ABSTRACT

Title of Thesis: TEMPORAL TRACKING URBAN AREAS USING GOOGLE STREET VIEW

Ladan Najafizadeh, Master of Science, 2016

Thesis Directed By: Professor Jon E. Froehlich
Department of Computer Science

Tracking the evolution of built environments is a challenging problem in computer vision due to the intrinsic complexity of urban scenes, as well as the dearth of temporal visual information from urban areas. Emerging technologies such as street view cars, provide massive amounts of high quality imagery data of urban environments at street-level (e.g., sidewalks, buildings). Such datasets are consistent with respect to space and time; hence, they could be a potential source for exploring the temporal changes transpiring in built environments. However, using street view images to detect temporal changes in urban scenes induces new challenges such as variation in illumination, camera pose, and appearance/disappearance of objects.

In this thesis, we leverage Google Street View's new feature, "time machine", to track and label the temporal changes of accessibility features (e.g., existence of curb-ramps, condition of sidewalks). The main contributions of this thesis are: (i) initial proof-of-concept automated method for tracking accessibility features through panorama images across time, (ii) a framework for processing and analyzing time series panoramas at scale, and (iii) a geo-temporal dataset including different types of accessibility features for the task of detection.

TEMPORAL TRACKING URBAN AREAS USING GOOGLE STREET VIEW

by

Ladan Najafizadeh

Thesis 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
Master of Science
2016

Advisory Committee:
Professor Jon E. Froehlich, Chair
Professor Hal Daumé III
Professor Jeffry Foster
Professor David Jacobs

To my parents,

Soheila & Abbas

Acknowledgements

First and foremost, I would like to thank my wonderful advisor, Jon E. Froehlich, for always taking chances on me, from the beginning and throughout my studies here at UMD. Your precious advice and guidance encouraged me to not only be a better researcher, but also be a better version of myself. Thank you for being patient with me, and understanding me, when I needed.

I would also thank my committee members, Hal Daumé III, Jeffrey Foster, and David Jacobs. Hal, you are an incredible teacher, and I feel lucky to have had a chance to take your classes and having you in my committee. Jeff, thank you for always supporting me one way or another. David, your valuable feedback, and suggestions truly helped me look at problems from different angles. Thank you.

Throughout my studies at UMD, I took some courses that definitely helped me in my research and in my professional life. Leah Findlater, Tom Goldstein, and David Mount, thank you for sharing your knowledge with others and me.

I could not make this far without the support of my friends and colleagues. Kotaro Hara, thank you for always giving me great advice and helping me figuring out my research questions. You are a true definition of O.G. A big thank you to my UMD friends: Jin Sun (for helping me at such a short notice), Matt Mauriello (for your wisdom and for making us laugh once in a while), Lee Stearns (for your help and advice, and Sir, you indeed have eagle eyes), Seokbin Kang (for your collaboration), Manaswi Saha (for your friendship), Leyla Norooz (for our worldwide adventures), Michael Gubbels (for our deep conversations), Sudha Rao (for the long-hours we spent struggling with homework questions, and of course, for your friendship), Meethu Malo (for your friendship), Jonggi Hong, Liang He, Majeed Kazemitabaar, and Soheil Behnezhad. Also a big shout out to my Persian friends at UMD, especially: Saba Ahmadi (for being such a great friend and a good listener), Kiana RoshanZamir (for always keeping me in the loop, and for your friendship and kindness), Mahsa Derakhshan, Ali Shafahi, and Sina Dehghani, as well as to my non-UMD friends, Venus Saatchi, and Elham Alikhani for always cheering me up.

At last but not least, I would like to thank my parents, Soheila and Abbas for their unconditional love. You taught me to fight for my dreams and to never give up on them. Your way of living has inspired me to think independently and passionately. Laleh thank you for being the best sister I could ever ask for. You always have been my role model. Ali, you are my brother, and my best friend at the same time, how cool is that? Thank you for being there, when I needed help. Our family would not have been complete without Sasan, and Sahar. Thank you for brining more joy to our family ☺

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

Evolution of built environments, whether occurring naturally or artificially, is an unavoidable process, which affects the elements of urban areas (*e.g.,* plants, buildings, sidewalks, climate, *etc.*) in terms of transformation, deterioration, amelioration, and construction [1]–[3]. Across urban environments, trees and plants transform into different states due to earth's axial tilt (*i.e.,* seasonal change), existing infrastructures deteriorate with time and usage, and new infrastructures are constructed in response to the demands.

Indeed, studying the evolution of urban environments is critically important to government policy, urban studies, as well as citizens (*e.g.,* for understanding gentrification, land use, predicting real-estate prices, *etc.*). Tracking temporal changes of built environments and visualizing the changes at scale would allow us to build better models of urban behavior across time (Figure 1-1). Tracking and visualizing accessibility problems, specifically, will help reveal how and where cities invest in improving accessibility infrastructure, how often that infrastructure is changes/improved, and whether certain parts of a city are systematically overlooked.

Land use **Gentrification**

Figure 1-1: Examples of temporal tracking built environments in urban studies.

Tracking built environments has been around for a while, and has been investigated by several studies in computer science, geo science, and urban planning over the years (details can be found in Chapter 2). For instance, in urban planning, inspecting the condition of street/sidewalk in a particular area requires taking multiple snapshots of that area over a specified time period. These snapshots then may help urban planners to gather information on how often street/sidewalks need to be changed/updated, or if constructing new infrastructure is in demand. These types of inspections are usually done via street audits, which are labor intensive, and do not address every issue of sidewalks such as accessibility of sidewalks. In this regard, the dearth of mechanisms to track the accessibility features in built environments at scale, motivated this thesis with the following research questions:

- How can we track the changes of accessibility features in urban environments across time, and automatically label them?
- How can we perform such a task at relatively large scale, let's say for the entire city?

The Lack of accessibility in urban areas directly impacts the lives of individuals with mobility impairments in many ways [4], [5]. The problem is not just that sidewalk accessibility affects where and how people travel in cities, but also that there are few mechanisms to determine accessible areas of a city a priori. A newly published report by the National Council on Disability stated that no comprehensive information can be found on the degree to which sidewalks are accessible across the US [6].

Recent studies such as Project Sidewalk [7], proposed the use of crowdsourcing to locate and assess sidewalk accessibility problems via Google StreetView imagery. This

thesis extends work in Project Sidewalk [7]–[9]. Project Sidewalk focused on scalable methods to map the accessibility of the world by semi-automatically classifying features in panoramic map imagery such as Google StreetView, where only the current state of accessibility infrastructure is being captured. In contrast, our primary focus, in this research, is on developing scalable methods to track accessibility features in the built environment over time. This is, arguably, a much harder problem because we have to scale both spatially (lots of location) as well as temporally (over time). Thus, our dataset is much larger. In this thesis, we provide a proof-of-concept investigation of studying how to back propagate labels of accessibility features in time.

The types of elements in urban environments that we are interested in are accessibility problems (*i.e.,* poorly conditioned sidewalks, missing curb-ramps, objects on path), as well as accessible sidewalks (*e.g.,* sidewalks with no accessibility problems) to check whether the accessibility problems resolved (Figure 1-2). Tracking these elements is particularly a hard problem in computer vision, since they might change in terms of structure and texture. Take, for instance a poorly conditioned sidewalk that got updated over the course of few years, and became an accessible sidewalk. This type of changes are textural rather than structural, since the geometric shape of the sidewalk remains the same, while the changes only occurred on its surface (*i.e.,* change in color or intensity). On the other hand, appearance and disappearance of objects on sidewalks represent a structural change.

Figure 1-2: Types of accessibility features in built environments

Emerging technologies such as street view cars provide massive amounts of high quality imagery data of built environments that gets updated frequently. Google Street View (GSV) is an example of such technologies that contains a feature, called "Time machine", which allows for a possibility of going back in time and exploring how built environments evolve over time (currently from 2007 to 2015) [10]. Moreover, GSV covers nearly every region of cities in the US [11], which makes it to be a potential source for exploring the evolution of urban areas, especially those neighborhoods that receive less attention in terms of maintenance of the pedestrian infrastructure.

This massive amounts of GSV imagery data as a less conventional data source is used in this thesis to track the progression of urban infrastructures in terms of accessibility features at scale, which otherwise would be expensive and difficult. However, similar to many other datasets, GSV images come with their own challenges such as, different

(lighting, weather, and season) conditions, as well as different camera viewpoints, and oftentimes containing occlusions (*e.g.,* a parked car obstructing the region of interest). These challenges are particularly hard to deal with from the computer vision perspective, where the aim is to detect the changes that have taken place in urban scenes across time, with a reasonable accuracy (Figure 1-3).

