and svVerify.

5.5.3 Tohme Study Results

We first present high-level descriptive statistics of the MTurk HITs before focusing on the comparison between Tohme *vs.* our baseline approach (pure manual labeling with svLabel). We provide additional analyses that help explain the underlying trends in our results.

Descriptive Statistics of MTurk Work. To gather data for our analyses, we hired 242 distinct turkers for the svLabel pipeline and 161 turkers for the svVerify pipeline (Table 5.3). As noted previously, all 1,046 GSV scenes were fed through both workflows. For svLabel, turkers completed 1,270 HITs (6,350 labeling tasks) providing 17,327 curb ramp labels and 3,462 missing curb ramp labels. For svVerify, turkers completed 582 HITs (5,820 verification tasks) and verified a total of 42,226 curb ramp labels. On average, turkers eliminated 4.9 labels per intersection (SD=2.9). We hired an average of 6.1 (SD=0.6) turkers per intersection for svLabel and 5.6 (SD=0.6) for svVerify.

Evaluating Tohme's Performance. To evaluate Tohme's overall performance, we first examined how well each pipeline would perform on its own across the entire dataset (1,046 scenes). This provides two baselines for comparison: (i) the *svDetect + svVerify* results show how well Tohme would perform if the svControl module passed *all* work to this pipeline and, similarly, (ii) the *svLabel* results show what would happen if we only relied on manual labor for finding and labeling curb ramps.

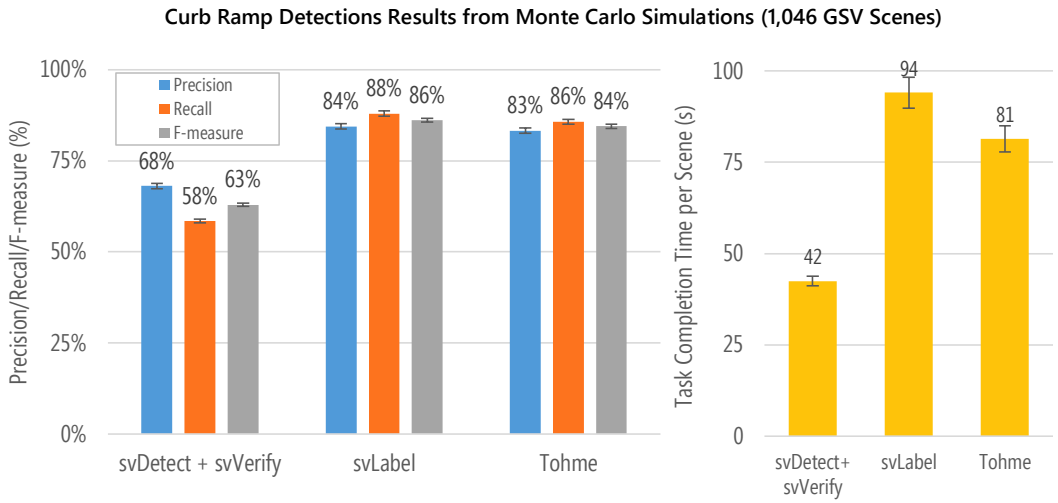


Figure 5.14: Tohme achieves comparable results to a manual labeling approach alone but with a 13% reduction in task completion time cost. Error bars are standard deviation.

We found that Tohme achieved similar but slightly lower curb ramp detection results compared to the manual approach alone (F-measure: 84% *vs.* 86%) but with a much lower time cost (13% reduction); see Figure 5.14. As expected, while the svDetect + svVerify pipeline is relatively inexpensive, it performed the worst (F-measure: 63%). These findings show that the svControl module routed work appropriately to maintain high accuracy but at a reduced cost. Tohme reduces the average per-scene processing time by 12 seconds compared to svLabel alone. The overall task completion times were 12.3, 27.3, and 23.7 hours for svDetect + svVerify, svLabel, and Tohme respectively.

The above results were calculated using the aforementioned Monte Carlo method. If we, instead, use only the *first* turker to arrive and complete the task, our results are largely the same. The F-measures are 63%, 86%, and 85% respectively for svDetect + svVerify, svLabel, and Tohme with a 10% drop in cost for Tohme (rather

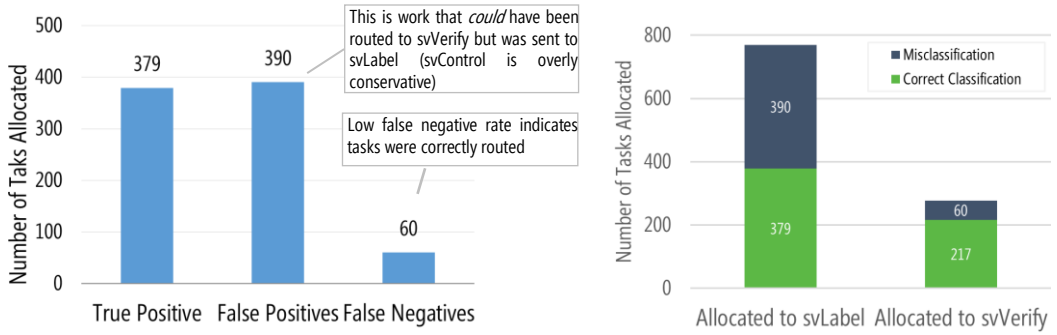


Figure 5.15: svControl allocated 769 scenes to svLabel and 277 scenes to svVerify. 379 out of 439 scenes (86.3%) where svDetect failed were allocated "correctly" to svLabel. Recall that svControl is conservative in routing work to svVerify because false negative labels are expensive to correct; thus, the 86.3% comes at a high false positive cost (390).

than 13%). This includes 65 distinct turkers for svDetect + svVerify, 97 for svLabel, and 149 for Tohme.

Task Allocation by svControl. As the workflow scheduler, the svControl module is a critical component of Tohme. Because the svVerify interface does not allow for labeling (*e.g.*, correcting false negatives), the svControl system is conservative—it routes most of the work to svLabel otherwise many curb ramps would possibly remain undetected. Of the 1,046 scenes, svControl predicted svDetect to fail on 769 scenes (these results are the same as presented in the svControl section but with the 40 ground truth scenes removed). Thus, 73.5% of all scenes were routed to svLabel for manual work and the rest (277) were fed to svVerify for human verification (Figure 5.15). Again, svControl's true positive rate is high: 86%. However, if svControl worked as a perfect classifier, 439 scenes would have been forwarded to svLabel and 607 to svVerify. In this idealized case, Tohme's cost drops to 27.7% compared to a manual labeling approach with the same F-measure as before (84%). Thus, assuming limited

improvements in CV-based curb ramp detections in the near future, a key area for future work will be improving the workflow control system.

Where Humans and Computers Struggle. The key to improving both CV and human labeling performance is to understand *where* and *why* each sub-system makes mistakes. To assess the detection accuracy of human labelers, we calculated the average F-measure score per scene based on the average number of true positives (TP), false positives (FP), and false negatives (FN). For example, if the average for a scene was (TP, FP, FN) = (1, 1, 2), then (Precision, Recall, F-measure) = (0.5, 0.3, 0.4). For CV, we simply used the F-measure score for each scene based on our svDetect results. We sorted the two F-measure lists and visually inspected the best and worst performing scenes for each. For the top and bottom 10, the average F-measure scores were 99% and 0% for CV and 100% and 25% for human labeling respectively. Common problems are summarized in Figure 5.16.

Crowd workers struggled with labeling distant curb ramps (scale) or due to placement and angle (viewpoint variation). To mitigate this, future labeling interfaces could allow the worker to "walk" around the intersection to select better viewpoints (similar to [77]); however, this will increase user-interaction complexity and labeling time. Perhaps as should be expected, crowd workers were much more adept at dealing with occlusion than CV—even if a majority of a curb ramp was occluded, a worker could infer its location and shape (*e.g.*, middle occlusion picture). CV struggled for all the reasons noted in Figure 5.16. Given the tremendous variation in curb ramp design and capture angles, a larger training set may have improved our results. Moreover,

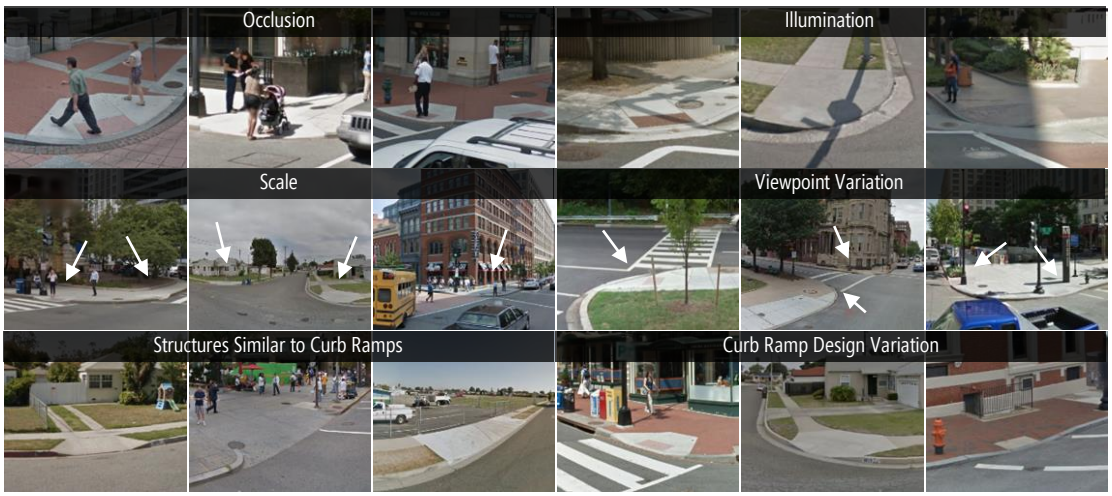


Figure 5.16: Finding curb ramps in GSV imagery can be difficult. Common problems include occlusion, illumination, scale differences because of distance, viewpoint variation (side, front, back), between class similarity, and within class variation. For between class similarity, many structures exist in the physical world that appear similar to curb ramps but are not. For within class variation, there are a wide variety of curb ramp designs that vary in appearance. White arrows are used in some images to draw attention to curb ramps. Some images contain multiple problems.

because multiple views of a single intersection are available in GSV via neighboring panoramas, these additional perspectives could be combined to potentially improve scene structure understanding and mitigate issues with occlusion, illumination, scale, and viewpoint variation. The semantic issues—*e.g.*, confusing structures similar to curb ramps—are obviously much more difficult for CV than humans. We describe other areas for improvement in the Discussion.

Effect of Area Overlap Threshold on Performance. As noted previously, there is no universal standard for selecting an area overlap threshold in CV; this decision is often domain dependent. To investigate the effect of changing the overlap threshold on performance, we measured precision, recall, and F-measure at different values from 0-50% at a step size of 10% (Figure 5.17). For *overlap=0%*, at least 1px of a detected bounding must overlap with a ground truth label to be considered correct.

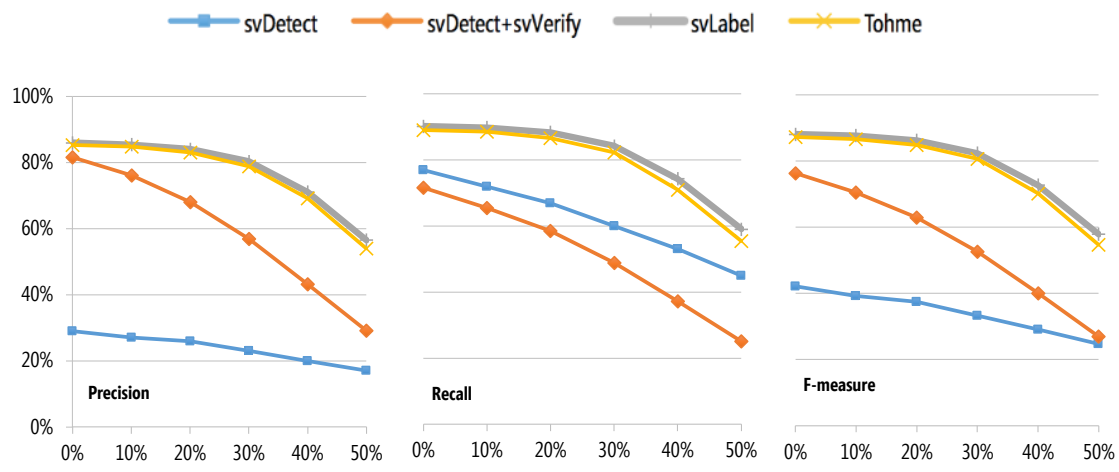


Figure 5.17: As expected, performance drops as the area overlap threshold increases; however, the relative difference between Tohme and baseline (svLabel) remains consistent.

A few observations: first, as expected, performance decreases as the overlap threshold increases; however, the relative performance difference between Tohme and baseline (svLabel) stays roughly the same. For example, at 0% overlap, the (Precision, Recall, F-measure) of Tohme is (85%, 89%, 87%) and (86%, 90%, 88%) for svLabel and at 50% overlap, (54%, 55%, 55%) *vs.* (57%, 59%, 58%). Thus, Tohme's relative performance is consistent regardless of overlap threshold (*i.e.*, slightly poorer performance but cheaper). Second, there appears to be a more substantial performance drop starting at ~30%, which suggests that obtaining curb ramp label agreement at the pixel level between human labelers and ground truth after this point is difficult. Finally, though svDetect + svVerify has much greater precision than svDetect alone, this increase comes at a cost of recall—a gap which widens as the overlap threshold becomes more aggressive. So, though human verifiers help increase precision, they are imperfect and sometimes delete true positive labels.

Figure 5.18: In the *quickVerify* interface, workers could randomly verify CV curb ramp detection patches. After providing an answer for a given detection, the patch would "explode" (bottom left) and a new one would load in its place. Though fast, verification accuracies went down in an experiment of 160 GSV scenes and 59 turkers.

5.6 Discussion

Our research advances recent work using GSV and crowdsourcing to remotely collect data on accessibility features of the physical world (*e.g.*, [73,77,80,81]) by integrating CV and a machine learning-based workflow scheduler. We showed that a trained CV-based curb ramp detector (svDetect) found 63% of curb ramps in GSV scenes and fast, human-based verifications further improved the overall results. We also demonstrated that a novel machine-learning based workflow controller, svControl, could predict CV performance and route work accordingly. Below, we discuss limitations and opportunities for future work.

5.6.1 Improving Human Interfaces

How much context is necessary for verification? We were surprised that verification tasks were only 2.2x faster than labeling tasks. Though we attempted to design both interfaces for rapid user interaction, there is some basic overhead incurred by panning and searching in the 360-degree GSV view. In an attempt to eliminate this overhead, we have designed a completely new type of verification interface, *quickVerify*, that simply presents detected bounding boxes in a grid view (Figure 5.18). Similar to the facial recognition verifier in Google Picasa, these boxes can be rapidly confirmed or rejected with a single-click and a new bounding box appears in its place. In a preliminary experiment using 160 GSV scenes and 59 distinct turkers, however, we found that accuracy with quickVerify dropped significantly. Unlike faces, we believe that curb ramps require some level of surrounding context to accurately perceive their existence. More work is needed to determine the appropriate amount of surrounding view context to balance speed and accuracy.

***Improving human labeling*.** Human labeling time could be reduced if point-and-click interactions were used for labeling targets rather than outlining; however, as demonstrated in Figure 5.16, curb ramps vary dramatically in size, scale, and shape. Clicking alone would be insufficient for CV training. Moreover, labeling will *always* be more costly than verification because it is a more difficult task (*i.e.*, finding elements in an image requires visual search and a higher mental load). With that said, we currently discard *all* svDetect bounding boxes—even those with a high confidence score—when a scene is routed to svLabel. Future work should explore how to, instead,

best utilize this CV data to improve worker performance (*e.g.*, by showing detected bounding boxes with high scores to the user or as a way to help *verify* human labels). Finally, similar to quickVerify, future work could explore GSV panorama labeling that is not projected onto a 3D-sphere but is instead flattened into a 2D zoomable interface (*e.g.*, [106]) or specially rendered to increase focus on intersection corners.

5.6.2 Improving Automated Approaches

As the first work in automatically detecting curb ramps using CV, there are no prior systems with which to directly compare our performance. Having said that, there is much room for improvement and advances in CV will only increase the overall efficacy of our system.

Improving CV-based curb ramp detection. Interesting areas of future work include: (i) *Context integration.* While we use some context information in Tohme (*e.g.*, 3D-depth data, intersection complexity inference), we are exploring methods to include broader contextual cues about buildings, traffic signal poles, crosswalks, and pedestrians as well as the precise location of corners from top-down map imagery. (ii) *3D-data integration.* Due to low-resolution and noise, we currently use 3D-point cloud data as a ground plane mask rather than as a feature to our CV algorithms. We plan to explore approaches that combine the 3D and 2D imagery to increase scene structure understanding (*e.g.*, [87]). If higher resolution depth data becomes available, this may be useful to directly detect the presence of a curb or corner, which would likely improve our results. (iii) *Training*. Our CV algorithms are currently trained using GSV scenes

from all eight city regions in our dataset. Given the variation in curb ramp appearance across geographic areas, we expect that performance could be improved if we trained and tested per city. However, in preliminary experiments, we found no difference in performance. We suspect that this is due to the decreased training set size. In the future, we would like to perform training experiments to study the effects of per-city training and to identify minimal training set size. Relatedly, we plan to explore active learning approaches where crowd labels train the system over time.

Improving the workflow controller. While our current workflow controller focuses on predicting CV performance, future systems should explore modeling and predicting human worker performance and adapting work assignments accordingly. For example, struggling workers could be fed scenes that are predicted to be easy, or hard scenes can be assigned to more than one worker to take majority vote [47,99]. Similar to CV detection, per-city training and active learning should also be explored.

Who pays? The question of who will pay for data collection (or if payment is even necessary) in the future is an important, unresolved one. Our immediate plans are to build an open website where anyone can contribute voluntarily as described in the next section. From conversations with motor impaired (MI) persons and the accessibility community as a whole (*e.g.*, non-profit organizations, families of those with MI), we believe there is a strong demand for this system. For example, with a public version of Tohme, a concerned, motivated father could easily label over 100 intersections in his neighborhood in a few hours. A website akin to walkscore.com could then visualize

the accessibility of that neighborhood using heatmaps and also calculate accessible pedestrian routes.

5.6.3 Limitations

There are two primary limitations to our work. First, there is a workload imbalance between svLabel and svDetect. svLabel gathers explicit data on both curb ramps *and* missing curb ramps while svDetect only detects the former. It is likely that if the svLabel task involved only labeling curb ramps, the labeling task completion time would go down, which would affect our primary results. And, while the *lack* of a detected curb ramp could be equated to a missing curb ramp label for svDetect, we have not yet performed this analysis. Clearly, more explorations are needed here but we believe our initial examinations are sufficient to show the potential of Tohme.

Second, there is no assessment of how our curb ramp detection results compare to traditional auditing approaches (*e.g.*, performed by city governments). Anecdotally, we have found many errors in the DC government curb ramp dataset [178]; however, more research is necessary to uncover whether our approach is faster, cheaper, and/or more accurate. Ultimately, Tohme must produce sufficiently good data to enable new types of accessibility-aware GIS applications (*e.g.*, pedestrian directions routed through an accessible sidewalk path).

5.7 Summary

This chapter introduced our preliminary work on the design and evaluation of new crowd-powered data collection methods. Completely manual methods were introduced to show the feasibility of using crowdsourcing and GSV to collect accessibility data. We have also introduced a data collection tool, Tohme, for semi-automatically detecting curb ramps in GSV images using crowdsourcing, computer vision, and machine learning. Thus far, we have shown that paid crowd workers recruited from Amazon Mechanical Turk can find and label accessibility attributes in GSV with accuracy of 81%. We have further shown that by combining crowdsourcing, CV, and ML-based smart workflow controller, we can increase data collection efficiency by 13% without sacrificing accuracy. In the next chapter, we describe the research we propose to work on next.

Chapter 6 Volunteer-sourced Accessibility Data Collection and Development of Assistive Location-based Technologies

This chapter describes our work on design and development of volunteer-based street-level accessibility data collection system.

6.1 Introduction

Despite the proliferation of location-based services and tools driven by rich geographical information (*e.g.,* car navigation [147], GIS-based urban environment modeling [138]), existing technologies have largely ignored to support people with mobility impairments [79,135]. As we explored in the formative interview study in Chapter 3, lack of capabilities to query and explore accessibility of places affect mobility impaired people's decisions to travel.

Absence of accessibility-aware location-based technologies—what we call assistive location-based technologies (ALTs)—is predominantly due to the lack of comprehensive data about the accessibility of the physical environment [132,176]. Emerging work (*e.g.,* [3,73] as well as our own work described in the previous chapters) are starting to address this issue by introducing methods to collect street-level accessibility data with paid crowdsourcing. However, even paid micro task crowdsourcing can be insufficiently scalable, and it remains expensive for creating a large dataset [97].

Because the cost is bound to the fact that we rely on paid-crowdsourcing, we investigate the feasibility of utilizing volunteer contribution to collect the street-level accessibility information. Our goal is similar to the existing *volunteered geographic information* (VGI) platforms (*e.g.,* OpenStreetMaps) that elicit contribution of many anonymous online volunteers to collect geographical information [74,75,76,140,141,142]. In this chapter, we build a VGI system by extending the crowdsourcing system described in the previous chapters, performed a pilot study with volunteers, and study how the said system is being used by the volunteers. As a preliminary study of the VGI system, we focus on studying the volunteers' activities and their labeling accuracies.

As a pilot study, we invited volunteer contribution via word-of-mouth and by emailing government organizations in Washington, D.C. from June 2016. As of August 2016, 154 volunteers contributed and we have collected 13,782 accessibility features from 2,864 street segments in the D.C. neighborhoods, which is equivalent to 20% of the entire D.C. streets.

To show the value of the street-level accessibility data that are collected with the VGI, we demonstrate two ALTs. First, we develop an online map visualization tool that shows Washington, D.C.'s street-level accessibility levels. Second, we use the accessibility data as an analytic tool to investigate relationship between neighborhoods' accessibility levels and other socio-economic characteristics (*e.g.,* income levels). The ALTs that we demonstrate are enabled by a repository of street-level accessibility data that is made publicly available via online REST APIs.

In summary, the contributions of this chapter are: (i) development and preliminary deployment of the VGI system that enable us to build a large repository of street-level accessibility data, (ii) the first of its kind neighborhood accessibility data that is made publicly available through the REST APIs, (iii) a pilot study using volunteers and an analysis of their system use and collected data, and (iv) demonstration of the utility of the collected accessibility data through embodiment of two ALTs

6.2 Study Site

To illustrate the utility of the Project Sidewalk platform, we selected Washington, D.C. as a study site (Figure 6.1). D.C. is uniquely suited for this research because of the city's economic and geographical characteristics. According to the U.S. census (2015), 672k people live in D.C. [22], many commute daily into the city from the commuter towns, and over twenty million people travel into the city every year [49]; making the capital one of the biggest city in the U.S. and important site to be accessible. The large city area (158km^2 / 61mi^2 [22]) makes collecting data less trivial, so it is a good test site to study the feasibility of our data collection method at scale. Finally, close proximity to the University of Maryland campus makes it easy for us to physically visit the neighborhood if needed.

