Road Surface Obstacle Detection using Vision and LIDAR for Autonomous Vehicle

Nathalie Joy Sucgang, Manuel Ramos Jr., Nicolette Ann Arriola

Abstract—**The project aims to present a system that detects and estimates road surface obstacles - potholes and speed humps using low-cost camera and LIDAR sensor. Regions are determined using histogram-based data from gray-scale image. Image segmentation and spectral clustering are used for identification and rough estimation of potholes. Speed hump; on the other hand, are detected by integrating the LIDAR measurements in time, relative to the motion of the vehicle. The algorithm is implemented in C++ with Open Source Computer Vision (OpenCV) Library. The pothole detection system can detect potholes 1.6m to 5m away from the vehicle with 86.18 percent accuracy, while the speed hump detection system can detect speed humps at an optimal distance of 4.13m with an accuracy of 98.36 percent. Errors during the detection are due to the algorithms implemented, and hardware limitations. When the two systems are cascaded, the resulting system is reliable with 80.83 percent accuracy in pothole detection, and 98.46 percent accuracy in speed hump detection. However, this is only true if the number of potholes and speed humps to be detected are minimal. The limitation of the cascaded system is imposed by the single execution capability of the processing module. Thus, to be able to use the system together, separate processing module should be used for each system.**

Index Terms – **Autonomous Vehicle, Pothole Detection, Speed Hump Detection, Single Camera and 2D LIDAR System**

I. INTRODUCTION

OVER the past few years, many developments have been made in the field of automated vehicles. From identifying navigation paths to obstacle detection, efforts are continuously being made to improve the system, paving way to driverless vehicle technology. Perception to the environment is one of the key tasks of an autonomous vehicle because it is the input needed on how the vehicle should behave. Focusing on urban areas, identifying road surface obstacles such as potholes and speed humps should be addressed in order to prevent vehicular damages and accidents. Potholes, being the most common and damaging pavement distress, are defined as bowl-shaped depression with a minimum plan dimension of 150 mm [5]. These result from poor road maintenance and natural calamities.

Speed humps, on the other hand, are rounded elevation with height varying from 5 to 18 cm [6]. Unlike potholes, speed humps are strategically placed across the road to control track. However, in autonomous vehicle path planning, both potholes and speed humps are considered as road obstacles that should be detected. Nowadays, several detection methods have been developed. These existing methods can be divided into vibration-based methods, 3D reconstruction-based methods, and vision-based methods [5]. Vibration-based methods rely on accelerometers and require small storage that is advantageous in real-time processing, however, vibration-based methods are prone to provide wrong results due to the limitation of the accelerometers. 3D reconstruction-based methods use reflected laser pulses to establish accurate 3D surface models, but are costly and require high computational efforts. Due to the said reasons, recent developments are focusing on vision-based methods. These methods include 2D image-based and video-based approaches [7]. The collected image and video data are analyzed and processed using segmentation, stereo geometry and other techniques. Vision-based methods are cost-effective and can potentially be as accurate as 3D reconstruction-based methods.

In this paper, an effective pothole and speed hump detection system is discussed. The captured data are processed by the system. For pothole detection, image segmentation followed by spectral clustering is performed. After which, pothole extraction is done [8]. For speed hump detection, set of data from the LIDAR is continuously compared to each other to estimate the change in z-axis. The data is considered a speed hump if the change in z-axis resembles an increase-decrease pattern within 5 to 18cm [6]. As the final result, information such as type of obstacle and placement can be gathered from the system.

II. METHODOLOGY

A. Hardware Design and Implementation

Vision provides sufficient scene information, such as colors, shapes, and texture. However, the distance estimation using a single camera is not good enough. With that being said, single camera and LIDAR was used to detect potholes, and speed humps. The computing hardware used was the Hardkernel ODROID-XU4 single board computer that has Samsung Exynos5422 CortexTM-A15 2Ghz and CortexTM-A7 Octacore CPUs [1]. ODROID-XU4 is energy-efficient, has an open source support, and can run in Linux – Ubuntu 15.04 and Android 4.4 KitKat and 5.0 Lollipop, which makes it suitable for this project. The camera used was the ODROID-XU4 compatible oCam:

Manuscript received December 22, 2016; revised January 13, 2017.