Figure 1-3: The challenges and limitations of GSV temporal images. All locations are located in Washington, DC. The physical address from left to right is as follows: 1899 Lang PI NE, 2702 28th St NE, 1733 Lang PI NE, 1701 V St SE.

1.1 Summary of Contributions

In this thesis, our focus is to explore the possibility of tracking accessibility features in urban areas across time using Google StreetView images as our dataset. Towards this goal, we have collected temporal images of nearly 400 locations from different neighborhoods of Washington, DC and the state of Maryland. We formulate the problem

into two parts: (1) image classification for classifying the types of accessibility problems, and (2) object detection to localize the accessibility problems within all snapshots.

To address this problem, the proposed system works as follows: the system identifies and labels the accessibility problems (*e.g.,* object in path, missing curb-ramps) in the most recent image at the location of interest. Next, it searches for the identified problems in the previous snapshots of that location, to see whether the identified accessibility problems have been resolved, or they still exist. The details of our proposed framework can be found in Chapter 3.

The main contributions of this thesis are:

(i) Initial proof-of-concept automated method that can be used to track accessibility problems through panorama images across time

(ii) Development of a preliminary framework for processing and analyzing time series panoramas at scale.

(iii) A geo-temporal dataset including different types of accessibility features for the task of object detection/image classification with respect to accessibility problems.

Chapter 2: Background and Related Work

The purpose of this chapter is to provide a review of studies that are most related to this thesis. We first review the computer vision aspect of urban tracking using satellite imagery, aerial imagery, street view imagery, and photos from the Internet (section 2.1). Next, we go over the studies on street-level accessibility, how cities invest in pedestrian infrastructures in terms of accessibility, and finally what semi-automated methods are currently being used to track urban accessibility (section 2.2).

2.1 Tracking Urban Changes

Change detection in urban areas has always been a challenging problem in computer vision. The goal of change detection is to identify the significant differences between the pixels of one image to the pixels of previous images, where all the images are referring to the same scene but taken at different times [12]. The difference is defined based on the application and the type of changes that are of interest (*e.g.,* in urban studies, the targets are usually buildings, roads, street signs, and vegetation). Change in weather and lighting conditions, the structure of the scene itself, followed by the variation of camera parameters in terms of viewpoint, resolution, and the distance between the camera and the scene, all together make the change detection problem very challenging. As a result, there is no solid or unique recipe for addressing the problem, but on the bright side, narrowing down the problem into smaller sets can help to achieve an optimal solution. To better categorize the related work, with respect to image data, we break down the change detection problem into two categories: stationary cameras, and non-stationary cameras.

Stationary Cameras. In this case, the sequence of images of a same scene, over time, is captured from a fixed viewpoint, which means that the acquired images are more or less aligned. Therefore, the challenges are mainly related to the illumination and/or geometric changes of wanted objects/regions of interest in the images. Jacobs et al. proposed a method for understanding the changes in the time-lapse sequences of static outdoor webcams (AMOS dataset) in terms of illumination such as the time of the day, or the weather condition [13]. Other methods to address change detection in videos with a stationary cameras, are probabilistic models, where pixels are modeled as a Gaussian mixture model, and are adapted to slow variations of object's position [14], [15]. A more detailed explanation about change detection can be found in [12]. On the application side, time-lapse photography was used to examine the spatio-temporal of snow cover of a particular area, where the camera was fixated [16].

Non-Stationary Cameras. The goal, in this case, is to reason about the temporal changes that are taken place in the same scene, but have been captured either with different cameras, or via vehicle-mounted cameras. Typically, in urban planning and related fields, high resolution satellite imagery and remote sensing technologies are used to track changes in urban areas with respect to land coverage, congestion, transportation, and infrastructures over time [17]. For example, the remote sensing data can be used to evaluate the traffic pattern of crowded locations, or the conditions of roads. Traditionally, detecting changes in the pattern of urban environments is done by human observation, which is time consuming, expensive, and with high error rate. To automate the change detection procedure, Pacifici et al. proposed a Neural-nets method for high-resolution imagery (Figure 2-1) [18]. Temporal tracking (or change detection in this context) using

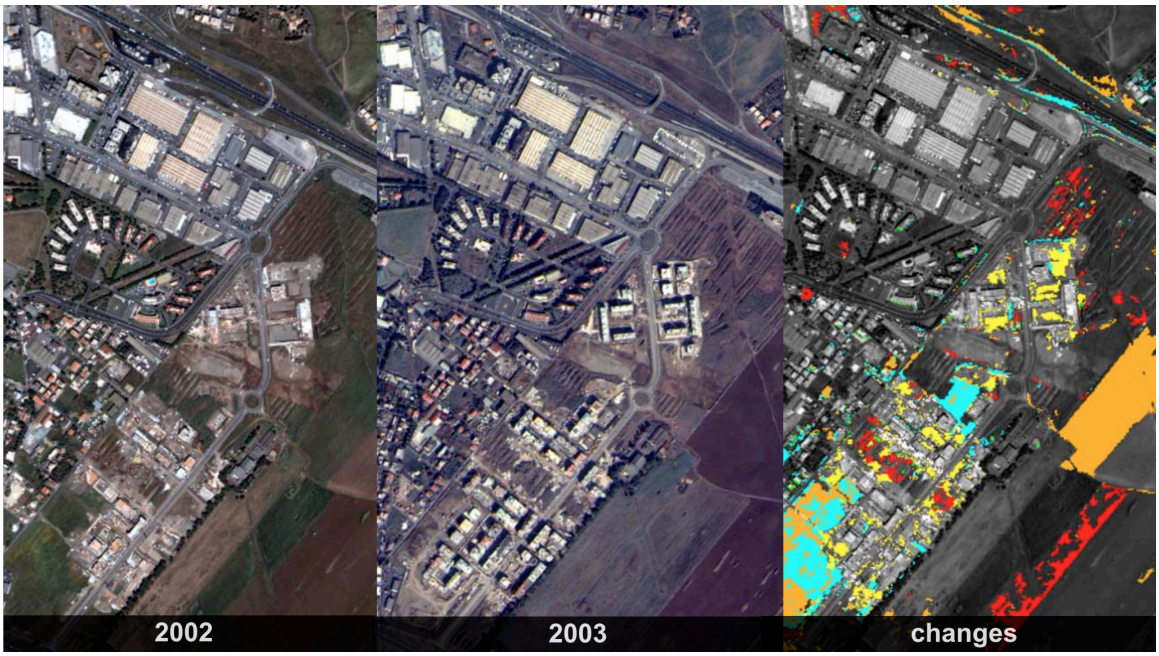

Figure 2-1: Change detection on satellite imagery for an area by comparing two consecutive years. The changes are shown in different colors, where each color represents the transition between two states. For example, the orange represents the transition from vegetation to soil.

satellite image data brings its own challenges (*e.g.,* atmospheric conditions, satellite sensor angles, and sensor noise). A broad body of work has been done to overcome these challenges, which is beyond the scope of this thesis, but interested reader is encouraged to read [19]–[21]. Regarding aerial images, a study use images of a same scene over time, by placing cameras at arbitrary but known positions, and treated the change detection problem as a probabilistic three dimensional (3D) voxel model, where new images are compared with old images at voxel-level, and get updated accordingly [22].

In regards to ground-based images acquired by non-stationary cameras, because the images are taken from different perspectives, many studies have attempted to first align the images before going through the change detection process. This alignment process is called image registration, and when the images are almost planar, Scale-Invariant Feature Transform (SIFT) feature matching [23] followed by a homography would suffice. Nonetheless, when the images are parallax (*e.g.,* variation in depth in the image), SIFT

features are not sufficient for the alignment, since they are invariant to affine deformation and to drastic changes in viewpoint, which are typical in temporal images of built environments. Hence, if the number of snapshots of the same scene is relatively large at each time stamp, a 3D reconstruction of a set of images using Structure from Motion (SfM) techniques [24] would be employed to align the images with each other along the time axis. Most previous work [25]–[33] follow the image registration step, but the change detection stage, along with the data collection procedure distinguish them from one another.

Posterior to the image registration, using SfM, the next step is to reason about the temporal changes (types of changes differ with respect to their task). When the historical photos are available but are undated, reconstructing a 3D probabilistic temporal model from the images, and reasoning about the visibility of the points in the 3D domain, has shown to help in determining the temporal ordering of the images [29], [33]. Further, to create a smooth time-lapse video from Internet photos, Martin-Brualla *et al.* computed a global depthmap of the input images, warped them according to one virtual camera, and applied a temporal regularization on the output [25]. Similarly, Matzen *et al.* proposed a method to create a time-lapse sequence of temporal changes in planar structures (textural changes) of cities such as billboards, and street arts, by reasoning about the point clouds in terms of space and time [28]. To detect the tempo-structural changes in urban scenes, Sakurada *et al.* and Taneja *et al.* used videos taken from a vehicle-mounted camera within a period of time, and transformed the data into 3D domain [27], [31]. By warping the recent 3D model into the previous one via reprojection, the changes in appearance

then revealed. A similar approach was applied on Google StreetView panoramas, where the cadastral models were available [30].