We describe our work on mapping the accessibility of 179 D.C. neighborhoods (Figure 6.1b). Using the web tool that we describe below, we ask volunteers to explore the streets in these neighborhoods. Volunteers are instructed to label street-level

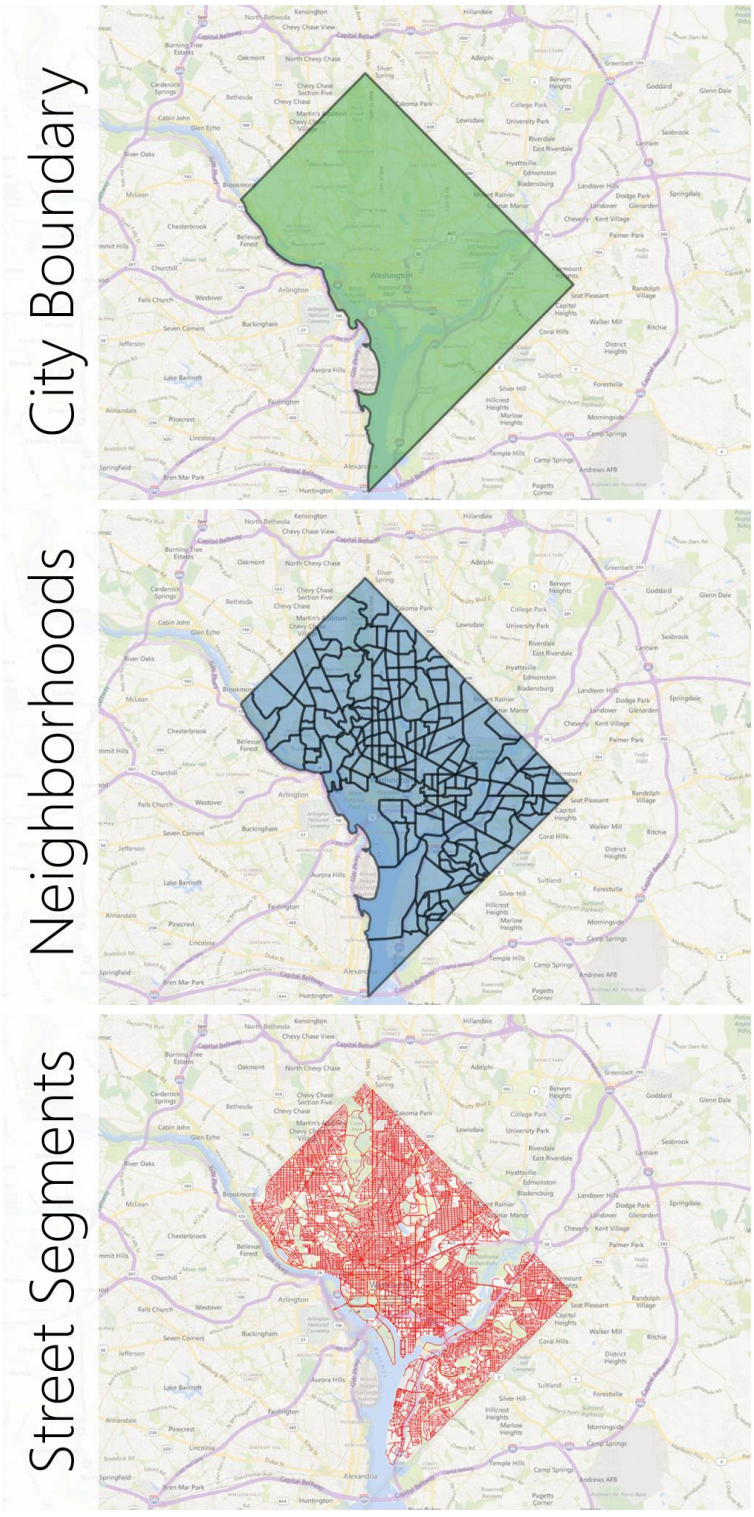


Figure 6.1. Geometry data used in this study: (a) D.C. city boundary, (b) neighborhoods, and (c) street segments.

accessibility features in Google Street View (GSV) images (a process that we call *accessibility audits*). In total, we had 15,014 street segments (Figure 6.1c) in D.C. according to the data downloaded from OpenStreetMap (we describe the process of extracting street segments from the dataset in the next section). The total street length is 1,874km (1,164mi). Because our accessibility data collection method relies on the presence of Street View images, we filter out 892 street segments where GSV images are not available (*e.g.*, streets within government facilities and hospitals), which reduced the street distance to 1,740 km (1,081mi).

6.3 VGI System for Accessibility Data Collection

Informed by our four-year iterative design experience in building GSV-based accessibility data collection tools, we designed and developed a VGI system to collect the street-level accessibility data. Volunteers were asked to explore the streets in D.C. and find and label accessibility attributes using SVLabel v.2—a web application that allows users to explore the Street View environment and find and label accessibility features in GSV images (Figure 6.2). SVLabel v.2 extends the previous version of the labeling interface [84]: (i) it allows users to label accessibility features such as obstacles, surface problems, and missing sidewalks in addition to curb ramps and missing curb ramps that were available in v.1 described in the previous chapter; (ii) it uses the geographical dataset downloaded from OpenStreetMap to provide guidance (*e.g.*, navigation message) to navigate users to walk along the streets to explore the street-level accessibility.

6.3.1 Geographical Dataset

Keeping track of which streets have been audited by volunteers requires us to have data about streets in D.C. To this end, we use OpenStreetMap street data from the area within the city boundary (Figure 6.1a). We extract `<way>` elements with `trunk`, `primary`, `secondary`, `tertiary`, or `residential` tag that represent topology of major streets [136], as well as accompanying `<node>` and `<nd>` elements that contained geographical information (*i.e.*, latitude-longitude coordinates). Because a `<way>` element often represents a street ranging multiple city blocks, we split it into multiple segments at each intersection. As a result, we had 15,014 street segments (Figure 6.1c). The total street length was 1,874km (1,164mi). The distance was computed after projecting the coordinates into EPSG:26918—a geographical coordinate system for the eastern U.S. region [150]. Because our accessibility data collection relies on Street View images, we filtered out 892 street segments where GSV were not available (*e.g.*, streets within government facilities and hospitals). This made the total street distance 1,740 km (1,081mi).

Having street segment data not only allows us to keep track of which street segments have been audited, but also aids us to manage distributed micro human work. To partition a large task of auditing the entire D.C. into smaller discrete subtasks, we use neighborhood polygonal data and street segment data (Figure 6.1b and c). The city area was broken down into 179 neighborhoods based on the 2010 Washington, D.C. census tracts [54]. We describe how we use this data to manage micro tasks in more details in Section 6.3.3.

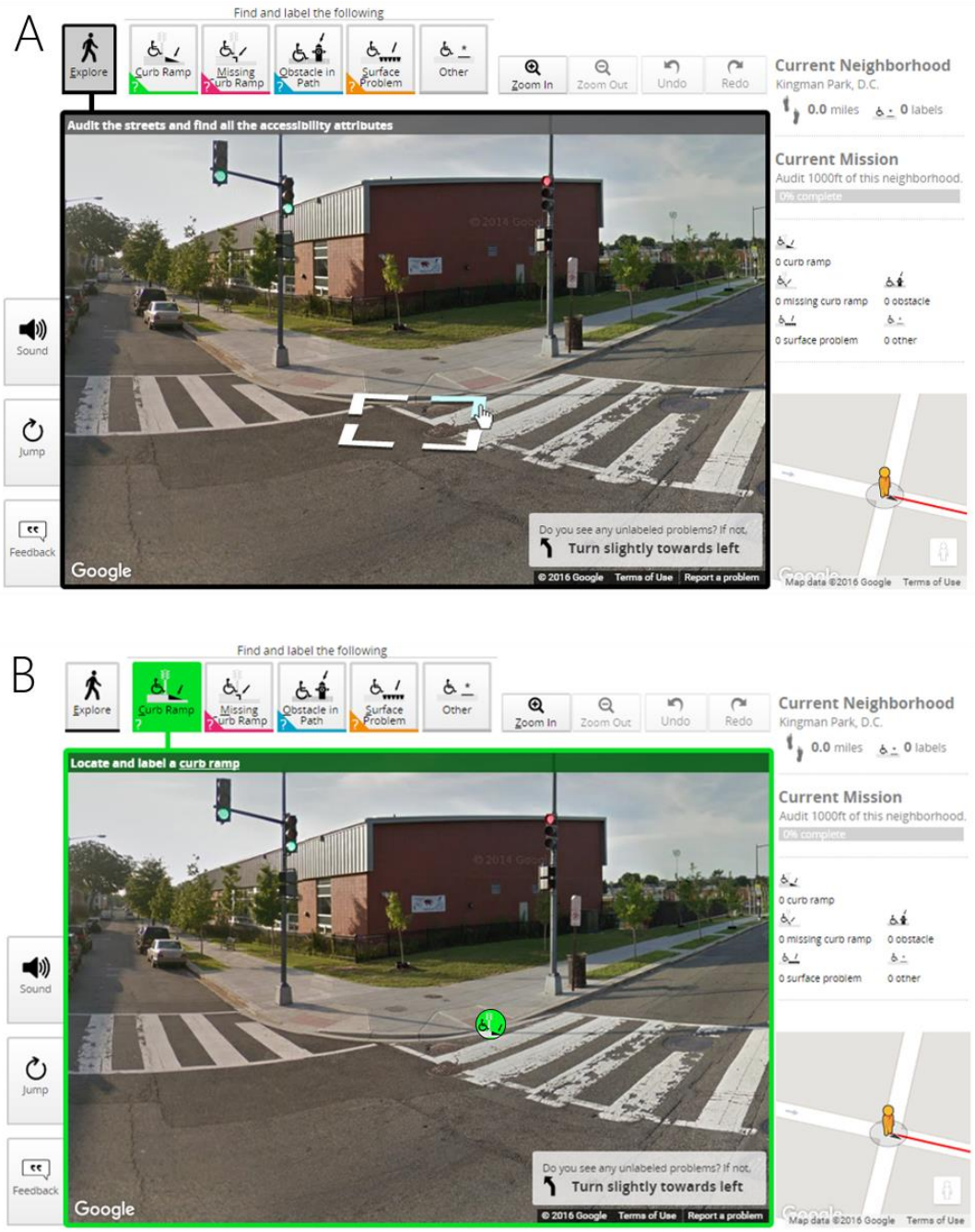


Figure 6.2. SVLabel v.2 has two modes. (a) Users can use the Explorer Mode to pan around to explore the location and click white arrows to move to the adjacent Street View locations. (b) Switching to the Labeling Mode allows them to label curb ramps, missing curb ramps, obstacles, surface problems, and other accessibility features.

The neighborhood polygons and street segments were stored in PostgreSQL database with a PostGIS spatial database extension. Both neighborhood polygonal data and street segment data were transformed into Polygons and LineStrings in Well-Known Binary (WKB) geometry format in an EPSG:4326 coordinate system—a commonly used format for storing GIS data.

6.3.2 Exploration and Labeling in SVLabel v.2

SVLabel v.2 is an interactive browser-based application for finding and labeling street-level accessibility features (Figure 6.2). SVLabel v.2 builds on the previous version of the image labeling tool described in Chapter 5. The tool provides interactive 360 degree views of the Street View panoramic image. The interface lets the users to label accessibility feature of the following types: Curb Ramp, Missing Curb Ramps, Obstacle, Surface Problems, and Other. Under the Other category, there are sub-categories Can't See Sidewalk, No Sidewalk, and Other, where users can describe the type of accessibility feature.

Similar to the prior version of the tool, SVLabel v.2 has two primary modes of interaction: *Explorer Mode* and *Labeling Mode* (Figure 6.2 a&b). When the user first drops into a scene, s/he defaults into Explorer Mode, which allows for exploration using Street View's native control. Users are instructed to pan around to observe the street-level environment and are navigated to walk along streets. Users could either double click the Street View images, click white arrows (Figure 6.2a), or hit arrow keys to move to desired directions.

When the user clicks on one of the accessibility feature buttons in the ribbon menu, the interface switches automatically to Labeling Mode. Here, mouse interactions no longer control the camera view or allow the user to walk to another Street View location. Instead, the cursor changes to a stamp that indicates the selected accessibility feature type, allowing the users to drop the stamp on the visual target (Figure 6.2b). Unlike the previous version where users were asked to outline the area, the interface instructed the users to simply click on the visual target. The labeling interaction was simplified to optimize for the speed; while the decision has a drawback that we cannot collect granular data to train computer vision-based accessibility feature detection algorithms, interaction is faster (about 23% faster for labeling [80]) and the collected data is sufficient to identify the geolocations of accessibility features.

Once the user labels an accessibility feature on a Street View image, a context menu pops up and prompts the user to provide optional fine-grained properties including severity rating, description, and checkbox to indicate a temporary problem (Figure 6.3). The collected labels are submitted to our server periodically as well as upon browser unload (*e.g.,* closing the browser tab).

Once the user labels the target on the Street View image, the labeled accessibility feature is also visualized on Google Maps pane (Figure 6.4). Our labeling application uses three types of data to estimate the geolocation of the labeled accessibility features: the labeled stamp's XY-coordinate on the Street View image, the Street View's 3D point cloud data (Figure 6.5), and the geographical coordinate of the Street View camera. To project the label on the Street View image to a point on the

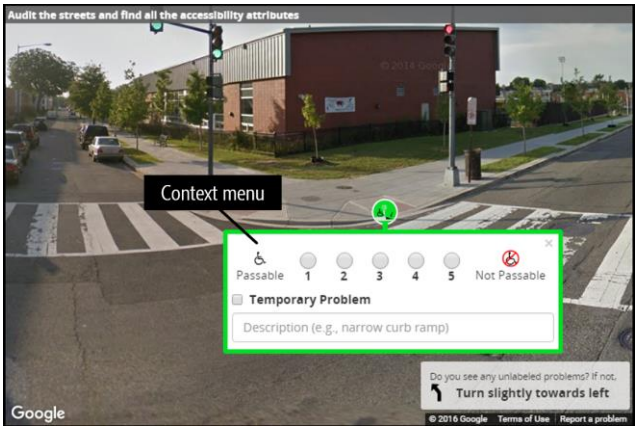


Figure 6.3. A context menu prompts the user to provide additional information for the labeled feature, including its quality/severity, temporariness, and description

Figure 6.4. The feature labeled on the image is projected to geographical coordinate and visualized on the map

map, our application takes the following steps: (a) find the label's XY-coordinate on the GSV image (Figure 6.6a); (b) find the corresponding point on the 3D point cloud data and extracts the point's displacement from the Street View camera center (Figure 6.6b); and (c) compute the label's latitude-longitude coordinate from the Street View camera's geographical coordinate and the displacement information (Figure 6.6c). Note, the 3D point cloud data is interpolated with bilinear interpolation because the data is 676x coarser compared to the Street View image. For example, GPS positioning alone could cause 8m of error [214].

Interactive Tutorial. Because navigating the Street View environment and labeling accessibility features are complex user interactions, a volunteer who uses the interface for the first time is greeted by an interactive tutorial (Figure 6.7 a-d). The a step-by-step tutorial was designed to teach how to (a) select label types, (b) label accessibility features on Street View images, (c) pan-and-zoom to look around and find accessibility features, and (d) move from one Street View image to another. The design of the

Figure 6.5. Our JavaScript application downloads Google Street View's 3D-point cloud data and use this compute the geographical coordinates of the labeled accessibility features.

tutorial was informed by our experiences building tutorials in prior work [77,84]. Once the tutorial was completed, we automatically positioned the volunteer in one of the audit area and interface initiated the auditing task.

6.3.3 Guiding Volunteers in the Accessibility Audit Task

Allowing volunteers to walk around in the Street View environment is the major change from the previous version of the SVLabel [84]. While this change lets users to observe sidewalk accessibility features from multiple angles and allow us to delegate complexity of selecting which Street View locations to audit, it adds an additional complexity to user interaction. Therefore, the application needs to break down the task into smaller, more consumable subtasks and provide guidance in order to support volunteers to complete audit tasks [33,104].

Mission and Progress Feedback. Providing people with a clear goal and providing feedback upon task completion can provides increase performance and offers a more playful, enjoyable experience [116,118]. We introduce *missions* in which volunteers are asked to audit predefined distance in a neighborhood (Figure 6.8a). Missions include auditing 1000ft, 2000ft, 4000ft, and every half a mile of the streets in each neighborhood. Upon completing each mission, the interface provides the

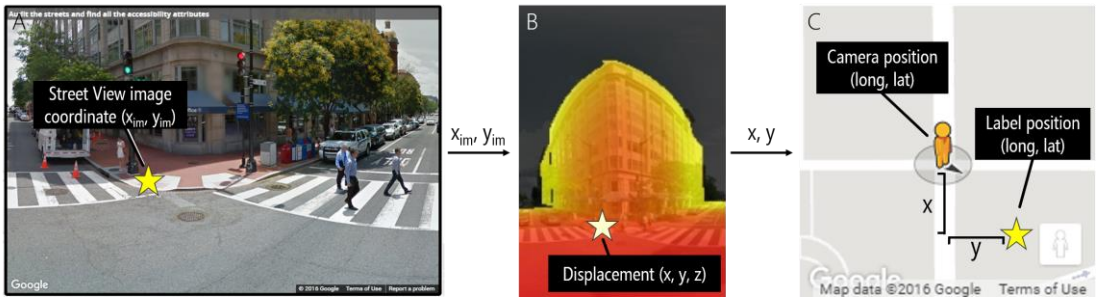


Figure 6.6. Computing a label's geographical coordinate. (a) Find the label's image coordinate on the Street View image (x_{im}, y_{im}). (b) Find the corresponding point on the 3D point cloud data and extract the displacement of the label point from the Street View camera center (x, y, z). (c) Compute the label's latitude-longitude coordinate from the Street View camera's latitude-longitude coordinate and the label's displacement (x, y).

summary of the mission contribution, which visualizes the audited streets, numbers of accessibility features collected, and distance audited (Figure 6.8b).

Neighborhoods. When a volunteer participates for the first time, our application selects one of the 179 D.C. neighborhoods in a round-robin fashion and assign it to the volunteer. Because we expected some volunteers would prefer to audit specific neighborhoods, they could select which neighborhoods in D.C. they want to audit (*e.g.*, volunteers who lived in one neighborhood wanted to audit the accessibility of their neighborhood).

Audit Routes. Street segments intersecting the assigned neighborhood are used to determine a route to audit. The volunteers are instructed to follow the routes and audit the accessibility of the streets along the way. The audit route is computed greedily from a collection of the street segments; street segments that have not been audited by the user were selected to form the route. Occasionally, the generated route guided volunteers to walk outside of the neighborhood or lead to dead-end where street segments are not available. In those cases, volunteers were jumped back into the assigned neighborhood.

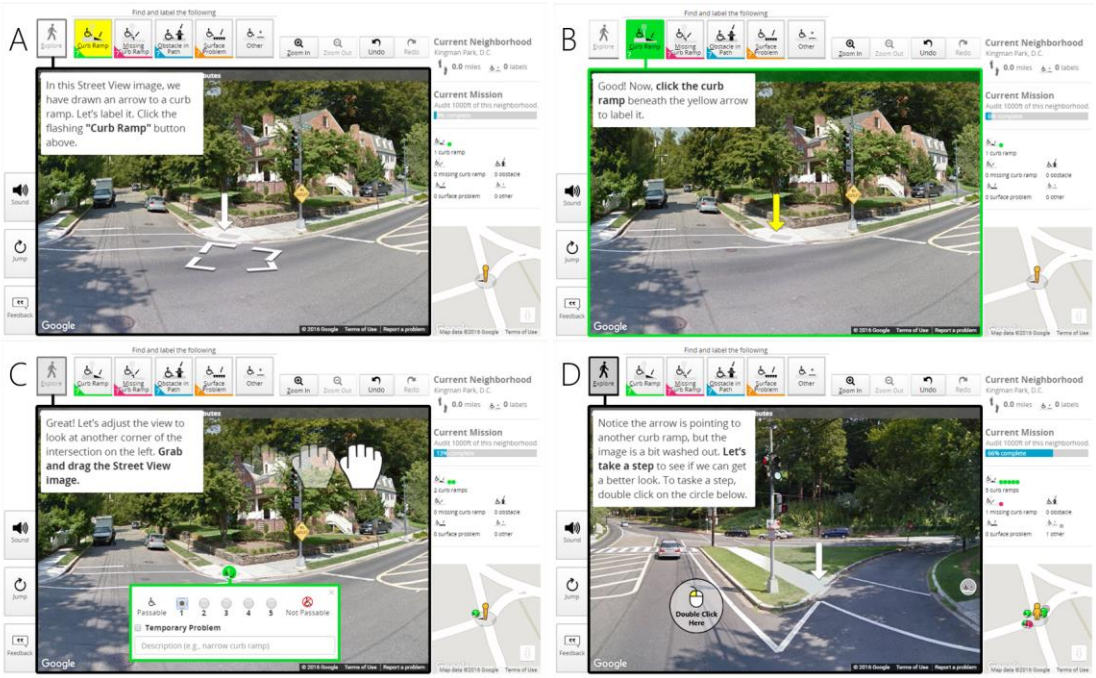


Figure 6.7. The interactive onboarding tutorial. The tutorial progressively teaches volunteers (a) to select accessibility feature types from the menu, (b) click on the Street View images to label accessibility features, (c) drag the Street View to look around the environment, and (d) double click on the Street View to move to different locations.

To present the audit route to the volunteers, the application visualizes the audit route on the Google Maps pane and the *compass* describes which direction to walk to (Figure 6.9). On the Google Maps pane, the unaudited street segments are visualized as red paths and the audited paths are colored green. A message such as "Walk straight" and "Turn right" were showed in the compass to instruct the user which way to walk.

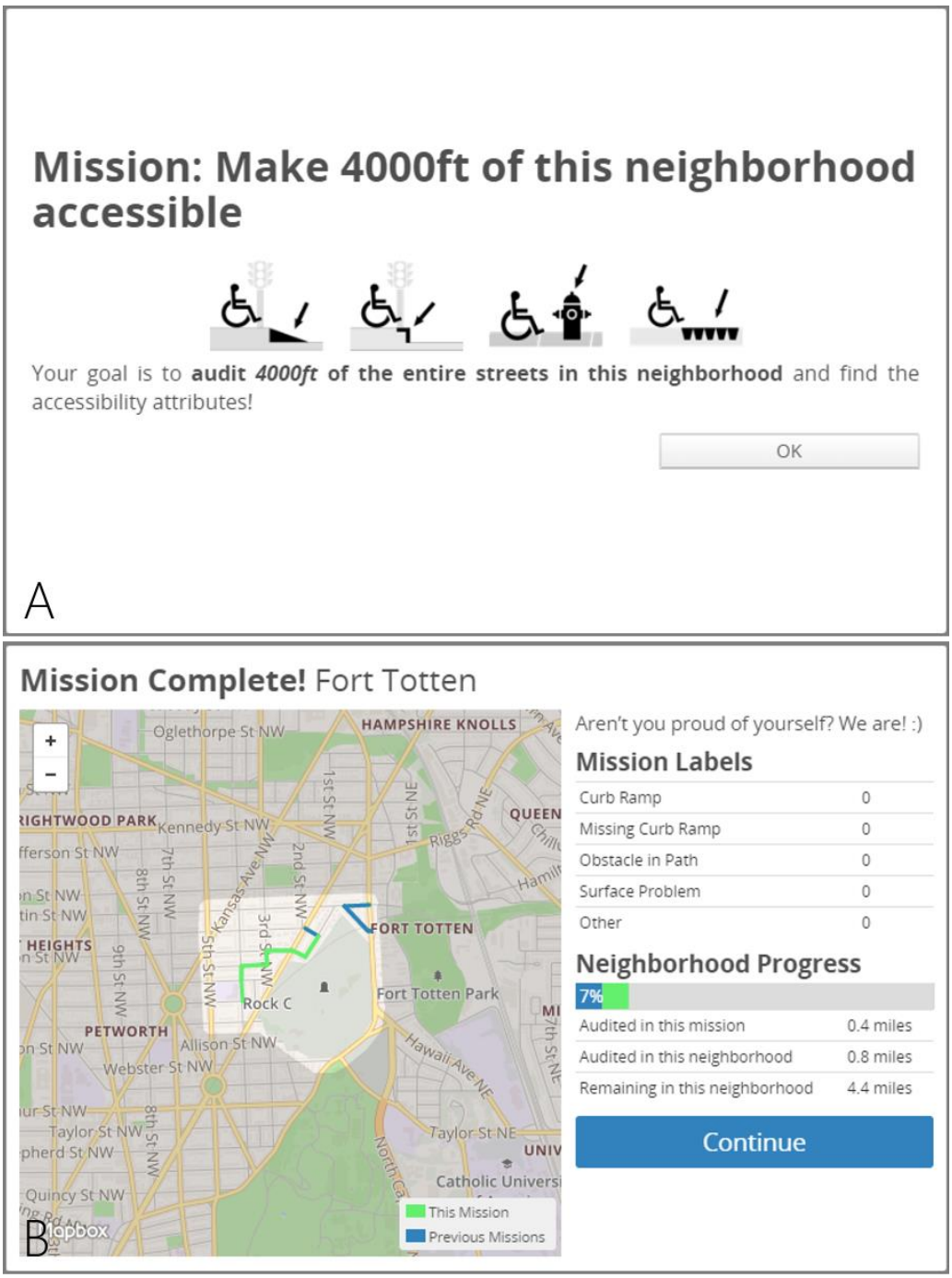


Figure 6.8. The mission information. (a) The interface presents users the *mission* which describes the immediate objective. (b) Upon mission completion, the interface presents the summary of the accessibility audit tasks completed during the mission.