Nathalie Joy Sucgang was a student at the Electrical and Electronics Engineering Institute, University of the Philippines-Diliman, Quezon City, Metro Manila, Philippines. (email: nbsucgang@up.edu.ph).

Manuel Ramos is a professor at the Electrical and Electronics Engineering Institute, University of the Philippines-Diliman, Quezon City, Metro Manila, Philippines. (email: manuel@eee.upd.edu.ph).

Nicolette Ann Arriola is an instructor at Electrical and Electronics Engineering Institute, University of the Philippines-Diliman, Quezon City, Metro Manila, Philippines. (email: nicolette.arriola@eee.upd.edu.ph).

ISBN: 978-988-14047-3-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

5MP USB 3.0 Camera [2] set to provide medium-sized file, 640x480 images at 30fps. The laser scanner was the low-cost LIDAR Lite v1 [3] with scanning range of 40m. Arduino Uno was used to get data from the LIDAR [4] due to some I2C communication failure experienced in ODROID-XU4. Data from Arduino Uno were transferred to ODROID-XU4 using the serial communication established between them. In totality, the hardware chosen were low-cost, energy-efficient, has a high performance, an open support, and can easily be integrated with each other.

B. Pothole Detection

The pothole detection system used can be divided into three major steps: (1) Image Segmentation, (2) Shape Extraction, (3) Pothole Identification. The method took advantage of the three distinctive visual characteristics of potholes such as: (1) shape of potholes are approximately elliptical, (2) one or more shadows found in potholes are darker than the surrounding region, and (3) the texture of potholes are coarser and grainer than the texture of the surrounding region [18].

Images were segmented into defect (foreground) and non-defect (background) regions using histogram Otsu-based thresholding. The geometric properties of the defect region were used to describe potential pothole shape through spectral clustering. The pixels inside the potential pothole was then extracted and analyzed to determine if the potential pothole is an actual pothole.

1) Image Segmentation: Based on the intensity of the pixels in the image, the optimal threshold was calculated using Global thresholding and Otsu's thresholding. Global thresholding finds the mean threshold, while Otsu's thresholding computes for the threshold value that minimizes the interclass variance. The average of the two thresholds was used as the final threshold to segment the image into foreground and background portion. This was done to remove noise without using additional filters.

During image segmentation, two cropped images were formed. The first image is cropped from the top of the original image. The second image, on the other hand, is cropped from the bottom of the original image. The amount of cropped image is computed such that when the cropped images were compared, the depth of the pothole in the image could be emphasized in the latter part of the detection.

Binary image was then formed by comparing the final threshold to the difference of the pixels from the first and second cropped images. If the difference is greater than the calculated final threshold, the pixel in the binary image will be set to 1. If it's less than, the value will be set to 0.

Very small areas as well as those connected to the boundaries of the image were also removed. This was done to put emphasis on the object of interest.

2) Spectral Clustering: Spectral clustering was done to group the pixels in the original image based on their connectivity and similarity. This gives us an idea about which areas appear to be on the same layer.

To cluster the original image, general algorithm for spectral clustering is shown below [7]:

- Form affinity matrix, S, based on the adjacency and similarity of the pixels
- Create degree matrix, D. D = diag(d_i), where $d_i = \sum_{j=1}^{n} S_{ij}$
- Calculate the normalized laplacian matrix, L. $L = D^{\frac{-1}{2}}\, x\, L1\, x\, D^{\frac{1}{2}}$, where L1 = D – S
- Find the eigenvalues, λ, and eigenvectors, v. $L\, x\, v = \lambda\, x\, v$
- Get the largest eigenvectors, and form matrix U.
- Normalize matrix U. Let the resulting matrix be normalized matrix Y. $Y_{ij} = \frac{U_{ij}}{(\sum_{l=1}^{k} U_{il}^{2})^{\frac{1}{2}}}$, where i and j = 1 corresponds to the points in matrix U.
- Normalized matrix Y is clustered using k-means algorithm

3) Pothole Identification: Pothole is identified by overlapping the results from image segmentation and spectal clustering. Those marked as possible pothole in both resulting images were marked as the detected pothole in this stage.