Recent studies [26], [32] proposed De-Convolutional Neural Networks (deConvNets) and Convolutional Neural Networks (ConvNets) for detecting changes in urban scenes, respectively, where the data was collected using vehicle-mounted camera with additional information. For example, in [26], the street view videos are used to detect the structural changes in urban areas using deConvNests (Figure 2-2). A summary of data collection and change detection methods of previous studies is provided in Table 2-1.

Figure 2-2: Structural change detection using De-convolutional Neural Networks.

We now highlight the differences between our work and previous ones. First, our dataset is limited to Google Street View, with no additional information provided, other than the location of the images. Second, since there is no API for the "time machine" feature in Google Street View, the data is collected manually, which means that the data is limited. Finally, in this thesis, we specifically focus on the accessibility features of urban scenes at street-level (*e.g.,* missing curb-ramps, surface problems), which means both the structural and the textural changes are important, and we need methods that can handle both types of changes.

Paper	Data Collection Method	Change Detection Technique
Schindler et al., 2007	Historical images	Visibility of feature points in the 3D domain
Schindler et al., 2010	Undated historical photos from 19th-20th centuries	Visibility of feature points in 3D domain
Taneja et al., 2011	Vehicle-mounted camera	Warping the recent 3D model onto previous ones
Taneja et al., 2013	Google Street View panoramas + cadastral model view	Warping the recent 3D model onto previous ones
Sakurada et al., 2013	Vehicle-mounted camera	Probabilistic model at pixel-level
Matzen et al., 2014	Geo-tagged photos from the Internet	Clustering 3D point clouds with respect to space & time
Martin-Brualla et al., 2015	Geo-tagged photos from the Internet	One general depthmap + temporal regularization at pixel-level
Sakurada et al., 2015	Vehicle-mounted camera + GPS sensor	CNN features + super-pixel segmentation
Alcantarilla et al., 2016	Vehicle-mounted camera (videos)	De-convolutional Neural Networks

Table 2-1: Summary of data collection and change detection techniques using non-stationary cameras.

2.2 Tracking Sidewalk Accessibility

The notion of accessibility in urban areas can be interpreted in two ways: (1) being locally close to opportunities such as jobs, health, and education, and (2) building urban infrastructures (*e.g.,* sidewalks) that follow the "universal design" principles, where universal design, in this context, refers to infrastructures that can be used/crossed by as many people as possible, including people with disabilities [34], [35]. Unfortunately, the latter interpretation is overlooked in most cities, in which the lack of accessibility at street-level (*e.g.,* poorly conditioned sidewalks, or absence of curb-ramps at intersections)

has brought and continues to bring significant challenges to the lives of people with mobility impairments. Inaccessible sidewalks vastly has impacted the lives of 30.6 million individuals with physical disabilities across the US [36]. Despite civil rights legislation for American with disabilities, ironically, inaccessibility at street-level still exists and has ostracized people with mobility impairments from the society. Missing curb-ramps at intersections, narrow or uneven sidewalks, existence of utility poles on sidewalks, and having poorly conditioned sidewalks or no sidewalk at all, reflect only a small fraction of the barriers people with mobility impairments face during navigation. Several lines of research in accessibility and urban planning have been dedicated to not only understand the severity of the problem but to improve the accessibility of sidewalks, accordingly [37]–[40]. In order to grasp the difficulties that people with mobility impairments face, when navigating the city, a considerable amount of surveys, interviews, and street audits have been conducted. For instance, Brookfield et al. have conducted a study with older adults to see how they would choose a route based on their physical condition via Google Street View [41].

Recently, Hara *et al.,* by combining computer vision techniques and crowdsourcing, proposed a semi-automated mechanism for identifying accessibility problems in cities, remotely by using Google Street View [8]. Similarly, Prandi *et al*. developed a system for mobile phones that suggests accessible paths to the user, using the data collected by crowdsourcing, geo-referenced social websites [42].

The dearth of interactive tools for obtaining information about the accessible areas within urban environments exacerbates the situation for individuals with mobility impairments; otherwise, they would have been prepared for upcoming challenges on the

route, prior to their trip. The most recent attempt to address this issue is the Project Sidewalk, in which people around the globe can remotely contribute in identifying the accessibility features within cities via Google Street View [7].

These types of mechanisms are suitable for identifying the most likely accessible route, or detecting current accessibility problems within the city that require repairing/updating. Identifying accessibility features in cities are as important as tracking their temporal changes. Despite the major advances in computer vision and urban planning, temporal tracking of accessibility features (*e.g.,* curb-ramps) in urban areas has received little to no attention, and to our knowledge, we are the first to address this issue.

Chapter 3: Methodology

In this chapter we present our proposed approach for temporal tracking accessibility problems in urban environments. First we describe the dataset, the procedures for data collection along with its limitations. Then, we explicitly define our approach for tackling the problem of temporal tracking accessibility problems in urban environments.

3.1 Dataset

To track the evolution of built environment with respect to accessibility problems we took advantage of Google StreetView new feature, "Time machine". To this date, the available images to view cover the period of 2007 to 2015, including arbitrary gaps between the dates. For instance, for some locations the available images are 2007, 2009, 2011, 2012, and 2014. Note that the process of updating urban scenes in GSV does not equally take place among all locations, meaning that the number of available temporal images differs per location (Figure 3-1).

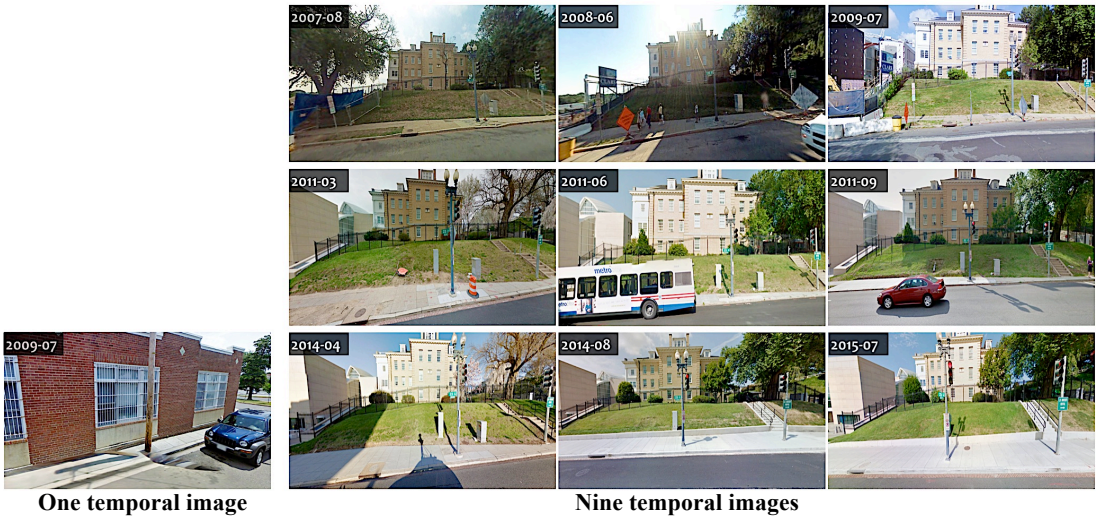

Figure 3-1: Difference in the number of temporal images available in GSV for various locations.

Perceiving the accessibility problems, especially in images, is subjective, which causes the data collection procedure to be ambiguous. To control the ambiguity of accessibility problems detection, we employed the guidelines of US Department of Transportation [43], the US Access Board [44], and followed the definition indicated in [9]. Accordingly, we categorized the accessibility features at street-level of urban areas into 5 main categories, as listed below:

1. Missing curb-ramps (including narrow and poorly conditioned curb-ramps)
2. Objects in path
3. Surface problems (*i.e.,* narrow/uneven/poorly conditioned sidewalks)
4. Accessible sidewalks
5. Accessible curb-ramps

3.1.1 Data Collection

The procedure of data collection was done manually, since Google has not yet offered an API for its "time machine" feature. We chose Washington, DC, and the state of Maryland as our primary source of data because of our first-person knowledge of those areas and their use in our previous work [8], [9].We collected temporal images of built environments by randomly walking through the streets of Washington, DC, and the state of Maryland, using Google StreetView, and by taking advantage of "Project Sidewalk" crowdsourced data to locate the areas containing accessibility problems [7]. To this aim, we randomly dropped the pegman of the Google maps on a random street, and started walking from there. From our experience in the data collection phase, the probability of encountering accessibility problems is higher in relatively poor neighborhoods. In Washington, DC, for instance, as we moved towards southeast and northeast areas, the

number of accessibility problems increased. To randomize and diversify our data, we took screenshots of locations based on the following rules:

- If a location contained accessibility problems, but over time the accessibility problems are resolved (Figure 3-2a).
- If a location still contained accessibility problems (Figure 3-2b).
- If a location did not contain accessibility problems within the available time frame (Figure 3-2c).
- If a location contained accessibility problems and occlusion sometime within the available time frame (Figure 3-2d).

