SENIOR DESIGN FINAL REPORT

SIDEWALK SLOPE MONITORING SYSTEM

Version 1.0 - 04/12/2021

ENGINEERING

CITY OF LOS ANGELES

Team Members:

Alexis Pena
Jan Bautista
Hua Chen
Abigail Garcia
Ana Guardado
Cristina Munteanu
Pabasara Navaratne
Beatriz Ruiz
Aoqian Wang

Faculty Advisor:

Dr. Jung "Soo" Lim

Liaisons:

Ted Allen
Irvin Nguyen

TABLE OF CONTENTS

I. Introduction

A. *Background*

The City of Los Angeles Bureau of Engineering is responsible for the upkeep of approximately 472 square miles of sidewalk within LA County. With over 11,000 miles of concrete, keeping sidewalks ADA compliant can be a difficult task. This project aims to aid the Bureau of Engineering in properly identifying, monitoring, and expediting sidewalk repair to ensure they remain wheelchair accessible.

The project works to eliminate the need for a city official to go out in the field and manually measure and calculate the slope severity of each sidewalk by developing a fully functioning rover which does the task with greater efficiency. The rover's functionality would allow it to roll down a sidewalk and measure its slope while also collecting crucial data such as images and GPS coordinates--with very minimal user guidance. Our group has picked up where the last one left off and worked on transferring the initial hardware into a new platform provided to us by the Bureau and expanding the project with the necessary front and backend software to ensure the rover functions as required.

The project is divided into four separate tasks: overseeing the transfer of data to the rover's web interface, which we are creating (Web application); managing software utilizing artificial intelligence to process the images collected by the rover and determine a sidewalk's severity (Image processing); creating a user interface to control the rover and display live data (Rover User Interface); and building a comprehensive database to store all information collected by the rover (Backend Database). A rover equipped with these capabilities decreases the time a city employee would need to spend analyzing and collecting sidewalk data themselves. This rover speeds up the process by alerting the city to which sidewalks need fixing and what to prioritize.

B. *Design Principles*

The rover's hardware/software was built to be autonomous, robust, and modular. The Leo Rover is a very capable platform which met all our needs. Its modular build allowed us to expand its battery capacity which in return allowed us to expand the onboard hardware with a leveler, lidar, and depth cameras. The onboard software was created with minimal set of perquisites so to easily modify, upgrade, and understand it. The software is meant to serve its purpose and not rely on the rest of the platform to function as expected. The backend side of things takes into consideration that data will be collected from several pieces of hardware and has various autonomous steps to help with storing the crucial data in an easily accessible database.

The rover is not fully autonomous and requires city official to survey the city's sidewalks. However, the rover was designed to be operated by someone with little to no technical skills or even a working internet connection.

C. *Design Benefits*

The rover's modularity allows for future expansion of capabilities and quick repairs. The rover can be expanded to meet the future needs of the BOE and could be equipped to handle jobs outside of the current spectrum of tasks without compromising its current capabilities. By utilizing the built in Robot Operating System (ROS), the rover can take advantage of the vast number of libraries available. The software we designed was also built around the BOE's current portfolio of tools which should substantially help with costs and its learning curve.

D. *Achievements*

A working prototype of the Leo Rover has been developed. The current iteration of the Leo Rover is capable of being controlled remotely via our user interface, collecting the necessary data to decide if its ADA compliant, and store the data in our backend database. The early stages of better analyzing the data with AI is also there. Despite the immense amount of time dedicated to design, implement, and testing, the project is still a work in progress.

E. *Team Contributions*

The following table lists the roles and contributions of each member.