On the resulting image from image segmentation, every 50^{th} point from those pixels whose values were set to 1, were selected. These points were labeled as seeds and projected onto the image from spectral clustering.

On the clustered image, vertical extraction was performed. The top and bottom vertical points for each seed that have the same value (belong to the same group in spectral clustering, and have a value of 1 in image segmentation) were selected. All connecting points (belong to the same group in spectral clustering, and have a value of 1 in image segmentation) between the top and bottom points were also selected.

After the vertical extraction, horizontal extraction was then performed. All points selected during the vertical extraction were considered as new seeds. From there, the leftmost and rightmost points for each seed that have the same value (belong to the same group in spectral clustering, and have a value of 1 in image segmentation) were selected. All connecting points (belong to the same group in spectral clustering, and have a value of 1 in image segmentation) between the leftmost and rightmost points were also selected.

Pothole identification was completed by plotting all selected points during extraction.

C. Speed Hump Detection

Fixed vertical angle enables the estimation of road surface. When the vehicle is approaching a speed hump, the Z axis measurement starts to rise, then falls rapidly.

The speed hump detection algorithm can be divided into three parts: (1) Data Acquisition, (2) Data Filtering, and (3) Data Processing.

1) Data Acquisition: Acquisition of LIDAR datapoints were done continuously. Each set of data was composed of fifteen data points for better estimation of slope.

2) Data Filtering: To remove noise and out-of-place changes in the distance, median filtering was used. Three data samples were taken from the distance array: i-1, i, and i+1, and sorted from increasing order using quicksort algorithm. The median replaced the current datapoint, i.

ISBN: 978-988-14047-3-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

3) Speed Hump Detection: Average of each set was taken and compared to its two neighboring sets. If the distance between each set was from 5 to 18 cm, the data was marked as speed hump.

D. System Testing

Processed images were validated by determining the number of true positive (TP), false positive (FP), true negative (TN), and false positive (FP). Accuracy, precision, true positive rate, and false positive rate were also computed.

The pothole detection system and the speed hump detection system were validated by gathering data around the University of the Philippines - Diliman Campus. Ten (10) or more speed hump sets of data are processed. Manual checking and comparison were also done. Processing time was then computed.

III. Results and Analysis

A. Pothole Detection

Fig. 1. Original Image

Fig. 2. Segmented Image

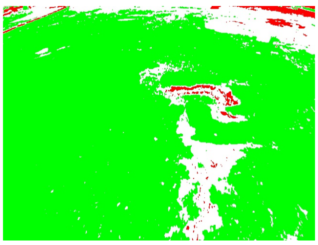

Fig. 3. Clustered Image

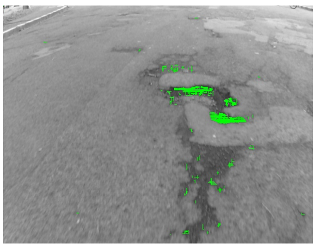

Fig. 4. Detected Pothole

Sample images are shown above. Image captured by the camera goes through image segmentation, spectral clustering, and lastly pothole detection.

Six hundred ninety two images were processed during system testing, and the results are summarized in the table below.

Table I. Performance Evaluation of the Pothole Detection System

True Positive	113
True Negative	65
False Positive	492
False Negative	32
True Positive Rate	77.93%
True Negative Rate	11.66%
Accuracy	86.18%
Precision	63.48%
Processing Time (in seconds)	1.013

Given the processing time and the accuracy of the whole system during the testing, the method proposed can be used in real-time pothole detection, and gives accurate results for vehicles with speed from 10 to 15 kph.

