CITY-WIDE ROAD DISTRESS MONITORING WITH SMARTPHONES

Christoph Mertz, Srivatsan Varadharajan, Sobhagya Jose, Karan Sharma, Lars Wander, and Jina Wang

Carnegie Mellon University

5000 Forbes Ave

tel.: 412-2687955

cmertz@andrew.cmu.edu, svaradha@andrew.cmu.edu, sobhagyajose@gmail.com, karans@andrew.cmu.edu, lars.wander@gmail.com, jina.jinan@gmail.com

ABSTRACT

In this paper we present a smartphone-based system that can assess road damage by determining the amount of cracks in the road. The smartphone can be mounted in vehicles like personal cars, buses and garbage trucks that travel the roads on a regular basis. It collects images and tags them with GPS and other information, which is then either displayed to inspectors or automatically analyzed to give a road distress score. With this system the whole street network of a city can be continuously monitored at a much lower initial and operating cost than traditional methods. A pilot test of this system is on its way in the City of Pittsburgh and first results are shown.

Keywords: *Road damage, machine vision, road monitoring, smartphone*

INTRODUCTION

State and local maintenance departments are tasked with keeping roads in good repair, which includes monitoring the roads to detect the presence of cracks, potholes, and other distress. Currently, this is done by inspectors, who drive out to each location and record the conditions on paper. This is a tedious and often inconsistent process. Using specialized vehicles outfitted with sensors [1] can automate the data collection process significantly. However, these vehicles are expensive, costing as much as $500,000. Another way of collecting data is through citizen reports; but these are often only about severe and acute problems. In this paper, we present a system to monitor the condition of roads on a continuous basis, at low cost, with consistent quality, and with minimal human intervention.

Our approach involves using images or videos collected by commodity devices such as smartphones to obtain information about the road condition. These devices can be mounted in service-vehicles like cabs, garbage trucks, police cars, etc., which already make regular rounds on city roads for other purposes. Therefore no dedicated vehicles or drivers are needed, which significantly reduces the cost of the system. We also present a machine vision analysis pipeline to analyze such images automatically, and pass the resulting distress scores to the asset management system.

There have been other approaches to detect road damage using automatic analysis. [3] and [5] propose using smartphone-based data collection to detect potholes, but in contrast to our approach, they use the accelerometer disturbances caused by riding e.g. over potholes to detect them. Our approach, instead, is based on vision and can therefore be extended to cover visually recognizable pavement conditions such as cracking, which may often not cause any significant disturbance in the accelerometer data. [6], [7] and [8] describe systems using image analysis to detect cracks like we do. However, these works have as a key limitation that the camera has to be mounted facing the road, somewhere outside the vehicle. This requires weatherproofing, damage protection for the camera and its mounting, experts for installation and regular maintenance, all of which significantly increase the cost and complexity of the system. In comparison, our method involves mounting the camera inside the vehicle on the windshield, using a very simple mount. We move the complexity from the hardware to the software component through our intelligent analysis algorithm.

DATA COLLECTION

Data collection with smartphone and app

The data collection setup consists of a smartphone running our data collection app. The smartphone is mounted on the windshield of the vehicle with a suction cup and is powered by the cigarette lighter (Figure 1). The camera can easily be removed; it is screwed on a quick-release mount. A black shield extends from the mount to the windshield in order to block reflections. The driver can start the app at the beginning of the trip and have it record data throughout the whole trip. The app collects time-stamped images or videos of the road scene as viewed from the driver’s perspective, along with GPS, accelerometer, gyro, orientation, and timing data.

Figure 1 Data collection system: The smart camera is connected to a quick-release mount and fastened to the window with a suction cup. The camera is powered from the cigarette lighter. One can see the collection app running on the system.