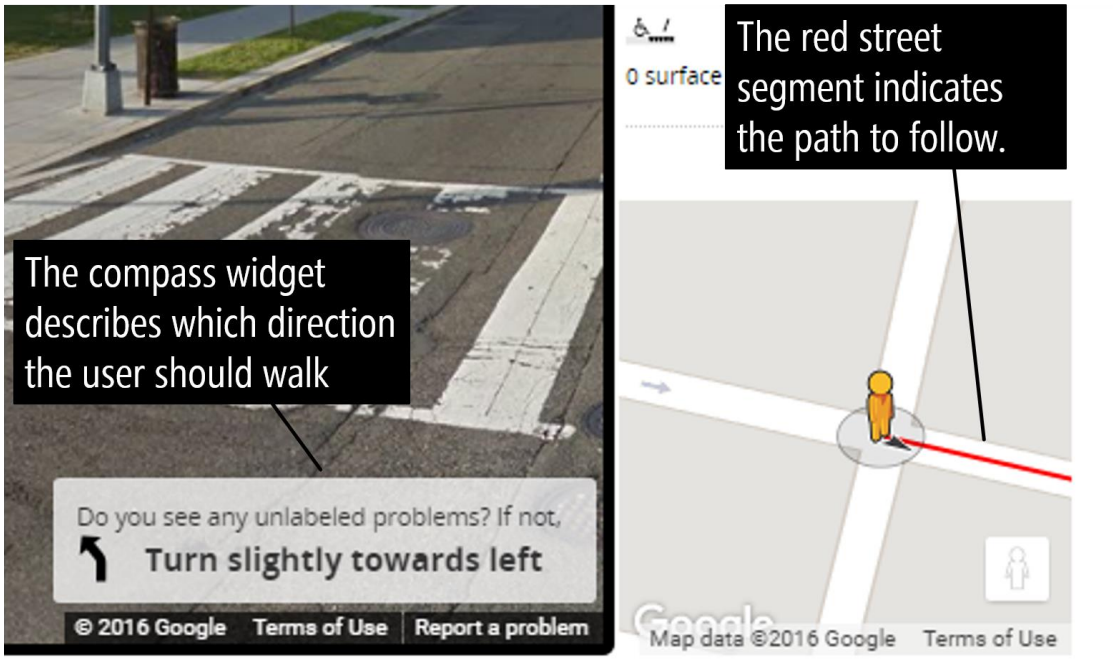


Figure 6.9. User guidance. The SVLabel interface navigates the user along the computed route with a compass which shows a directional icon and a description of which way to walk (left) to and path visualization on the Google Maps pane (right).

6.4 Evaluation of Volunteered Geographical Information

We deployed our system and invited volunteer contributors to help us collect accessibility data in Washington, D.C. (*sidewalk.umiacs.umd.edu*). As a preliminary work, we performed a soft-rollout with small number of volunteers. To invite volunteers to contribute to the data collection, we advertised the platform via word of mouth as well as contacted D.C. government organizations. We also reached out to undergraduate students at the University of Maryland to participate in the study for extra credit assignments.

6.4.1 Volunteer Participation

As of July 24th 2016, 154 volunteers participated and covered 20% of the street segments in Washington, D.C. 56 were anonymous volunteers with distinct IP addresses and 98 were registered volunteers. Of the 98 registered volunteers, 56 were undergraduate students who completed at least 2mi of accessibility audit. These students were compensated with extra credits for their courses.

In total, 2,864 street segments, which is worth 346 km (215 miles) of streets, were audited by at least one volunteer and 13,782 accessibility labels were collected (10,298 curb ramps, 1,199 missing curb ramps, 549 obstacles, 455 surface problems, 1,221 No Sidewalk, 40 Occlusion, and 20 Other types). Of 179 neighborhoods, 1 of them were 100% covered. On average, 23.5% of streets in each neighborhood were covered (the figure is slightly higher than the overall average due to the streets shared by multiple neighborhoods). Of 2,864 street segments, 2,734 were audited by registered volunteers. Each registered volunteer audited 31.8 street segments on average (*SD*=20.4, *max*=95, *min*=1).

6.4.2 Accessibility Data Accuracy

To assess the accuracy of the collected accessibility data, we randomly selected 100 street segments that were audited by volunteers. Fifty four distinct registered volunteers provided the accessibility data for these street segments, The accessibility features labeled in these streets are compared against the researcher-generated ground truth labels. To create the ground truth labels, one researcher labeled each of the 100 street

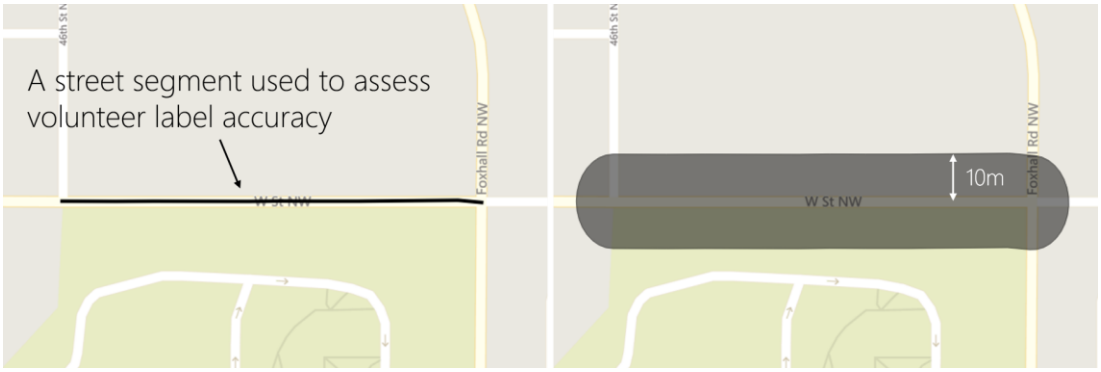


Figure 6.10. A street segment (left) and segment buffer (right). For each street segment used in the accuracy assessment, we created a 10m buffer polygon and checked presence accessibility features in this buffer.

segments. Because volunteer labels and ground truth labels could be labeled from different Street View images, we first projected all the accessibility features labeled on GSV images to latitude-longitude coordinates using Street View's 3D point cloud data (Figure 6.5 and Figure 6.6).

Based on the ground truth label, 97 streets (out of 100 streets) had curb ramps, 33 streets had missing curb ramps, 32 streets had obstacles, 27 had surface problems, and 35 had no sidewalks. On the other hand, volunteers identified 95 streets with curb ramps, 59 streets with missing curb ramps, 45 streets with obstacles, 36 streets with surface problems, and 26 no sidewalks. Volunteers also reported occlusions in 6 streets and 3 user provided accessibility feature types were submitted, but we omitted them from analysis because of their small numbers.

Following the image-level evaluation in Chapter 4, we calculate the accuracy, precision, and recall based on the presence and absence of the accessibility features. Instead of assessing the presence or absence of labels within curated static images, we

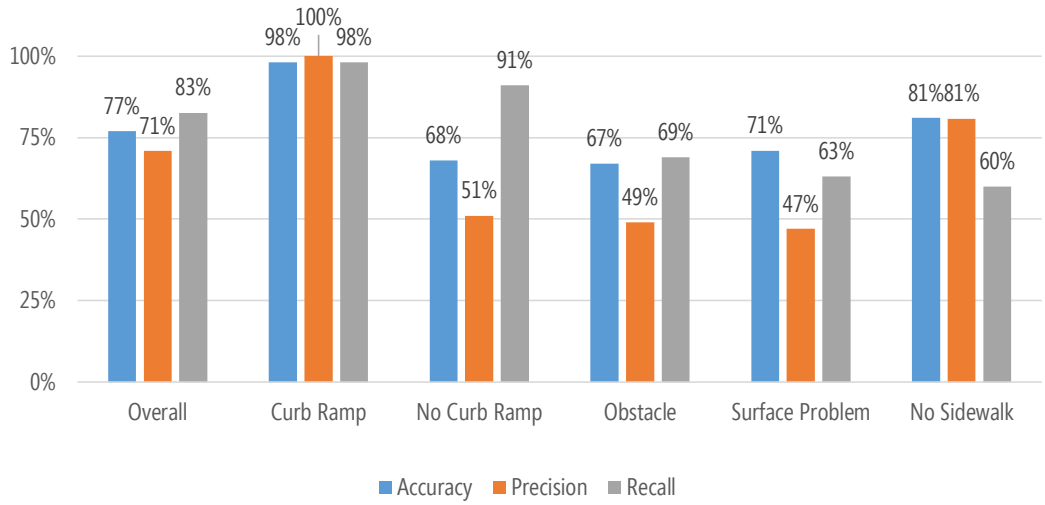


Figure 6.11. Accessibility audit accuracy. Overall accuracy was 77% when compared to researcher labels. Volunteers accurately labeled curb ramps, but label accuracy for other label types were lower. For the most of the accessibility problems, recall were higher than precision, indicating the over labeling characteristics of volunteer labels.

assessed the presence and absence of accessibility labels in a street segment buffer (Figure 6.10).

We found that the overall accuracy of the volunteer labels was 76% when compared to the ground truth label (Figure 6.11). Accuracy for each label type was (Curb Ramp, No Curb Ramp, Obstacle, Surface Problem, No Sidewalk) = (98%, 68%, 67%, 71%, 81%). The most dominant curb ramp labels were labeled accurately. We also computed precision and recall for overall and per label type—see Figure 6.11. Although we cannot directly compare, the results of the image-level evaluations Chapter 4, we look into the results to contrast the results. There, we had image-level accuracies of (No Curb Ramp, Obstacle, Surface Problem, No Sidewalk) = (79%, 73%, 85%, 85%). Note that we did not measure the curb ramp accuracy in the evaluation in Chapter 4, and we used a label type Prematurely Ending Sidewalk instead of No

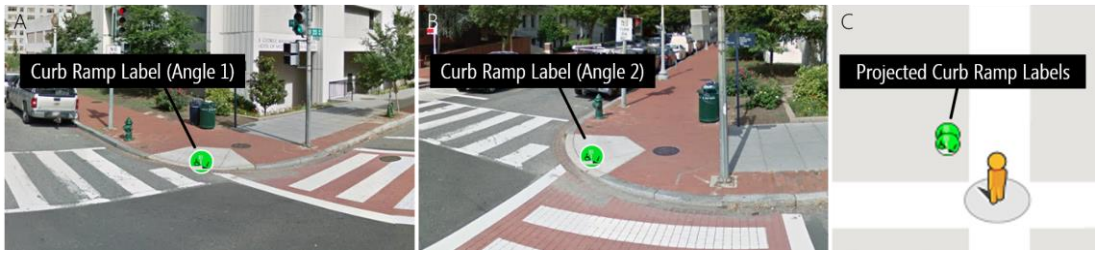


Figure 6.12. A curb ramp labeled from multiple angles. (a&b) A single curb ramp was labeled in two consecutive GSV images. (c) The two labels are projected to latitude-longitude coordinates and plotted on Google Maps as two distinct curb ramps, so they need to be clustered together to avoid double counting.

Sidewalk. The accuracies were consistently higher in the previous study. We believe this is because we used the curated static images for the labeling tasks in Chapter 4 whereas volunteers were asked to explore, find, and label accessibility features in this study, which is arguably more difficult. We discuss the results in more detail in the Discussion section.

6.5 Accessibility Data Repository

The collected street-level accessibility data is processed and served to client applications as either a set of accessibility features or Access Scores—abstract scores that represent the accessibility levels of given regions. In this section, we describe how we process the collected accessibility data, methods for computing Access Scores, and the designs of APIs that are used to serve the data to client applications.

6.5.1 Accessibility Data Processing

The accessibility features that are labeled by volunteers are processed to be served to client applications. First, because volunteers could label accessibility features in different Street View locations, multiple labels that are projected to latitude-longitudes

could represent a single sidewalk accessibility feature (Figure 6.12). To remove duplicate labels, we use their latitude-longitude coordinates to cluster and merge them into a single label. Two or more labels that are apart by less than a given threshold are clustered together. The threshold distance is set to 10m to take into account of GPS errors (see [214]).

6.5.2 Access Score

We design *Access Scores*, abstract quantitative measures of the built environment accessibility levels. Access Scores are computed with the processed street-level accessibility data. As this is the first work that uses a large geographical data to quantify accessibility, we introduce two simple computational methods to quantify *per-street* and *per-neighborhood* Access Scores—*Access Score: Street* and *Access Score: Neighborhood.* Both scores have ranges between [0, 1], where 0 represents inaccessible and 1 represents accessible.

Access Score: Street (AS_{street}) models the accessibility level of a given street. Inaccessible streets with many accessibility problems should be scored low, and vice versa. To reflect this heuristics, we count the number of accessibility features along the streets. A buffer with a 10m (32ft) radius is created around a given street segment (Figure 6.10), and the accessibility features within this buffer are counted to construct the *accessibility feature vector (*$\boldsymbol{x_a}$*)*. For example, if there are 6 curb ramps, 5 missing curb ramps, 0 obstacle, and 1 surface problem, then $\boldsymbol{x_a} = (6, 5, 0, 1)$**.** We then take a dot product of the accessibility feature vector and a user-provided *significance vector*

$(\boldsymbol{w_s})$, a vector that represents the importance of each accessibility feature type. Each element of the significance vector has a value between 0 and 1, and its polarity (+/-) depends on whether it is a positive or negative accessibility feature (*i.e.*, a curb ramp is a positive feature and all the other accessibility problems are negative features). Because the range of the dot product could be anywhere between $(-\infty, \infty)$, we map it to $(0, 1)$ using a sigmoid function. To be concrete, AS_{street} of a given street is computed by:

$$AS_{street} = 1/1 + \exp(-w_s \cdot x_a) \qquad \text{(Eq. 6.1)}$$

For example, with a significance vector (Curb Ramp, No Curb Ramp, Obstacle, Surface Problem) = (1.0, -1.0, -1.0, -1.0) and accessibility feature vector (6, 5, 0, 1), the resulting Access Score is 0.27. This reflects the fact that the street is less accessible due to the multiple missing curb ramps and a surface problem.

An accessibility level of a neighborhood should take into account of accessibility levels of all the streets within the area. To this end, we compute Access Score of a given neighborhood ($AS_{neighborhood}$) by taking the mean AS_{street} of all the streets intersecting the given neighborhood polygon. It can be written as:

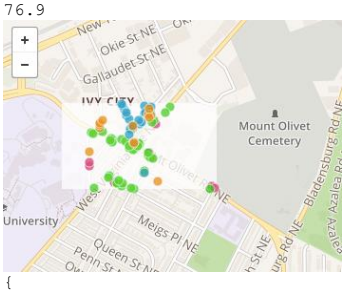
$$AS_{neighborhood} = \frac{1}{n}\sum AS_{street} \qquad \text{(Eq. 6.2)}$$

Here, n represents a number of street segments intersecting the given neighborhood. In the next section, we describe the APIs that serve the Access Score data to client applications.

6.5.3 API

(a) Access Features

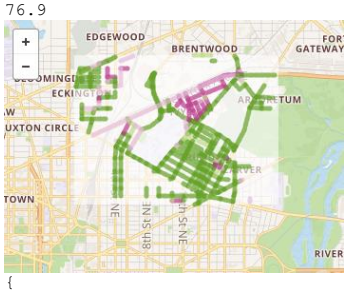
GET:/v1/access/features?lat1=38.9&lng1=-76.91&lat2=38.91&lng2=-76.9




```
{
 type: "FeatureCollection",
 features: [
  {
   type: "Feature",
   geometry: {
    type: "Point",
    coordinates: [
     [-76.9851, 38.9116]
    ],
    properties: {
    label_type: "NoCurbRamp",
    panorama_id:
     "uoEs-ZwoiNQyx9tcZlgOiQ"
   }
  },
  …
 ]
}
```


(b) Access Score: Streets

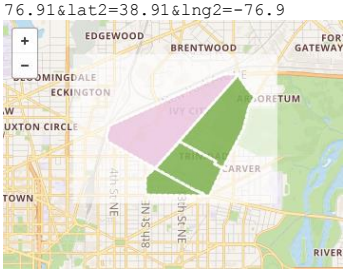
GET:/v1/access/streets?lat1=38.9&lng1=-76.91&lat2=38.91&lng2=-76.9




```
{
 type: "FeatureCollection",
 features: [
  {
   type: "Feature",
   geometry: {
    type: "LineString",
    coordinates: [
     [-76.981969, 38.9007234],
     [-76.9823201,38.9006129],
     …
    ]
   },
   properties: {
    street_edge_id: 93,
    score: 0.9820,
    significance: {
     CurbRamp: 1,
     NoCurbRamp: -1,
     Obstacle: -1,
     SurfaceProblem: -1
    },
    feature: {
     CurbRamp: 6,
     NoCurbRamp: 2,
     Obstacle: 0,
     SurfaceProblem: 0
    }
   },
   …
  ]
}
```


(c) Access Score: Neighborhoods

GET:/v1/access/neighborhoods?lat1=38.9&lng1=-76.91&lat2=38.91&lng2=-76.9




```
{
 type: "FeatureCollection",
 features: [
 {
  type: "Feature",
  geometry: {
   type: "Polygon",
   coordinates: [
    [
     [-76.9808, 38.9040],
     [-76.9809, 38.9040],
     …
    ]
   ],
  },
  properties: {
   region_id: 273,
   region_name: "Trinidad",
   score: 0.9927,
   significance: {
    CurbRamp: 1,
    NoCurbRamp: -1,
    Obstacle: -1,
    SurfaceProblem: -1
   },
   feature: {
    CurbRamp: 7.8658,
    NoCurbRamp: 1.8780,
    Obstacle: 0.4268,
    SurfaceProblem: 0.6463
   }
  },
  …
 ]
}
```


Table 6.1. REST APIs to serve accessibility information. (a) *Access Features API* serves location data of accessibility features with their accessibility feature type. (b) *Access Score: Streets API* serves a set of street segments with corresponding Access Scores.. (c) *Access Score: Neighborhoods API* serves a set of neighborhood polygons with corresponding Access Scores.

Clients access the repository of the collected accessibility data through a RESTful API, loading the appropriate endpoint URL and receiving GeoJSON data in return [24]. To serve accessibility data of varying geographical precision and data granularity, the repository provides three API endpoints serving data of varying geographical precision and data granularity: (i) Access Features, (ii) Access Score: Streets, and (iii) Access

Score: Neighborhoods. These APIs enable varying assistive location-based technologies. All APIs require a bounding box defined by a pair of latitude-longitude coordinates (*i.e.*, (minlat, minlng), (maxlat, maxlng)) as a parameter to specify the region of interest. In addition, Access Score: Streets and Access Score: Neighborhoods APIs take optional significance vector as a parameter.

i. **Access Features API** serves a set of geographical coordinates that represent *where* and *what* accessibility features exist (Table 6.1a). The data served by this API enables a client application to present what makes a region of interest accessible or inaccessible. The data is represented as a Feature Collection of Points in the GeoJSON format.

ii. **Access Score: Streets API** serves street segments enclosed in a given bounding box together with AS_{street} computed using the Eq. 6.1. Because significance of each accessibility feature varies between people with different mobility levels, the API allows users to specify each feature's significance in a scale of [0, 1]. The API returns the data as a Feature Collection of LineStrings in the GeoJSON format (Table 6.1b).

iii. **Access Score: Neighborhoods API** provides neighborhood polygons enclosed in a given bounding box with corresponding $AS_{neighborhood}$ computed by Eq. 6.2. Similar to Access Score: Streets API, users can specify each feature's significance in a scale of [0, 1]. The data is represented as a Feature Collection of Polygons in the GeoJSON format (Table 6.1c).

6.6 Assistive Location-based Technologies

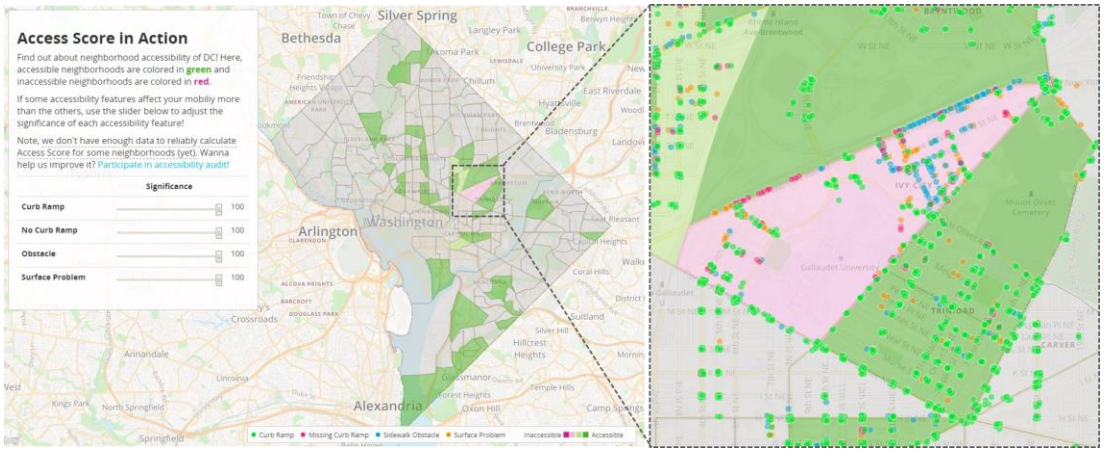


Figure 6.13. Access Map. The choropleth map visualizes accessibility levels of the D.C. neighborhoods using the data from Access Score: Neighborhoods API. The neighborhoods are colored in green they are accessible and red if they are inaccessible. When a user zoom in, accessibility feature points from Access Feature API are visualized. The neighborhoods with audit coverage < 50% are colored in gray to show that we do not have sufficient data to compute $AS_{neighborhoods}$.

The assistive location-based technologies enabled by the collected street-level accessibility data could be used by people with mobility impairments and other interested users such as policy makers who are responsible for compliance with ADA. In this section, we present two proof-of-concept ALTs to demonstrate the value of the collected accessibility data.

6.6.1 Access Map

Access Score: Neighborhoods API and Access Features API enable an *Access Map*, a choropleth map that allows mobility impaired people to quickly explore accessibility of different parts of D.C. (Figure 6.13). The geographical visualization could be useful to find a neighborhood that is easy to live/stay and locate cafes and stores in accessible neighborhoods. The GeoJSON data served by the APIs is visualized with our proof-of-

concept web application, thought data can also be visualized with existing GIS tools (*e.g.*, QGIS [227]). The application reads in the served neighborhood accessibility data and visualizes neighborhood polygons; it uses four colors to indicate accessibility levels from very inaccessible (red, AS $\in [0, 0.25)$) to very accessible (green, AS $\in [0.75, 1]$). Neighborhoods that have less than 50% audit coverage are colored gray to indicate that the data is not available yet.

Because the impacts of different accessibility features vary among people with different mobility levels, our proof-of-concept application provides a set of range sliders to adjust the significance of each accessibility feature. The Access Map gets updated dynamically when the significance changes.

In addition to the choropleth map that visualizes the overview of neighborhood accessibility, Access Map also lets users to explore *why* a certain neighborhood has high/low Access Score by visualizing data from Access Features API. The accessibility features' locations are visualized as points on the map (Figure 6.13). These points are set visible only when the user zooms into an area of a map to reduce visual clutter.

6.6.2 Accessibility Analytics

Researchers are often interested in understanding the relationship between particular characteristics of neighborhoods, such as the neighborhood walkability and the real estate values [44], the socio-economic status and the health of the neighborhoods [66], and the means of transportation and the street connectivity [162]. For example, Saelens *et al.* surveyed studies that investigated correlation between people's walking/cycling

activities and neighborhood characteristics such as population density, street connectivity, land use mix, and neighborhood socio-economic status to better understand what facilitates higher rates of walking/cycling [163]. Their survey revealed that population density is among the most consistent positive correlates of walking trips. As we can see in Saelen's investigation and the studies that they surveyed [30,144,154], revealing the correlation (or lack thereof) between two factors advances our understanding of the relationship between them.

To demonstrate the value of the collected accessibility data and Access Score as neighborhood accessibility analytic tools, we conduct a preliminary investigation of the relationship between $AS_{neighborhood}$ and 45 socio-economic statistics for D.C. neighborhoods from 2010 census (*e.g.*, average household income, race and ethnic distribution, and unemployment)—see Table 6.2. When socio-economic data of the same type were available from multiple points in time, we used the most recent data. For example, Total Population for each D.C. census tract is available for 1980, 1990, 2000, and 2010. In this case, we used the data from 2010.

Because some neighborhood socio-economic measures are non-continuous ordinal variables (*e.g.*, a number of schools present in a neighborhood), we use Spearman's rank correlation (r_s) to assess the relationship between them and the $AS_{neighborhood}$. We note that our intention is not to make a causal statement, but simply to use this correlation to validate the value of the information contained in our accessibility data.

