SQUID-Bike to digitally measure citywide Bike Lane Infrastructure

Felipe Diego Gonzalez and Nicola Macchitella and Geoff Perrin and Sichen Tang
Center for Urban Science and Progress Mentors: Varun Adibhatla *Advanced Research in Government Operations* Sponsors: Advanced Research in Government Operations

Abstract / Executive Summary

Urban bicycle usage has gained in importance across many cities with progressive transportation policies. According to [1], as of this paper's publication, "there are more than 450,000 daily bike trips in New York City, up from 170,000 in 2005, an increase that has outpaced population and employment growth". About one in five bike trips is by a commuter. Biking serves as an important transportation option to many around the world and we argue that the need for effective bicycle lane maintenance should be a top concern for municipalities.

Conventionally, street maintenance is an expensive, often inaccurate, and time intensive process that either uses subjective data prone to error or uses very precise data that is very expensive to collect citywide. There exists a clear need for cities to adopt a cost-effective and data intensive maintenance practice that can scale to the entire city and be performed frequently. These needs have been explored through the Street Quality Identification also known as the SQUID project [2] to develop standardized methods for digital street inspection. This work extends the SQUID project and repurposes it for citywide bike lane measurements.

In this paper, we describe the development of a data and analytics framework to measure bicycle lane quality using street imagery and accelerometer data obtained from an open source smartphone application, OpenStreetCam (OSC) [3] . This framework can be used in crowd-sourced or situated settings with the overall purpose being the standardized measurement of citywide bike lane quality.

Introduction

In recent years, bike based mobility in cities around the world has been growing rapidly. Compared to cars, bikes are cheaper, safer, more sustainable, allow for higher traffic flows, and require far less space for parking. Many city governments have invested in a permanent bike-sharing infrastructure and programs in an effort to improve transit conditions within the urban core.

Besides surface streets, the main urban infrastructure supporting bike ridership is a well maintained bike-lane network. The quality of bike lanes has an enormous impact in bike ridership. Lack of suspension in most bikes make them more susceptible to imperfections in the road. At the same time, bike damage, such as a flat tire or uneven wheels, can be linked to poor road conditions. City governments interested in fostering bike ridership as a mode of transportation should also ensure an effective infrastructure maintenance program.

However, this maintenance creates a set of problems for city governments around resource allocation constraints. The challenge therefore, is to come up with a cost-effective way of performing citywide bike-lane maintenance. A key aspect of this is to detect which bike-lanes need maintenance more than others. While there exist techniques to solve this problem, they are often are time intensive and costly that do not allow cities to perform routine bike-lane quality surveys. We argue that these surveys should be audited frequently for cities to be proactive around maintenance and avoid the large costs of deferred maintenance.

Our SQUID-Bike solution is a new standardized measure of the general condition of a city-wide bike lane infrastructure by integrating digital street imagery and ride quality data using readily available, inexpensive, and open source technologies. Our approach enables municipal agencies to answer a simple question: *Which bike lanes are worse off than others in a municipality?* This empowers agencies to engage in a proactive and equitable maintenance program by relying on digital assessments of all bike lanes in a city. Using frequently updated, longitudinal data from SQUID-bike would allow municipality agencies to observe bike lane degradation over time and could be used to power an anticipatory maintenance program and avoid the huge financial and political burdens of deferred maintenance.

LITERATURE REVIEW AND RELATED PREVIOUS WORK

Cycling (along with walking) are considered to be "the most economically, socially, and environmentally sustainable forms of human locomotion". [4] The literature shows that the introduction of a bike-lanes causes increase of bike ridership, ranging from +21% up to +171% [5]. At the same time, in city centers a trip made by bike can take less time than using a car [6]. Furthermore, studies measuring street volume capacity show that the number of people who can be transported in a 3.5 m wide lane per hour on a bike ranks third most after trains and walking: bikes are able to move over 14,000 people, with trains moving 22,000 and walking able to handle 19,000 people. All of these rank above buses and cars, which are able to only accommodate 9,000 and 2,000 people, respectively [7].