Figure 3-2: Examples of locations with various transitions in terms of accessibility features: (a) the missing curb-ramps changed to accessible curb-ramps, (b) the missing curb-ramps still exist, (c) the sidewalk remains accessible, (d) vehicles obstruct the view of missing curb-ramps in 2009-07.

For each location, the screenshots were captured throughout the entire available time frame, along with their metadata. The metadata for each location is a JSON file containing the address, the GPS coordinates, URL, number of temporal snapshots, camera's yaw/pitch/field-of-view information followed by the date (year-month) for each snapshot. For locations with multiple accessibility problems, all accessibility problems were indicated in the their metadata separated by comma. Note that if only one of the snapshots from a same location contained an accessibility problem, we still treated the


```
{
  "reason": "surface problem",
  "location" : {
    "city": "Washington",
    "state": "DC",
    "address": "3052 Douglas St NE",
    "lat": 38.9234437,
    "long": -76.9657801,
    "url": "https://www.google.com/maps/@38.9234437,-76.9657801,3a,75y,1.54h,81.9t/data=!3m6!1e1!3m4!
        1s3iVZm0oAyJwEpbff0vlxig!2e0!7i13312!8i6656"
  },
  "num_timeshots": 4,
  "timeshots":[
    {
      "year": 2007,
      "month": 11,
      "img_info": [
        83.41,
        1.32,
        75
        ]
    },
    {
      "year": 2009,
      "month": 7,
      "img_info": [
        80.05,
        2.45,
        85
        ]
    },
    {
      "year": 2011,
      "month": 7,
      "img_info": [
        79.66,
        0.79,
        75
        ]
    },
    {
      "year": 2014,
      "month": 8,
      "img_info": [
        81.9,
        1.54,
        75
        ]
    }

  ]

}
```


Figure 3-3: A view of JSON file for a location with physical address as: 3052 Douglas St NE, Washington, DC.

location as being inaccessible with respect to the identified accessibility problem. A sample of JSON file is illustrated in Figure 3-3. Our geo-temporal tagged data will be available for public download.

We have collected 376 locations total, in which 90% of them contain accessibility problems. We mostly covered the DC area (88%) because of its diverse neighborhoods. The total number of all images regardless of location is 1633 (Table 3-1). As mentioned previously, the number of available temporal images is different per location, but the average number of available snapshots in our dataset is 4, meaning that most locations contain 4 available temporal images (Table 3-2).

To understand the quantity of each accessibility feature within our collected data, we used two metrics: per location, and per image. For a certain location, if an accessibility feature exists at least in one of the available temporal snapshots of that location, we consider the location as having that accessibility feature. For *per image,* we discard the locations, and calculate the existence of accessibility features within all images in the dataset (1633 images). In other words, for a given image, regardless of its location, we calculated how many accessibility features it contains. Using Matlab image labeling tool [45], we manually labeled/annotated the accessibility features for our *ground truth*, and computed the total number of each accessibility features category on all images (Table 3-3).

	Overall	DC	MD
# Locations	376	332	44
# Images	1633	1464	169
# Locations with accessibility problems	341	296	45

Table 3-1: Distribution of dataset

Avg # of snapshots	STD	Median	Min # of snapshots	Max # of snapshots
4	1	4	1	9

Table 3-2: Overall distribution of snapshots per location

Category	Per Location	Per Image
Missing curb-ramps	52	231
Objects in path	96	449
Surface problems	123	374
Accessible sidewalk	35	285
Accessible curb-ramp	70	267

Table 3-3: A breakdown of labels with respect to location and image. Per location refers to existence of accessibility features on at least one of the snapshots, and per image represent the existence of accessibility features on all images, regardless of their locations and time.

3.1.2 Dataset Limitations

The primary application of Google StreetView, and similar street view online tools are to provide remote navigation tools; hence using such datasets for tracking the temporal changes of built environments brings their own challenges and limitations to the table. In general, the common challenges of street view images (*e.g.,* GSV images) are variation in illumination (due to weather and lighting conditions), and camera pose among temporal images of street view cars (*i.e.,* images are not being captured from the same distance, or from the same spots). As a result, the temporal snapshots of a same scene are not aligned, which would exacerbate the temporal tracking problem.

Furthermore, the distance between the panorama images captured by Google Street cars is not consistent, and varies based on the location and time. Therefore, the regions of interest (ROI), in this case accessibility problems, might not be visible from the exact same geo-location for all images across time. In this case, the images are captured from nearest available spot.

Finally, due to the manual procedure of data collection, and since the regions of interest are accessibility problems; the number of available images is limited. This limitation directly affects the categories such that the number of one type might be comparatively different than other types. For instance, although the number of "missing curb-ramps" problem per image is relatively high (N=231), it is not comparable with the number of "objects in path"(N=449), or "surface problem" (N=374), which leads the dataset to be imbalanced. In the next section, methods for handling the abovementioned dataset limitations are discussed in details.

3.2 Our Approach

Given multiple snapshots of a same location over time, the goal is to automatically identify and label accessibility features in all snapshots, based on the labeled accessibility features in the most recent snapshot. More simply, if we label an accessibility problem (*e.g.,* a power pole on the middle of the sidewalk) on the most recent dated GSV image of a certain location, the goal is to see the evolution of the specified accessibility problem, in this case back propagating, whether the power pole existed on that sidewalk, or it has been installed recently. Our framework consists of 3 stages (Figure 3-4):

Stage 1. For each location, the most recent image is sent as an input to the framework. The accessibility feature is manually labeled via Matlab image labeling tools, and the resulting patch is sent to stage 2.

Stage 2. The category classification determines the category of the patch, and sends the result to the next stage.

Stage 3. Based on the result of category classification, the trained object detector examines all previous images of the same location to localize the specified accessibility feature within each image.

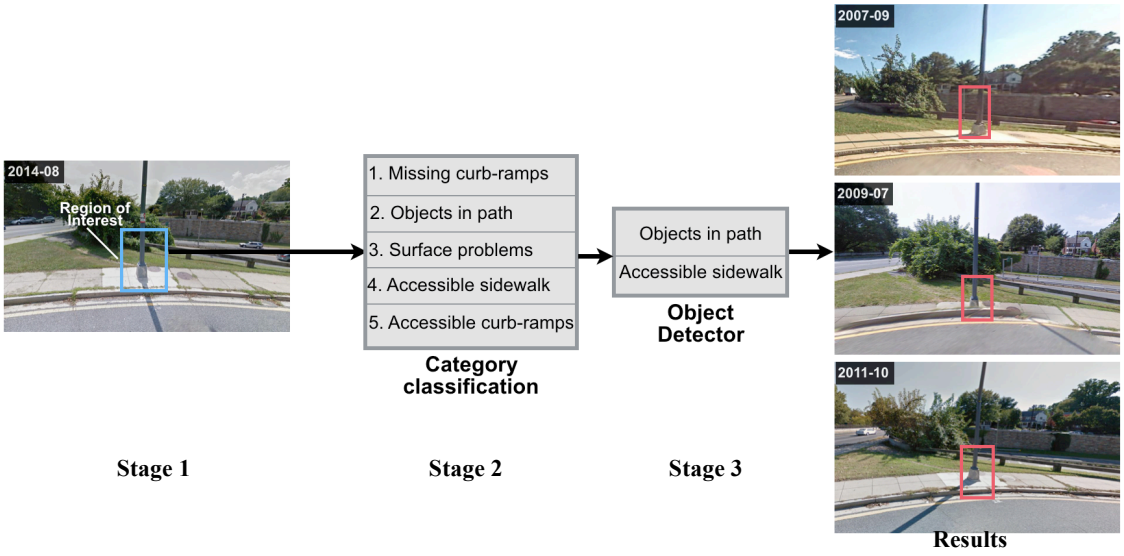

Figure 3-4: The stages in our framework: (stage1) the accessibility problem (*e.g.*, "object in path") is manually labeled, (stage 2) the category classifier classifies the labeled area, and (stage 3) the object detector localizes the accessibility problem in all snapshots over time of that location.