Mounting on service vehicles

Figure 2: Service vehicles that can be used for data collection

One of the key ideas behind our data collection system is that it can be easily mounted on any vehicle, especially those that drive on the roads on a regular basis, e.g. garbage trucks drive through every neighborhood once a week. It is therefore possible to collect data frequently without the need for a dedicated vehicle and driver. The amount of human involvement in the data collection process is very minimal. The drivers only need to launch the app and press the start button on the app when they start driving and press the stop button when they finish the drive. Because the hardware is inexpensive, such systems can be set up on multiple vehicles at a reasonable cost.

Tabletop inspection

The collected images are already useful for maintenance as they allow for inspection of the roads without having to visit the roads in person. The images can be displayed in the asset management system of the department or with free software. An example is shown in where the data is displayed on Google Earth (Figure 3). The inspector can see where images are available and then display images of the chosen road by clicking on the yellow arrows. It is also possible to select images from past years by adjusting the time slider in the upper left corner of the display.

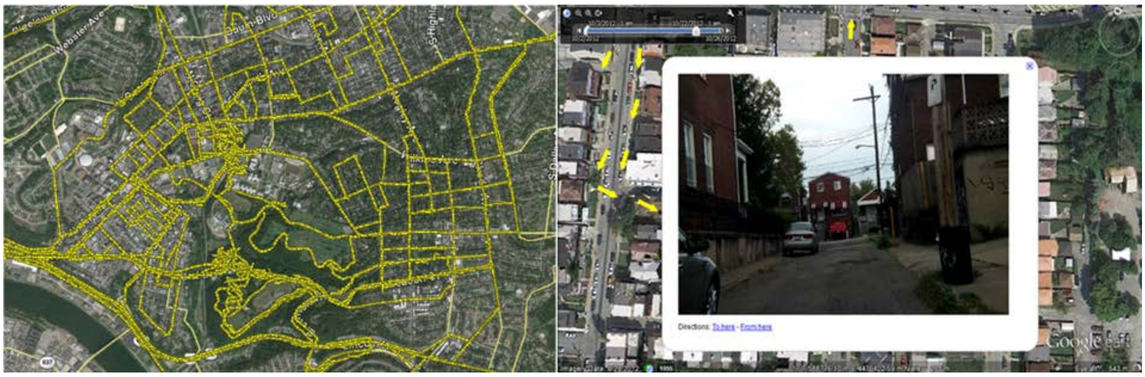

Figure 3 Example of road image displayed on Google Earth. Left: The streets where images were collected are shown in yellow. Right: The small yellow arrows on the street are markers pointing in the driving direction. When clicking on the marker the corresponding image appears.

Selecting images suitable for analysis

The proposed data collection setup allows us to collect information about each location over multiple runs within a week or a month. This redundancy allows us to discard data that is difficult to analyze and keep only those images that can be analyzed easily. For instance, bright sunshine, shadows and precipitation while imaging often cause machine vision algorithms to fail. Since we have the GPS and time-stamp for each image, by querying public weather sources, we obtain the weather information for each image. Within each 10m window, we select the image that is captured during the daytime, under cloudy or overcast weather, and with the least blur. Such images are the most ideal for automatic as well as manual analysis.

ANALYSIS

The automatic analysis of the images to determine the amount of damage of the road is an interesting and challenging computer vision problem. Our companion paper [2] describes the computer vision system in detail. The objective of the computer vision system is to take an input image from our data collection setup and identify the regions in the image containing cracked road areas. Figure 4 shows on the left a typical image with cracks and on the right patches with cracks where the intensity of red indicates the severity of the cracks.

Figure 4 - Left: A typical road image with cracks in the road. Right: Classification result. The color indicates the severity of distress: blue = no cracks, light to dark red = light to severe cracks.

The vision analysis consists of multiple stages. A high-level description of each stage is given in the following paragraphs.