Property	r_s	Property	r_s
Occupied housing units, 2010	0.41	Property crimes, 2011	0.05
% pop. 16+ yrs. employed, 2010-14	0.39	Number of schools, 2013	0.04
Avg. family income, 2008-12	0.37	Number of DCPS schools, 2013	0.03
Population, 2010	0.36	% subprime loans, 2006	-0.02
% foreign born, 2010-14	0.36	% HHs with a car, 2010-14	-0.02
% Asian/P.I. non-Hispanic, 2010	0.34	% seniors in poverty, 2010-14	-0.03
Loans per 1,000 housing units, 2006	0.32	% same house 5 years ago, 2000	-0.05
% Hispanic, 2010	0.32	Trustee deed sale rate, 2013	-0.12
% change in avg. family income, 2000 to 2008-12	0.22	SF homes, 2013	-0.12
% white non-Hispanic, 2010	0.19	Persons receiving food stamps, 2014	-0.16
% change senior population, 2000 to 2010	0.16	% persons without HS diploma, 2010-14	-0.17
Median borrower income, 2006	0.16	Violent crimes, 2011	-0.19
% HHs with a phone, 2010-14	0.15	% low weight births (under 5.5 lbs), 2011	-0.21
Charter school enrollment, 2013	0.15	% births to teen mothers, 2011	-0.22
% seniors, 2010	0.14	% black non-Hispanic, 2010	-0.29
Number of charter schools, 2013	0.11	% children, 2010	-0.30
% change population, 2000 to 2010	0.10	% change child population, 2000 to 2010	-0.31
SF homes/condos receiving foreclosure notice, 2013	0.10	Rental vacancy rate (%), 2010-14	-0.32
Foreclosure notice rate, 2013	0.09	% female-headed families with children, 2010-14	-0.35
Total school enrollment, 2013	0.08	% children in poverty, 2010-14	-0.35
Homeownership rate (%), 2010-14	0.08	Unemployment rate (%), 2010-14	-0.36
DCPS school enrollment, 2013	0.07	Poverty rate (%), 2010-14	-0.36
Number of sales, 2015	0.06		

Table 6.2. Correlation between neighborhood statistics and Access Score: Neighborhoods.

We summarize the list of correlates and the corresponding Spearman's rank correlation indices on Table 6.2. Of the 45 neighborhood socio-economic statistics, the measures that indicated the strongest correlations with $AS_{neighborhood}$ were Occupied housing units, 2010 (r_s=0.41), followed by Percentage of Population 16+ years employed, 2010-14 (0.39) and Average Family Income, 2008-12 (0.37). Although there is no definitive measure of what is considered as a "strong correlation," these indices represent moderate positive correlations according to Tayler's definition [175]. On the other hand, the three characteristics that had strongest negative correlations are: Poverty rate (%), 2010-14 (-0.36), Unemployment rate, 2010-14 (-0.36), and Percentage of Children in Poverty 2010-14 (-0.35). While no definitive conclusion can be drawn from this analysis, the result suggest neighborhood accessibility is correlated with the wealth of neighborhoods.

6.7 Discussion and Future Work

We developed and deployed the system to collect street-level accessibility information, conducted the preliminary evaluation of the collected data, and the demonstrated two proof-of-concept ALTs. As this is the first work that uses VGI system to collect street-level accessibility data and demonstrates ALTs that utilize the collected accessibility information, it poses areas of improvements and opens up future research opportunities.

Accuracy of the accessibility feature labels. We showed that the accuracy of the accessibility feature labels are 77% when compared to researcher provided ground truth. While the most dominant curb ramp labels were 98% accurate, other accessibility feature data had accuracy below 81%. Recall was higher than precision for missing curb ramps, obstacles, and surface problems, showing that volunteers tend to over label these problems. The fact that we can get accurate data for curb ramps with a single volunteer suggest that we only need to allocate a single volunteer to audit accessible neighborhoods. On the other hand, we should assign multiple volunteers to audit inaccessible neighborhoods to collect reliable information about accessibility problems. Future work should also investigate how to improve label consistency between multiple volunteers. For example, the web site should implement a better set of practice tasks as well as "talk" feature on the web site where novice can learn from experienced volunteers through a discussion board [131]. It is also important to evaluate if the accuracy or other data qualities of the data collected by volunteers and paid crowd workers differ.

How can we increase data collection throughput? We need further investigation of how to effectively collect volunteer participation. We deployed the VGI system and announced it to the small group of people (*e.g.*, undergraduate students). We will fully deploy and investigate the effect of advertisement of the system to the contribution to the data collection.

Another intriguing area of research is how to computationally optimize the amount of contribution made by volunteers. As we observed, areas that are accessible require minimal audits. Allocating less workers to accessible neighborhoods increases data collection throughput. The question is, then, "how can we (semi-)automatically identify accessible and inaccessible neighborhoods prior to allocating volunteers?" Future work should investigate the feasibility of using existing neighborhood statistics (*e.g.*, correlates discussed in the accessibility analytics) to predict the accessibility of the neighborhoods and use them to prioritize the volunteer allocation.

Improving Access Score. We developed two neighborhood accessibility indicators AS_{street} and $AS_{neighborhood}$. Since this was the first work quantifying accessibility of geographical regions using the crowdsourced accessibility data, we employed simple methods in which we counted number of accessibility features in a given area. There are limitations and future work should address them. First, we only took into account of presence and absence of curb ramps, sidewalk obstacles, and surface problems. Other features such as presence and absence of sidewalks, terrain information, temporary accessibility barriers such as vehicle/pedestrian traffic, and Walk Score should be considered as potential features to compute Access Scores. Also,

future work should look into methods to incorporate the severity ratings of the accessibility features that we collected.

Future ALTs. We could design and develop more ALTs. For example, combining the street-level accessibility data and sidewalk network data enables accessibiltiy-aware pedestrian navigation system that can be used by people with mobility impairments to plan travel routes. The presented proof-of-concept applications could also be improved. For example, neighborhood accessibility analysis should be conducted again once we have fully covered the D.C. neighborhoods. Because the insights emerged may only apply to D.C., it is also important to extend the study to multiple cities. Future work should also investigate how people use these ALTs (*e.g.*, would Access Scores impact planners' decisions on alterations in city infrastructures?).

6.8 Conclusion

In this chapter, we (i) developed VGI system that lets volunteers to contribute to accessibility data collection, (ii) invited volunteers to populated the accessibility data repository, (iii) conducted a preliminary evaluation of the collected accessibility data, (iv) developed backend system that serves the collected accessibility data to clients through three REST APIs, and (v) designed and developed Access Map and demonstrated a preliminary accessibility analysis. As a whole, this chapter shows the utility of the accessibility data collection methods and the value of the large accessibility data repository.

Chapter 7 Conclusion

The primary goal of this dissertation was to design, develop, and evaluate scalable methods to remotely and accurately collect street-level accessibility data. In this chapter, we first briefly summarize the threads of research in this dissertation before describing the main contributions and outlining promising directions for future work.

To fulfill the thesis goal, we conducted four threads of research. In Chapter 3, the formative interview study revealed how people currently assess accessibility of the physical environment. The study also identified 10 key design features and 6 data qualities for future designs of assistive location-based technologies. Findings from this study, in combination with previous work [21,135,167], motivated us to design accessibility data collection methods that use Google Street View (GSV) as a massive source of street-level accessibility information. In Chapter 4, we designed, developed, and evaluated an online image labeling system where crowd workers can view and label accessibility features in GSV images. The study showed that with appropriately designed interfaces, minimally trained crowd workers can provide accessibility data with an accuracy of 81% and up to 93% with quality control mechanisms. To increase the efficiency of the crowdsourced data collection methods, we introduced a semi-automated data collection system, Tohme, which combines crowdsourcing, computer vision, and machine learning in Chapter 5. We showed that we can increase the accessibility data collection efficiency by 13% without sacrificing the accuracy. In Chapter 6, we developed, deployed, and evaluated a VGI system that collects street-

level accessibility data. Also, we designed and developed proof-of-concept assistive location-based technologies with the collected data. The developed system showed the value of the proposed data collection methods.

7.1 Summary of Contributions

In this section, we restate the contributions listed in the Introduction and summarize how each of these contributions were achieved.

7.1.1 Characterization of How People with Mobility Impairments Assess Accessibility of the Physical Environment

We conducted a formative interview study (Chapter 3) with 20 people with mobility impairments. The findings from the study highlight common accessibility barriers and facilitators in the built environment, the impact of those barriers, and methods to mitigate or avoid accessibility problems, which reaffirm and extend prior work (*e.g.*, [129,135,152,167]). We also uncovered how modern technology is used to assess accessibility. For example, online imagery such as GSV and satellite imagery are used by people with mobility impairments to visually assess the physical accessibility of locations of their interests.

Through participatory design activities, we identified ten desired features and six essential data qualities for ALTs. The top three most desired features were providing detailed descriptions, accessibility-aware routing, and top-down map-based views of street-level accessibility. Data quality attributes—granularity, relevance,

credibility, recency of information, coverage, and location-precision—often related to features (*e.g.*, high granularity of data corresponds to the detailed description feature). No prior research has enumerated desired features and data qualities of ALTs. Our findings have direct implications for the design of ALTs.

7.1.2 A Novel Crowd-powered Method for Collecting Accessibility Data

Another contribution of this dissertation is the design, development, and evaluation of a novel method for collecting street-level accessibility information by combining crowdsourcing and GSV imagery. First, we assessed the viability of using GSV imagery as a data source for street-level accessibility information. Six dedicated workers, three wheelchair users and three researchers, went through curated set of Street View images and identified accessibility problems in the images. We observed high concordance between the accessibility problems identified by researchers and wheelchair users. This shows that (i) dedicated people can consistently find accessibility problems in Street View imagery and (ii) what they consider as accessibility problems correspond to what mobility impaired people consider as accessibility barriers.

Second, we designed and developed three types of interfaces to label accessibility features in Street View images. The three designs, point-and-click, rectangular bounding box, and outline interface, were designed with consideration of tradeoff between interaction speed and data granularity. With a study with 153 crowd

workers, we quantitatively evaluated speed and accuracy for different types of interfaces, which informed the design of image labeling tools in the for data collection.

Third, we showed that minimally trained crowd workers from Amazon Mechanical Turk can accurately find and label accessibility problems. The study with 402 crowd workers showed that minimally trained workers can provide accessibility data with an accuracy of 81% and this figure increased to 93% with quality control mechanisms like majority voting.

We also note that our data collection approach is generalizable to other domains and could be used to collect a variety of urban data for public health and city planning purposes. For example, we used crowdsourcing to collect bus stop landmark information from GSV in our previous work [77,78]. The collected data, such as presence of bus stop signs and shelters at a given bus stop, could be used in a navigation tool to support people with visually impairments to localize bus stop; people with visual impairments could identify what landmarks to look for when they are searching for the bus stop. Future work should extend this approach to collect other important data such as urban vegetation, city cleanliness, and bicycle signs and lanes.

7.1.3 A New Approach for Combining Crowdsourcing and Automation

To overcome the sole reliance on human labors in labeling Street View images, which limits scalability, we introduced Tohme, a semi-automated system for remotely collecting geo-located curb ramp data using a combination of crowdsourcing, computer vision, machine learning, and online map data. Tohme lowers the overall human time

cost of finding accessibility problems in GSV while maintaining result quality. The main contribution of Tohme is the design, development, and evaluation of the overall system. Through a study with 403 crowd workers from Amazon Mechanical Turk, we showed that Tohme can detect curb ramps in Street View images much more accurately compared to computer vision system alone (F-measure: 84% vs. 67%), but at a 13% reduction in human time cost compared to completely manual labeling approach.

7.1.4 VGI system and Proof-of-Concept ALTs

In the final part of this dissertation, we (i) developed volunteered geographical information (VGI) system for collecting the street-level accessibility data and conducted a preliminary deployment and evaluation of the system, and (ii) designed and developed two proof-of-concept ALTs to demonstrate the value of the collected street-level accessibility information. For our initial evaluation, we invited a small number of volunteers and students (who received extra credit in their classes) to use our VGI system to find and label street-level accessibility information via word-of-mouth. At the time of analyzing the data (July 24th, 2016), 154 volunteers (of which 56 were undergraduate students who received extra credit) contributed and we gathered data from 20% of the streets in Washington, D.C. In our preliminary evaluation of the collected accessibility data, we showed that the overall data accuracy is 77%.

To demonstrate the value of the accessibility data collection methods and the collected street-level accessibility data, we developed Access Map and conducted accessibility analytics. Access Map was designed to support people with mobility

impairments to easily explore neighborhood accessibility. The tool could be used when mobility impaired users are deciding on where to live. Accessibility analytics revealed relationship between neighborhoods' socio-economic characteristics and accessibility. While preliminary, the study provides directions for more rigorous analysis to investigate what neighborhood characteristics make neighborhood (in)accessible.

7.2 Cost Estimation for Large-Scale Data Collection

The data collection methods introduced in this dissertation enable us to gather street-level accessibility data at scale. The *in situ* audit that used to take years to perform could be done in days with remote accessibility audits. For example, to collect the street-level accessibility information from the 1,200 mi of the roads in Washington, D.C., it would only take 152 human-hours (based on the average audit speed of the researcher: 7.9 mi per hour). Even we solely rely on paid crowd workers, this would only cost $1.1k (with federal minimum wage $7.25 per hour), and this figure will go down even further by incorporating help from volunteers and increasing audit efficiency by combining automation. This is much faster and cheaper compared to *in situ* auditing; to put it into context, the *in situ* ADA compliance audits conducted by DC DOT by three field auditors in the last three years only covered 15% of the streets in DC.

While this dissertation provides the first step towards large scale street-level accessibility data collection, research and practical challenges remain open as future work. For example, while we have started collecting street-level accessibility

information from Washington, D.C., the total street distance in the city only amounts to 0.1% of the entire roads in the U.S. urban areas (1,200 mi of 1.2m mi) [187], which is estimated to take 6.3k human-days and will cost $1.1 million if we relied on paid crowd sourcing with federal minimum wage. This figure will increase if we employ quality control mechanisms such as majority voting used in this dissertation (*e.g.,* the cost will be tripled if we use three labeler majority vote). Therefore, increasing the data collection efficiency by incorporating more volunteer contributions, integrating automation, and increasing data quality with less human work will be the major challenges. We discuss future research directions to address this issue in the next section.

7.3 Directions for Future Research

In this section, we cover the limitations of this dissertation to both better frame and scope our contributions as well as to highlight opportunities for future work. We first discuss how our crowdsourcing-based accessibility data collection methods could be made more efficient. Second, we discuss potential approaches for combining crowdsourcing and computer vision to extend the work presented in Chapter 5. Finally, we discuss design, development, and evaluation of future assistive location-based technologies enabled by our data collection method.

7.3.1 Crowdsourcing

We showed that it is feasible to accurately collect street-level accessibility data from GSV. The accuracy of crowdsourced data, however, came with the cost of redundancy in quality control methods (*i.e.*, majority voting or manual verification) that are time consuming and labor intensive. For the data collection to scale even further, crowdsourced data collection and quality control methods need to be more efficient. In this section, we discuss potential areas of future work in (i) increasing per-worker accuracy to reduce quality control cost, (ii) designing more efficient quality control mechanisms, and (iii) designing of more efficient interaction methods for data collection.

Training and Feedback for Crowdsourcing Tasks. For our crowdsourcing tasks, novice workers were guided to watch a video tutorial (Chapter 4) or complete interactive tutorials (Chapter 5 & 6) that explained the motivation of the task, how to interact with the user interfaces, and what constitutes accessibility features. While we found that these tutorials were sufficient for our online workers to complete the crowdsourcing tasks, suboptimal per-worker accuracy necessitated quality control. This is a common problem in crowdsourcing and online citizen science projects focused on data collection [131]. To lower the cost while maintaining the accuracy, future work should investigate more efficient mechanisms to obtain high-quality accessibility data from crowd workers.

One interesting future work is to investigate how to effectively and efficiently train workers and/or give better feedback on their performance to increase per-worker

accuracy. Existing research in crowdsourcing and MOOCs can guide the design of training methods. For example, recent work reported that providing means of communication among distributed peers have the benefits of increased productivity of crowd workers [170], increased quality of work [59,131], and facilitates collaborative learning [40]. For example, Zhu *et al.* showed that reviewing other people's work is an effective way of learning how to conduct a task [217]. Dow *et al.* showed that crowd workers produce better results when they self-assessed and received external assessment of one's work compared to the case where there is no feedback [55]. Learning from the above approaches, future work should explore the following: (i) would providing novice crowd workers ways to ask experienced workers what accessibility features to label increase overall accuracy of the data? (ii) Would letting novice crowd workers see what experienced workers labeled (which could be treated as a part of verification tasks) improve their understanding about what accessibility features to label in GSV?

We imagine, however, training to label some accessibility features will be harder than others. For example, it is hard to make a decision on whether to report a missing curb ramp in some cases. Imagine a busy intersection with no signal lights; those intersection were designed so pedestrians are not supposed to cross. In fact, the ADA Standards for Accessible Design requires "(curb ramps) shall be provided wherever an accessible route crosses a curb" [189], leaving where accessible routes should be installed ambiguous. This is especially hard when there is no clear indication of walkways at intersection (*i.e.*, crosswalks).

Efficient Quality Control of the Accessibility Data. Even with careful training, however, unavoidable errors in labeling accessibility features (*e.g.,* confusion due to difficult labeling tasks) necessitate quality control of the collected data [10,92,121]. Nevertheless, prior research in crowdsourcing suggests that the methods we employed, majority vote and manual verification, are not cost effective [100,149,203] and can be made more efficient. For example, Whitehill *et al.* [203] and Raykars *et al.* [149] designed and evaluated unsupervised machine learning methods to assess workers' labeling skills from their mutual agreement. Baba and Kashima extended this so that the methods not only measures labelers' skills but also models verifiers' efficacy [10]. While these methods use workers' responses to specified tasks as a sole signal for measuring their efficacy, Rzeszotarski and Kittur showed that it is also possible to model worker quality using behavioral information (*e.g.,* scrolling, mouse movements, completion time) [161].

Using the assessed worker quality, we could either filter out data from less reliable workers [121], adaptively assign difficult future tasks to more reliable workers [46,100], or adjust the number of workers assigned to a single task [99,202]. These techniques would make quality control more cost-effective compared to naively taking majority vote of a fixed number of workers or asking a fixed number of workers to verify labels. Since the crowd tasks in above literatures are different from ours and often use datasets curated for lab experiments, it is important to explore how much these quality control mechanisms can increase the efficiency of our data collection methods.

Efficient Interaction Methods. The labeling and verification interfaces presented in this dissertation could be improved; it is important to explore more efficient interface designs and interaction methods to reduce the data collection cost. For example, we designed quickVerify interface in Chapter 6 that allowed users to quickly verify computer vision detections of curb ramps. In our preliminary examination of quickVerify with 56 turkers, however, we found that this interface actually reduced verification recall from 60.5% (which is by the curb ramp detector alone) to 23.4% because correct detections were erroneously rejected by turkers, most likely because of lack of enough visual context in the presented images. We should therefore explore other interface designs. One potential approach for increasing the efficiency of the labeling and verification tasks is to eliminate the cost of panning and walking in GSV. For example, we could stitch together multiple Street View images that are then played back as a movie. In the early prototype that we created (Figure 7.1), workers could use this interface to simply label perceived problems or verify labeled features as they are quickly "driven" through the street scenes. With this approach, we minimize the time they have to interact to "walk" in the Street View environment. Unlike the existing video annotation research (*e.g.,* marking players in a recording of a basketball game) [197,198], however, videos generated from 360-degree Street View images will not necessarily have a "good" camera angle. For example, we could set the camera's heading angle perpendicular to the driving direction to show the street sides, but it is not clear what would be the best vertical angle (pitch) to assess the sidewalk accessibility. Thus it would be interesting to investigate what constitutes a

Figure 7.1. A prototype time-lapse video created from consecutive GSV panoramas. The camera automatically moves along the street and faces towards the street side, so the user could assess presence/absence of accessibility features such as sidewalks, curb ramps, and obstacles.

"good" angle by learning from existing field such as camera position/angle optimization in 3D game design and automatically optimize the camera view port accordingly [35]. Other research opportunities include the investigation of how fast such video could be played to let workers accurately label accessibility features and how many videos workers could process in parallel (like [126]).

7.3.2 Computer Vision

In chapter 5, we initiated research into combining crowdsourcing and automatic curb ramp detection. Although some recent research has explored use of computer vision for locating accessibility features like cross walks (*e.g.,* Ahmetovic et al. [3]), the space is still largely underexplored. Therefore, in addition to evaluating accuracy of state-of-the art object detection and scene understanding algorithms (*e.g.,* [165,166]) for accessibility feature detection, future work should push forward the state-of-the-art computer vision research by investigating (i) how to combine computer vision and crowdsourcing to accurately and efficiently find accessibility features and (ii) how to assess fine grained information about the accessibility.

Combining of Crowdsourcing and Computer Vision. The key to effectively integrate computer vision and crowdsourcing is to understand the performance of computer vision algorithms and adaptively use crowd work [158,215]. In chapter 5, we described a method to use an ML-based supervised workflow controller to assess the difficulty of each object detection task, which allowed us to adaptively allocate work to different crowd workflows to reduce human cost. Similar approaches have been taken in recent computer vision. For example, Zhang *et al.* used a supervised machine learning algorithms to detect computer vision failure in semantic image segmentation and vanishing point detection [215]. Russakovsky *et al.* introduced a method to combine a variety of crowdsourcing tasks with object detection algorithms using Markov decision process that automatically balances human cost and object detection accuracy [158].

The workflow controllers in these literatures (including ours), however, required data preprocessing and/or manually defined hyper parameters to assess expected computer vision performance. For example, we defined the criteria for the computer vision failure (*i.e.,* presence of false negatives), and used a set of preprocessed data to let the controller detect failures and decide workflow (manual labeling *vs.* CV + verification). We believe these manual processes could be integrated into automated learning using reinforcement learning. For example, instead of explicitly defining what the computer vision failure is, we could provide overall accuracy and cost as input to let the workflow controller learn what constitute computer vision failures and what features to use to assess those failures. Automating the manual processes will increase the generalizability of the techniques and could make integration of computer vision algorithms and crowdsourcing components for systems designers.

Automatically Retrieving Fine-Grained Accessibility Information. This dissertation showed that crowdsourcing and computer vision can be used to identify the presence curb ramps in a street-level environment. However, we did not investigate if these technologies can accurately assess more fine grained properties of other accessibility features. For example, it is often difficult to make precise quantitative judgments about the obstacle size in an image, or assess whether the incline of a curb ramp is too steep. Future work should investigate the use of high-precision satellite imagery and 3D point cloud data collected via LiDAR data to assess fine grained details of the accessibility features. For example, if a user labels a pole as an obstacle, we can

measure the width of the obstructed path if we have precise 3D point cloud data. Alternatively, because each street view scene often has multiple picture angles, CV-based mensuration techniques like structure-from-motion.

Modeling Indoor Accessibility. We focused on finding outdoor accessibility features from GSV. Equally important is assessing indoor accessibility of points-of-interest for people with mobility impairments. Often times, building owners provide limited, if any, information about the accessibility of their buildings. Crowd-powered projects like Wheelmap and Axsmap are making progress in providing more detailed accessibility information, but the information has low location precision. That is, the applications tell the users whether the building has accessible entrance or not, but it does not tell which entrances are accessible if there are multiple of them. One future research direction include feasibility assessment of combining our data collection approach with indoor Street View imagery (Figure 7.2); we should investigate what useful indoor accessibility information could be semi-automatically extracted by crowd workers and computer vision algorithms (*e.g.,* can we semi-automatically detect locations of entrances and table heights?). Another potential avenue of future research is to make the indoor accessibility data more *confirmable* for the users. For example, we expect recent advancement in automatic 2D floorplan or 3D indoor model generation [25,36,90,115,208] could be useful for people with mobility impairments; using the generated 2D/3D indoor maps, mobility impaired people could easily assess which parts of the building are accessible and inaccessible.

Figure 7.2. Indoor Street View imagery of public places (*e.g.*, restaurants) contains potentially useful accessibility information such as presence and location of accessible entrances and height of tables. See https://goo.gl/maps/4LZ3GRHvdEK2 for the original Street View image.