The technologies that currently exist to measure bike-lane quality are few, and in our opinion, insufficient. The first approach uses an expensive array of lasers to remotely sense and

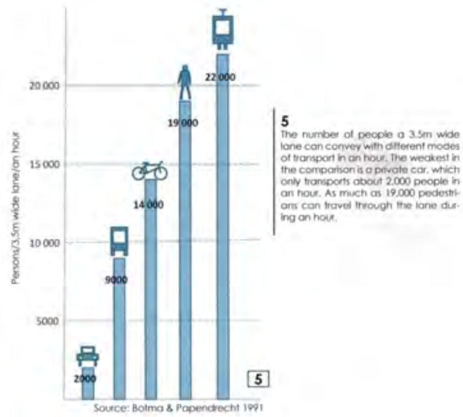

Figure 1. Space efficiency comparison. Bike is the third most efficient mode of transit.

detect pavement imperfections and defects [8]. While this approach offers a comprehensive and precise result, the main disadvantage is in the cost and time required to perform this assessment. Only the wealthiest and biggest city governments are able to afford it, and if they are able to afford, this survey can only be performed once every few years limiting what the city can do with the data.

Another technology that has been proposed is a crowdsourced approach using inexpensive accelerometers called Street Bump. Street Bump was developed in the Boston Mayor's Office of New Urban Mechanics [9]. A limitation with this approach is that is designed for car owners, potentially biasing results. Another limitation of using Street Bump to survey city streets is the absence of street imagery. Without imagery to backup the data, Boston has no way of knowing the "ground truth" of street quality. The Street bump dataset appears to be contain only accelerometer readings that profile a small section of the road and can suffer from calibration issues.

The SQUID approach extends Street Bump as it operationalizes the common adage, "A picture is worth 1,000 words" [10], and if pictures of the entire width of a street segment were captured, street quality for the entire width of the street can be measured in manner that is holistic and empirical.

Figure 2. A 2x2 showing where SQUID-Bike fits

NYC Bike Lane Profiling

This project uses the DOT Bike Lane Shapefile [11] as a primary source of local administrative data. this data is provided to the public for free by the New York City Department of Transportation and is also used to publish the annual NYC Bike Map, a physical map booklet. Over 375,000 are distributed annually at bicycle shops, libraries, and schools. DOT also distributes geographic data of NYC bike routes through shapefiles. This dataset contains information on length, Borough , class, street from and street to. This data is integrated with data collected from the Open Street Cam application using the spatial join method. A summary of our findings include:

NYC, as measured by individual segments, has 22,757 bike lanes in total. 4,569 bike lanes in Brooklyn, 4,044 bike lanes in Manhattan, 3,328 bike lanes in Queens, 2,405 bike lanes in the Bronx, and 634 bike lanes in Staten Island. There 13 bike lane types.

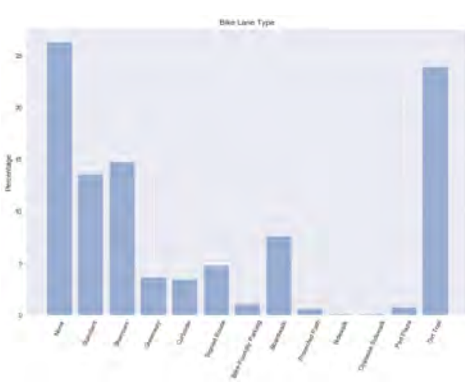

Figure 3. The types of Bike Lanes as classified by the NYC DOT shapefile

This analysis also includes an integration with the NYC Neighborhood Tabulation Area shapefile that contains boundaries of neighborhoods in NYC.

A neighborhood score is calculated based on the type of Bike Lane, and then normalized the score by the length of the bike lane.

Using this score, the top ten neighborhoods that have the best bike lanes are:

1) West Farms-Bronx River, Bronx
2) Soundview-Bruckner
3) Old Astoria, Queens
4) Stuyvesant Town-Cooper Village, Manhattan
5) Lower East Side, Manhattan
6) Whitestone
7) Laurelton
8) Hunts Point
9) Mott Haven-Port Morris
10) Brownsville, Brooklyn

Figure 4. Profiling NYC's Bike Lanes By Neighborhood

Data description

The OpenStreetCam data originates from 3 sensors on a smartphone to:

- The Camera to capture imagery at a set frequency.
- Accelerometer to collect ride quality data at a set frequency.
- GPS to collect location data at a set freqeuncy

Open Steet Camera is designed by the company Telenav. Telenav designed OSC to gather street imagery for the OpenStreetMap platform to offer open sourced street imagery data for several mapping applications.