Ground detection

The first step in analyzing the image consists of detecting the pixels belonging to the ground region. Since we are interested in only finding cracked regions on the road, any area that does not belong to the ground does not need to be analyzed. To detect the ground region, we use the Geometric Context detection algorithm described in [4]. This algorithm takes an input image and produces a mapping as shown in Figure 5 (left), dividing the image into three regions: ground, vertical structures and sky.

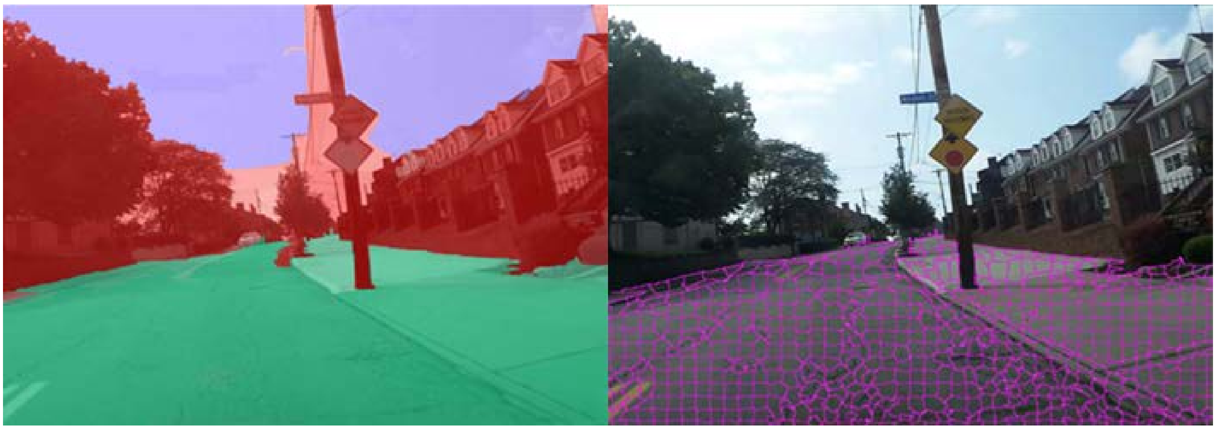

Figure 5 - Left: Ground detection. The ground is colored green, vertical surfaces red, and the sky blue. Middle: Segmentation (superpixels) of ground region.

Segmentation

The second major stage involves grouping the pixels belonging to the ground regions into larger segments, called *"superpixels"*, with the property that within each *superpixel*, all pixels have similar color and local texture. Performing the analysis at the superpixel level is usually more accurate and faster than pixel level analysis. Figure 5 (middle) shows the superpixels created on the ground region.

Computing features and classification

Once we have divided the ground region into superpixels, we use a machine learning classifier to build a model of the cracked-road and non-cracked or non-road superpixels from a sub-set of images that were manually annotated (see Figure 6). Using these models, we can make predictions on superpixels in any image, as to whether they contain cracked-road regions or not. The model is created by extracting information like color, shape and texture from each superpixel. For a detailed description, see [2]. The machine learning classifier outputs a value in the range of 0 to 1 for each superpixel, which indicates the probability that the superpixel contains cracked-road areas. Figure 4 (right) shows the classification results for each superpixel. The superpixels for which the classifier output is close to 1 are shaded red, while the ones for which the output is close to 0 are shaded in blue; the other values in between correspond to varying lighter shades of red.

EXPERIMENTS

Summary of collected data

We have mounted the system on a personal vehicle and collected data for about two years. The routes included daily commutes and other typical day-to-day driving, together about 250 hours of driving on 600 miles of unique roads. Using our image selection procedure, we automatically selected a subset of the data with favorable conditions (daytime, overcast, no precipitation, least blur).

Collecting annotations using Mechanical Turk

To train and test the automatic analysis system, we selected a subset of around 400 images. We obtained annotations for these images, marking the cracked and not-cracked regions (as described in the previous section), in each image. These annotations were obtained by crowd-sourced data collection using Amazon's Mechanical Turk service. Multiple people were asked to label each image. Figure 6 shows an example of the positive and negative labels obtained through crowdsourcing.