During the manual testing of each image in dataset, errors in classification are due to shadows, marks, and objects that have very different pixel values as the road and are not connected to the border of the image. Uneven color of the road after the rain produces the same error. This is because the algorithm used was focused on the differences in the pixel characteristics. The use of single camera limits the verification of the detected pothole since it has no idea on the actual depth of the surface that was labeled as pothole.

B. Speed Hump Detection

Three different sets of tests were made to determine the optimal distance in detecting the speed humps. The first one has an average distance of 1.9m from the vehicle to the speed humps. The second one has an average distance of 4.13m, while the last one has an average distance of 6m.

Performance of the system can be evaluated by classifying the results into: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). The 3.7km route was taken and divided into 7.5m to evaluate true negative values. Results are shown below.

ISBN: 978-988-14047-3-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Table II. Performance Evaluation of the Speed Hump Detection System

	At 1.9m	At 4.13m	At 6m
True Positive	13	11	6
True Negative	471	470	457
False Positive	2	3	16
False Negative	3	5	10
True Positive Rate	86.67%	78.57%	37.5%
False Positive Rate	0.63%	1.05%	3.38%
Accuracy	98.87%	98.36%	94.68%
Precision	81.25%	68.75%	27.27%
Processing Time (in seconds)	0.44	0.45	0.49

It can be observed that at 1.9m, the accuracy, precision, and true positive rate are high, while the false negative rate is low. However, at the said distance, the vehicle won't have ample time to react according to the detection so another trial was performed at 4.13m. At 4.13m, it can be seen that the accuracy was decreased by 0.61%, precision by 12.5%, and true positive rate by 8.1%, while the false positive rate was increased by 0.42%. The system is still reliable at this rate, and provides ample time to react according to the detection. Another test was made to check if there is a better distance. At 6m, the accuracy was decreased by another 3.68%, precision by 41.48%, and true positive rate by 41.07%, while the false positive rate was increased by 2.33%. The increase in overall error of the system at 6m leads us to conclude that the optimal distance is at 4.13m. Long-range readings produce more errors due to cars and street pavements that were misread as speed humps.

In general, the errors in classification of the system were observed to be constantly occurring during turns and on uneven road. This is because the whole system takes advantage of the changes in slope. Hence, the more flat and straight the road is, the more accurate detection there is. Another cause of error is when the speed exceeds 15kph. The said cause was due to the limitations of the LIDAR Lite v1. When stationary, the distance reading of LIDAR has a discrepancy of 3 to 5m, which increases with the movement of the LIDAR and as the distance to be read gets farther.

C. Cascaded System

The two systems were cascaded and tested around University of the Philippines - Diliman Campus. Route taken was 7.3km long, and divided into 7.5m road segments to evaluate true negative values during speed hump detection analysis. One thousand one hundred twenty seven images were processed. The following results are summarized in the table below.

Table III. Performance Evaluation of the Cascaded System

	Individual System		Cascaded System	
	Pothole Detect	**Speed Hump Detect (At 4.13m)**	**Pothole Detect**	**Speed Hump Detect (At 4.13m)**
True Positive	113	11	256	15
True Negative	492	470	718	947
False Positive	65	3	76	5
False Negative	32	5	77	7
True Positive Rate	77.93%	78.57%	59.88%	54.54%
False Positive Rate	11.66%	1.05%	10.53%	0.52%
Accuracy	86.18%	96.36%	80.83%	98.46%
Precision	63.48%	68.75%	70.11%	70.58%
Average Processing Time (in seconds)	1.013	0.45	1.53	

Based on the results of the cascaded system, it can be seen that the average time is approximately equivalent to the sum of the processing time of the two individual systems. This is due to the alternate execution of each algorithm. For the pothole detection, the accuracy was decreased by 5.35%, while for the speed hump detection; the accuracy was increased by 0.10%. However, the increase in accuracy was due to the longer route taken. Taking a look at the number of false negative under the speed hump detection on the cascaded system, it can be seen that the number of missed speed hump is almost half the number of speed humps to be detected. Again, this is due to the alternate execution of each algorithm. Hence, it can be said that when the two systems are cascaded, the results are still reliable and accurate, however the number of potholes and speed humps missed increases. This observation on the system will be problematic as the number of potholes and speed humps to be detected increases. Thus, the system can be cascaded as long as the potholes and speed humps are minimal.