TABLE 1 Team contributions

Task	Name	Role	Contributions
Web application	Hua Chen	Quality assurance/Quality control lead	• Researched CSS/HTML features for website • Developed front end of web application
	Ana Guardado	Customer liaison/requirements lead	• Researched DWG automation • Developed back end of web application
Image processing	Pabasara Navaratne	Architecture/design lead	• Researched Image Processing methods • Developed image segmentation and thresholding, and canny edge detection.
	Jan Bautista	Demo/presentation lead	• Captured sidewalk images • Developed Grabcut algorithm • Developed algorithm to calculate xy displacement.
Rover user interface	Beatriz Ruiz	N/A	• Researched Rover Operating System • Developed Rover Interface control functionalities
	Cristina Munteanu	Documentation lead	• Developed Rover Interface data table • Designed Rover controllers

Task 4 name	Alexis Pena	Project lead	• Designed, developed, and tested:
			  ○ Azure Database
			  ○ Database automation
			  ○ Image Storage
			• Assembled Leo Rover

	Abigail Garcia	Components lead	• Researched Automation Methods • Researched cloud tools to be used to save images. • Designed Developed and tested Azure Database

II. Related Works and Technology

As described, the project is composed of four major tasks: web application, image processing, rover user interface, and the backend database. The following sections outline the related work and technology associated with each specific task.

A. *Web Application*

1) *NavigateLA* One of the existing mapping applications that we reference is NavigateLA. NavigateLA was built by BOE and it visualizes data using reports produced by the city, county, and other associated agencies. Data is broken down into layers so that users can select the layers they would like to see or even load their own layers. As part of the web application task, we needed to develop scripts that could overlay the sidewalk data collected by the rover onto NavigateLA. This section of the task is explained in detail in Web Application.

Figure 1 NavigateLA, BOE's mapping application

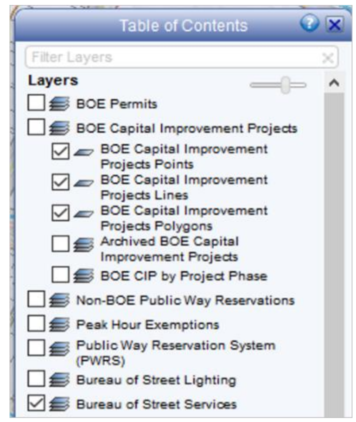


Figure 2 NavigateLA, example of layers that users may select from

2) *Google Maps API* In the web application, we also wanted to display a small section of the city using a map. Because there is a need for the user to view where in the City the rover image was captured, we used Google Maps' Application Programming Interface (API) to return a snippet of the map. Rather than generating a static image that constantly queried NavigateLA or BOE's databases, the Google Maps' API returned an interactive frame of the map that users can control.

B. *Image Processing*

1) *Read/Writes Image Data* Currently we read and write images from computer file system. Eventually, the functionality will process the rover chosen image.

2) *Isolate sidewalk on Image* Image processing functionalities use the rectangle mode of the GrabCut method to isolate the sidewalk on the image by deleting anything that is not the sidewalk. These functionalities have been programmed with OpenCV library in Python.

3) *Calculate Displacements* First, align colored and depth frames to map the pixels together. Then, get the two 3D coordinates and perform euclidean distance to get the displacement in meters. Finally, convert to centimeters.

C. *Rover User Interface* Before developing the Rover User Interface, we must review what is available in collecting the data. The existing tools in measuring the sidewalk slope and gathering the data are done manually using a Digital Level, CS8900 Access Profiler and a Light Survey Profiler. In removing the manual aspect, we have developed a User Interface for the Leo Rover which will replace the manual aspect of measuring and collecting the data. The Rover UI comprises of:

1) *Rover Control functionalities* This allows for the user to control the rover movement and speed.

2) *Data Table Display* Display the data collected/ Close the table for space conservation.

3) *Export Data* Export data to local field agent computer.

D. *Backend Database*The backend database is an Azure based SQL database. It's currently comprised of three tables: GoProEXIF, RoverData, and NavLA Sidewalks. The tables collect data from the rovers GoPro, Level, Lidar Cameras, and GPS Module. The data is upload to the DB via csv files through the Blob Storage.

III. System Architecture

Each section below outlines each tasks' architecture, workflow, and data exchange.