Figure 6 - Top: Positive labels (road cracks) as drawn by multiple annotators, Bottom: Negative labels (non-cracked ground regions) as marked by a single annotator

Performance of automatic analysis vs. human annotators

The labeled images were divided into training and test sets: the set of training images were used by the algorithm to build a model of road distress regions while the set of test images were used to evaluate the accuracy of the algorithm. We compare the performance of the vision system to the human accuracy, which we computed by considering how often each human annotator's labels agreed with the majority of the remaining human annotators. We found that compared to other visual recognition tasks like recognizing a car or a pedestrian, identifying cracked regions on the road was a subjective task – in the sense that the human annotators tended to disagree with each other more often, indicating that the "extent" of cracked regions are decided by each person very differently. The algorithm's performance came quite close to that of the overall human performance. Some of the resulting segmentations from the algorithm are shown in Figure 7. The failure cases shown in Figure 7 are caused by weak texture cues (bottom-left and bottom-right) and the presence of certain markings on the road that may have been missing in the training data (bottom-middle).

There were also some failures when there were objects on the road, when the vehicle came to a "T" junction, or when there were shadows or wet spots on the road. The last ones are not a failure of the cracks finding algorithm itself, rather a failure of the pre-selection. In future versions of our system we will change the automatic pre-selection so that we only select images when the sky is fully overcast, it has not rained in the hour before the image was taken, and when the road does not end in front of the vehicle.

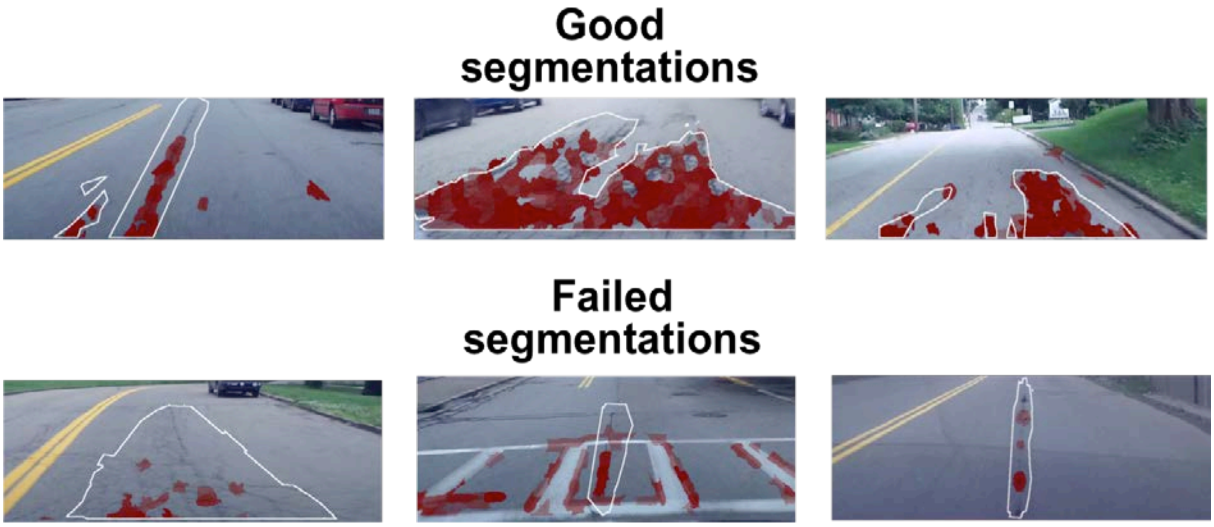

Figure 7 Some segmentations produced by the vision system - The white boundaries indicate the true extent of cracks obtained from the majority of human annotators.

The complete algorithm takes about a minute to run for each image on a desktop workstation with an Intel Core i7 quad-core processor and 8GB of memory. Work on optimizing the most computationally intensive sections of the vision system is currently in progress.

