xIlkogretim Online - Elementary Education Online,2021; Vol 20 (Issue 1): pp.2270-2275
http://ilkogretim-online.org
doi: 10.17051/ilkonline.2021.01.252

Contour Based Pothole Detection & Updation Using Raspberry Pi & Edge Detection

D. Akhil Varma, Department of Electronics and Communication Engineering, SRM Institute of Science and Technology, SRM Nagar, Kattankulathur, Kancheepuram,603203, Chennai, TN, India. dv1871@srmist.edu.in
S.Krithiga, Department of Electronics and Communication Engineering, SRM Institute of Science and Technology, SRM Nagar, Kattankulathur, Kancheepuram,603203, Chennai, TN, India.
Authors E-mail: krithigs@srmist.edu.in
D.Vijayalakshmi, Department of Electronics and Communication Engineering, SRM Institute of Science and Technology, SRM Nagar, Kattankulathur, Kancheepuram,603203, Chennai, TN, India. vijayald@srmist.edu.in

Abstract-Potholes are the major causes of road accidents. There is no proper maintenance in the local areas and reporting is done manually .The model is designed to be installed in a vehicle to report the potholes using Image-Processing with the help of Raspberry-Pi. This is achieved using various image processing edge detection algorithms using Open CV software in python language. Edge detection techniques have been applied by the combination of various morphological filtering techniques using the combination of various smoothing filters

Keywords: Micro-Controller, GPS, Cloud, image-processing, web server, Smart devices.

I. INTRODUCTION

Potholes are detected along with their severity using built in accelerometer. The variations present on the road are de tested using an accelerometer along with an ultrasonic sensor. Potholes are also detected by using the lasers which scans and measures the depth of the pothole. This paper presents about building a real-time prototype for the detection and reporting of Potholes by examining the road with a camera. Potholes are detected based on edge detection algorithms. It Collects information about the coordinates of the pothole location with the help of a GPS module and stores the location and the image in the database server for future references using the Wi-Fi module integrated along with the micro controller. This study aims at providing a smart detection using image processing and simultaneously reporting the detected potholes using a web server. This process ensures a stable Wi-Fi connection. Smoothing filters such as Median blur [1] is applied to remove the noise and simultaneously preserving the edges of the image. Thresholding techniques [2] are applied to adjust the color variants. Morphological techniques such as opening are applied to remove the small edges and thicken the required images. Canny edge algorithm [3], along with arc length techniques, is used to detect the pothole using image processing.

II. METHODOLOGY

A. Overview of the System

The system should detect the pothole and should simultaneously upload the GPS location using a web server. Figure 1 shows the complete overview of the system it consists of a Raspberry pi for the image processing schemes to take place.

For this a camera is attached to capture the images in front of it and the captured image is sent to the raspberry pi for the processing to take place. It also consists a GPS module to get the latitude and longitude locations of the detected potholes and is uploaded using a stable WiFi connected to the raspberry pi with the help of the web server. With the images obtained from the camera the raspberry pi starts detecting potholes, if a pothole is detected then buzzer starts indicating a presence of a pothole and simultaneously the location is taken with the help of GPS and updated it to the server. Steady WiFi connection should be present to send the data automatically

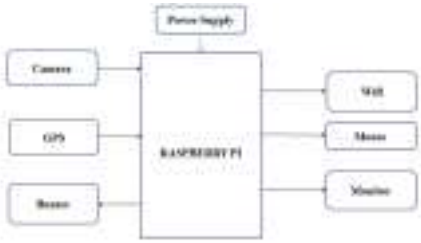

Figure 1. Overview of the proposed system

B. Detection of Potholes

The location and position of the camera play an essential role in accurately detecting the potholes. In order to obtain optimized results, it is best to place it near the rear mirror. The position of camera plays a major role in detecting the potholes. The camera captures at a rate of 30 frames per second . The Whole process is done in real time. The process starts by removing the unwanted objects from the image such as sidewalks and pavement of the side road . The extracted image then undergoes binary to gray conversion then the noise is removed using smoothing techniques in this median blur filter is used to remove salt and pepper noise and to preserve the edges of the image. The extracted image then undergoes thresholding techniques [2] followed by morphological tech niques [4] such as erosion and dilation, which in this case, is called opening. Erosion is used to remove the small gravel and small irregularities, and then dilation is used to thicken the required edges. Then canny edge algorithm is used to lower the error rate and decrease the multiple responses on the single edge. It also provides good localization when compared to other edge detection algorithms. For the extracted image, arc length techniques are applied to detect the contours [5] and from this pothole images are detected. Median blur is achieved using the mathematical expression