A. Web Application

The purpose of the web application is to provide a user-friendly method to view images captured by the rover. Because the rover images and data will be stored in a database, it is preferred that users interact with a website that can perform searches and display the data in a legible manner. This removes any overhead with training users on how to access, use, and query the data stored in the database. Instead, the web application will perform the necessary queries to retrieve the information for the user.

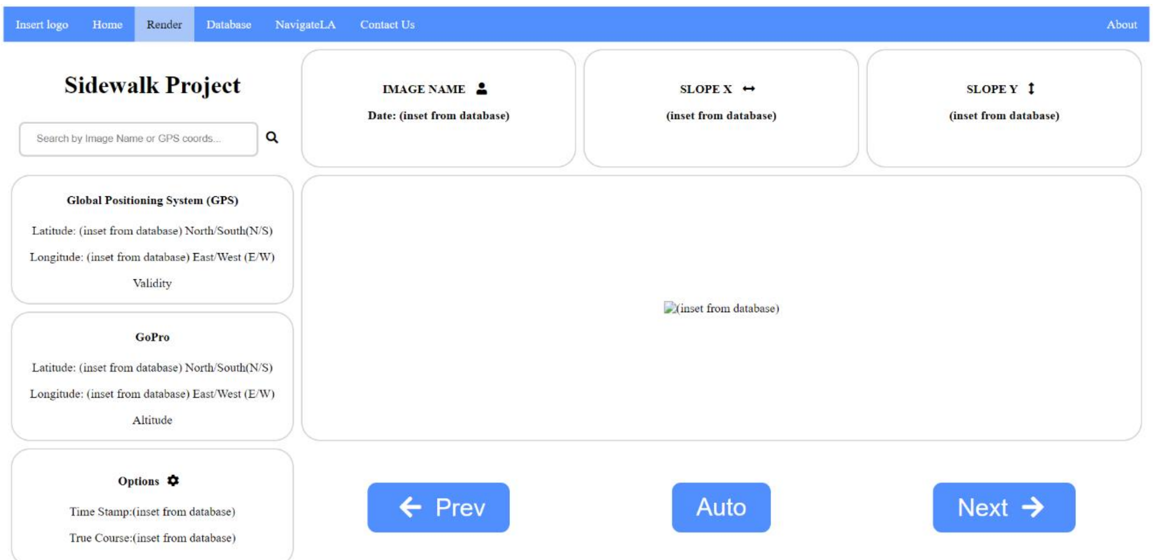


Figure 3 Web application overview without any data

The web application is dependent on a database that supports spatial queries. The database must also contain images captured by the rover and the related EXIF data for each image. The web application must also be hosted on a server provided and maintained by BOE.