3.2.1 ROI labeling

In order to track the temporal changes of accessibility features in one location, we manually label the area containing the accessibility feature on the most recent image of that location. The reason behind choosing the most recent image rather than the earliest image is that the oldest image often has poorer quality in terms of resolution, and lighting (Figure 3-5). Also, the number of temporal snapshots varies for each location, therefore, the most recent image was chosen for labeling the ROI, and sending the ROI patch to the next stage to be classified.

Figure 3-5: Comparing the quality of the earliest image (left) to the most image (right) of a location.

3.2.2 Category Classification

In category classification stage, the goal is to find local interest points (*i.e.,* keypoints) in images that could distinguish the accessibility features from one another. Keypoints refer to geometrical or textural features that are unique to the accessibility problem's general shape or appearance. For instance, most accessible curb-ramps have trapezius shape, which can discern them from other accessibility problems in urban areas. However, this is not the case for missing curb-ramps, such that they are not geometrically discernable, and might be mistaken as surface problems.

Furthermore, due to variation in illumination (*e.g.,* lighting, weather condition) as well as variation in street view camera pose, make the classification even harder.

Recently, Convolutional Neural Networks (CNNs) have become the dominant approach for image classification [46]. However, a massive amount of training data is required to avoid over-fitting [47], [48]. Our dataset, on the contrary, is not large enough to train CNNs for category classification.

With all this in mind, we have used a well-known technique for categorization, called "bag-of-visual-words" (BoVW) [49], which is derived from the "bag-of-words" method that is used in natural language processing for information retrieval. The idea behind the

BoVW method is to create a vector of most frequent local features that represent each category. Although, the BoVW method does not depend on the spatial information of the ROI and can be used on the entire image, we used ROI patches (*i.e.,* only the accessibility features), because of the similarities between the elements in built environments. In other words, the visual information extracted from the entire image is a mixture of ROI and background, which affects on the real appearance of the ROI. Therefore, we have used the ROI patches of each category as our dataset (Figure 3-6).

Figure 3-6: The ROI patches that are used in training the category classification.

The BoVW method works as follows:

Feature extraction. The local features that are repeatable and invariant to image transformations (*i.e.,* translation, rotation, affine deformation, and scaling) are extracted from the image patches in the training set, and formed feature vectors (*e.g.,* SIFT descriptors [23]).

Clustering. The vectors of extracted local features are then mapped into the nearest cluster centers that contain similar features using k-means clustering algorithm [50]. Each cluster center represents a visual word vocabulary.

Visual BoW histograms. The frequencies of occurrence of visual words are mapped to vectors (*i.e.,* histograms) reflecting the categories.

To train accessibility feature categorizer (5 categories), 5 Support Vector Machines (SVMs) [51] were trained, where each SVM distinguishes one category from the rest.

The BoVW method does not localize the ROI on an image. Therefore, the next step towards reaching our goal is to do localization, which can be done using object detection algorithms.

3.2.3 Localization

In localization, the aim is to identify and detect the ROI (in this case an accessibility problem) within the image. The state-of-the-art object detection algorithms use a bounding box to scan the entire image and search for keypoints that are similar to the ROI. Here, we used Viola-Jones algorithm for object detection (*a.k.a.,* cascade object detector) [52], which is based on boosting [53].

When it comes to detecting accessibility problems at street-level in built environments, scanning the entire image is redundant, since accessibility problems are located at ground-level. This is not true for all GSV images, due to the variation in camera pose, and the street view car position. On the other hand, since the goal of this thesis is to label the accessibility features on images of a same location over time, the

Figure 3-7: Consistency of the location of accessibility features (*e.g.*, accessible curb-ramps) within all images over time of a same scene.

approximate location of the accessibility problem remains the same within all temporal images (Figure 3-7). Therefore, we can reduce the search area for object detector based

on the location of labeled area in the most recent dated image. This not only helps the object detector to detect and localize the accessibility feature faster and more accurately, but also reduces the number of false positives (i.e., falsely labeling an area on the image that does not contain the specified accessibility problem).

To train our cascade detector, for each accessibility feature category, we provided a large set of negative examples (*i.e.,* snapshots of urban areas that do not contain the

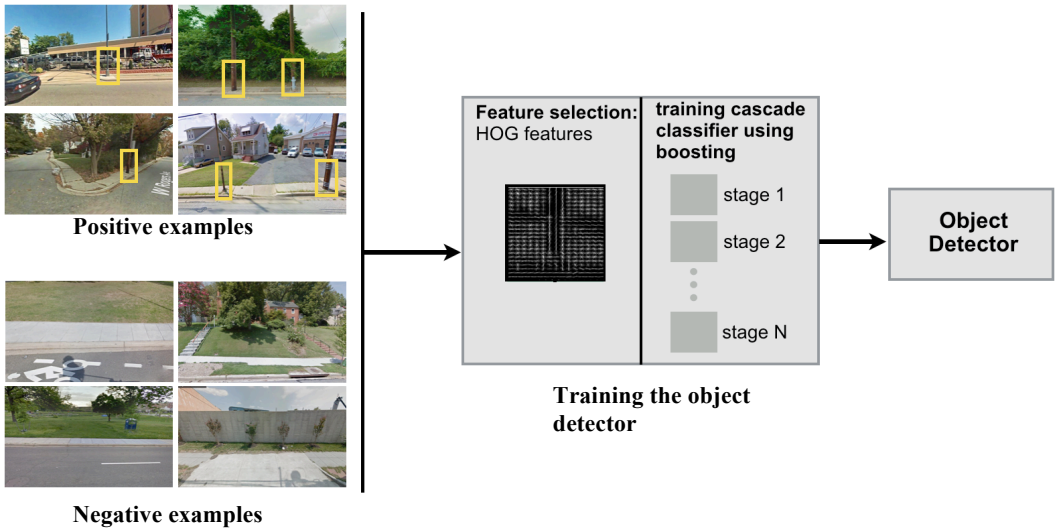

Figure 3-8: The overall procedure of Cascade object detector.

targeted accessibility problem), along with set of positive examples with the accessibility features labeled in each image. The number of negative examples is roughly twice the number of positive examples. The accessibility features are labeled manually in positive examples using Matlab image labeling toolbox [45]. In the training phase, the HOG (Histograms of Orientated Gradients) features of input images are selected, and are sent to the cascade classifier. The cascade classifier is a set of stages, where at each stage an ensemble of weak classifiers is trained to be a highly accurate one using the information from its previous stage. The number of stages depends on the size of dataset. For our

limited dataset, we have tested several stages to train the cascade detector, and found that 13-15 stages reduce the false positive rate (*i.e.,* the percentage of labeled areas that do not contain the specified accessibility features). The procedure of cascade object detector is shown in Figure 3-8.

In addition, accessibility features differ from one another in terms of aspect ratio (Figure 3-9). However, they maintain their aspect ratio within their category, meaning that "objects in path" aspect ratio remains approximately the same for majority of examples in the "objects in path" category. This increases the chance of detecting each category correctly.

Note that the cascade object detector has to be trained on all five categories of accessibility features, in order to localize them. Therefore, we trained five object detectors for five accessibility features. Finally, localizing the specified accessibility features on previous snapshots is not sufficient for tracking their changes over time. As a result, for each location, if the specified accessibility features are not detected, then object detectors for other accessibility features will be scanning the snapshots to see whether the

Figure 3-9: Consistency of aspect ratio within the examples of "Objects in path" category.

reason for the failed localization was due to the transition of the accessibility features or the object detector's fault.

Finally to reduce the number of bounding boxes predicted, we removed the overlapping detected windows by averaging the overlapped regions between the windows, and comparing them with a threshold.

3.2.4 Handling Occlusion

Vehicles and people are parts of urban environments; therefore, they exist in street view images, and might obstruct the regions of interest (*i.e.,* accessibility problems). A car parked in front of a sidewalk, where the sidewalk is the potential ROI, is a simple example of occlusion. In GSV imagery data, the amount of data for a scene that contains occlusion is sparse. If we consider the street view car's movement, for each sidewalk there are three snapshots available (Figure 3-10): before the street view car arrives (right view), when its in front of the sidewalk (frontal view), and after it passes the sidewalk (left view). Since, the goal in here is tracking the changes of accessibility problems over time, information from all temporal images is essential, and therefore, handling occlusion is required.

Left-view **Front-view** **Right-view**

Figure 3-10: Occluded an area of sidewalk by a vehicle (highlighted in blue). The left-view, and the right-view give enough information regarding the hidden area in front-view.