Detecting Changes in Accessibility over Time. Changes in streetscape could affect the accessibility of street-level environment. Dynamic characteristics of urban environment such as pedestrian density, construction, and snow accumulation affect the accessibility of the sidewalks [129]. Long-term changes like construction of sidewalks and curb ramp installment can often improve sidewalk accessibility for mobility impaired people. Whether the changes include accessibility improvement or degradation, the accessibility information needs to be kept updated so that we can provide accurate information to the users. Therefore, an important future work would be to design efficient methods to track these changes. An interesting area of future research includes the development of methods that incorporate recent advancement in computer vision research in detecting environmental changes over time (*e.g.*, [8,169,183]). Using frequently updated geo-localized images and 3D models (which could be collected through LiDAR or generated from structure-from-motion) of the built environment, we could use background-subtraction to automatically detect

Figure 7.3. Three form factors of accessibility-aware navigation tool. (a) A smart phone based navigation system similar to existing applications like Google Maps and Apple Maps. (b&c) Google Glass and smart watch-based navigation applications; we expect these form factors are easier for manual wheelchair users to use while they are on-the-go and their hands are occupied.

changes in accessibility features in the street-level environment. As we believe the completely automated methods would not provide sufficiently accurate change detection, the key challenge here would be, again, to design methods to effectively combine inaccurate but efficient computer vision algorithms with accurate but more costly human work so that we can collect data with low cost while maintaining high accuracy.

7.3.3 Design, Development and Evaluation of Applications

Our accessibility data collection methods enable a variety of new assistive location-based technologies for people with mobility impairments. This opens up rich research opportunities for designing technologies for people with mobility impairments and beyond.

Design and Development of Assistive Location-based Technologies. It is important for us to extend our formative study described in Chapter 3 that explored what location-based technologies could be useful for people with mobility impairments. Following the design approach in HCI, we should iteratively design and

develop mid- to high-fidelity technologies and conduct participatory design with potential users with mobility impairments to refine the applications. During this design process, it is crucial to investigate preferred accessible form factors of the technologies. For instance, for an accessibility-aware navigation tool—technology desired by the majority of our interview study participants—we expect wheelchair users to prefer Google Glass- or smart watch-based interfaces over a traditional smart phone-based design, because the former designs would be useable even the wheelchair users are on-the-go and their hands are occupied by rolling the chair—see lo-fidelity system prototypes on Figure 7.3.

Data Quality Requirement Analysis. In evaluating each of the future technologies, it is important to assess the required levels of accuracy and data granularity, as well as investigate what the budget needed to achieve those levels. For example, while neighborhood-level visualization such as Access Score visualization described in Chapter 6 does not require high location precision (the application needs to know the presence of accessibility problems in a given area larger than localization errors introduced by GPS inaccuracy), accessibility-aware navigation systems may need higher location precision. It is also important to investigate the necessary data accuracy for each application; while we achieved high-accuracy (*e.g.,* 93% for image level accuracy), it is unclear if this level of accuracy is enough for the applications to be used by people with mobility impairments. Because more accurate and precise data would require careful quality control, the cost of data collection would increase.

Applications in Urban Planning and Public Health. The street-level accessibility information that is collected with our methods not only support people with mobility impairments, but could also be used for urban design and planning for policy makers, public health researchers, and urban planners [15,157]. Studies in the above fields have found associations between specific neighborhood characteristics (*e.g.,* cleanliness, perceived safety) and cardiovascular disease [41], self-rated health [2], walking and other forms of physical activity [86], and obesity [17]. However, the effect of street-level accessibility to these health, social, and psychological factors has not been studied at a large scale, presumably because it has not been possible without comprehensive data about the accessibility of the built environment. Therefore, an important piece of research would be to investigate if the street-level accessibility data could be used as a source of good indicators for above factors. And if so, the work should also explore how we can empower public health researchers and practitioners like urban designers to use the data through technologies.

7.3 Final Remarks

We have provided insights into how to scalably collect street-level accessibility data using crowdsourcing and automated methods from GSV through development and evaluation of crowd-powered systems. We believe this dissertation serves as the first step towards making technologies that enable us to characterize accessibility of the physical environment of the world.

Appendix A

Formative interview study materials

Includes:

- Background survey
- Semi-structured interview script
- Participatory design session scenarios
- Participatory design session templates
- Design probes

Background Survey

1. Your name

2. Age

3. Gender

4. Please describe your mobility impairment and the way that it affects your movement. If you have a specific diagnosis, please list that as well (e.g., spinal cord injury, level c5)

5. How long have you had your mobility impairment (e.g., 5 years)?

6. What mobility aids do you use?
   [Manual wheelchair / Electric wheelchair / Scooter / Cane / Walker / Other]

7. What is your main means of transportation for everyday tasks (e.g., to grocery store, to cafe, to a park).
   [Private vehicle / Paratransit / Public transportation (e.g., Metrobus) / Wheelchair or walk / Other]

8. How often do you leave your home to take trips in your city (regardless of transportation mode)?
   [Never / Rarely (once a week) / Sometimes (a few times a week) / Often (nearly everyday) / Everyday]

9. Do you have any other impairments? (e.g., vision impairment)
   [None / Vision impairment / Hearing impairment / Upper body motor impairment / Other]

10. Do you use a computer? [Yes / No]

11. If so, how often do you use a computer every week on average?
    [Rarely (once a week or less) / Sometimes (a few times a week) / Often (nearly everyday) / Everyday]

12. Do you use any assistive technologies to use a computer? For example, a trackball mouse? Please describe

13. Do you own a mobile phone? [Yes / No]

14. Is your mobile phone a smartphone? [Yes / No]

15. Do you use any assistive technologies to use a mobile phone? (e.g., a mouth stick or Apple VoiceOver, etc.). Please describe.

16. How did you learn about this interview study?
    [From my family and/or friends / Email from an accessibility organization / Email from the research team / Email from the University of Maryland (e.g., FYI UMD) / Other]

Semi-structured Interview Script

Preparation

- Interview study script
- Design activity instruction for the participant
- Pen and paper
- Consent form
- Payment form

Introduction

Hello, I am [YOUR NAME]. First, I would like to thank you for participating in our study.

Our team is designing new methods and tools to inform people about inaccessible areas of a city. For example, places could be inaccessible due to lack of sidewalks, absence of curb ramps at intersections, or inaccessible building entrance.

The goal of this study is to better understand how you currently cope with the accessibility problems and what technologies could improve the way you plan a trip and navigate the city.

1. The first stage is an interview study. I will ask how you get around the city. For example, we want to know if you ever look up accessibility information about the built environment, and if you do, what methods you use.
2. The second stage is a design activity. We would like you to brainstorm and explore the design of potential map applications that could improve the way you navigate a city.

The brainstorming activity will involve sketching potential map applications that could help you navigate unfamiliar places. If you are not comfortable sketching using a pen and paper, you could describe the potential map applications verbally so we could

sketch on your behalf. We have also brainstormed and prepared some early designs of potential map tools. I will also ask for your feedback about them.

The whole study session should take about 60 minutes. Your data will be kept anonymous. You have the right to stop participating in a study, for any reason, and at any time. We will be audio/video recording. For the video recording, your face will not be captured and we do not intend to take identifiable images of you. Before we begin the interview, we need to complete a consent form and basic background survey.

Is there any question?

[Start recording once the participant signed the consent form.]

Part 1: Methods for Planning a Trip and Navigating through a City

I would like to ask you how you currently learn about accessibility of unfamiliar places and neighborhoods when you travel. For example, I want to know if you look up whether sidewalks exist, curb ramps are installed at intersections, sidewalks are in good conditions and not obstructed, or there are any accessible entrance at a building you visit.

1. Tell me how you get around the city. [Ask how they go to grocery stores, how often, and with who, if the interviewee is stuck.]

2. What would you do if you don't know the neighborhoods? Let's say you changed a dentist and you are visiting a new place.
    - How do you find a new dentist (or any other places to go) in the first place?
    - When you are in an unfamiliar area, what would be your strategy to navigate from a point A to your destination? For example, do you use paper maps or technologies like Google Maps' navigation to find a route?

3. What are the anxieties? What are the challenges for traveling to unfamiliar places?

4. When you visit an unfamiliar place, do you check if the place is accessible?
    - If yes,
        - How? [Ask the following questions if the interviewee does not describe them.]
            - Do you look up accessibility of a building you visit, for example, by calling the place you are visiting?
            - Do you use any existing technologies like mobile app to find accessibility information of the place you visit?
            - What's the preferred method to look up accessibility information?
    - If no,

        - Why not?

5. Do you factor in accessibility of the neighborhoods' built-environment like sidewalks and streets when you are deciding a place to visit?
    - How? Do you use Google Street View?
    - Or why not?

6. Have you ever had any problem because you did not check the accessibility of a place or a route? Could you explain?

Part 2: Brainstorming & Design Session

To explore what tools we could design to help people learn about accessibility of built environment like sidewalks and streets, I would like you to brainstorm ideal map technologies that we could develop.

1. I will present three scenarios. In each scenario, you will be asked to work on a task related to planning a trip.
2. I will ask you to brainstorm ideas for potential technologies that could support you to complete the task in the scenario. Note that the potential tools do not have to exist today; I would like to know what tools could help you rather than what is possible with today's technology.
3. I want you to use a pen and paper to sketch the ideas for about 5-7 minutes. While sketching, I want you to speak aloud so we know what you are thinking. If you are not comfortable sketching, please describe your ideas verbally so I can sketch on your behalf.

Is there any question?

This should be a fun activity! I want us to design the future of accessibility-aware map tools together!

Part 2.1: Exploration of City Accessibility

Let's think about how you could **explore the accessibility of the city**. The tool allows you to quickly browse how accessible city areas are. Please sketch design of the tool that can support you in the following scenario:

Scenario

You are planning to rent a room in an unfamiliar city that you will move to a few months. Imagine that there is a website that provides accessibility information about the city. What should that website look like?

As a start, I've provided a map-based interface below with a few apartments indicated with black icons. Please sketch your ideas below. To help with this task, think about the information you would like to know in order to make a decision about where to live. Imagine that the tool has access to any information that you want!

Part 2.2: Accessibility-aware Location Search

Let's design **location search tool**; the next generation of Yelp that allows you to search businesses with all kinds of accessibility information (*e.g.*, accessibility of building entrance, access to nearby public transportation). Please design a tool that can support you in the following scenario:

Scenario

Your friends are visiting you and you want to take them to an Italian restaurant in Washington, DC. You want to find a popular restaurant, and you also want to make sure the business and its surrounding areas are accessible for you. On a web browser, you choose to ***search for "Italian restaurants in Washington, DC that are accessible."***

Part 2.3: Accessibility-aware Navigation

Now let's think about **routes**, an awesome tool or new features in Google Maps that provides information about accessible routes. Imagine the following scenario:

Scenario

You came to an unfamiliar city for your holiday. You remember there is a natural science museum in the city and decide to visit there. You open a navigation tool on your computer to find accessible routes from the hotel you are staying to the museum.

Part 3: Design Probe

This is the last part of the study. First, I want your feedback on prototype tools that we have designed. Then, we will ask a few questions about the methods we use to collect accessibility information.

Part 3.1: Exploration of City Accessibility

Design Probe

Please remember the scenario where you were planning to rent a room in an unfamiliar city. [Show the sketches of map applications for accessibility exploration.]

- Are tools that show accessible and inaccessible areas of a city useful in this scenario?
- What level of detail do you expect from the application? Do you want to know just an abstract level of street accessibility, or do you want to know where exist what types of accessibility barriers?

Part 3.2: Accessibility-aware Location Search

Design Probe

Please remember the scenario where you were asked to find an accessible Italian restaurant. [Show the sketches of potential map applications and ask following questions.]

- Are tools that allow you to search and sort businesses based on accessibility level useful?
- What level of detail do you expect from the application? Do you want the indoor accessibility and outdoor accessibility to be quantified separately?

Part 3.3: Accessibility-aware Navigation

Design Probe

Please remember the scenario where you were asked to find an accessible route to a museum. [Show the sketches of potential map applications and ask following questions.]

- Are tools that allow you to search accessible routes useful?
- Would both indoor and outdoor navigation be useful?

End of the Study

- Thank the participant for participating.
- Have participant complete payment form.
- Pay participants.
- Ask participants if they will be willing to participate in the future research.

Participatory design session scenarios

Part 2: Brainstorming & Design Session

To explore what tools we could design to help people learn about accessibility of built environment like sidewalks and streets, I would like you to brainstorm ideal map technologies that we could develop.

I will present three scenarios. In each scenario, you will be asked to work on a task related to planning a trip.

I will ask you to brainstorm ideas for potential technologies that could support you to complete the task in the scenario. Note that the potential tools do not have to exist today; I would like to know what tools could help you rather than what is possible with today's technology.

I want you to use a pen and paper to sketch the ideas for about 5-7 minutes. While sketching, I want you to speak aloud so we know what you are thinking. If you are not comfortable sketching, please describe your ideas verbally so I can sketch on your behalf.

Is there any question?

This should be a fun activity! I want us to design the future of accessibility-aware map tools together!

Part 2.1: Exploration of City Accessibility

Let's think about how you could **explore the accessibility of the city**. The tool allows you to quickly browse how accessible city areas are. Please sketch design of the tool that can support you in the following scenario:

Scenario

You are planning to rent a room in an unfamiliar city that you will move to in a few months. Imagine that there is a website that provides accessibility information about the city. What should that website look like?

Part 2.2: Accessibility-aware Location Search

Let's design a **location search tool**; the next generation of Yelp that allows you to search businesses with all kinds of accessibility information. Please design a tool that can support you in the following scenario:

Scenario

Your friends are visiting you and you want to take them to an Italian restaurant in Washington, DC. You want to find a popular restaurant, and you also want to make sure the business and its surrounding areas are accessible for you. What should the application look like?

Part 2.3: Accessibility-aware Navigation

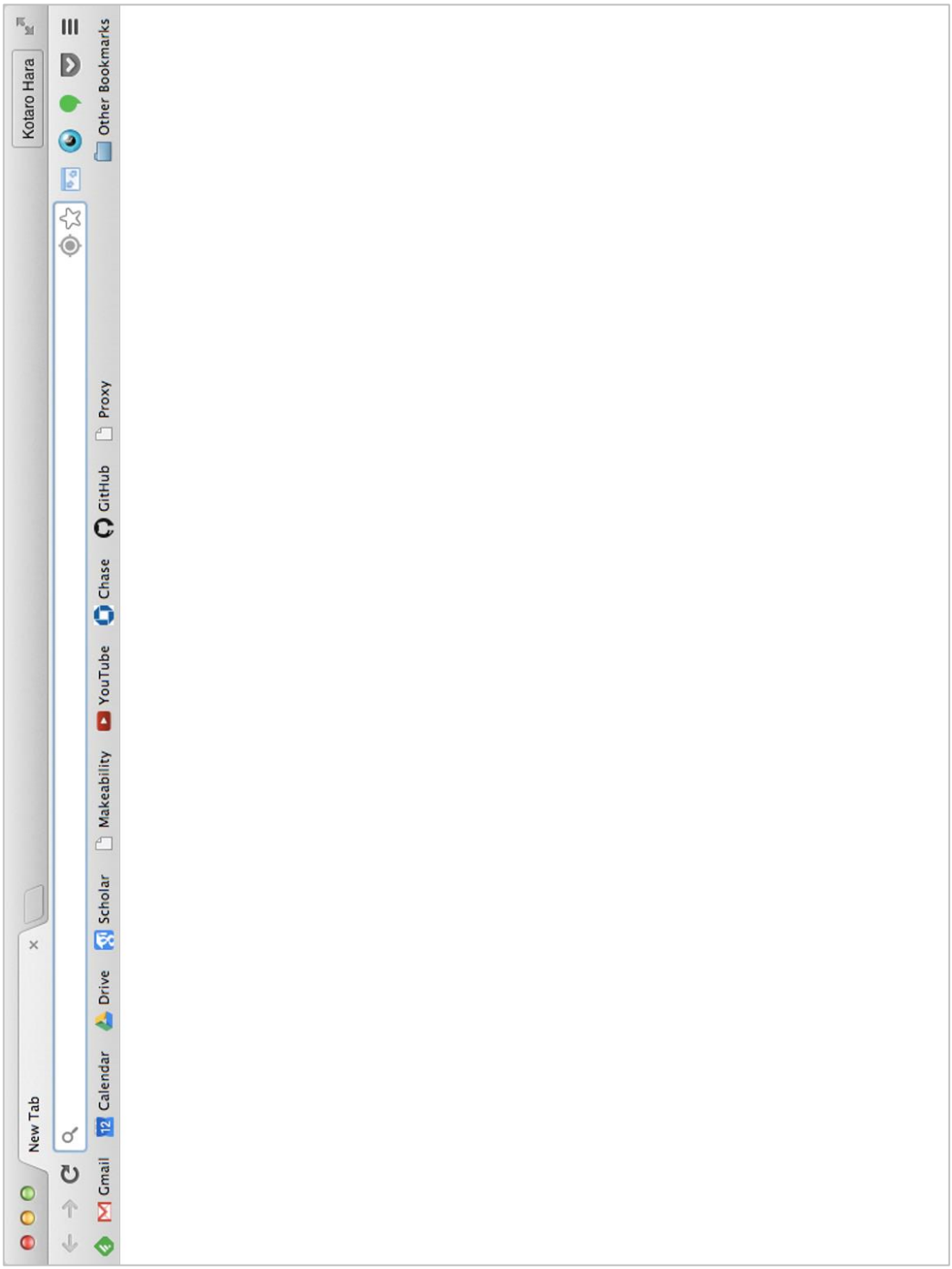
Now let's think about **routes**, an awesome tool or new features in Google Maps that provides information about accessible routes. Imagine the following scenario:

Scenario

You came to an unfamiliar city for your holiday. You remember there is a natural science museum in the city and decide to visit there. You open a navigation tool on your computer to find accessible routes from the hotel you are staying to the museum.

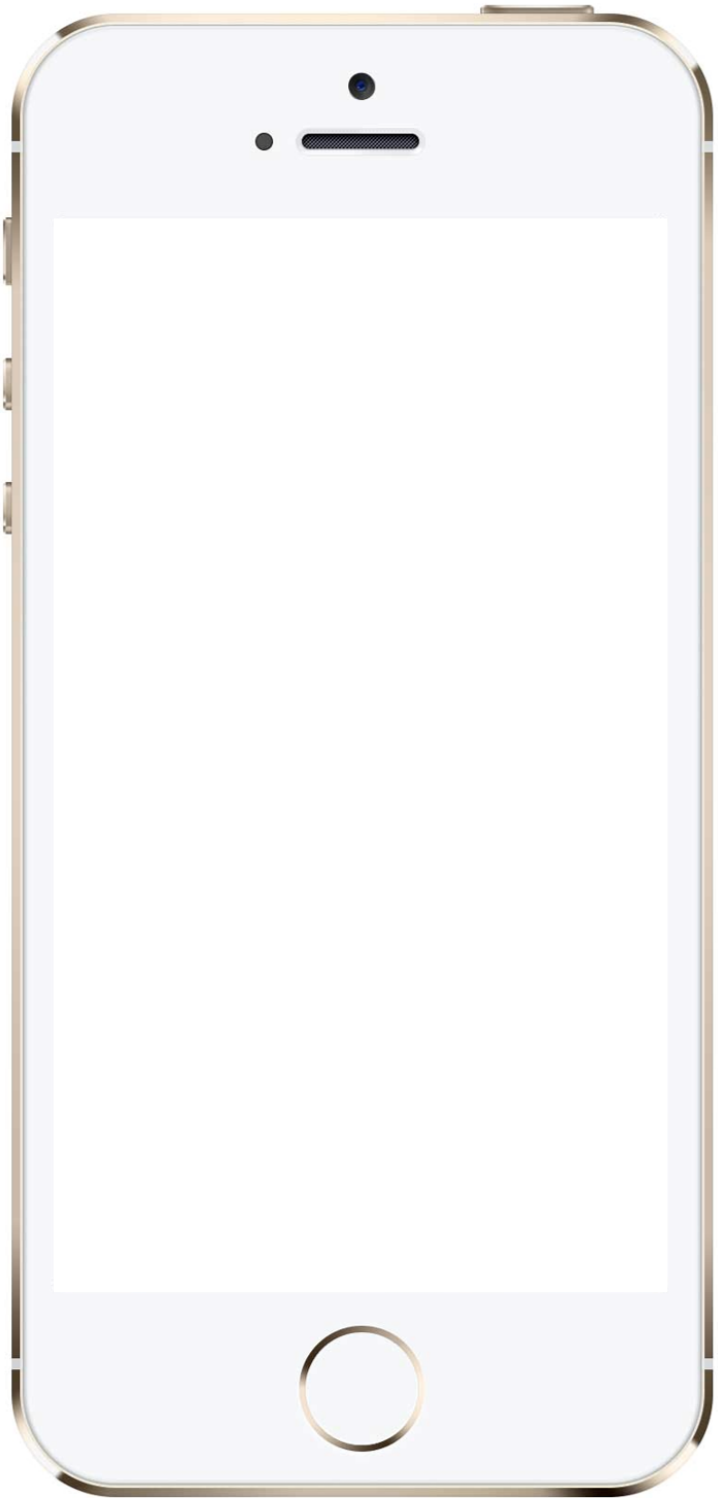
What should the application look like?

Participatory Design Session Template

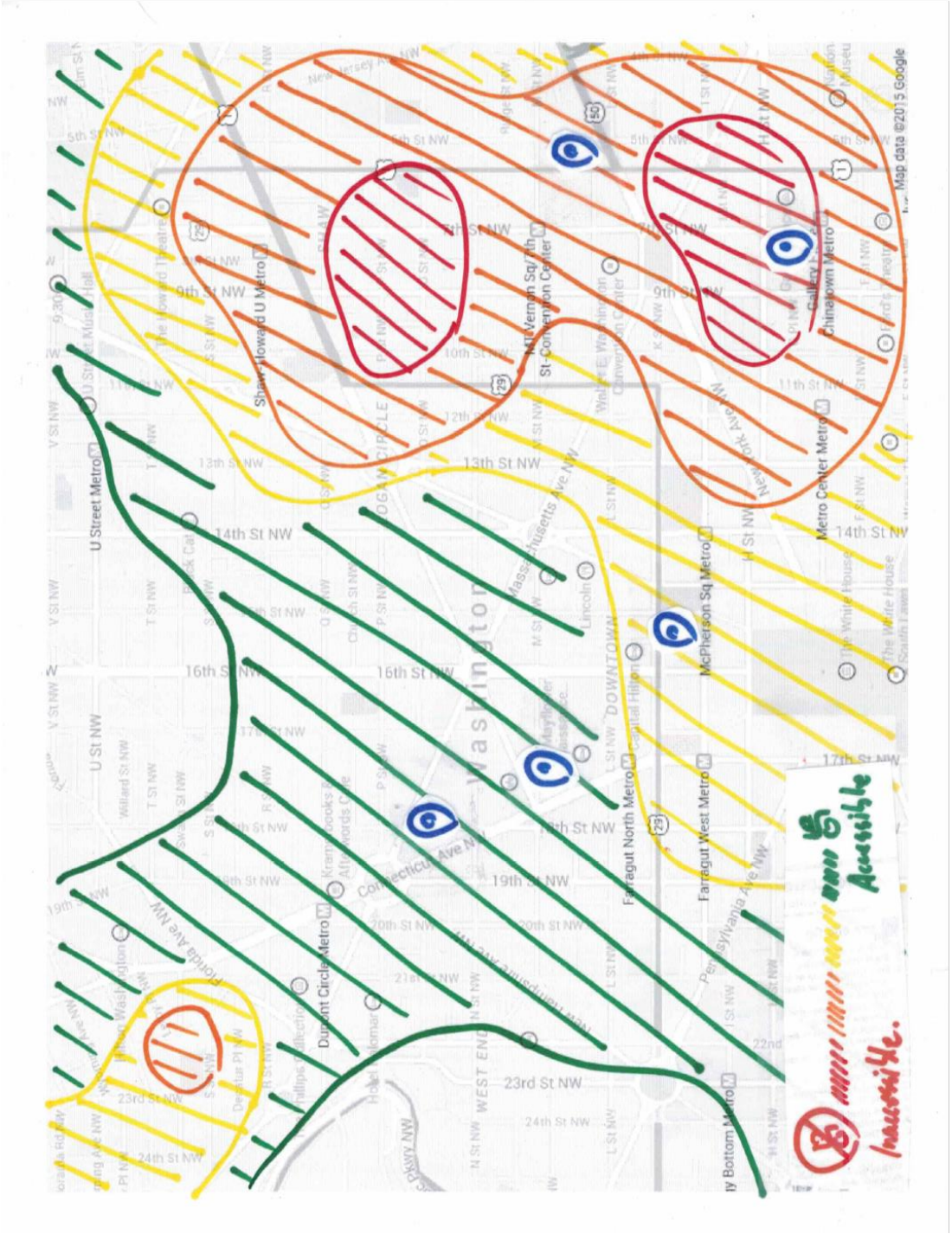


Kotaro Hara
New Tab
Gmail
Calendar
Drive
Scholar
Makeability
YouTube
Chase
GitHub
Proxy
Other Bookmarks
Apartment