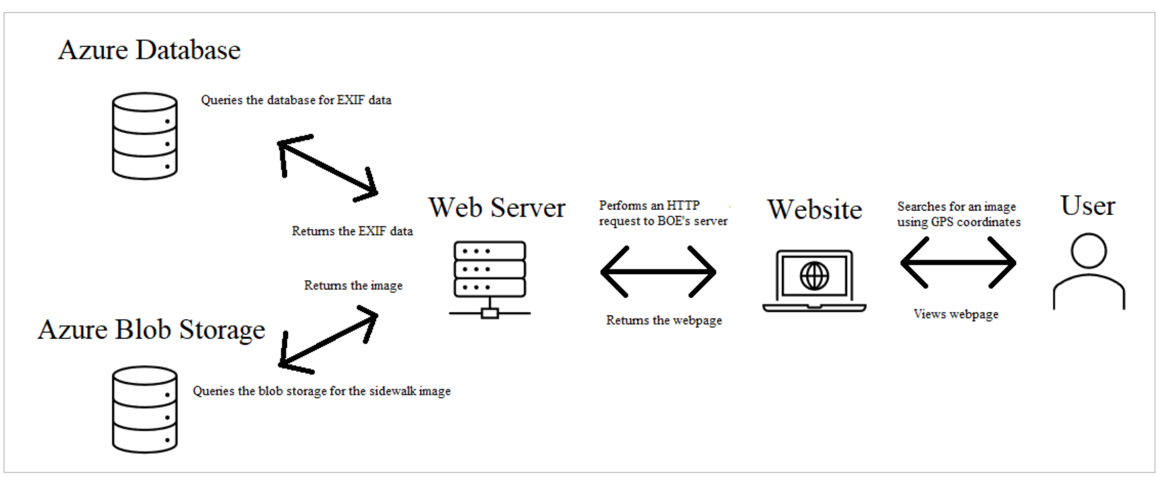


Figure 4 Example of how a user may interact with the system

BOE currently uses NavigateLA to visualize their mapping data. The main difference between NavigateLA and our web application is that NavigateLA visualizes data in a 2d manner, whereas our web application will display the actual images of sidewalk segments captured by the rover. An alternative to our web application was to develop automation scripts that would generate DWG files. These DWG files could be hosted in NavigateLA, allowing BOE to continue using only one site for their data visualizations.

Sidewalk Cross-slope	Sidewalk Cross-slope
≥ 20%	5
< 20% to ≥10%	4
< 10% to ≥ 5%	3
< 5% to > 2%	2
≤ 2%	1

Figure 5 Color code score guide using sidewalk cross slope from BOE's report

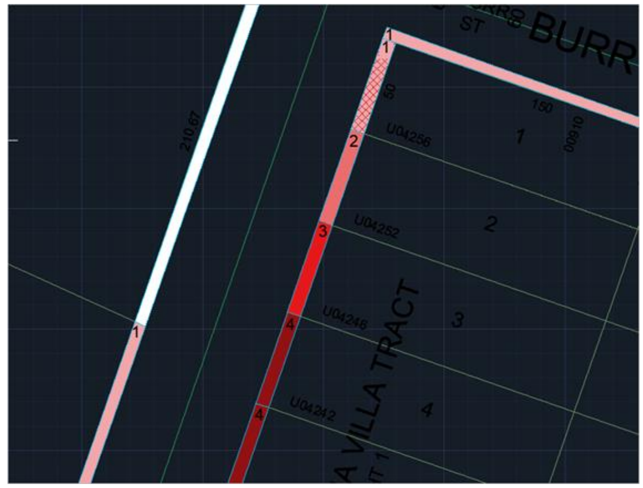


Figure 6 Example of DWG using color code

This solution required a server from BOE that hosted our DWG files. This solution also required automation scripts in AutoCAD's command language and Python. We would use Python to automate the collection of EXIF data. We would then use AutoCAD's command language to automatically create DWG files using the coordinates from the EXIF data. Although the web application team had started to conduct research on this solution, BOE decided this was not a solution they desired at the moment. This effort was put on hold indefinitely.