Figure 5. Cellphone running the OpenStreetCam app mounted on handle

The OSC data is stored in the cellphone and uploaded to OSC's website and servers. A restful API allows us to query this data based on a trip id identifier. GPS and accelerometer data are stored in a text file on the web and the pictures are stored on the OpenStreetCam server, which is accessible from a URL provided by the API query.

Example of data on Open Street Cam after uploading

The OSC app reads accelerometer data once every 0.01 seconds (100 Hz), GPS position every 1 second (1 Hz), and takes pictures every 2 seconds (0.5 Hz). This changes as a function of the velocity of the bike. When the bike is stationary, no pictures and accelerometer data are captured.

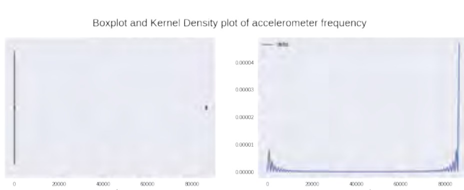

Figure 6. This plots show the frequency in seconds of the measurements for the accelerometer in a trip. 75% of cases are below 0.01 seconds. Some outliers have higher frequency.

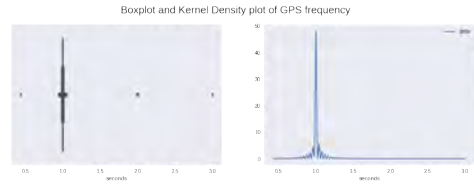

Figure 7. This plots show the frequency in seconds of the measurements for the GPS readings in a trip. Some outliers have higher frequency.

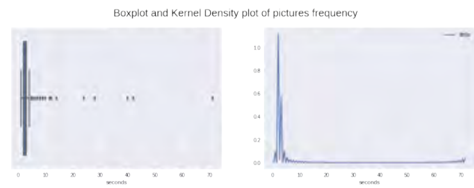

Figure 8. This plots show the frequency in seconds of the measurements for the camera readings in a trip. Some outliers have higher frequency.

METHODOLOGY

Introduction

One of the tasks of this work is to measure how "*fluid*" a bike ride is as a proxy for the overall quality of the bike-lane using just accelerometer data. To this end, a cell phone running Open Street Cam is mounted to the handlebars of a bike and used the phone's sensors to measure the activity in 3 axes (X, Y, Z). Fluidity implies a riding experience that has the least amount of abrupt shocks along the way. An highly fluid bike ride is a straight line along a bike lane, with no interruptions, no bumps, no sudden shifts in directions or the need to suddenly apply the brakes. Large variations in accelerometer readings indicates that a bike ride along a street segment is not fluid at that point. We also assume that the quality of the bike ride depends heavily on the overall quality of the bike-lane where that bike rides takes place.

Accelerometer data

An accelerometer "is a device that measures proper acceleration; proper acceleration is not the same as coordinate acceleration (rate of change of velocity) [12]. For example, an accelerometer at rest on the surface of the Earth will measure the acceleration due to Earth's gravity of g [?] 9.81 m/s^2 in the up direction. By contrast, accelerometers in free fall (falling toward the center of the Earth at a rate of about 9.81 m/s^2) will measure zero". We refer the Starlino accelerometer guide [13] on describing how to use accelerometer data. To better understand this type of data, It is useful to imagine a box with a ball inside. Next, imagine that each wall is pressure sensitive. If we suddenly move the box to the left (we accelerate it with acceleration 1g=9.8m/s^2), the ball will hit the wall X-. We then measure the pressure force that the ball applies to the wall and output a value of -1g on the X axis. Please note that the accelerometer will actually detect a force that is directed in the opposite direction from the acceleration vector. This force is often called Inertial Force or Fictitious Force .