Kotaro Hara
New Tab
Gmail
Calendar
Drive
Scholar
Makeability
YouTube
Chase
GitHub
Proxy
Other Bookmarks

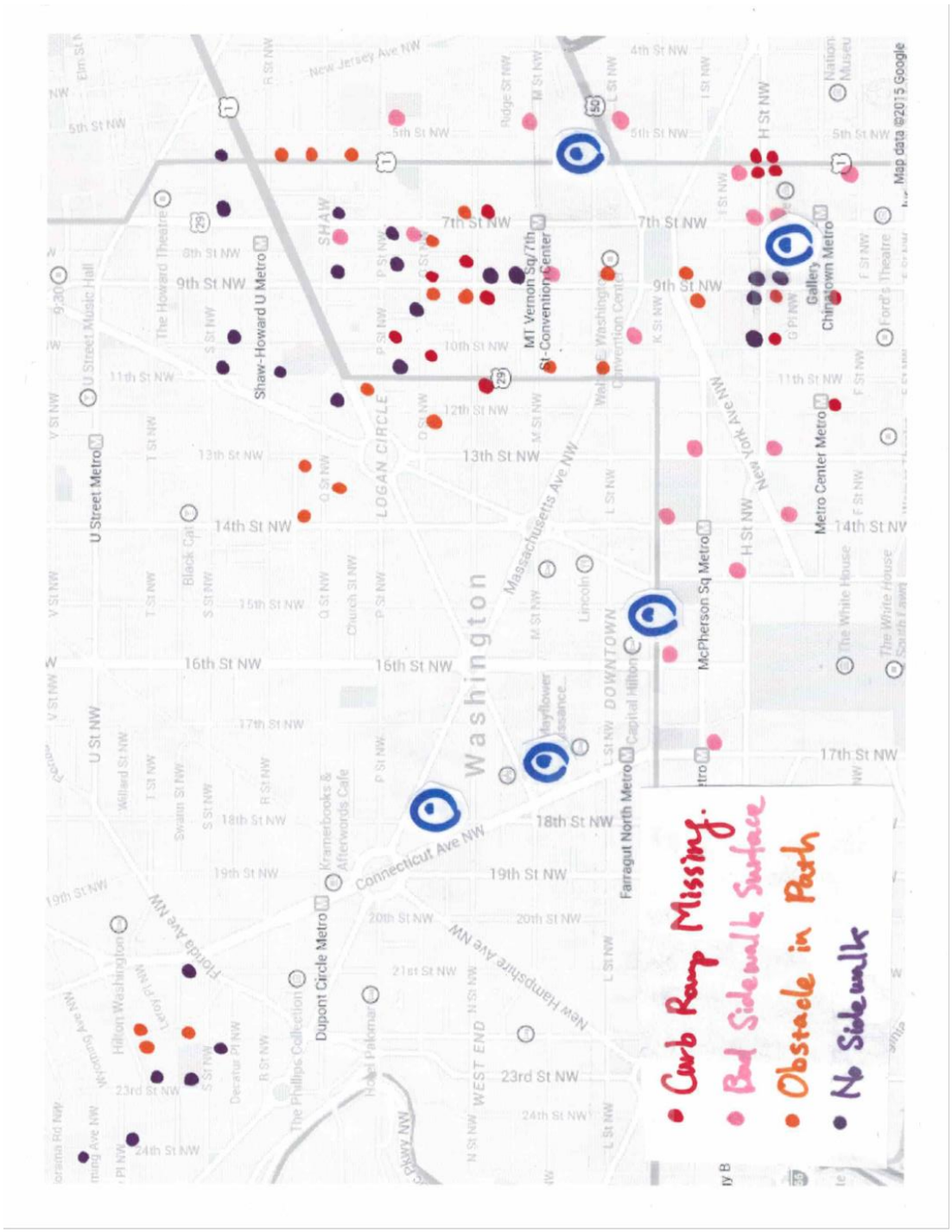


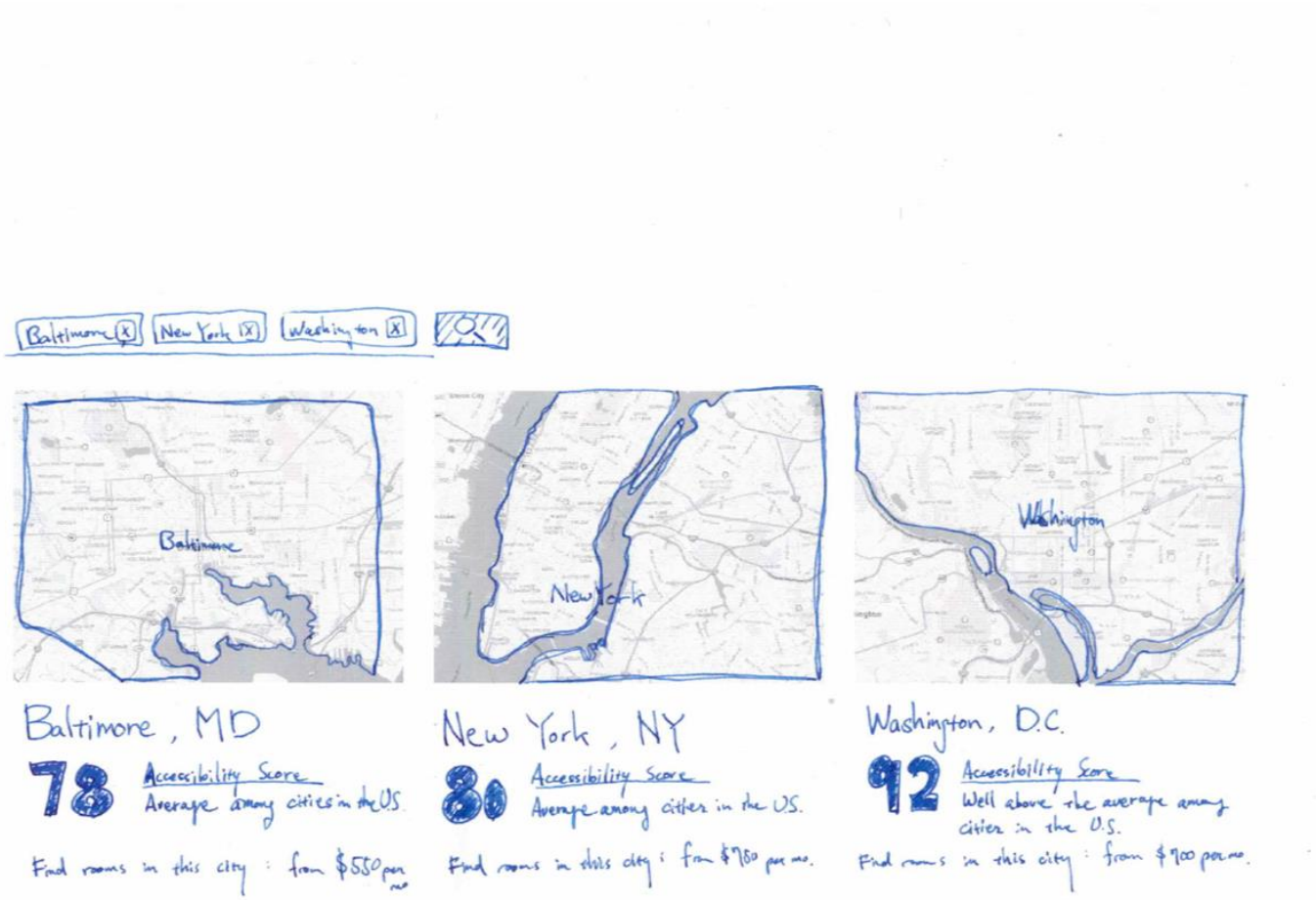
Design Probes




U Street Metro
14th St NW
16th St NW
U St NW
Connecticut Ave NW
23rd St NW
WEST END
McPherson Sq Metro
Metro Center Metro
Farragut West Metro
13th St NW
9th St NW
7th St NW
Map data ©2015 Google


• Accessibility Problem.
Washington
DOWNTOWN
LOGAN CIRCLE
SHAW
WEST END
Connecticut Ave NW
Massachusetts Ave NW
New Hampshire Ave NW
Pennsylvania Ave NW
New York Ave NW
Florida Ave NW
7th St NW
9th St NW
13th St NW
14th St NW
16th St NW
17th St NW
18th St NW
19th St NW
23rd St NW
Shaw-Howard U Metro
U Street Metro
Dupont Circle Metro
Farragut North Metro
Farragut West Metro
McPherson Sq Metro
Metro Center Metro
Gallery Chinatown Metro
MT Vernon Sq/7th St-Convention Center
Map data ©2015 Google


• Curb Ramp Missing.
• Bad Sidewalk Surface
• Obstacle in Path
• No Sidewalk
Washington
DOWNTOWN
LOGAN CIRCLE
SHAW
WEST END
Connecticut Ave NW
Massachusetts Ave NW
New Hampshire Ave NW
Florida Ave NW
New York Ave NW
7th St NW
9th St NW
13th St NW
14th St NW
16th St NW
17th St NW
18th St NW
19th St NW
23rd St NW
U Street Metro
Shaw-Howard U Metro
Mt Vernon Sq/7th St-Convention Center
McPherson Sq Metro
Metro Center Metro
Gallery Chinatown Metro
Dupont Circle Metro
Fairagut North Metro
Map data ©2015 Google


Baltimore
New York
Washington
Baltimore
New York
Washington
Baltimore, MD
78
Accessibility Score
Average among cities in the US
Find rooms in this city : from $550 per mo
New York, NY
80
Accessibility Score
Average among cities in the US.
Find rooms in this city : from $750 per mo.
Washington, D.C.
92
Accessibility Score
Well above the average among cities in the U.S.
Find rooms in this city : from $700 per mo.

Map
1625 Eye Street
704
Accessible Bus Stop
Inaccessible Bus Stop

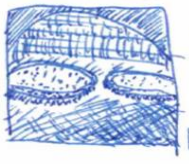
Find Italian Near Washington, D.C

Sort By
Best Match
Highest Rated
Most Reviewed
Most Accessible

Neighborhoods
- ☐ Dupont Circle
- ☐ Georgetown
- ☐ Capitol Hill
- ☐ U Street Corridor

1. Tortino
394 reviews
$$ · Italian

Logan Circle
1228 11th St NW
Washington, DC 20005
(202) 312-5590

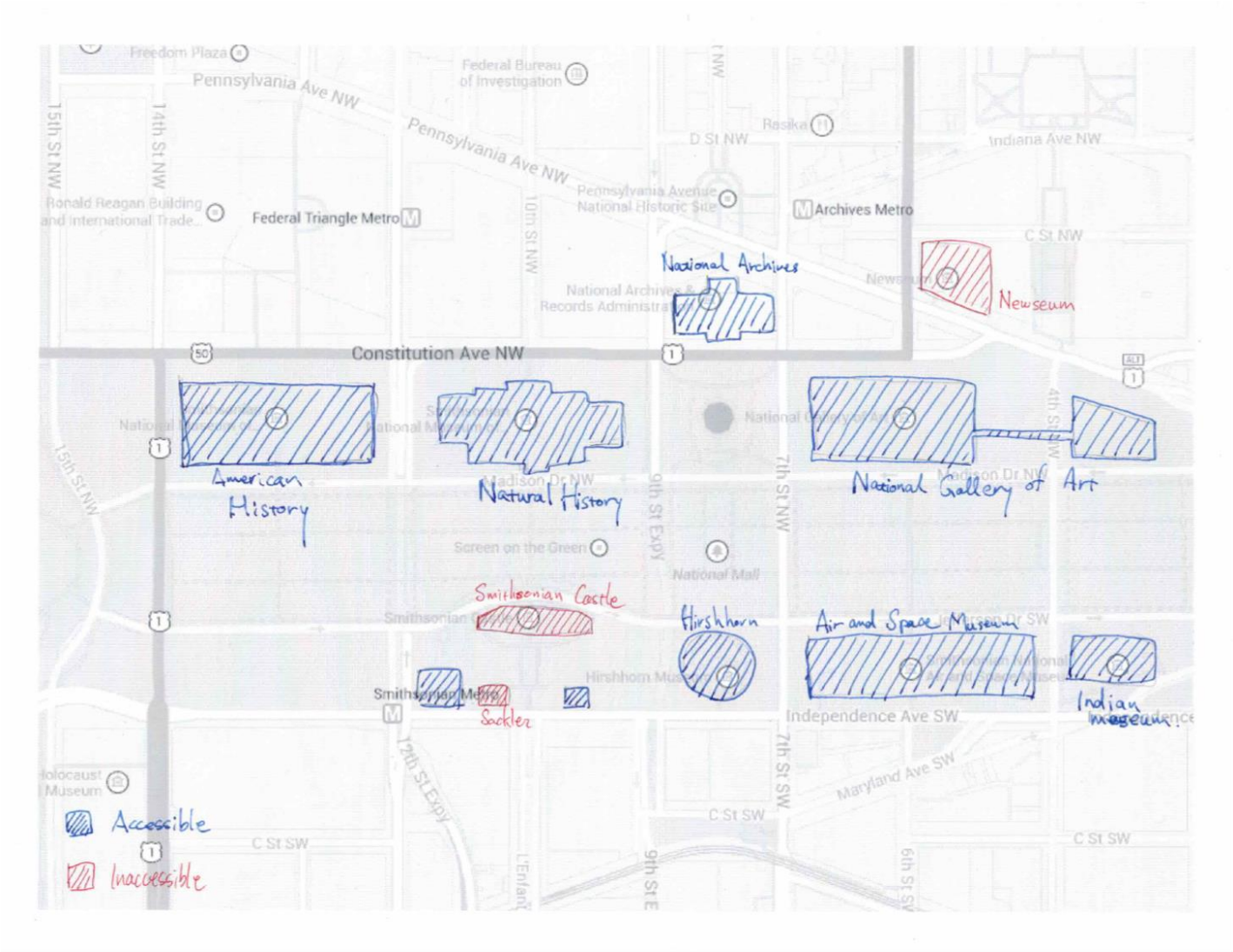


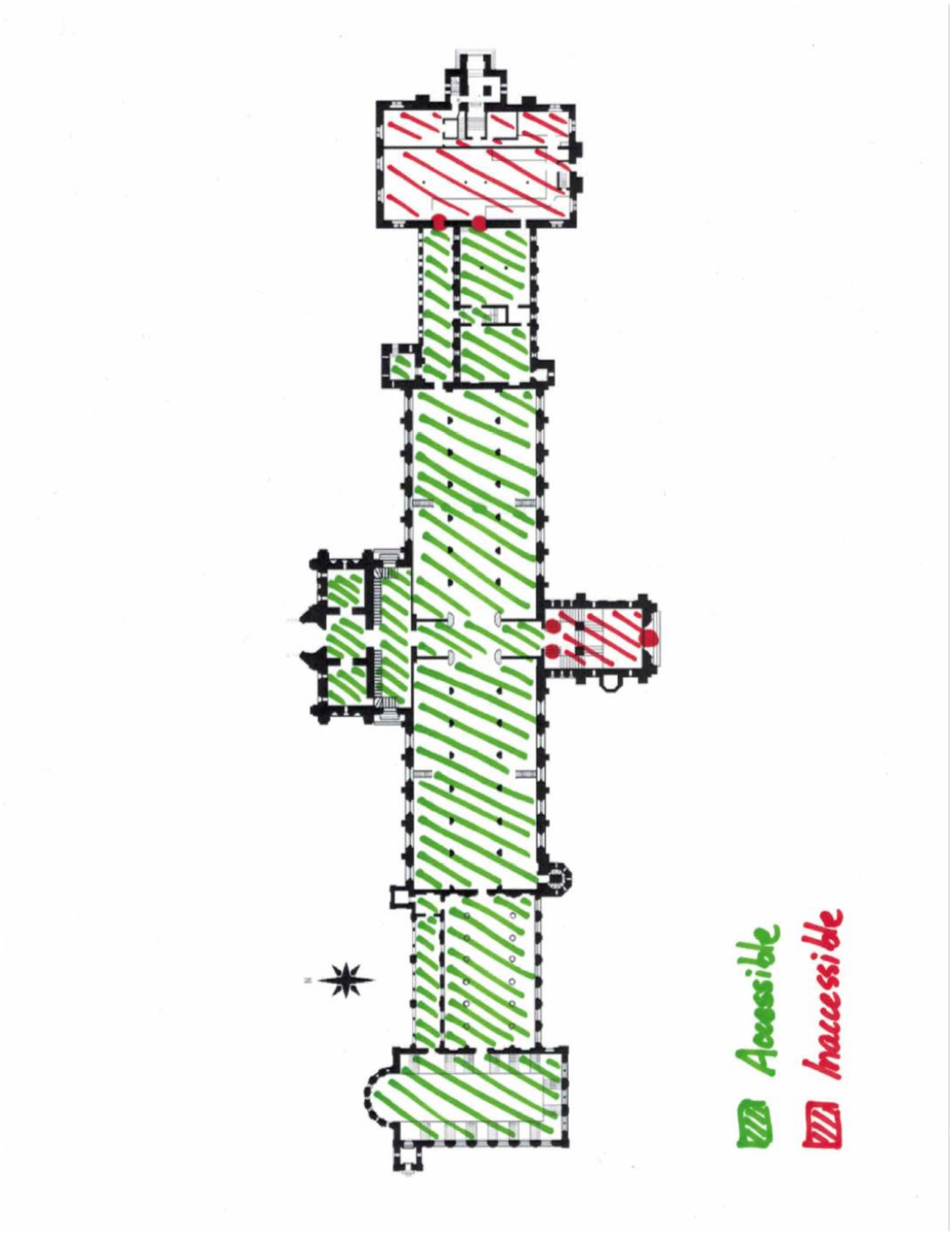
2. il Canale
788 reviews
$$ · Italian, Pizza

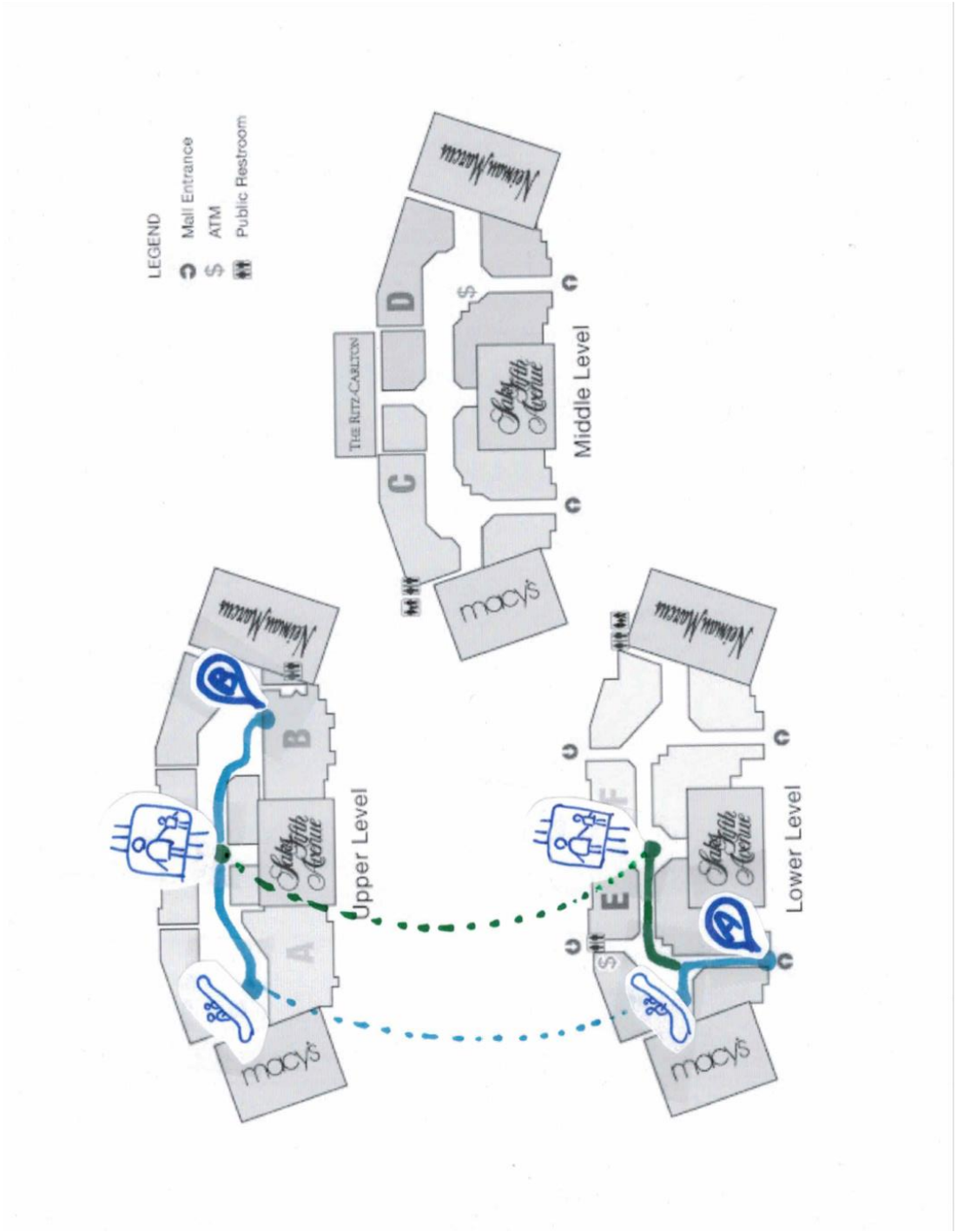
Georgetown
1063 31st St NW
Washington, DC 20007
(202) 337-4444

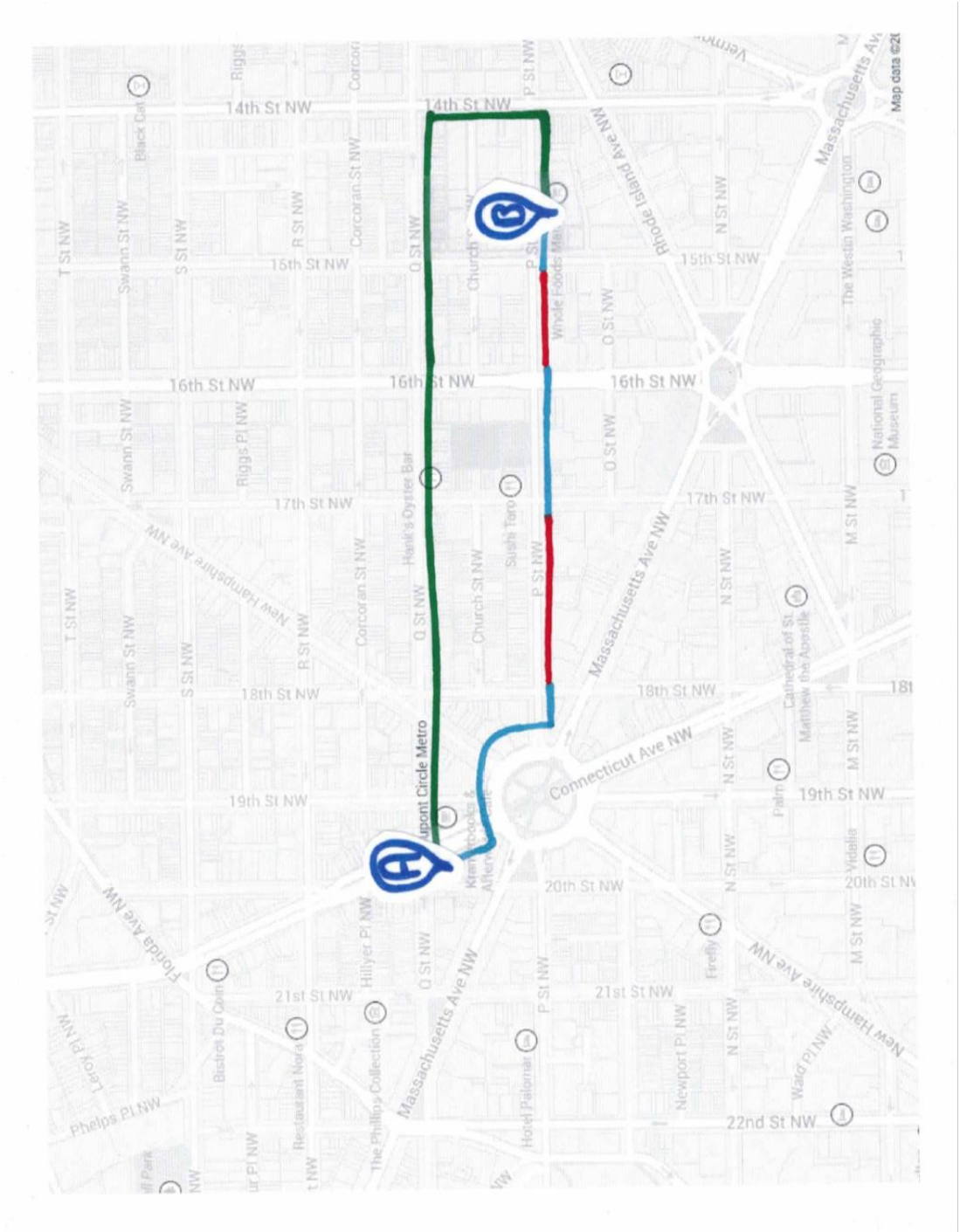
3. Filomena Ristorante
1030 reviews
$$$ · Italian, Buffets