IV. Summary and Conclusion

The Whole Detection System used oCam OV5640 USB 3.0 Camera, PulsedLight LIDAR Lite v1, Arduino Uno, and Hardkernel Odroid XU4 Single-board Computer. It was designed to detect potholes, and speed humps for autonomous vehicle to aid in free path planning. The performance of each system was evaluated. Based on the results, potholes can be detected from 1.6m to 5m away from the vehicle with 86.18% accuracy within 1.013s, while the speed humps can be detected efficiently with an average distance of 4.13m with 98.36% accuracy within 0.45s.

Inaccuracies are due to the main idea on which each algorithm is focused, and the hardware limitations. For the pothole detection system, the algorithm is focused on pixel differences and cannot be verified using single camera, as it has no perception of depth. While during speed hump detection, the algorithm is focused on the changes in the slope, which produces errors on turns and uneven roads. In addition to that, LIDAR Lite v1 introduces another limitation as the reading becomes unstable during fast movement, and father distances. In totality, the detection system is reliable when the speed of the vehicle is from 10 to 15kph.

When the systems were cascaded, the pothole detection accuracy was decreased by 5.35%, while the speed hump detection accuracy was increased by 0.10%. However, noting that the number of the missed potholes and speed humps is almost as half as the number of potholes and speed humps to be detected, it can be concluded that the behavior of the system will be problematic as the number of potholes and speed humps to be detected increases. The alternate execution of the algorithms during the cascaded testing was the reason behind the increased missed detection. This is also true for the longer processing time. Thus, the cascaded system is reliable and accurate as long as the numbers of potholes and speed humps to be detected are minimal.

REFERENCES

[1] Odroid-XU4 Datasheet [Online]. Available: https://www.hardkernel.com/main/products/prdt_info.php?g_code=G14345223982.

[2] oCam: 5MP USB 3.0 Camera Datasheet [Online]. Available: http://www.hardkernel.com/main/products/prdt_info.php?g_code=G145231889365.

[3] Lidar Lite v1 Datasheet [Online]. Available: https://cdn.sparkfun.com/datasheets/ Sensors/Proximity/LIDAR-Lite-Laser-Datasheet.pdf.

[4] Arduino Uno Datasheet [Online]. Available: https://www.arduino.cc/en/Main/ArduinoBoardUno.

[5] T. Kim and S. Ryu. Review and Analysis of Pothole Detection Methods. In Journal of Emerging Trends in Computing and Information Sciences, 2014 CIS Journal, pages 603–608, August 2014.

[6] D. F. Llorca I. Parra R. Quintero A. G. Lorente L. Vlacic C. Fernandez, M. Gavilan and M. A. Sotelo. Free Space and Speed Humps Detection using LIDAR and Vision for Urban Autonomous Navigation. In Intelligent Vehicles Symposium (IV), pages 698–703, June 2012.

[7] E. Buza S. Omanovic and A. Huseinovic. Pothole Detection with Image Processing and Spectral Clustering. In 2nd International Conference on Information Technology and Computer Networks, pages 48–53, 2013.

[8] A. Vaswani R. Jotwani H. Punjabi, R. Nanwani and Dr. A. Kunte. Intelligent Pothole De- tection System. In International Journal of Emerging Technology and Advanced Engineering, 2014 IJETAE, pages 590–595, July 2014.

ISBN: 978-988-14047-3-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)