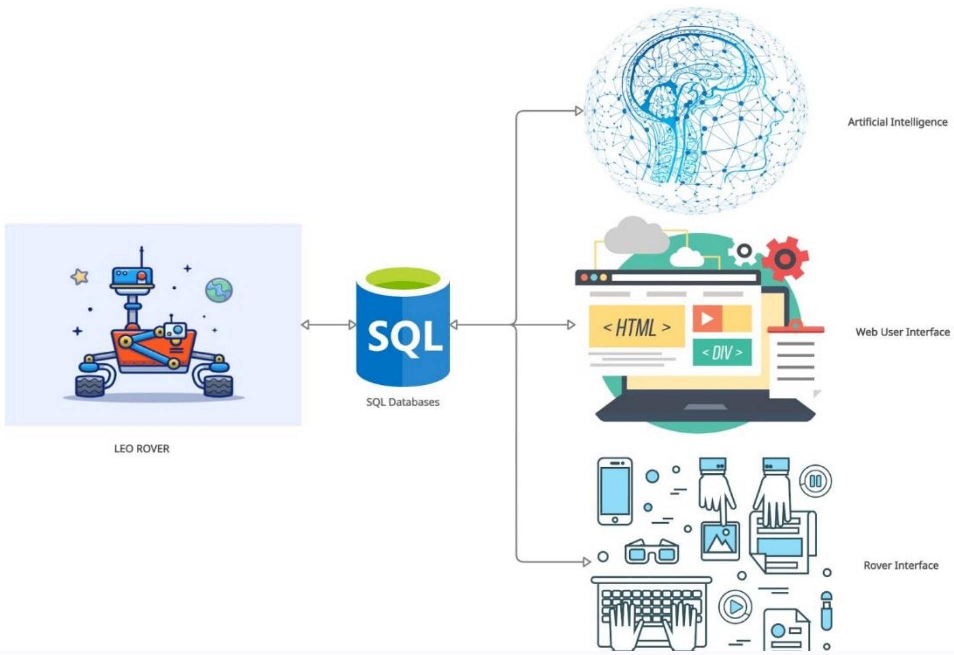
B. Image Processing

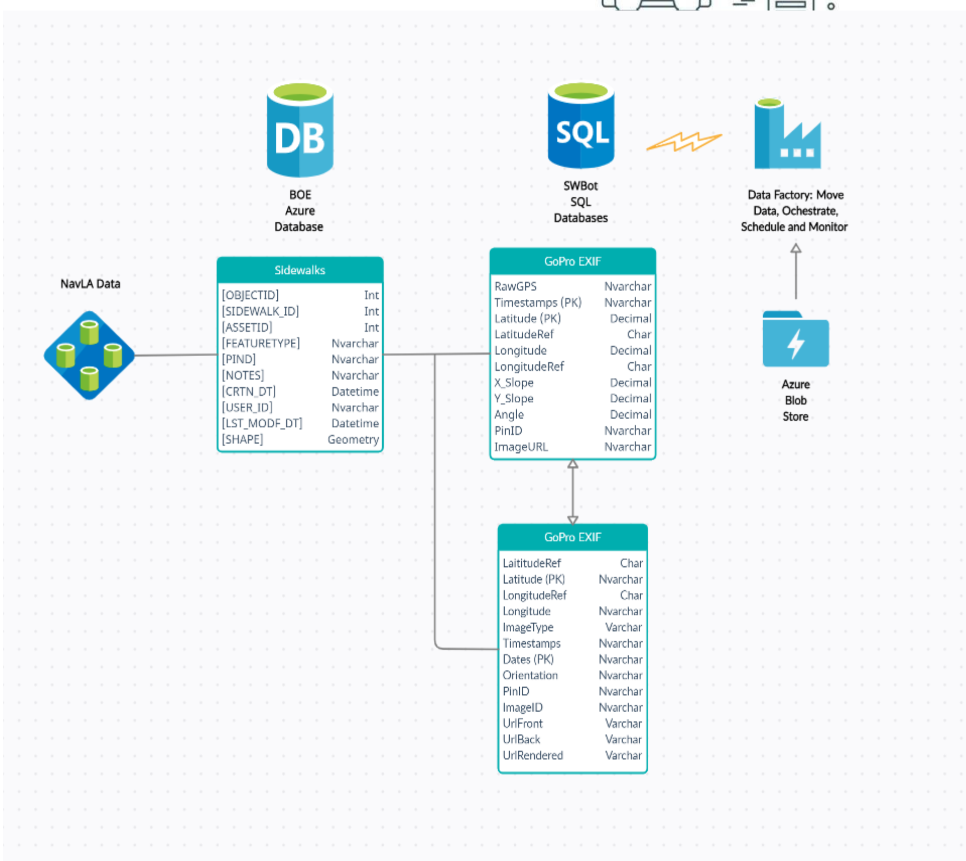
Image Processing task provides the functionality to process the images taken from the rover camera (LiDar) to assist BOE in determining the ADA compliance for horizontal and vertical displacements of a sidewalk. LiDar images will be processed to isolate the sidewalk and measure the said displacements of the damages on the sidewalk. This year's image processing task has been the first task that has developed image analysis and manipulation for this project comparing to previous years.
First, read both the colored and depth data and align them. Then, select two 2D coordinates and the algorithm will obtain the 3D coordinates. Finally, perform Euclidean distance between those 3D coordinates and the console will display the displacement in meters, which will then be converted to centimeters.