CITY-WIDE RESULTS

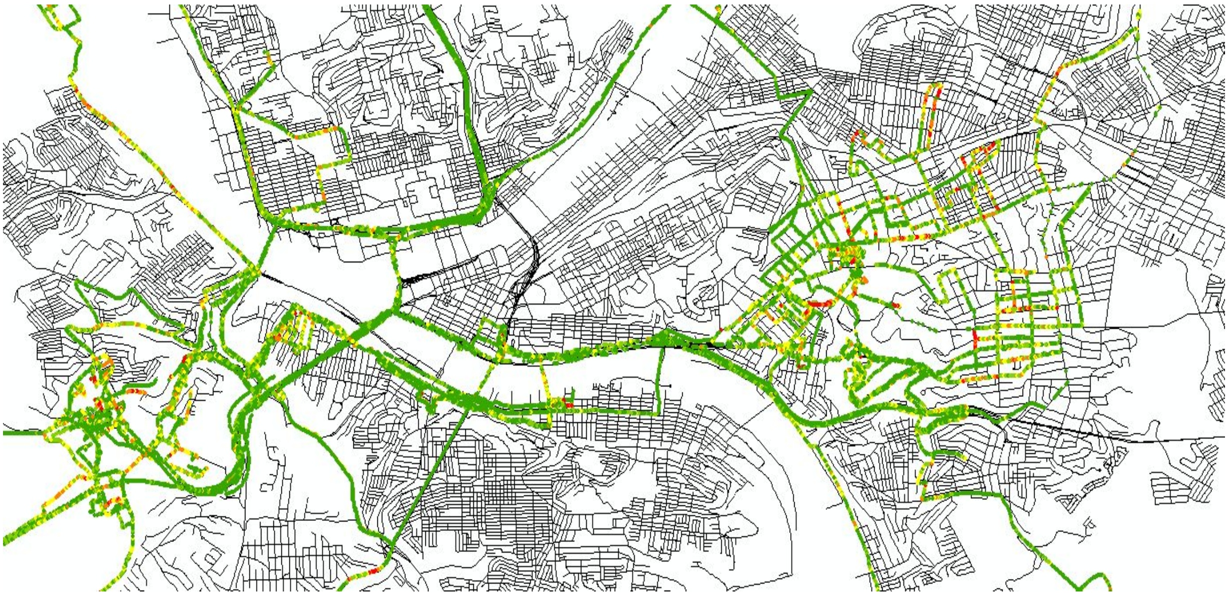

Figure 8 - Distress score overlaid on a GSI street map of Pittsburgh. Green, yellow, and red are low, medium, and high amount of cracks, respectively.

From the classifications, we calculate a distress score for each image as the fraction of the area that is covered by cracks in front of the vehicle. Because we have the GPS for each image, we can plot these scores on a map. We selected around 30,000 images captured in and around the city of Pittsburgh and ran them thorough the automatic analysis. For this data we calculated the distress score, and overlaid the scores on the GIS street-center lines used by the city of Pittsburgh (see Figure 8).

We also received the road ratings from the City of Pittsburgh. They contained the Overall Condition Index (OCI) and rehabilitation dates for each city road segment. We did find correlations between the OCI and our score, however, these two are difficult to compare. OCI contains measures like rutting and raveling whereas our score is only based on cracks. Instead, we show in Figure 9 the relationship of the time since the observed road has been rehabilitated and the average damage score. With this we want to evaluate the crack detection algorithm, so we manually discarded images where the pre-selection failed. It is expected that the road deteriorates over time and the score increases accordingly. One can see that it is indeed the case.