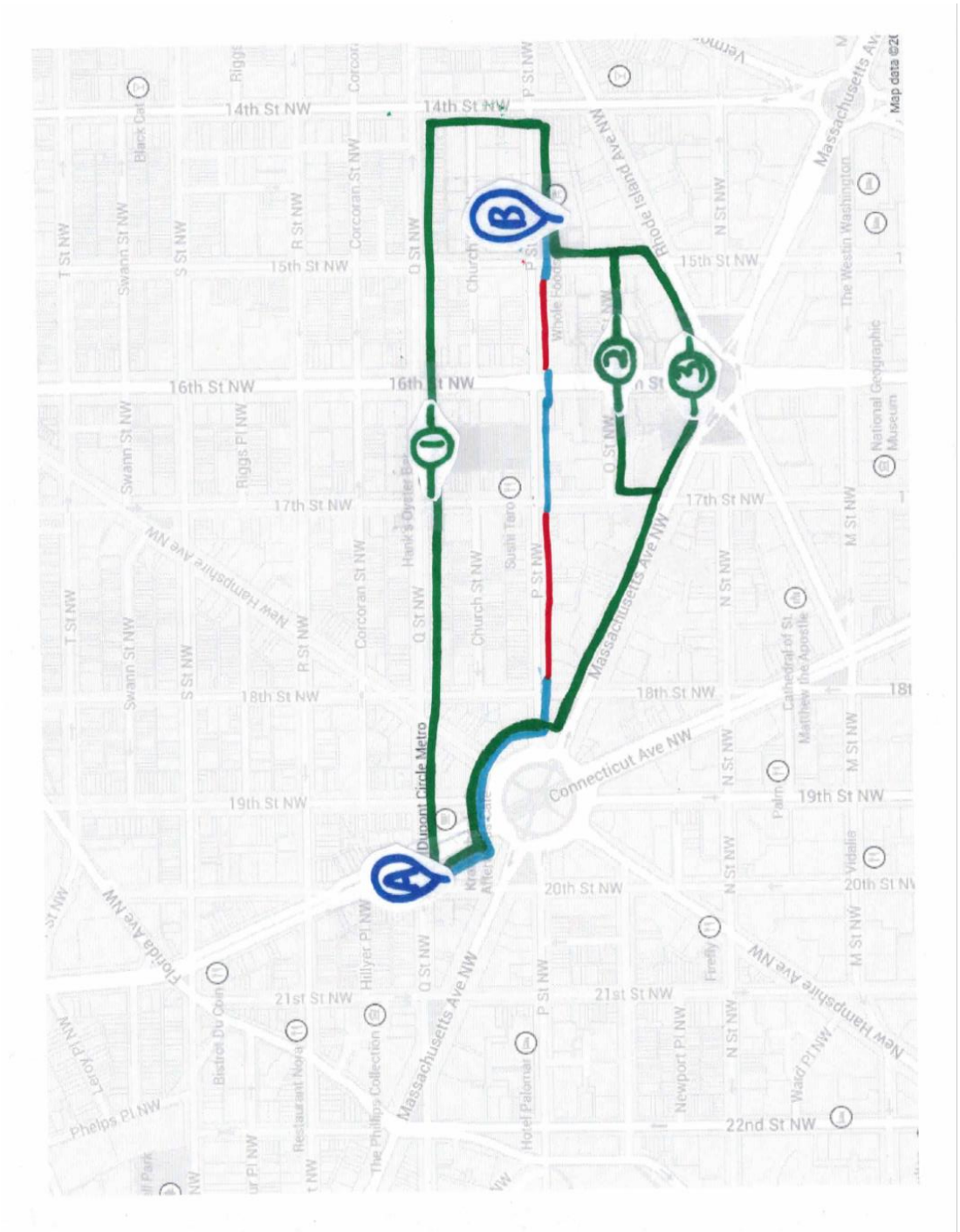
Georgetown
1063 Wisconsin Ave NW
Washington, DC 20007
(202) 338-8800


Freedom Plaza
Pennsylvania Ave NW
Federal Bureau of Investigation
15th St NW
14th St NW
Ronald Reagan Building and International Trade...
Federal Triangle Metro
10th St NW
D St NW
Rasika
Pennsylvania Avenue National Historic Site
Archives Metro
Indiana Ave NW
C St NW
National Archives
Newseum
Constitution Ave NW
American History
Natural History
National Gallery of Art
4th St NW
Madison Dr NW
9th St Expy
7th St NW
Screen on the Green
National Mall
Smithsonian Castle
Hirshhorn
Air and Space Museum
Smithsonian Metro
Sackler
Independence Ave SW
Indian Museum
12th St Expy
Maryland Ave SW
Holocaust Museum
C St SW
L'Enfant
6th St SW
Accessible
Inaccessible


Accessible
Inaccessible


LEGEND
Mall Entrance
ATM
Public Restroom
Middle Level
Upper Level
Lower Level
THE RITZ-CARLTON
macy's
Saks Fifth Avenue
A
B
C
D
E
F


14th St NW
15th St NW
16th St NW
17th St NW
18th St NW
19th St NW
20th St NW
21st St NW
22nd St NW
Massachusetts Ave NW
Connecticut Ave NW
Rhode Island Ave NW
New Hampshire Ave NW
Florida Ave NW
Dupont Circle Metro
Corcoran St NW
Swann St NW
Riggs Pl NW
Hillyer Pl NW
Phelps Pl NW
Leroy Pl NW
Newport Pl NW
Ward Pl NW
Church St NW
O St NW
P St NW
N St NW
M St NW
R St NW
S St NW
T St NW
Hank's Oyster Bar
Sushi Taro
Whole Foods Market
The Westin Washington
National Geographic Museum
Cathedral of St Matthew the Apostle
Hotel Palomar
The Phillips Collection
Bistrot Du Coin
Restaurant Nora
Firefly
Vidalia
Map data ©2


14th St NW
15th St NW
16th St NW
17th St NW
18th St NW
19th St NW
20th St NW
21st St NW
22nd St NW
Massachusetts Ave NW
Connecticut Ave NW
Dupont Circle Metro
Rhode Island Ave NW
New Hampshire Ave NW
Florida Ave NW
Swann St NW
Riggs Pl NW
Corcoran St NW
Church St NW
P St NW
O St NW
N St NW
M St NW
R St NW
S St NW
T St NW
Hillyer Pl NW
Phelps Pl NW
Leroy Pl NW
Newport Pl NW
Waid Pl NW
The Westin Washington
National Geographic Museum
Cathedral of St Matthew the Apostle
Hotel Palomar
Restaurant Nora
Bistrot Du Coin
The Phillips Collection
Sushi Taro
Hank's Oyster Bar
Whole Foods
Firefly
Vidalia
Palm
Map data ©2

Bibliography

1. 3rd Circuit, C. of A. *Kinney v. Yerusalim, 1993 No. 93-1168.* 1993.
2. Agyemang, C., van Hooijdonk, C., Wendel-Vos, W., Lindeman, E., Stronks, K., and Droomers, M. The association of neighbourhood psychosocial stressors and self-rated health in Amsterdam, The Netherlands. *Journal of Epidemiology and Community Health 61*, 12 (2007), 1042–1049.
3. Ahmetovic, D., Manduchi, R., Coughlan, J.M., and Mascetti, S. Zebra Crossing Spotter: Automatic Population of Spatial Databases for Increased Safety of Blind Travelers. *The 17th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS 2015)*, (2015).
4. von Ahn, L. and Dabbish, L. Labeling Images with a Computer Game. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM (2004), 319–326.
5. von Ahn, L. and Dabbish, L. Designing Games with a Purpose. *Commun. ACM 51*, 8 (2008), 58–67.
6. von Ahn, L., Liu, R., and Blum, M. Peekaboom: A Game for Locating Objects in Images. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM (2006), 55–64.
7. Airbnb. ADA and FHA Compliance. https://www.airbnb.com/help/article/898/ada-and-fha-compliance.
8. Alcantarilla, P., Stent, S., Ros, G., Arroyo, R., and Gherardi, R. Street-View

Change Detection with Deconvolutional Networks. *Robotics: Science and Systems (RSS), Michigan, USA*, (2016).

9. Anguelov, D., Dulong, C., Filip, D., et al. Google Street View: Capturing the World at Street Level. *Computer 43*, 6 (2010), 32–38.
10. Baba, Y. and Kashima, H. Statistical Quality Estimation for General Crowdsourcing Tasks. *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM (2013), 554–562.
11. Badland, H.M., Opit, S., Witten, K., Kearns, R.A., and Mavoa, S. Can virtual streetscape audits reliably replace physical streetscape audits? *Journal of urban health : bulletin of the New York Academy of Medicine 87*, 6 (2010), 1007–16.
12. Bakhshi, S., Kanuparthy, P., and Shamma, D.A. Understanding Online Reviews: Funny, Cool or Useful? *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, ACM (2015), 1270–1276.
13. Batty, M., Hudson-Smith, A., Milton, R., and Crooks, A. Map mashups, Web 2.0 and the GIS revolution. *Annals of GIS 16*, 1 (2010), 1–13.
14. Beale, L., Field, K., Briggs, D., Picton, P., and Matthews, H. Mapping for Wheelchair Users: Route Navigation in Urban Spaces. *Cartographic Journal, The 43*, 1 (2006), 68–81.
15. Ben-Joseph, E., Lee, J.S., Cromley, E.K., Laden, F., and Troped, P.J. Virtual and actual: Relative accuracy of on-site and web-based instruments in auditing the environment for physical activity. *Health & Place 19*, 0 (2013), 138–150.
16. Bernstein, M.S., Little, G., Miller, R.C., et al. Soylent: A Word Processor with

a Crowd Inside. *Proceedings of the 23Nd Annual ACM Symposium on User Interface Software and Technology*, ACM (2010), 313–322.

17. Boehmer, T.K., Hoehner, C.M., Deshpande, A.D., Brennan Ramirez, L.K., and Brownson, R.C. Perceived and observed neighborhood indicators of obesity among urban adults. *Int J Obes 31*, 6 (2007), 968–977.

18. Bragg, J., Kolobov, A., and Weld, D.S. Parallel Task Routing for Crowdsourcing. *Proc. of HCOMP '14*, (2014).

19. Branson, S., Wah, C., Babenko, B., et al. Visual Recognition with Humans in the Loop. *European Conference on Computer Vision (ECCV)*, (2010).

20. Braun, V. and Clarke, V. Using thematic analysis in psychology. *Qualitative Research in Psychology 3*, 2 (2006), 77–101.

21. Bromley, R.D.F., Matthews, D.L., and Thomas, C.J. City centre accessibility for wheelchair users: The consumer perspective and the planning implications. *Cities 24*, 3 (2007), 229–241.

22. Bureau, U.S.C. QuickFacts District of Columbia. 2015.

23. Burges, C.C. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery 2*, 2 (1998), 121–167.

24. Butler, H., Daly, M., Doyle, A., Gillies, S., Schaub, T., and Schmidt, C. The GeoJSON Format Specification. http://geojson.org/geojson-spec.html.

25. Cabral, R. and Furukawa, Y. Piecewise Planar and Compact Floorplan Reconstruction from Images. *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society (2014),

628–635.

26. Cardinal, B.J. and Spaziani, M.D. ADA Compliance and the Accessibility of Physical Activity Facilities in Western Oregon. *American Journal of Health Promotion 17*, 3 (2003), 197–201.

27. Cardonha, C., Gallo, D., Avegliano, P., Herrmann, R., Koch, F., and Borger, S. A Crowdsourcing Platform for the Construction of Accessibility Maps. *Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility*, ACM (2013), 26:1--26:4.

28. Carr, L.J., Dunsiger, S.I., and Marcus, B.H. Walk score™ as a global estimate of neighborhood walkability. *American journal of preventive medicine 39*, 5 (2010), 460–463.

29. Caughy, M.O., O'Campo, P.J., and Patterson, J. A brief observational measure for urban neighborhoods. *Health & Place 7*, 3 (2001), 225–236.

30. Cervero, R. Mixed land-uses and commuting: Evidence from the American Housing Survey. *Transportation Research Part A: Policy and Practice 30*, 5 (1996), 361–377.

31. Chen, D., Bilgic, M., Getoor, L., and Jacobs, D. Dynamic Processing Allocation in Video. *IEEE Transactions on Pattern Analysis and Machine Intelligence 33*, (2011), 2174–2187.

32. Cheng, C., Koschan, A., Chen, C.-H., Page, D.L., and Abidi, M.A. Outdoor Scene Image Segmentation Based on Background Recognition and Perceptual Organization. *IEEE Transactions on Image Processing 21*, 3 (2012), 1007–

1019.

33. Cheng, J., Teevan, J., Iqbal, S.T., and Bernstein, M.S. Break It Down: A Comparison of Macro- and Microtasks. *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, ACM (2015), 4061–4064.

34. Chin Jr., G., Rosson, M.B., and Carroll, J.M. Participatory Analysis: Shared Development of Requirements from Scenarios. *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, ACM (1997), 162–169.

35. Christie, M., Olivier, P., and Normand, J.-M. Camera control in computer graphics. *Computer Graphics Forum*, (2008), 2197–2218.

36. Chu, H., Wang, S., Urtasun, R., and Fidler, S. HouseCraft: Building Houses from Rental Ads and Street Views. *European Conference on Computer Vision*, (2016), 500–516.

37. Church, R.L. and Marston, J.R. Measuring Accessibility for People with a Disability. *Geographical Analysis 35*, 1 (2003), 83–96.

38. Cicchetti, D. V and Fleiss, J.L. Comparison of the Null Distributions of Weighted Kappa and the C Ordinal Statistic. *Applied Psychological Measurement 1*, 2 (1977), 195–201.

39. Clarke, P., Ailshire, J., Melendez, R., Bader, M., and Morenoff, J. Using Google Earth to conduct a neighborhood audit: reliability of a virtual audit instrument. *Health & place 16*, 6 (2010), 1224–9.

40. Coetzee, D., Lim, S., Fox, A., Hartmann, B., and Hearst, M.A. Structuring Interactions for Large-Scale Synchronous Peer Learning. *Proc. of CSCW 2015*, (2015).

41. Cohen, D.A., Farley, T.A., and Mason, K. Why is poverty unhealthy? Social and physical mediators. *Social Science & Medicine 57*, 9 (2003), 1631–1641.

42. Cohen, J. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement 20*, 1 (1960), 37–46.

43. Conrad, E. Tulsans Hope Sidewalk Improvements Make City More Accessible (Viewed Nov 2014). *Oklahoma's Own*, 2014. http://www.newson6.com/story/26407524/tulsans-hope-sidewalk-improvements-make-city-more-accessible.

44. Cortright, J. Walking the Walk: How Walkability Raises Home Values in U.S. Cities OR - CEOs for Cities. 2009. citeulike-article-id:5541951.

45. Cox, J., Oh, E.Y., Simmons, B., et al. Defining and Measuring Success in Online Citizen Science: A Case Study of Zooniverse Projects. *Computing in Science & Engineering 17*, 4 (2015).

46. Dai, P., Daniel, M., and Weld, S. Decision-theoretic control of crowd-sourced workflows. *In the 24th AAAI Conference on Artificial Intelligence (AAAI'10*, (2010).

47. Dai, P., Mausam, and Weld, D.S. Artificial Intelligence for Artificial Artificial Intelligence. *AAAI*, AAAI Press (2011).

48. Dalal, N. and Triggs, B. Histograms of oriented gradients for human detection.

IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005., (2005), 886–893 vol. 1.

49. DC, D. *2014 Visitor Statistics Washington, D.C.* .

50. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. *CVPR09*, (2009).

51. Deng, J., Russakovsky, O., Krause, J., Bernstein, M., Berg, A., and Fei-Fei, L. Scalable Multi-label Annotation. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*, (2014), 3099–3102.

52. Ding, C., Wald, M., and Wills, G. A Survey of Open Accessibility Data. *Proceedings of the 11th Web for All Conference*, ACM (2014), 37:1--37:4.

53. Ding, D., Parmanto, B., Karimi, H.A., et al. Design Considerations for a Personalized Wheelchair Navigation System. *29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2007*, (2007), 4790–4793.

54. District of Columbia Open Data. Washington, D.C. Census Tracts 2010. http://opendata.dc.gov/datasets/6969dd63c5cb4d6aa32f15effb8311f3_8.

55. Dow, S., Kulkarni, A., Klemmer, S., and Hartmann, B. Shepherding the Crowd Yields Better Work. *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*, ACM (2012), 1013–1022.

56. Duncan, D.T., Aldstadt, J., Whalen, J., Melly, S.J., and Gortmaker, S.L. Validation of Walk Score for estimating neighborhood walkability: An analysis of four US metropolitan areas. *International journal of environmental research*

and public health 8, 5 (2011), 4160–4179.

57. Efron, B. and Tibshirani, R.J. *An introduction to the bootstrap*. CRC press, 1994.

58. Elwood, S. Geographic Information Science: Visualization, visual methods, and the geoweb. *Progress in Human Geography 35*, 3 (2011), 401–408.

59. Embiricos, A., Rahmati, N., Zhu, N., and Bernstein, M.S. Structured Handoffs in Expert Crowdsourcing Improve Communication and Work Output. *Proceedings of the Adjunct Publication of the 27th Annual ACM Symposium on User Interface Software and Technology*, ACM (2014), 99–100.

60. Everingham, M., Van~Gool, L., Williams, C.K.I., Winn, J., and Zisserman, A. The {PASCAL} {V}isual {O}bject {C}lasses {C}hallenge 2012 {(VOC2012)} {R}esults. .

61. Everingham, M., Van~Gool, L., Williams, C.K.I., Winn, J., and Zisserman, A. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision 88*, 2 (2010), 303–338.

62. Felzenszwalb, P., McAllester, D., and Ramaman, D. A Discriminatively Trained, Multiscale, Deformable Part Model. *IEEE Conference on Computer Vision and Pattern Recognition*, (2008).

63. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., and Ramanan, D. Object Detection with Discriminatively Trained Part-Based Models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 32*, 9 (2010), 1627–1645.

64. Ferrari, V., Fevrier, L., Jurie, F., and Schmid, C. Groups of Adjacent Contour Segments for Object Detection. *Pattern Analysis and Machine Intelligence,*

IEEE Transactions on 30, 1 (2008), 36–51.

65. Flegenheimer, M. Suit Seeks to Make Sidewalks More Accessible for Disabled New Yorkers (Viewed Nov 2014). *New York Times*, 2014. http://www.nytimes.com/2014/07/31/nyregion/group-sues-new-york-city-for-upgrades-to-sidewalks.html.

66. Frank, L.D., Sallis, J.F., Saelens, B.E., et al. The development of a walkability index: application to the Neighborhood Quality of Life Study. *British Journal of Sports Medicine 44*, 13 (2010), 924–933.

67. Girshick, R.B., Felzenszwalb, P.F., and McAllester, D. Discriminatively Trained Deformable Part Models, Release 5. .

68. Goh, D.-L., Sepoetro, L., Qi, M., et al. Mobile Tagging and Accessibility Information Sharing Using a Geospatial Digital Library. In *Asian Digital Libraries. Looking Back 10 Years and Forging New Frontiers SE - 38*. Springer Berlin Heidelberg, 2007, 287–296.

69. Goh, D.-L., Sepoetro, L., Qi, M., et al. Mobile Tagging and Accessibility Information Sharing Using a Geospatial Digital Library. In D.-L. Goh, T. Cao, I. Sølvberg and E. Rasmussen, eds., *Asian Digital Libraries. Looking Back 10 Years and Forging New Frontiers SE - 38*. Springer Berlin Heidelberg, 2007, 287–296.

70. Goodchild, M.F. and Li, L. Assuring the quality of volunteered geographic information. *Spatial Statistics 1*, 0 (2012), 110–120.

71. Gray, D.B., Gould, M., and Bickenbach, J.E. Environmental barriers and

disability. *Journal of architectural and planning research 20*, 1 (2003), 29–37.

72. Gurari, D., Jain, S., Grauman, K., and Betke, M. Pull the Plug? Predicting If Computers or Humans Should Segment Images. *IEEE International Conference on Computer Vision (ICCV)*, (2016).

73. Guy, R. and Truong, K. CrossingGuard: exploring information content in navigation aids for visually impaired pedestrians. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*, ACM (2012), 405–414.

74. Haklay, M. How good is volunteered geographical information? A comparative study of OpenStreetMap and Ordnance Survey datasets. *Environment and Planning B: Planning and Design 37*, 4 (2010), 682–703.

75. Haklay, M., Basiouka, S., Antoniou, V., and Ather, A. How many volunteers does it take to map an area well? The validity of Linus' law to volunteered geographic information. *The Cartographic Journal 47*, 4 (2010), 315–322.

76. Haklay, M. and Weber, P. OpenStreetMap: User-Generated Street Maps. *Pervasive Computing, IEEE 7*, 4 (2008), 12–18.

77. Hara, K., Azenkot, S., Campbell, M., et al. Improving Public Transit Accessibility for Blind Riders by Crowdsourcing Bus Stop Landmark Locations with Google Street View. *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility Technology*, (2013), 16:1-16:8.

78. Hara, K., Azenkot, S., Campbell, M., et al. Improving Public Transit

Accessibility for Blind Riders by Crowdsourcing Bus Stop Landmark Locations with Google Street View: An Extended Analysis. *ACM Trans. Access. Comput. 6*, 2 (2015), 5:1--5:23.

79. Hara, K., Chan, C., and Froehlich, J.E. The Design of Assistive Location-based Technologies for People with Ambulatory Disabilities: A Formative Study. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, (2016), 1757–1768.

80. Hara, K., Le, V., and Froehlich, J. A Feasibility Study of Crowdsourcing and Google Street View to Determine Sidewalk Accessibility. *Proceedings of the 14th international ACM SIGACCESS conference on Computers and accessibility (ASSETS '12), Poster Session*, ACM (2012), 273–274.

81. Hara, K., Le, V., and Froehlich, J. Combining Crowdsourcing and Google Street View to Identify Street-level Accessibility Problems. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM (2013), 631–640.

82. Hara, K., Le, V., Sun, J., Jacobs, D., and Froehlich, J. Exploring Early Solutions for Automatically Identifying Inaccessible Sidewalks in the Physical World Using Google Street View. *Human Computer Interaction Consortium*, (2013).

83. Hara, K., Sun, J., Chazan, J., Jacobs, D., and Froehlich, J. An Initial Study of Automatic Curb Ramp Detection with Crowdsourced Verification using Google Street View Images. *Proceedings of the 1st Conference on Human Computation and Crowdsourcing (HCOMP'13), Work-in-Progress*, (2013).

84. Hara, K., Sun, J., Jacobs, D.W., and Froehlich, J.E. Tohme: Detecting Curb Ramps in Google Street View Using Crowdsourcing, Computer Vision, and Machine Learning. *Proceedings of the 27th annual ACM symposium on User interface software and technology*, (2014).

85. Hicks, G.C. *But It's Your Sidewalk! Sidewalk Repair and Liability*. 2014.

86. Hoehner, C.M., Brennan Ramirez, L.K., Elliott, M.B., Handy, S.L., and Brownson, R.C. Perceived and objective environmental measures and physical activity among urban adults. *American Journal of Preventive Medicine 28*, 2, Supplement 2 (2005), 105–116.

87. Hoiem, D., Efros, A., and Hebert, M. Putting Objects in Perspective. *International Journal of Computer Vision 80*, 1 (2008), 3–15.

88. Holone, H., Misund, G., and Holmstedt, H. Users Are Doing It For Themselves: Pedestrian Navigation With User Generated Content. *The 2007 International Conference on Next Generation Mobile Applications, Services and Technologies (NGMAST 2007)*, (2007), 91–99.

89. Hruschka, D.J., Schwartz, D., St.John, D.C., Picone-Decaro, E., Jenkins, R.A., and Carey, J.W. Reliability in Coding Open-Ended Data: Lessons Learned from HIV Behavioral Research. *Field Methods 16*, 3 (2004), 307–331.