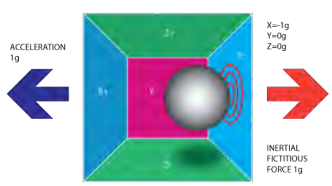

Figure 9. Accelerometer values in the X-axis

But acceleration is not the only concern. There is another that is of special interest to our bike riding measurement case. The accelerometer measures acceleration indirectly through a force that is applied to one of the walls (according to our model, it might be a spring or something else in real life accelerometers). This force can be caused by the acceleration , but as we'll see in the next example it is not always caused by acceleration.

If our model is placed on the earth's surface, the ball will fall onto the Z- wall and will apply a force of 1g on the bottom wall. If the accelerometer is laid on top of a table, this will measure 9.81 m/s^2 or -1g. In this instance, the box isn't moving but a reading of -1g on the Z axis is still observed. The pressure that the ball has applied on the wall

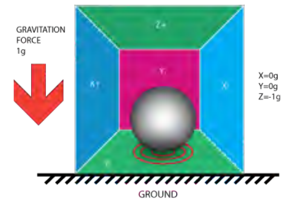

Figure 10. Accelerometer values in the Z-axis

was caused by the gravitational force. This is meant to illustrate that in essence accelerometer measures force not acceleration. It just happens that acceleration causes an inertial force that is captured by the force detection mechanism of the accelerometer. In this example, if the table is struck from the bottom, an abrupt shock is registered on the Z axis. This reading will look like this:

Figure 11. Accelerometer measure of a shock in Z axis

Following the previous example for the case of measuring street defects, the X-axis is considered when we turn the handle swiftly left or right or on shocks. A sudden shift in lateral direction implies a lack of fluidity of the bike-lane, and affects the overall quality of the bike ride. It is therefore within relevant to include abrupt shocks in the X axis.

The Y axis considers the direction of the bike's velocity. A sudden application of the brakes will cause an abrupt shock to the Y axis. This could include a sudden brake due to a pothole, an obstacle , or a cause related to the fluidity of the bike-lane. In all of these cases, is it important to measure the particular braking event. However, the event could also be due to a cyclist braking at a red light. To avoid this, the Y-axis measurement can be weighed. A sudden acceleration will return the same abrupt shock measurement but this type of activity is not expected with bikes due to limited speed capabilities. Precisely measuring ride quality involves registering measurements that are related to abrupt shocks or swift changes in direction as opposed to slight changes over elongated periods of time.

Our objective is to develop a robust measurements using all 3 axis of the accelerometer that adequately describes abrupt shocks and changes as a method to infer the fluidity of bike ride.

The Ride Quality (V) statistic

A key objective is to allow for a single measure of overall ride quality. The difference between the measurements along 3 axes at time t (V_t), against the measurement on the same axes for the previous time period ($Vt-1$) is considered. To this end, the vector magnitude of these 3 values give us our Ride Quality statistic.

Another way to look at this is to think the problem as an euclidean distance in a n-dimensional space [14]. So our final vector is compounded of the square difference in all of the three axis:

$$V = \sqrt{(x_t - x_{t-1})^2 + (y_t - y_{t-1})^2 + (z_t - z_{t-1})^2}$$
)

Another task is to match a given picture and the defects it captures to the stream of accelerometer readings taken between location and image data points. To do this, all V values are summed and assigned to a specific location point that also reference an image.

The resultant structure takes on the following form:

Row ID	Location (Lat,Long)	Image ID	Ride Quality metric
1	lat,long	link to image	V statistic

Table 1
The data structure combining Location, Ride Quality and Imagery from OSC

Data Cleaning - Snapping GPS points to a line

When using a smartphone as a primary data collection device, there can be some amount of error or noise associated with location data. There are a number of reasons for imprecise location data such as the GPS sensor not having a lock on a minimum number of satellites, current weather conditions or in the particular case for cities with high rise buildings, the Urban Canyon affect. [15]

Location data containing a "cloud of points" around a street instead of being on than the street centerline can severely affect results. To remedy this, a "snapping " process has been developed to ensure that readings that stray from the street or bike lane are corrected.

Shapely [16], open source python package for spatial data manipulation contains a function *interpolate()* that allows for snapping points to the nearest line using linear referencing. An issue with this approach is that there may be several competing lines for a given point (ie. all bike-lanes in the vicinity). The objective is to snap a point onto the bike lane that is currently being ridden.