For a snapshot with occlusion, we looked at the previous snapshots and the later snapshots. If the accessibility feature existed between the former and the next snapshots, the occluded snapshot is ignored (Figure 3-11); otherwise, we looked at the occluded

Figure 3-11: Tracking the missing curb-ramps in a location, where the curb is occluded in one of the snapshots (2009-07). However, the curb still exists in 2011-08, meaning that the occluded snapshot can be ignored.

snapshot from different viewpoints to see whether the accessibility feature has changed on the date the snapshot was captured (Figure 3-10).

3.3 Summary

In order to track the temporal changes of accessibility features in urban areas, we manually collected temporal snapshots of roughly 400 locations in Washington, DC, and the state of Maryland, using GSV. The locations in the dataset are selected according to two criteria: maintaining the randomness, and balancing dataset with respect to the number of images per category. To meet the first criteria, we chose random neighborhoods in DC, and started to inspect their accessibility features by walking through their streets via GSV. For the latter, we used Project Sidewalk's crowdsourced data to locate certain accessibility features.

Our framework consists of three stages: (1) labeling the accessibility features in the most recent snapshot of a location, (2) classifying the labeled area as one of five accessibility feature categories, and (3) localizing the classified patch in all previous

snapshots of that location. This process only considers one location at a time, but has the potential to support scalability.

Chapter 4: Study Results

In this chapter, first, we evaluate the performance of category classification by itself, and report the analysis of our framework using standard measures such as precision, recall, and F_1- score. Next, we present the results for our framework tested on different locations.

4.1 Category Classification

To evaluate the overall performance of our category classifier, we used k-fold cross validation approach for k=5, with each fold consisting of 326 image patches of accessibility features. The confusion matrix of the performance of the category classification is shown in Table 4-1. The diagonal cells of the confusion matrix refer to the percentage of correct classification for each class (predicted category = actual category), and the off-diagonal cells represent the misclassifications for each category (predicted category ≠ actual category).

	Missing curb-ramps	Objects in path	Surface problems	Accessible sidewalk	Accessible curb-ramps
Missing curb-ramps	66.3%	1.6%	0.1%		12.6%
Objects in path		97.3%			0.6%
Surface problems	12.8%		81.4%	14.1%	3.3%
Accessible sidewalk	11.7%		17.2%	85.7%	9.3%
Accessible curb-ramps	10.2%	1.1%	0.3%	0.2%	74.2%

Table 4-1: Category classification confusion matrix. Each cell indicates the percentage of images assigned to a predicted category (column) for each actual category (row).

According to the confusion matrix, the performance of "missing curb-ramps" category is relatively poor, and the reason, besides the scarcity of the dataset, is due to the similarity between curb-ramps, missing-curb-ramps, the curbs of accessible sidewalks, or similarity between the surface of missing curb-ramps and surface problems on sidewalk.

4.2 System Framework Evaluation

To evaluate the performance of classifier and object detector combined (our framework), we randomly split our dataset (per locations) into 70% training set (N=264), and 30% test set (N=112). Since our framework works per location, we ran the framework for 112 round (*i.e.,* the size of the test set). The input of the framework at each round is the patch consisting of the accessibility feature, which is done manually before running the experiment. The results of the framework then stored separately for each location.

To measure the correctness of our framework per location, we looked for the input patch in all previous snapshots of that location at feature-level. For each location, feature-level refers to appearance of the specified accessibility feature (*e.g.,* surface problem) within all previous snapshots of that location. Therefore, as long as the specified accessibility feature has been found within previous snapshots, we accept the result. If the location of the detected label was approximately close to the manually labeled region, we accept that as well. For instance, if the manually labeled region (input of the framework) was a "surface problem", then the resulted labels on the previous snapshots might be covering other parts of the sidewalk with the same accessibility problem, but not at the specific region, which was manually labeled. In this case, we accept the results.

To better understand the overall performance of our framework regardless of location, we measured the correctness at feature-level for all images in the test set. Since our dataset was small, we used human perception to evaluate the correctness of labels in all images. We measured the precision, recall, and F_1-score based on the following equations:

$$\text{Precision} = \frac{\#\text{ of True positives labels}}{\#\text{ of True positive labels} + \#\text{ of False positive labels}}$$

$$\text{Recall} = \frac{\#\text{ of True positives labels}}{\#\text{ of True positive labels} + \#\text{ of False negative labels}}$$

$$F_1\text{-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Where, true positive is defined as providing the correct label in the image, false positive is providing a label for a problem that does not exist in the image, and false negative is not providing a label for a problem that exist in the image. The performance for each category and overall performance are illustrated in Figure 4-1.

The performance of our system depends on the two stages of classification and localization. If the input image patch is classified falsely, the localization of the false category on the previous images reduces the accuracy of the system. According to the performance graph, the "object in path" category has relatively a high accuracy, and the reason is that we treated every cylindered shape on sidewalk as "objects in path", even if it is not obstructing the path of pedestrians. Also, temporal snapshots of a same location in GSV are not aligned. However, they are panorama images; hence, during the data collection phase, we changed the yaw/pitch/Field of View of the camera at each time to

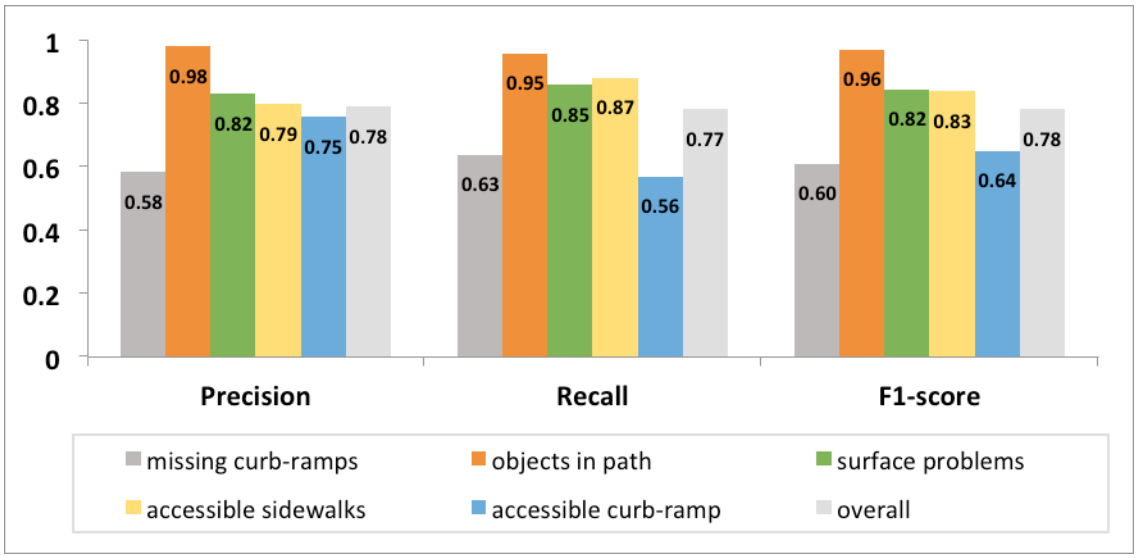

Figure 4-1: Test-time performance of our approach for each category and overall, with respect to Precision, Recall, and F1-score.

make the temporal snapshots aligned as much as it possible. Moreover, the search area for the object detector on each snapshot depends on the position of the manually labeled ROI (input patch). Thus, these results should be considered preliminary and likely represent the high-end of our framework's performance—they are under ideal conditions with manual tuning.

Furthermore, the object detector (Viola-Jones algorithm) has its own limitations. This algorithm works best on objects that do not have out-of-plane orientation, that's why the performance of the "object in path" category is comparatively high, to "missing curb-ramps" and "accessible curb-ramps" categories, since the orientation of curb-ramps differs among the street-view images.

4.3 Results

We have tested our framework on different locations from the dataset. The results are categorized into per location, and per image (Successful/failed).

4.3.1 Per Location Results

(i) Location: 2417 Hamlin St NW, Washington, DC (Figure 4-2).

The accessibility feature for this location is "Objects in path", and the framework successfully tracked the accessibility feature in all previous snapshots.

Figure 4-2: Framework results for "Objects in path" category. The blue label refers to the manually labeled ROI on the most recent snapshot (input of the framework), and red labels are localized by the framework, referring to the existence of accessibility problems on all previous snapshots.

(ii) Location: 6307 Brookville Rd, Washington, DC (Figure 4-3).

The accessibility feature for this location is "Accessible sidewalks". According to the results, the sidewalk in the snapshot "2012-03" does not have surface problems, and it

Figure 4-3: The most recent snapshot (top) has been manually labeled to specify the "Accessible sidewalk", and the result is shown on all previous snapshots (bottom).

was detected falsely, due to variation in illumination (false positive).

(iii) Location: 6202 Broad Branch Rd NW, Washington, DC (Figure 4-4).