90. Ikehata, S., Yang, H., and Furukawa, Y. Structured Indoor Modeling. *2015 IEEE International Conference on Computer Vision (ICCV)*, (2015), 1323–1331.

91. Imrie, R. and Kumar, M. Focusing on Disability and Access in the Built Environment. *Disability & Society 13*, 3 (1998), 357–374.

92. Ipeirotis, P.G., Provost, F., and Wang, J. Quality Management on Amazon Mechanical Turk. *Proceedings of the ACM SIGKDD Workshop on Human Computation*, ACM (2010), 64–67.

93. Iwasawa, Y., Nagamine, K., Matsuo, Y., and Eguchi Yairi, I. Road Sensing: Personal Sensing and Machine Learning for Development of Large Scale Accessibility Map. *Proceedings of the 17th International ACM SIGACCESS Conference on Computers & Accessibility*, ACM (2015), 335–336.

94. Iwasawa, Y. and Yairi, I.E. Life-Logging of Wheelchair Driving on Web Maps for Visualizing Potential Accidents and Incidents. In P. Anthony, M. Ishizuka and D. Lukose, eds., *PRICAI 2012: Trends in Artificial Intelligence: 12th Pacific Rim International Conference on Artificial Intelligence, Kuching, Malaysia, September 3-7, 2012. Proceedings*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, 157–169.

95. Jain, S.D. and Grauman, K. Predicting Sufficient Annotation Strength for Interactive Foreground Segmentation. *ICCV*, (2013), 1313–1320.

96. Jakobsson, U. and Westergren, A. Statistical methods for assessing agreement for ordinal data. *Scandinavian journal of caring sciences 19*, 4 (2005), 427–31.

97. Josephy, T., Lease, M., Paritosh, P., et al. Workshops held at the first AAAI conference on human computation and crowdsourcing: A report. *AI Magazine 35*, 2 (2014), 75–78.

98. Jung, J.-K. and Elwood, S. Extending the Qualitative Capabilities of GIS: Computer-Aided Qualitative GIS. *Transactions in GIS 14*, 1 (2010), 63–87.

99. Kamar, E., Hacker, S., and Horvitz, E. Combining Human and Machine Intelligence in Large-scale Crowdsourcing. *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*, International Foundation for Autonomous Agents and Multiagent Systems (2012), 467–474.

100. Karger, D.R., Oh, S., and Shah, D. Budget-Optimal Task Allocation for Reliable Crowdsourcing Systems. *Operations Research 62*, 1 (2014), 1–24.

101. Kasemsuppakorn, P. and Karimi, H.A. Pedestrian Network Data Collection through Location-Based Social Networks. *Sciences-New York*, .

102. Kirschbaum, J.B., Axelson, P.W., Longmuir, P.E., Mispagel, K.M., Stein, J.A., and Yamada, D.A. *Designing Sidewalks and Trails for Access, Part II of II: Best Practices Design Guide, Chapter 7*. 2001.

103. Kittur, A., Nickerson, J. V, Bernstein, M., et al. The Future of Crowd Work. *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*, ACM (2013), 1301–1318.

104. Kittur, A., Smus, B., Khamkar, S., and Kraut, R.E. CrowdForge: Crowdsourcing Complex Work. *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, ACM (2011), 43–52.

105. Koester, D., Lunt, B., and Stiefelhagen, R. Zebra Crossing Detection from Aerial Imagery Across Countries. *ICCHP*, (2016).

106. Kopf, J., Chen, B., Szeliski, R., and Cohen, M. Street Slide: Browsing Street Level Imagery. *ACM Transactions on Graphics (Proceedings of SIGGRAPH*

2010) 29, 4 (2010), 96:1-- 96:8.

107. Krippendorff, K.H. *Content Analysis: An Introduction to Its Methodology*. Sage Publications, Inc, 2003.

108. Krishna, R.A., Hata, K., Chen, S., et al. Embracing Error to Enable Rapid Crowdsourcing. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, ACM (2016), 3167–3179.

109. Landay, J.A. and Myers, B.A. Sketching interfaces: toward more human interface design. *Computer 34*, 3 (2001), 56–64.

110. Landis, J.R. and Koch, G.G. The Measurement of Observer Agreement for Categorical Data. *Biometrics 33*, 1 (1977), 159–174.

111. Lin, C.H., Daniel, M., and Weld, S. Dynamically switching between synergistic workflows for crowdsourcing. *In Proceedings of the 26th AAAI Conference on Artificial Intelligence, AAAI '12*, (2012).

112. Lin, T.Y., Cui, Y., Belongie, S., and Hays, J. Learning deep representations for ground-to-aerial geolocalization. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2015), 5007–5015.

113. Lin, T.Y., Cui, Y., Belongie, S., and Hays, J. Learning deep representations for ground-to-aerial geolocalization. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2015), 5007–5015.

114. Lintott, C.J., Schawinski, K., Slosar, A., et al. Galaxy Zoo: morphologies derived from visual inspection of galaxies from the Sloan Digital Sky Survey. *Monthly Notices of the Royal Astronomical Society 389*, 3 (2008), 1179–1189.

115. Liu, C., Schwing, A.G., Kundu, K., Urtasun, R., and Fidler, S. Rent3d: Floor-plan priors for monocular layout estimation. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2015), 3413–3421.

116. Locke, E.A., Shaw, K.N., Saari, L.M., and Latham, G.P. Goal setting and task performance: 1969--1980. *Psychological bulletin 90*, 1 (1981), 125.

117. Loukaitou-Sideris, A. Is it safe to walk? 1 neighborhood safety and security considerations and their effects on walking. *Journal of Planning Literature 20*, 3 (2006), 219–232.

118. Lucero, A., Holopainen, J., Ollila, E., Suomela, R., and Karapanos, E. The Playful Experiences (PLEX) Framework As a Guide for Expert Evaluation. *Proceedings of the 6th International Conference on Designing Pleasurable Products and Interfaces*, ACM (2013), 221–230.

119. Mackett, R.L., Achuthan, K., and Titheridge, H. AMELIA: A tool to make transport policies more socially inclusive. *Transport Policy 15*, 6 (2008), 372–378.

120. Malisiewicz, T., Gupta, A., and Efros, A.A. Ensemble of Exemplar-SVMs for Object Detection and Beyond. *ICCV*, (2011).

121. Marcus, A. and Parameswaran, A. Crowdsourced Data Management: Industry and Academic Perspectives. *Foundations and Trends® in Databases 6*, 1–2 (2015), 1–161.

122. Markesino, J. and Barlow, J. *Special Report: Accessible Public Rights-of-Way Planning and Designing for Alterations*. 2007.

123. Martin, D., Fowlkes, C., Tal, D., and Malik, J. A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics. July (2001).

124. Mashhadi, A., Quattrone, G., and Capra, L. Putting Ubiquitous Crowd-sourcing into Context. *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*, ACM (2013), 611–622.

125. Masli, M., Priedhorsky, R., and Terveen, L.G. Task Specialization in Social Production Communities: The Case of Geographic Volunteer Work. *ICWSM*, (2011).

126. Matejka, J., Grossman, T., and Fitzmaurice, G. Swifter: Improved Online Video Scrubbing. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM (2013), 1159–1168.

127. Matthews, H., Beale, L., Picton, P., and Briggs, D. Modelling Access with GIS in Urban Systems (MAGUS): capturing the experiences of wheelchair users. *Area 35*, 1 (2003), 34–45.

128. Menkens, C., Sussmann, J., Al-Ali, M., et al. EasyWheel - A Mobile Social Navigation and Support System for Wheelchair Users. *Eighth International Conference on Information Technology: New Generations (ITNG), 2011*, (2011), 859–866.

129. Meyers, A.R., Anderson, J.J., Miller, D.R., Shipp, K., and Hoenig, H. Barriers, facilitators, and access for wheelchair users: substantive and methodologic lessons from a pilot study of environmental effects. *Social Science & Medicine*

55, 8 (2002), 1435–1446.

130. Miller, G. Superpowering Runkeeper's 1.5 Million Walks, Runs, and Bike Rides, https://www.mapbox.com/blog/runkeeper-million-routes/. 2014.

131. Mugar, G., \Osterlund, C., Hassman, K.D., Crowston, K., and Jackson, C.B. Planet Hunters and Seafloor Explorers: Legitimate Peripheral Participation Through Practice Proxies in Online Citizen Science. *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing*, ACM (2014), 109–119.

132. National Council on Disability. *The Impact of the Americans with Disabilities Act: Assessing the Progress Toward Achieving the Goals of the ADA*. Washington, DC, USA, 2007.

133. Neis, P. and Zipf, A. Analyzing the contributor activity of a volunteered geographic information project—The case of OpenStreetMap. *ISPRS International Journal of Geo-Information 1*, 2 (2012), 146–165.

134. Nov, O., Arazy, O., and Anderson, D. Scientists@Home: What Drives the Quantity and Quality of Online Citizen Science Participation? *PLoS ONE 9*, 4 (2014), e90375.

135. Nuernberger, A. Presenting Accessibility to Mobility-Impaired Travelers (Ph.D. Thesis). 2008.

136. OpenStreetMap.org. OpenStreetMap, Key:highway. .

137. Palazzi, C.E., Teodori, L., and Roccetti, M. Path 2.0: A participatory system for the generation of accessible routes. *Multimedia and Expo (ICME), 2010 IEEE*

International Conference on, (2010), 1707–1711.

138. Parrott, R. and Stutz, F.P. Urban GIS applications. *Geographical information systems 2*, (1991), 247–260.

139. Powers, D.M.W. The Problem with Kappa. *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, Association for Computational Linguistics (2012), 345–355.

140. Priedhorsky, R., Jordan, B., and Terveen, L. How a Personalized Geowiki Can Help Bicyclists Share Information More Effectively. *Proceedings of the 2007 International Symposium on Wikis*, ACM (2007), 93–98.

141. Priedhorsky, R., Masli, M., and Terveen, L. Eliciting and Focusing Geographic Volunteer Work. *Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work*, ACM (2010), 61–70.

142. Priedhorsky, R. and Terveen, L. The Computational Geowiki: What, Why, and How. *Proceedings of the 2008 ACM Conference on Computer Supported Cooperative Work*, ACM (2008), 267–276.

143. Purciel, M., Neckerman, K.M., Lovasi, G.S., et al. Creating and validating GIS measures of urban design for health research. *Journal of Environmental Psychology 29*, 4 (2009), 457–466.

144. Quade, P.B. and Douglas, I. The Pedestrian Environment (Vol. 4A). *Portland, OR 1000*, (1993).

145. Quattrone, G., Mashhadi, A., Quercia, D., Smith-Clarke, C., and Capra, L. Modelling Growth of Urban Crowd-sourced Information. *Proceedings of the 7th*

ACM International Conference on Web Search and Data Mining, ACM (2014), 563–572.

146. Quinn, A.J. and Bederson, B.B. Human Computation: A Survey and Taxonomy of a Growing Field. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM (2011), 1403–1412.

147. Raper, J., Gartner, G., Karimi, H., and Rizos, C. Applications of Location-based Services: A Selected Review. *J. Locat. Based Serv. 1*, 2 (2007), 89–111.

148. Rashid, O., Dunabr, A., Fisher, S., and Rutherford, J. Users Helping Users: User Generated Content to Assist Wheelchair Users in an Urban Environment. *2010 Ninth International Conference on Mobile Business and 2010 Ninth Global Mobility Roundtable (ICMB-GMR)*, (2010), 213–219.

149. Raykar, V.C., Yu, S., Zhao, L.H., et al. Supervised Learning from Multiple Experts: Whom to Trust when Everyone Lies a Bit. *Proceedings of the 26th Annual International Conference on Machine Learning*, ACM (2009), 889–896.

150. Reference, S. EPSG:26918. http://spatialreference.org/ref/epsg/nad83-utm-zone-18n/.

151. Rettig, M. Prototyping for Tiny Fingers. *Commun. ACM 37*, 4 (1994), 21–27.

152. Rimmer, J.H., Riley, B., Wang, E., Rauworth, A., and Jurkowski, J. Physical activity participation among persons with disabilities: Barriers and facilitators. *American Journal of Preventive Medicine 26*, 5 (2004), 419–425.

153. Rogstadius, J., Kostakos, V., Kittur, A., Smus, B., Laredo, J., and Vukovic, M. An Assessment of Intrinsic and Extrinsic Motivation on Task Performance in

Crowdsourcing Markets. *ICWSM*, (2011).

154. Ross, C.L. and Dunning, A.E. *Land use transportation interaction: An examination of the 1995 NPTS data*. Georgia Institute of Technology, 1997.

155. Rosson, M.B. and Carroll, J.M. Scenario-Based Design. *Human-computer interaction. Boca Raton, FL*, (2009), 145–162.

156. Roux, A.V.D., Evenson, K.R., McGinn, A.P., et al. Availability of Recreational Resources and Physical Activity in Adults. *American Journal of Public Health 97*, 3 (2007), 493–499.

157. Rundle, A.G., Bader, M.D.M., Richards, C.A., Neckerman, K.M., and Teitler, J.O. Using Google Street View to audit neighborhood environments. *American journal of preventive medicine 40*, 1 (2011), 94–100.

158. Russakovsky, O., Li, L.-J., and Fei-Fei, L. Best of both worlds: human-machine collaboration for object annotation. *CVPR*, (2015).

159. Russell, B., Torralba, A., Murphy, K.P., and Freeman, W.T. LabelMe: a database and web-based tool for image annotation. *International Journal of Computer Vision 77*, 1–3 (2007), 157–173.

160. Ryan, R.M. and Deci, E.L. Intrinsic and Extrinsic Motivations: Classic Definitions and New Directions. *Contemporary Educational Psychology 25*, 1 (2000), 54–67.

161. Rzeszotarski, J.M. and Kittur, A. Instrumenting the Crowd: Using Implicit Behavioral Measures to Predict Task Performance. *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, ACM

(2011), 13–22.

162. Saelens, B.E., Sallis, J.F., Black, J.B., and Chen, D. Neighborhood-Based Differences in Physical Activity: An Environment Scale Evaluation. *American Journal of Public Health 93*, 9 (2003), 1552–1558.

163. Saelens, B.E., Sallis, J.F., and Frank, L.D. Environmental correlates of walking and cycling: Findings from the transportation, urban design, and planning literatures. *Annals of Behavioral Medicine 25*, 2 (2003), 80–91.

164. Sammer, G., Uhlmann, T., Unbehaun, W., et al. Identification of Mobility-Impaired Persons and Analysis of Their Travel Behavior and Needs. *Transportation Research Record: Journal of the Transportation Research Board 2320*, 1 (2012), 46–54.

165. Schwing, A.G. and Urtasun, R. Fully connected deep structured networks. *arXiv preprint arXiv:1503.02351*, (2015).

166. Simonyan, K. and Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR abs/1409.1*, (2014).

167. Sobek, A.D. and Miller, H.J. U-Access: a web-based system for routing pedestrians of differing abilities. *Journal of Geographical Systems 8*, 3 (2006), 269–287.

168. Sorokin, A. and Forsyth, D. Utility data annotation with Amazon Mechanical Turk. *First IEEE Workshop on Internet Vision at CVPR*, (2008).

169. Stent, S., Gherardi, R., Stenger, B., Soga, K., and Cipolla, R. Visual change detection on tunnel linings. *Machine Vision and Applications 27*, 3 (2016), 319–

330.

170. Stewart, K.J. and Gosain, S. The Impact of Ideology on Effectiveness in Open Source Software Development Teams. *MIS Q. 30*, 2 (2006), 291–314.

171. Stollof, E.R. and Barlow, J.M. *Pedestrian Mobility and Safety Audit Guide*. 2008.

172. Streets Wiki. Walk Audit. http://streetswiki.wikispaces.com/Walk+Audit.

173. Su, H., Deng, J., and Fei-Fei, L. Crowdsourcing Annotations for Visual Object Detection. *AAAI Technical Report, 4th Human Computation Workshop*, (2012).

174. Taylor, B.T., Fernando, P., Bauman, A.E., Williamson, A., Craig, J.C., and Redman, S. Measuring the Quality of Public Open Space Using Google Earth. *American Journal of Preventive Medicine 40*, 2 (2011), 105–112.

175. Taylor, R. Interpretation of the Correlation Coefficient: A Basic Review. *Journal of Diagnostic Medical Sonography 6*, 1 (1990), 35–39.

176. Team, T.G.C. *Data-Enabled Travel: How Geo-Data Can Support Inclusive Transportation, Tourism, and Navigation through Communities*. 2011.

177. Thapar, N., Warner, G., Drainoni, M.-L., et al. A pilot study of functional access to public buildings and facilities for persons with impairments. *Disability and Rehabilitation 26*, 5 (2004), 280–289.

178. The District of Columbia. Data Catalog http://opendata.dc.gov/. .

179. The District of Columbia. 311 Online http://311.dc.gov/. .

180. Tibshirani, R. Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society, Series B 58*, (1994), 267–288.

181. Timp, S. and Karssemeijer, N. A new 2D segmentation method based on dynamic programming applied to computer aided detection in mammography. *Medical Physics 31*, 5 (2004), 958–971.

182. Tohidi, M., Buxton, W., Baecker, R., and Sellen, A. User Sketches: A Quick, Inexpensive, and Effective Way to Elicit More Reflective User Feedback. *Proceedings of the 4th Nordic Conference on Human-computer Interaction: Changing Roles*, ACM (2006), 105–114.

183. Torii, A., Arandjelovic, R., Sivic, J., Okutomi, M., and Pajdla, T. 24/7 Place Recognition by View Synthesis. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2015).

184. Torkia, C., Reid, D., Korner-Bitensky, N., et al. Power wheelchair driving challenges in the community: a users' perspective. *Disability and Rehabilitation: Assistive Technology 10*, 3 (2015), 1–5.

185. U.S. Census Bureau. *Americans with Disabilities: 2010 Household Economic Studies*. 2012.

186. U.S. Department of Transportation, F.H.A. *Technological Innovations in Transportation for People with Disabilities*. 2011.

187. U.S. Department of Transportation, F.H.A. Highway Statistics 2014. 2015.

188. United States Access Board. *Proposed Accessibility Guidelines for Pedestrian Facilities in the Public Right-of-Way*. 2011.

189. United States Department of Justice. *2010 ADA Standards for Accessible Design*. 2010.

190. United States Department of Justice, C.R.D. *Wheelchairs, Mobility Aids, and Other Power-Driven Mobility Devices*. .

191. United States Department of Justice, C.R.D. *Americans with Disabilities Act of 1990, Pub. L. No. 101-336, 104 Stat. 328*. 1990.

192. Varadharajan, S., Jose, S., Sharma, K., Wander, L., and Mertz, C. Vision for road inspection. *IEEE Winter Conference on Applications of Computer Vision*, (2014), 115–122.

193. Vines, J., Wright, P.C., Silver, D., Winchcombe, M., and Olivier, P. Authenticity, Relatability and Collaborative Approaches to Sharing Knowledge About Assistive Living Technology. *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, ACM (2015), 82–94.

194. Viola, P. and Jones, M. Rapid Object Detection using a Boosted Cascade of Simple Features. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on 1*, (2001), 511.

195. Völkel, T. and Weber, G. RouteCheckr: Personalized Multicriteria Routing for Mobility Impaired Pedestrians. *Proceedings of the 10th International ACM SIGACCESS Conference on Computers and Accessibility*, ACM (2008), 185–192.

196. Völkel, T., Weber, G., Communication, M., and Gmbh, S. A New Approach for Pedestrian Navigation for Mobility Impaired Users Based on Multimodal Annotation of Geographical Data. *Access*, , 575–584.

197. Vondrick, C., Patterson, D., and Ramanan, D. Efficiently Scaling up Crowdsourced Video Annotation - A Set of Best Practices for High Quality, Economical Video Labeling. *International Journal of Computer Vision 101*, 1 (2013), 184–204.

198. Vondrick, C. and Ramanan, D. Video Annotation and Tracking with Active Learning. *Neural Information Processing Systems (NIPS)*, (2011).

199. WalkDenver. WALKscope. http://www.walkscope.org/.

200. Wang, R.Y. and Strong, D.M. Beyond Accuracy: What Data Quality Means to Data Consumers. *Journal of Management Information Systems 12*, 4 (1996), 5–33.

201. Welinder, P., Branson, S., Mita, T., et al. *Caltech-UCSD Birds 200*. 2010.

202. Welinder, P. and Perona, P. Online crowdsourcing: rating annotators and obtaining cost-effective labels. *In W. on Advancing Computer Vision with Humans in the Loop*, (2010).

203. Whitehill, J., Wu, T., Bergsma, J., Movellan, J.R., and Ruvolo, P.L. Whose Vote Should Count More: Optimal Integration of Labels from Labelers of Unknown Expertise. In Y. Bengio, D. Schuurmans, J.D. Lafferty, C.K.I. Williams and A. Culotta, eds., *Advances in Neural Information Processing Systems 22*. Curran Associates, Inc., 2009, 2035–2043.

204. Willett, K.W., Lintott, C.J., Bamford, S.P., et al. Galaxy Zoo 2: detailed morphological classifications for 304 122 galaxies from the Sloan Digital Sky Survey. *Monthly Notices of the Royal Astronomical Society*, (2013), stt1458.

205. Xiao, J. Princeton Vision Toolkit. http://vision.princeton.edu/code.html.

206. Xiao, J., Fang, T., Tan, P., Zhao, P., Ofek, E., and Quan, L. Image-based Fa\ÇAde Modeling. *ACM Trans. Graph. 27*, 5 (2008), 161:1--161:10.

207. Xiao, J., Fang, T., Zhao, P., Lhuillier, M., and Quan, L. Image-based Street-side City Modeling. *ACM Trans. Graph. 28*, 5 (2009), 114:1--114:12.

208. Xiao, J. and Furukawa, Y. Reconstructing the World's Museums. *International Journal of Computer Vision 110*, 3 (2014), 243–258.

209. Yang, S. and Mackworth, A.K. Route Planning and Scheduling for Wheelchair Users. *Proceedings of the Festival of International Conferences on Caregiving, Disability, Aging and Technology (FICCDAT), Toronto*, (2007).

210. Yao, B., Yang, X., and Zhu, S.-C. Introduction to a Large-Scale General Purpose Ground Truth Database: Methodology, Annotation Tool and Benchmarks. In A. Yuille, S.-C. Zhu, D. Cremers and Y. Wang, eds., *Energy Minimization Methods in Computer Vision and Pattern Recognition SE - 14*. Springer Berlin Heidelberg, 2007, 169–183.

211. Zamir, A.R., Darino, A., and Shah, M. Street View Challenge: Identification of Commercial Entities in Street View Imagery. *ICMLA (2)*, (2011), 380–383.

212. Zamir, A.R., Dehghan, A., and Shah, M. GMCP-Tracker: Global Multi-object Tracking Using Generalized Minimum Clique Graphs. *ECCV (2)*, (2012), 343–356.

213. Zamir, A.R. and Shah, M. Accurate Image Localization Based on Google Maps Street View. *Proceedings of the European Conference on Computer Vision*

(ECCV), (2010).

214. Zandbergen, P.A. Accuracy of iPhone Locations: A Comparison of Assisted GPS, WiFi and Cellular Positioning. *Transactions in GIS 13*, (2009), 5–25.

215. Zhang, P., Wang, J., Farhadi, A., Hebert, M., and Parikh, D. Predicting Failures of Vision Systems. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2014).

216. Zhang, Y.J. A review of recent evaluation methods for image segmentation. *Sixth International, Symposium on Signal Processing and its Applications, 2001*, (2001), 148–151 vol.1.

217. Zhu, H., Dow, S.P., Kraut, R.E., and Kittur, A. Reviewing Versus Doing: Learning and Performance in Crowd Assessment. *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing*, ACM (2014), 1445–1455.

218. Zooniverse. Galaxy Zoo. .

219. Zooniverse. Snapshot Serengeti. .

220. SeeClickFix http://seeclickfix.com/apps (Viewed Nov 2014). .

221. Handimap http://www.handimap.org/ (Viewed 2014). .

222. Walk Score. https://www.walkscore.com/methodology.shtml.

223. Wheelmap (Viewed Sept 2015). http://wheelmap.org/en/.

224. AXSMAP (Viewed Sept 2015). http://www.axsmap.com/.

225. Planet Labs. https://www.planet.com/.

226. Placemeter. https://www.placemeter.com/.

227. QGIS. http://qgis.org/en/site/.

228. Berkeley Segmentation Dataset and Benchmark, http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/. 2007.

229. San Francisco Municipal Transportation – Case Study. 2011. http://www.lytx.com/our-markets/fleet/case-study-san-francisco-municipal-transportation-agency.

230. OpenStreetMap Stats. 2016. http://wiki.openstreetmap.org/wiki/Stats#Registered_users.