$$f(\mathrm{X},Y) = median\{g(s,t)\} \quad (1)$$

Erosion is obtained using the expression

$$g = f - s \quad (2)$$

where f is the binary image and s is the structuring element similarly for dilation

$$g = f + s \quad (3)$$

Canny edge detection is a multi step algorithm as follows.

1. Applying Gaussian filter to reduce the noise and for smoothing the image.

$$g(m,n) = G_{\sigma}(m,n) * f(m,n) \quad (4)$$

Where m, n are the x and y co-ordinates and σ denotes standard the contour is detected using canny edge detection algorithm. Potholes can be detected using Stereovision [6], It can also be based on disparity map [7]

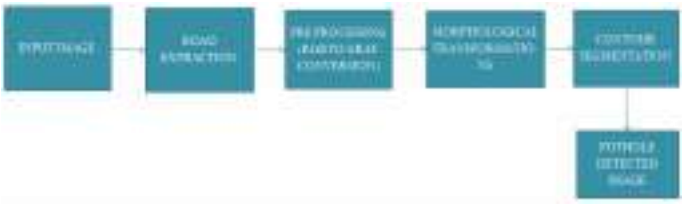

Figure 2. Block diagram of proposed system

C. Reporting of Potholes

The study focuses on automatic reporting of potholes. Raspberry pi microcontroller is used to capture the images, and It can capture up to 30 frames per second. The setup consists of the micro-controller along with GPS and a portable wifi module. If a pothole is detected, then the microcontroller automatically sends the location and the image to the web server using the inbuilt wifi. Deviation.

$$G_\sigma = 1\sqrt{2\pi\sigma^2}\exp - m^2 + n^2 2\sigma^2 \quad (5)$$

$$M(n, n) = \text{p}g^2{}_m(m, n) + g^2{}_n(m, n) \quad (6)$$

And

$$\theta(m, n) = tan^{-1}[g_n(m, n)/g_m(m, n)] \quad (7)$$

Locations are then sent to the micro-controller, and it sends the location of the potholes to the database.

D. System Integration

The main focus is on detecting and reporting of potholes simultaneously. The Flow chart of the process is shown in

3. Threshold of M: $M_T(m, n) = M(m, n)$ if $M(m, n) > T0$ otherwise (8)

Figure3. It mainly consists of 2 parts detecting and reporting of potholes. The micro-controller would capture the image in front of it continuously. The captured image is analyzed using Here *T* is chosen to preserve the edges and reducing most of the noise.
4. Edges might get broadened due to the thresholding in the above step so here we need to suppress the non-maxima points in the edges. To verify this compare each pixel value (Non - Zero) to its neighbours along the gradient direction. If the value is higher than retain it for next step otherwise set it to zero.
5. The modified image is then threshold using two different threshold values T1 and T2. If the obtained threshold value is greater than T1 then conform it as an edge. If the value is below T2 then it is not an edge and discards these values.
6. The values between T1 and T2 can best determined by their linkage. If the value between them is linked to T1 then save it as edge and the remaining values can be discarded.

Figure 2 shows an overview of how image processing techniques have been applied. First, the extra pavement parts are removed for better processing, and then the binary image is converted into gray scale. Then several morphological techniques like erosion and dilation are applied from which the image processing techniques if the pothole is detected, then GPS module would turn on and acquires the location of the pothole then the inbuilt wifi module would send together both the location and the image to the database. The installed buzzer turns on the successful detection of the pothole. The process continues to take place until the user desires to turn it off.