The accessibility feature for this location is "Missing curb-ramps". The last snapshot (2014-05) is misclassified with "surface problem". This can be due to the similarities

Figure 4-4: top snapshot is manually labeled as "Missing curb-ramps", and the four bottom snapshots are the result of framework. The yellow label in the last snapshot refers to misclassification of "Missing curb-ramps" and "Surface problems".

between the two categories (curbs are visible in patches of narrow sidewalks).

4.3.2 Per Image Results

(i) Successful results: the results of the framework on different locations (discarding the time), in which the accessibility features have successfully been identified and localized (Figure 4-5).

(ii) Failed results: the failure of the framework in either identifying the specified accessibility features correctly (category classification), or localizing them within the snapshots (localization). Also, since the framework looks for other accessibility features,

if the specified accessibility feature could not be identified/localized within the snapshots, other accessibility features might be identified and localized (Figure 4-6).

Figure 4-5: Successful results of our framework. The green labels refer to accessible sidewalks and accessible curb-ramps. The red labels refer to accessibility problems.

Figure 4-6: Failed results of our framework. The yellow labels refer to either misclassifying the accessibility features, or not identifying the specified accessibility features indicated by the red arrows.

Chapter 5: Discussion and Future Work

This thesis took the first exploratory step towards serving a greater goal of developing a scalable (semi)-automated method for temporal tracking of accessibility problems in built environments. Here, we summarize the main contributions of this work, along with limitations, and we end this thesis by providing insights for directions of future research.

5.1 Conclusion and Limitations

We have demonstrated an initial proof-of-concept automated method for tracking the accessibility problems in street view images over time. In this thesis, we took advantage of bag of visual words and cascade object detector to identify and localize the accessibility features within all snapshots of a given location. Our findings show that despite the challenges of street view images, they could be a valuable source for tracking the accessibility problems at street-level. Our framework based on each location, tracks the changes in accessibility features across time, depicting the fact that even temporal tracking accessibility features for one scene is a difficult task. The performance of our framework indicates that the nature of tracking accessibility features cannot be performed automatically, as analyzing the conditions of accessibility features requires human understanding, due to the structural and textural changes in accessibility features. However, by incorporating automated mechanism and crowdsourcing, the goal of scalability is achievable. At scale, temporal tracking accessibility features can inform us what areas in built environments the pedestrian infrastructures have been overlooked, and how long these features have not been maintained/updated. This information can also be

used with additional information, such as population of residents and passersby at each region to decide financial decisions on allocating budget for renewing pedestrian infrastructures. Our current framework has the potential to support scalability, by bringing human in the loop for verification.

Limitaions. In this thesis, the data was collected manually, which is labor intensive, and time consuming. Our small dataset (376 locations; 1633 total images) limited our choice of classification and object detection algorithms. Also, to simplify the problem, we made assumptions about the position, and the size of accessibility features within all temporal snapshots at each location. By limiting the search area for the object detector, and by manually aligning the GSV images before taking screenshots, we tried to meet the assumptions. That is a reason for the getting relatively high results. However, as we mentioned before, this thesis is an exploratory step towards achieving the optimal accuracy for temporal tracking accessibility features in built environments.

Moreover, we did not evaluate our framework per location, because our current dataset is small, and imbalanced towards some categories of accessibility features (# objects in path > # missing curb-ramps); hence, when splitting the data to training and test sets, some categories never occurred in the test set.

Another limitation is related to GSV images themselves such as poor quality of some images, especially in the foremost year, which reduced the performance of our framework.

Furthermore, the input of our framework is done manually (*i.e.,* labeling the ROI in the most recent snapshot of a given location), which is time consuming and does not support scalability. With enough training data, and more accurate classification methods,

however, one could observe the temporal changes of more locations with respect to accessibility features. In addition, we used the most recent snapshots as our baseline of choosing accessibility features. Nonetheless, if the accessibility features have been maintained or completely transformed, labeling those features is not possible on the most recent snapshot. One possible solution is to demonstrate both the most recent and the foremost snapshots at the beginning.

Finally, since the primary focus of this thesis is on accessibility features, the majority of locations in the dataset do not contain occlusion. Therefore, our approach for handling the occlusion is limited to our dataset.

5.2 Directions Towards Future Research

While, in this thesis, we captured the important role of street view images on the scalability of temporal tracking built environments, specifically accessibility features, there are still many unexplored paths that can be taken from this starting point. We list a few of future work in the following:

Heatmap Visualization of Temporal Changes. Our semi-automated method currently captures only the changes of accessibility features at each location. Visualizing these temporal changes on a map, and using variation of color intensity to illustrate how accessibility features deteriorate/update over time, would be a way to capture the essence of the scalability in this research (Figure 5-1)

Predicting future changes in terms of accessibility at street-level. By incorporating the current data with other available resources regarding the maintenance of pedestrian infrastructures, could be used to predict possible future changes in the condition of accessibility features. This could be a useful for urban planners, and

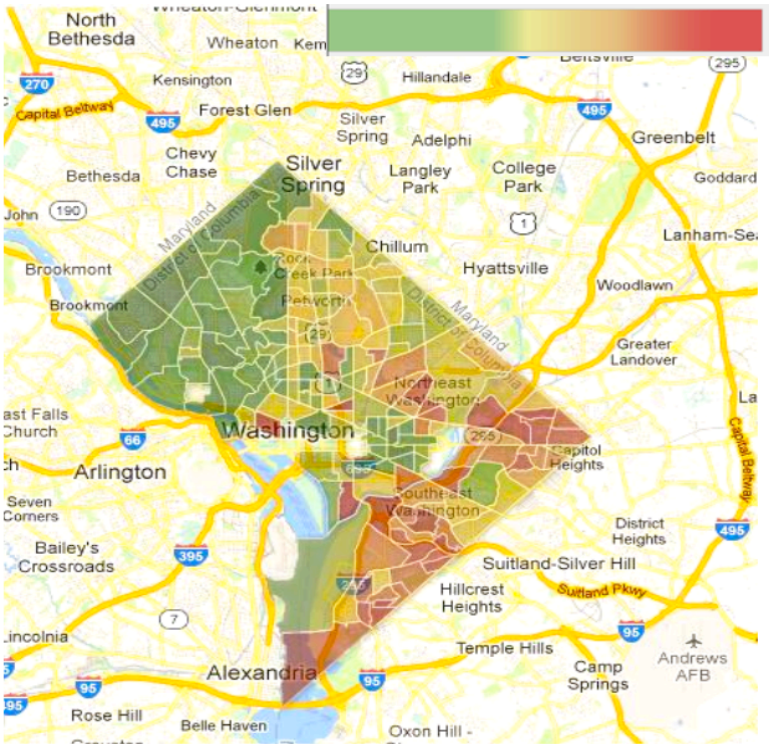

Figure 5-1: Possible heatmap visualization. The red refers to the period in which the accessibility features have not been maintained, and the green refers to the accessibility features being recently updated.

government officials to understand how often these features require maintain/update before they create barriers for citizen.

Time series labeling tool. The general theme of temporal tracking can be used to implement a tool that can automatically label a set of temporal images by only labeling one of the images, which could be useful in image labeling tasks.

Handling occlusion. Although we discussed about handling occlusion in this thesis, but future work can take advantage of the bird's eye view of Google StreetView [54], high-resolution satellite imagery, or aerial imagery to see the accessibility features from top-down view.

Bibliography

[1] S. M. Wheeler, “The Evolution of Built Landscapes in Metropolitan Regions,” *J. Plan. Educ. Res.*, vol. 27, no. 4, pp. 400–416, Jan. 2008.

[2] S. A. Changnon and S. A. Changnon, “Inadvertent Weather Modification in Urban Areas: Lessons for Global Climate Change,” *Bull. Am. Meteorol. Soc.*, vol. 73, no. 5, pp. 619–627, May 1992.

[3] J. G. Masek, F. E. Lindsay, and S. N. Goward, “Dynamics of urban growth in the Washington DC metropolitan area, 1973-1996, from Landsat observations,” *Int. J. Remote Sens.*, vol. 21, no. 18, pp. 3473–3486, Jan. 2000.

[4] I. M. Lid and P. K. Solvang, “(Dis)ability and the experience of accessibility in the urban environment,” *ALTER - Eur. J. Disabil. Res. / Rev. Eur. Rech. sur le Handicap*, vol. 10, no. 2, pp. 181–194, 2016.

[5] D. Gamache, S.;Vincent, C.; Routhier, F.; James McFadyen, B.; Beauregard, L.; Fiset, “DEVELOPMENT OF A MEASURE OF ACCESSIBILITY TO URBAN INFRASTRUCTURES: A CONTENT VALIDITY STUDY,” *Med. Res. Arch.*, vol. 4, no. 5, p. 603, 2016.