C. Rover User Interface

This factor allows the user to control the Leo Rover and displaying the data that is being collected.

*D. Backend Database*The database grabs data from the rover and feeds to the other tasks. The data from our tables is linked together via PinID which we get from the existing BOE Sidewalks table.


Artificial Intelligence
SQL
SQL Databases
LEO ROVER
< HTML >
< DIV >
Web User Interface
Rover Interface


DB
BOE
Azure
Database
SQL
SWBot
SQL
Databases
Data Factory: Move
Data, Ochestrate,
Schedule and Monitor
Azure
Blob
Store
NavLA Data
Sidewalks
[OBJECTID] Int
[SIDEWALK_ID] Int
[ASSETID] Int
[FEATURETYPE] Nvarchar
[PIND] Nvarchar
[NOTES] Nvarchar
[CRTN_DT] Datetime
[USER_ID] Nvarchar
[LST_MODF_DT] Datetime
[SHAPE] Geometry
GoPro EXIF
RawGPS Nvarchar
Timestamps (PK) Nvarchar
Latitude (PK) Decimal
LatitudeRef Char
Longitude Decimal
LongitudeRef Char
X_Slope Decimal
Y_Slope Decimal
Angle Decimal
PinID Nvarchar
ImageURL Nvarchar
GoPro EXIF
LaititudeRef Char
Latitude (PK) Nvarchar
LongitudeRef Char
Longitude Nvarchar
ImageType Varchar
Timestamps Nvarchar
Dates (PK) Nvarchar
Orientation Nvarchar
PinID Nvarchar
ImageID Nvarchar
UrlFront Varchar
UrlBack Varchar
UrlRendered Varchar

Results and Conclusion

A. *Results*

1) *Web Application* The web application was completed locally and cannot be accessed from the internet. The web application's front page and map page we completed. The backend was partially completed, including database requests and Google Maps API requests.

2) *Image Processing* We learned and researched image processing library techniques to implement Python based program. Then, we have programed the image manipulation function to segment the sidewalk and remove the background noise using OpenCV. The program can read both colored and depth data, and take two 2D coordinates as user input to display in the console the calculated xy displacement in centimeters.

3) *Rover User Interface* We have created a rover user interface that allows field users to control the functionalities of the Leo Rover. This allows for gathering the images and data and replacing the manual methods. The UI allows for interaction with the functions of the Rover without accessing its components, requiring extensive knowledge, or training.

4) *Backend Database* The backend database team has successfully managed to create, link, and automate the storage process of our database. We can collect data from the rover, automate its ingesting into the database, store it, and feed it back to our software for display and modification.

B. *Future*

1) *Web Application* The recommended next steps are to request a server from BOE to stand up the web application. This will enable BOE employees to either access the web application via their VPN if hosted within their network or via the Internet if the server is external facing.

2) *Image Processing* The future of Image processing task for this project should aim to automated ways of measuring the displacements. Additionally, the task should focus more of collecting data for Machine Learning and AI technologies.

3) *Backend Database* The Database will begin ingesting test data and see whether we can correlate the data from the rover to that from NavLA. Task 4 will also need to work on automating the data coming in to remove the manual aspect of uploading csv files.

V. References

M. Haklay, A. Singleton, and C. Parker, “Web Mapping 2.0: The Neogeography of the GeoWeb,” *Geography Compass*, vol. 2, no. 6, pp. 2011–2039, 2008, doi: https://doi.org/10.1111/j.1749-8198.2008.00167.x.

N. Shaparev and O. Yakubailik, “Usage of Web Mapping Systems and Services for Information Support of Regional Management,” *MATEC Web Conf.*, vol. 79, p. 01081, 2016, doi: 10.1051/matecconf/20167901081.