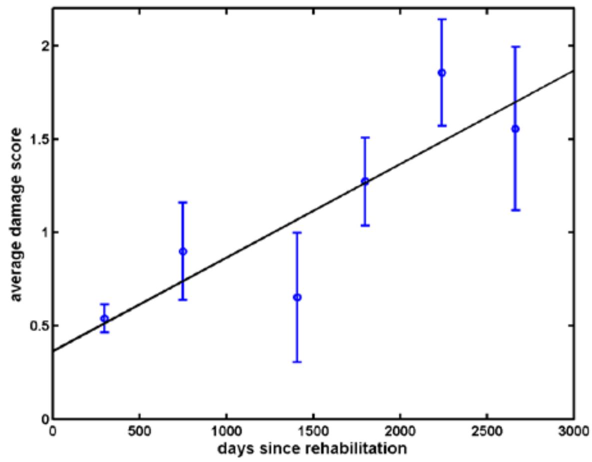

Figure 9 Average road cracks score vs. time since the last rehabilitation of the road. The error bars indicate the uncertainty of the averages.

The variations are large, as indicated by the error bars, because cracks are not evenly distributed along a road. One can find clusters of cracks and undamaged areas on the same road.

We are planning to expand the score to include the amount of potholes and patches by further image analysis and to include amount of shoving etc. by analyzing the accelerometer and gyro data of the smartphone that was collected at the same time as the images. With these measurements we will be able to determine a complete OCI.

CONCLUSION AND OUTLOOK

We have shown that it is possible to build a road distress monitoring system that can evaluate the surface quality of roads on a regular basis, close to human performance and at very low cost. The equipment can be purchased for less than $1000, the labor required is small, and the analysis can be done on a standard computer.

Having completed the development of a prototype system, we are in the process of testing it in a pilot project with the City of Pittsburgh. We will mount the data collection system on their vehicles, analyze the collected data and load the results into their asset management system. First indications are that the system will be very useful for them.

Further into the future we want to expand it into an all-purpose infrastructure inventory and monitoring system. We want it to be able to detect other things like traffic signs, lane markers, vegetation overgrowth or manholes to name just a few. It can also be converted into a system that monitors snow cover or weather conditions live.

Acknowledgement

This research was supported by the T-SET University Transportation Center sponsored by the US Department of Transportation under Grant No. DTRT12-G-UTC11

References

[1] Examples: http://www.pavemetrics.com/, http://www.greenwood.dk/profiler.php, http://www.roadware.com/products/ARAN-Subsystems/, LCMS - Laser Crack Measurement System

[2] S. Varadharajan, S. Jose, K. Sharma, L. Wander, and C. Mertz, "Vision for Road Inspection" IEEE Winter Conference on Applications of Computer Vision, March 2014.

[3] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan. Pothole patrol: Using a mobile sensor net- work for road surface monitoring. In *Proceeding of the 6th international conference on Mobile systems, applications, and services*, June 2008.

[4] D. Hoiem, A. A. Efros, and M. Hebert. Geometric context from a single image. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 654–661. IEEE, 2005.

[5] Tomonori Nagayama ; Akira Miyajima ; Shunya Kimura ; Yuuki Shimada and Yozo Fujino " Road condition evaluation using the vibration response of ordinary vehicles and synchronously recorded movies ", *Proc. SPIE* 8692, Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2013, 86923A (April 19, 2013); doi:10.1117/12.2010074;**http://dx.doi.org/10.1117/12.2010074**

[6] H. Oliveira and P. L. Correia. Automatic road crack segmen- tation using entropy and image dynamic thresholding. In *Proc. European Signal Process. Conf.(EUSIPCO'09)*, pages 622–626, 2009.

[7] H. Oliveira and P. L. Correia. Automatic road crack de- tection and characterization. *Intelligent Transportation Sys- tems, IEEE Transactions on*, 14(1):155 –168, march 2013.

[8] T. S. Nguyen, S. Begot, F. Duculty, and M. Avila. Free-form anisotropy: A new method for crack detection on pavement surface images. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 1069–1072, 2011.