[6] N. C. on Disability, “The impact of the American with disabilities act: asessing the progress toward achieving the goals of the ADA,” 2007.

[7] Makeability Lab, “Project Sidewalk,https://sidewalk.umiacs.umd.edu/,” 2016. [Online]. Available: https://sidewalk.umiacs.umd.edu/.

[8] K. Hara, J. Sun, R. Moore, D. Jacobs, and J. Froehlich, “Tohme,” in *Proceedings of the 27th annual ACM symposium on User interface software and technology - UIST ’14*, 2014, pp. 189–204.

[9] K. Hara, V. Le, and J. Froehlich, “Combining crowdsourcing and google street view to identify street-level accessibility problems,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI ’13*, 2013, p. 631.

[10] Https://googleblog.blogspot.com/2014/04/go-back-in-time-with-street-view.html, “Go back in time with Street View,” *Google official blog*, 2014. [Online]. Available: https://googleblog.blogspot.com/2014/04/go-back-in-time-with-street-view.html.

[11] Google, “Google Street View Coverage,” 2016. .

[12] R. J. Radke, S. Andra, O. Al-Kofahi, and B. Roysam, “Image change detection algorithms: a systematic survey,” *IEEE Trans. Image Process.*, vol. 14, no. 3, pp. 294–307, Mar. 2005.

[13] N. Jacobs, N. Roman, and R. Pless, “Consistent Temporal Variations in Many Outdoor Scenes,” in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–6.

[14] C. Stauffer and W. E. L. Grimson, “Adaptive background mixture models for real-time tracking,” in *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, 1999, pp. 246–252.

[15] P. KaewTraKulPong and R. Bowden, “An Improved Adaptive Background Mixture Model for Real-time Tracking with Shadow Detection,” in *Video-Based Surveillance Systems*, Boston, MA: Springer US, 2002, pp. 135–144.

[16] J. Parajka, P. Haas, R. Kirnbauer, J. Jansa, and G. Blöschl, “Potential of time-lapse photography of snow for hydrological purposes at the small catchment scale,” *Hydrol. Process.*, vol. 26, no. 22, pp. 3327–3337, Oct. 2012.

[17] J. R. Jensen and D. C. Cowen, “Remote Sensing of Urban/Suburban Infrastructure and Socio-Economic Attributes,” in *The Map Reader*, Chichester, UK: John Wiley & Sons, Ltd, 2011, pp. 153–163.

[18] F. Pacifici, F. Del Frate, C. Solimini, and W. J. Emery, “An Innovative Neural-Net Method to Detect Temporal Changes in High-Resolution Optical Satellite Imagery,” *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 9, pp. 2940–2952, Sep. 2007.

[19] F. Pacifici, F. Del Frate, C. Solimini, and W. J. Emery, “An Innovative Neural-Net Method to Detect Temporal Changes in High-Resolution Optical Satellite Imagery,” *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 9, pp. 2940–2952, Sep. 2007.

[20] P. Du, S. Liu, P. Gamba, K. Tan, and J. Xia, “Fusion of Difference Images for Change Detection Over Urban Areas,” *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 5, no. 4, pp. 1076–1086, Aug. 2012.

[21] S. C. van der Spek and C. M. van Langelaar, “USING GPS-TRACKING TECHNOLOGY FOR URBAN DESIGN INTERVENTIONS,” *ISPRS - Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. XXXVIII-4/, pp. 41–44, Aug. 2011.

[22] T. Pollard and J. L. Mundy, “Change Detection in a 3-d World,” in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–6.

[23] D. G. Lowe, “Object recognition from local scale-invariant features,” in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 1999, pp. 1150–1157 vol.2.

[24] S. Agarwal *et al.*, “Building Rome in a day,” *Commun. ACM*, vol. 54, no. 10, pp. 105–112, Oct. 2011.

[25] R. Martin-Brualla, D. Gallup, and S. M. Seitz, “Time-lapse mining from internet photos,” *ACM Trans. Graph.*, vol. 34, no. 4, p. 62:1-62:8, Jul. 2015.

[26] P. F. Alcantarilla, S. Stent, G. Ros, R. Arroyo, and R. Gherardi, “Street-View Change Detection with Deconvolutional Networks,” in *Robotics: Science and Systems XII*, 2016.

[27] A. Taneja, L. Ballan, and M. Pollefeys, “Image based detection of geometric changes in urban environments,” in *2011 International Conference on Computer Vision*, 2011, pp. 2336–2343.

[28] K. Matzen and N. Snavely, “Scene Chronology BT - Computer Vision – ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VII,” D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 615–630.

[29] G. Schindler, F. Dellaert, and S. B. Kang, “Inferring Temporal Order of Images From 3D Structure,” in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–7.

[30] A. Taneja, L. Ballan, and M. Pollefeys, “City-Scale Change Detection in Cadastral 3D Models Using Images,” in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 113–120.

[31] K. Sakurada, T. Okatani, and K. Deguchi, “Detecting Changes in 3D Structure of a Scene from Multi-view Images Captured by a Vehicle-Mounted Camera.” pp. 137–144, 2013.

[32] K. Sakurada and T. Okatani, “Change Detection from a Street Image Pair using CNN Features and Superpixel Segmentation,” in *Procedings of the British Machine Vision Conference 2015*, 2015, p. 61.1-61.12.

[33] G. Schindler and F. Dellaert, “Probabilistic temporal inference on reconstructed 3D scenes,” in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 1410–1417.

[34] L. Viegas, Jose; Martinez, “Urban Accessibility: perception, measurement and equitable provision.” .

[35] J. Hanson, “The Inclusive City: delivering a more accessible urban environment through inclusive design.”

[36] U.S. Census Bureau, “Americans with Disabilities: 2010 Household Economic studies,” 2012.

[37] L. Beale, K. Field, D. Briggs, P. Picton, and H. Matthews, “Mapping for Wheelchair Users: Route Navigation in Urban Spaces,” *Cartogr. J.*, vol. 43, no. 1, pp. 68–81, Mar. 2006.

[38] R. D. F Bromley, D. L. Matthews, and C. J. Thomas, “City centre accessibility for wheelchair users: The consumer perspective and the planning implications,” *Cities*, vol. 24, no. 3, pp. 229–241, Jun. 2007.

[39] D. B. Gray, M. Gould, and J. E. Bickenbach, “ENVIRONMENTAL BARRIERS AND DISABILITY,” *J. Archit. Plann. Res.*, vol. 20, no. 1, pp. 29–37, 2003.

[40] H. Matthews, L. Beale, P. Picton, and D. Briggs, “Modelling Access with GIS in Urban Systems (MAGUS): capturing the experiences of wheelchair users,” *Area*, vol. 35, no. 1, pp. 34–45, Mar. 2003.

[41] K. Brookfield and S. Tilley, “Using Virtual Street Audits to Understand the Walkability of Older Adults’ Route Choices by Gender and Age,” *Int. J. Environ. Res. Public Health*, vol. 13, no. 12, p. 1061, Oct. 2016.

[42] C. Prandi, P. Salomoni, and S. Mirri, “mPASS: Integrating people sensing and crowdsourcing to map urban accessibility,” in *2014 IEEE 11th Consumer Communications and Networking Conference (CCNC)*, 2014, pp. 591–595.

[43] Transportation US Department, “Designing sidewalks and trails for access.” .

[44] P. R.-O.-W. A. A. C. (PROWACC), “Street and sidewalk accessibility problems,” 2007. [Online]. Available: https://www.access-board.gov/guidelines-and-standards/streets-sidewalks/public-rights-of-way.

[45] Mathworks, “Matlab image labeling tool.” .

[46] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks.” pp. 1097–1105, 2012.

[47] L. Yao and J. Miller, “Tiny ImageNet Classification with Convolutional Neural Networks,” *vision.stanford.edu*.

[48] Matlab, “Machine Learning Challenges.”

[49] G. Csurka, G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, “Visual categorization with bags of keypoints,” *Work. Stat. Learn. Comput. VISION, ECCV*, pp. 1--22, 2004.

[50] P. Berkhin, “A Survey of Clustering Data Mining Techniques,” in *Grouping Multidimensional Data*, Berlin/Heidelberg: Springer-Verlag, 2006, pp. 25–71.

[51] C. Cortes and V. Vapnik, “Support-vector networks,” *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995.

[52] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, vol. 1, p. I-511-I-518.

[53] P. M. B. Vitányi and R. E. Schapire, *Computational learning theory : second European conference, EuroCOLT ’95, Barcelona, Spain, March 13-15, 1995 : proceedings*. Springer, 1995.

[54] Google, “Google bird’s eye view.” .