One approach could be to simply choose the closest line. However, that would require the calculation of distances from every point to every line which would be computationally very expensive. Instead, a buffer is generated around each bike-lane segment and then spatially join the points to that buffer. In addition to being more efficient, this method has the advantage of mapping of every point to a bike-lane segment.

The disadvantage is that a point can be joined to more than one bike-lane segment. For example, at a street intersection/corner with 2 bike-lanes, a point could theoretically belong to both intersecting bike-lanes. To correct for this, we choose the same bike-lane lane as the previous point.

Figure 11 illustrates the interpolation in action where the black colored dots are imprecise GPS points and colored dots are points snapped on to the street or bike lane. A clear limitation is that points too far away from the bike-lane (outside the chosen buffer) will not have a bike-lane assigned to it, and therefore they won't have a new point on the bike-lane. In the intersections, we can have competing candidate's bike-lanes and therefore, points may be assigned incorrectly.

Figure 12. Map showing misassigment of a point to a bike-lane

Bike Lane Imagery analysis

Data collection includes images of bike lanes as taken by the OSC app. The next key objective towards a unified quality metric is to evaluate the images to measure bike lane conditions. Two approaches are implemented . The first approach uses Microsoft's Custom Vision [17], to classify bike lane imagery. The second approach uses a custom built algorithm that uses open source computer vision and image processing functions.

- **Classifying Bike Lane Images using Microsoft's Computer Vision**

Microsoft Computer Vision is an online platform that allows users to label custom imagery as a training data set and then train a custom classifier on this data. A training process then reveals a prediction URL that can be used on test data to reveal probabilities of which labels best describe the test data.

For this project, a thousand images were uploaded and manually tagged as 'good', 'bad', or 'acceptable' as training data. A custom classifier was built on top of this training data and used to classify incoming bike lane imagery as in "good", "bad", or "acceptable" condition.

Figure 13. Microsoft's Custom Vision explained.

Figure 14. Tags used for labelling bike lane imagery in the Custom Vision product

Figure 15. Custom Vision - Image Prediction Result

Figure 16. Custom Vision - Good Image Prediction Result

Figure 17. Custom Vision - Bad Image Prediction Result

- **Classifying Bike Lane Images using Python's Image Processing and Computer Vision Libraries**

The Python programming language has a number of libraries that can implement powerful computer vision and image processing techniques — openCV [18] and SciPy [19] are the more widely known ones that we use here.

We attempt to measure bike lane quality by combining four approaches and borrowing from NACTO's "Urban Street Design Guide" [20] :

- **Ride Quality** as measured using cellphone accelerometer data from OpenStreetCam, as outlined above.
- **Color of the Bike lane** — Green lanes are higher in quality to un-painted bike lanes as they are more visible to bike riders.
- **Symbols and pavement markings** — Well marked bike lanes are higher in quality to poorly marked or unmarked bike lanes.
- **Visible street defects**, such as cracks and potholes. (*the fewer, the better!*)

It is important to note that this is a definitive scoring system and represents a first iteration of this process.

Detecting Color Features

According to [20], the best bike lanes are painted green. The green paint allows the bike lane to be more conspicuous and implies a separation from cars, and thus hopefully leading to greater space and safe space between bicyclists and automobiles.A first step in evaluating bike lane quality is classifying bike lane color.

When reading an image into openCV, they are stored as a color matrix using an RGB model. This is convenient, as it makes it possible to treat the image as a matrix of numbers to build a function that returns a numeric "color score". Our goal for this color score therefore is to calculate how "green" the bike lane image is. The hard part in this process is how to define this "bike lane green", since the bike lane green is a very specific color.

However, we should include a range of green colors in order to account for variations in the lighting of the images captured. In this step, we use image processing software (photoshop and gimp) to find the RGB components for the color of a sample bike lane and set lower and upper bound according to the RGB components. With lower and upper bounds in place, we can create a mask and perform a color filtering, leaving only the target green color behind. Which means that the resulting image's matrix after filtering will have non-green values set to 0, thus we can calculate the difference between original image and the resulting image. The more green kept, the more components kept and the higher the sum will be. The sum of the number of green pixels is our color score.

To summarize:

1) Convert image to HSV color model matrix.
2) Create color bound for "bike lane green".
3) Filter out colors that are not in the range.
4) Calculate color score based on total sum of green pixels after filtering

Detecting Line Markings

1) **Reading** the original image using SciPy.
2) **Cropping** the image to the lower half (*to remove non-bike lane features*), and clip the bottom 10% of the picture as well (*to remove my bike tire*)
3) **Filtering** the cropped image to only white pixels using a color threshold (any pixel that has a value greater than 200 in each of its red, green, and blue channels)

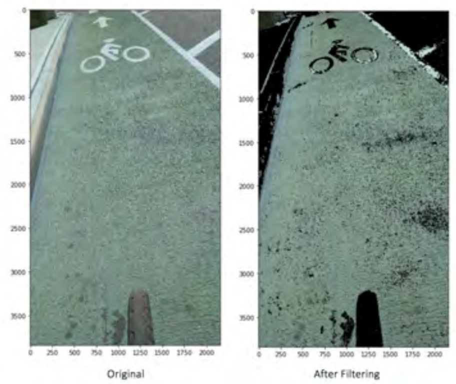

Figure 18. ColorScore = 553

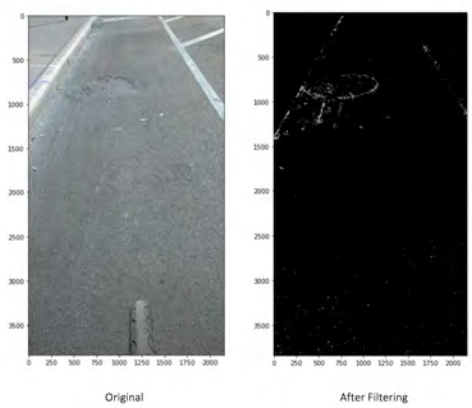

Figure 19. ColorScore = 138

4) **Blurring** the cropped+filtered image using gaussian blur, with a bandwidth value of 40 pixels.
5) **Binarizing** the cropped + filtered + blurred image i.e. setting the pixels that are lighter than blurred image all the way to 1, set all other pixels to 0.

Figure 20. Original Image

The final lane marking score of this image

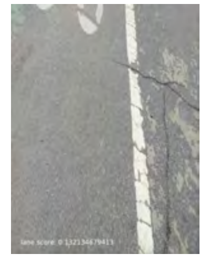

Figure 21. Cropped Image

Figure 22. Filtered Image

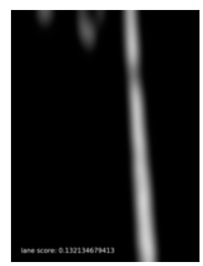

Figure 23. Blurring, after filtering

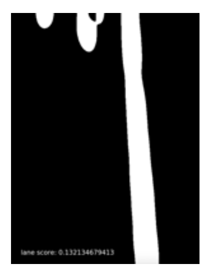

Figure 24. Binarizing, after Blurring and Filtering.

(cropped + filtered + blurring + binarizing the original image) is counted as a percentage value of white pixels aka lane markings.

In this case, 13.2% of the picture is filled by lane markings.

The final lane score is the percentage of this last picture that is filled by lane markings. In this case, 13.2% of the picture is filled by lane markings — this is a naive scoring system, certainly, but this is the first iteration.

Detecting Defects

1) **Reading** original image using SciPy.
2) **Cropping** the image to the lower half (*to remove non-bike lane features*), and clip the bottom 10% of the picture as well (*to remove bike tire from the view*).
3) **Filtering —** In this case, a median filter is applied to preserve edges.
4) **Converting** the image from RGB to HSV and filter image to preserve only darker (defect) pixels.
5) **Edge detection** using the canny edge detection.
6) A 3x3 gaussian filter is applied over the image and then perform erosion / dilatation operations on remaining white pixels to remove noise.

Figure 25. Original Image

Figure 26. Cropped Image

The final defect score of 6.95% is the percentage of white pixels in the image that remain after the process above. These algorithms can be improved to consider variations in light

Figure 27. Filtering the cropped image

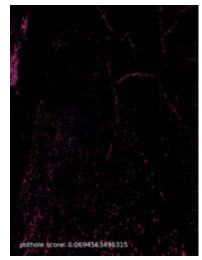

Figure 28. Conversion from RGB to HSV

Figure 29. Detecting edges on the HSV image

or camera angles. An evolution to these techniques may also include building a convolutional neural network and run each of our images through this network.

Data Workflow

Figure 30. The automated workflow powering this analysis

RESULTS AND IMPLICATIONS

The SQUID Bike project has demonstrated in this paper, an automated data workflow to interrogate the general conditions of New York City's Bike lane infrastructure. The authors have collected bike lane imagery data in excess of 50 miles consisting of over 5000 distinct images and over 500,000 accelerometer readings.

The results show that with further work, a robust standard to measure bike lane infrastructure for any city in the world can be made possible using a low-cost approach assuming that bike lane imagery is collected in a purposeful manner. Furthermore advances in Computer Vision may also lead to a fully automated inspection process.

The implications of a fully automated inspection process is that cities can collect high quality, ground truth data about their bike lane infrastructure quickly in a cost-effective manner. We argue that adopting such a practice empowers a city to enter into an anticipatory paradigm for bike lane maintenance and allow transportation agencies be more responsive to the 2-wheeled commuter.

LIMITATIONS AND IMPROVEMENTS

Various constraints involving hardware and software are limitations to this work in this current state. Most of which are solvable with further work. Image Quality, Battery life, and accelerometer discrepancies across phone models are some limitations on the data collection side, whereas inaccuracies in snapping points to bike lanes, among other general GPS inaccuracies, are concerns on the data analysis side. In the following paragraph we will explain how these limitations affected our analysis, how we overcame them and, in the rare cases we didn't, how we plan to solve them in the future.

Limitations to data collection

Most of the complexity around collecting data is around the collection of good quality bike lane imagery. While using a purposeful mount helped in acquiring stable imagery, going over very rough stretches of road or going over deep potholes created vibrations enough to loosen the mount and introduce additional movement of the phone which causes distortions to the imagery and accelerometer data. These situations were few and far between and constituted a small number of data points. Upon using a stable moun, this issue was successfully mitigated.

Recording images using OpenStreetCam consumes a lot of battery charge. A 30 min bike trip tends to drain close to half the battery of a standard Android phone. For this reason, the crowdsourcing approach for collecting data may be limited. Using an external battery pack will mitigate this isue but will require additional mounting on the bicycle.

Correcting for variance in accelerometer readings for the same defect from different devices is another limitation. However, these inconsistencies weren't high and the overall magnitudes for the accelerometer readings tended to be similar. This maybe mitigated by assuming some calibration settings for commonly used phones.

Limitations to analytics

Limitations to analytics include the precision of GPS data and the quality of the underlying shapefile. The naivete of the computer vision remains as the single largest limitation which can be improved with further maturation of computer vision and image processing techniques.

CONCLUSION

Expanding and maintaining a safe and comfortable bike lane infrastructure is key to supporting a progressive and climate friendly transit agenda. We believe that cities need to quickly adapt their core operations to deliver sustainable transit futures. Maintaining a sizable network of bike lane infrastructure is key to achieving these goals and contributing to overall improvements in quality of life indicators. To that end, we believe that SQUID-Bike is very relevant to realizing transit goals by empowering city agencies be more responsive and do more with fewer resources. Furthermore SQUID-Bike is also a tremendous opportunity for cities to engage with multiple stakeholders and pioneer a participatory inspection process where everyday citizens can be directly involved and actively contribute to city maintenance operations.

AUTHOR CONTRIBUTIONS

Felipe Gonzalez worked on data engineering (data gathering, cleaning, snapping function, overall coordination of python's processes)

Nicola Macchitella worked on data engineering (data gathering, GIS analysis and overall coordination of python's processes)

Geoff Perrin worked on the computer vision code (lane markings and defects classification), Tableau visualization, as well as setting up the amazon web service database and the code that pushed our data to the database.

Sichen Tang worked on data analysis to profile NYC's Bike Lane infrastructure, implement computer vision techniques (color score) in addition to implementing other front end components.

REFERENCES

[1] W. Hu, "More New Yorkers Opting for Life in the Bike Lane," *The New York Times*, jul 2017. [Online]. Available: https://www.nytimes.com/2017/07/30/nyregion/new-yorkers-bike-lanes-commuting.html

[2] V. Adibhatla, S. Fan, K. Litomisky, and P. Atwater, "Digitizing Municipal Street Inspections Using Computer Vision," *arXiv:1609.09582 [cs]*, sep 2016, arXiv: 1609.09582. [Online]. Available: http://arxiv.org/abs/1609.09582

[3] Telenav, "OpenStreetMap." [Online]. Available: http://blog.improve-osm.org/

[4] G. Rudinger, S. Poppelreuter, and K. Donaghy, *Social dimensions of sustainable transport: transatlantic perspectives*, ser. Transport and society. Aldershot, Hants, England ; Burlington, VT: Ashgate, 2005. [Online]. Available: https//catalog.hathitrust.org/Record/010629564

[5] C. Monsere, J. Dill, N. McNeil, K. J. Clifton, N. Foster, T. Goddard, M. Berkow, J. Gilpin, K. Voros, D. van Hengel, and others, "Lessons from the Green Lanes: Evaluating protected bike lanes in the US," 2014. [Online]. Available: http://pdxscholar.library.pdx.edu/cengin_fac/144/?utm_source=pdxscholar.library.pdx.edu%2Fcengin_fac%2F144&utm_medium=PDF&utm_campaign=PDFCoverPages

[6] A. Faghih-Imani, S. Anowar, E. J. Miller, and N. Eluru, "Hail a cab or ride a bike? A travel time comparison of taxi and bicycle-sharing systems in New York City," *Transportation Research Part A: Policy and Practice*, vol. 101, pp. 11–21, jul 2017. [Online]. Available: https://doi.org/10.1016%2Fj.tra.2017.05.006

[7] H. Botma and H. Papendrecht, "TRAFFIC OPERATION OF BICYCLE TRAFFIC," *Transportation Research Record*, no. 1320, 1991. [Online]. Available: https://trid.trb.org/view.aspx?id=365588

[8] E. Schnebele, B. F. Tanyu, G. Cervone, and N. Waters, "Review of remote sensing methodologies for pavement management and assessment," *European Transport Research Review*, vol. 7, no. 2, p. 7, jun 2015. [Online]. Available: https://link.springer.com/article/10.1007/s12544-015-0156-6

[9] City of Boston, "New Urban Mechanics," 2017. [Online]. Available: http://www.newurbanmechanics.org/

[10] Tess Flanders, "Speakers Give Sound Advice," *Syracuse Post Standard*, mar, year = 1911.

[11] Department of Transportation, City of New York, "NYC DOT - Data Feeds," 2017. [Online]. Available: http://www.nyc.gov/html/dot/html/about/datafeeds.shtml

[12] Wikipedia, "Accelerometer," jun 2017, page Version ID: 787517520. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Accelerometer&oldid=787517520

[13] starlino, "A Guide To using IMU (Accelerometer and Gyroscope Devices) in Embedded Applications. ≪ Starlino Electronics." [Online]. Available: http://www.starlino.com/imu_guide.html

[14] H. Lab, "'n'-Dimensional Euclidean Distance." [Online]. Available: https://hlab.stanford.edu/brian/euclidean_distance_in.html

[15] Mat Gordon, "Bad GPS Data: What/Why/How," oct 2013. [Online]. Available: http://support.strava.com/hc/en-us/articles/216917707-Bad-GPS-Data-What-Why-How

[16] S. Gillies, A. Bierbaum, and K. Lautaportti, "Shapely documentation." [Online]. Available: http://toblerity.org/shapely/

[17] Microsoft Azure, "Custom Vision Service | Microsoft Azure," 2017. [Online]. Available: https://azure.microsoft.com/en-us/services/cognitive-services/custom-vision-service/

[18] Open CV Team, "OpenCV library," 2017. [Online]. Available: http://opencv.org/

[19] SciPy developers, "SciPy.org," 2017. [Online]. Available: https://www.scipy.org/

[20] National Association of City Transportation Officials, "Urban Street Design Guide." [Online]. Available: https://nacto.org/publication/urban-street-design-guide/