III. PROTOTYPE AND TESTING

Table I describes the various metrics obtained from testing with various edged detection algorithms. The number of con tours shown below is the useful contours that are needed for detecting the potholes from the image. Canny edge shows the best number as it effectively removes the unwanted gravel and unessential small holes from the image. In contrast, the other two techniques show inaccuracy in describing the details. The sensor data can be taken [8]. Driving Conditions are taken into account [9]. The number of contours is obtained using the figure4.

The detected pothole location is shown in figure5.The sample output of the original image in comparison to the detected pothole image is shown in the figure6

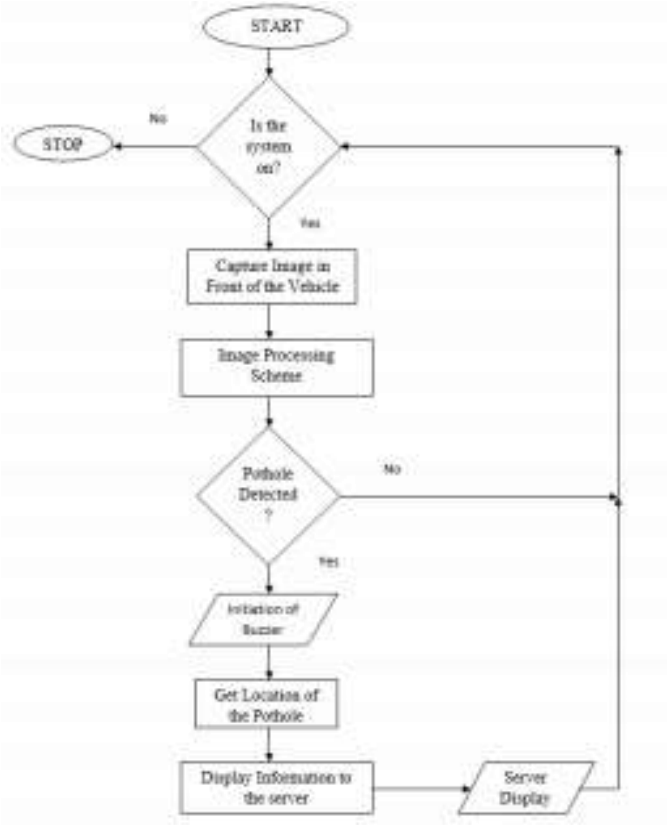

Figure 3. Flow diagram of proposed system

Figure 4. Sample Pothole

Table 1. Comparisons

Edge detection algorithm	Performance Metrics		
	contours detected	*advantages*	*disadvantages*
canny edge	24	removes unwanted gravel	more complex
sobel edge	98	Thick edges	direction specific
laplacian edge	98	best for simple images	not efficient

Figure 5. Pothole location

Figure 6. Detected Pothole

IV. CONCLUSION

This work on smart detection and automatic reporting of potholes is achieving excellent results. This work effectively made use of various algorithms including blurring techniques, morphological techniques and canny edge detection. The response time of the system is high so that it can be used for real-time applications. For this system, there is no need for a monitor so that this device can be portable. The accuracy and sensitivity of the system are found to be accurate. The reporting of the location is done without any errors. This can be effectively used in the day to day life and can be very useful for the motorists and cab drivers. The system was able to perform marginally in places where there are multiple potholes but only with a considerable error.

REFERENCES

1. G. George, R. M. Oommen, S. Shelly, S. S. Philipose and A. M. Varghese, *"A Survey on Various Median Filtering Techniques For Removal of Impulse Noise From Digital Image,"* 2018 Conference on Emerging Devices and Smart Systems (ICEDSS), Tiruchengode, 2018, pp. 235-238
2. A. K. Bhandari, A. Ghosh and I. V. Kumar, *"A local contrast fu sion based 3D Otsu algorithm for multilevel image segmentation,"* in IEEE/CAA Journal of Automatica Sinica,, January 2020, vol. 7, no. 1, pp. 200-213
3. Z. Xu, X. Baojie and W. Guoxin, *"Canny edge detection based on Open CV"*, 2017 13th IEEE International Conference on Electronic Measurement & Instruments (ICEMI), Yangzhou, 2017, pp. 53-56.
4. F. G. B. De Natale and G. Boato, *"Detecting Morphological Filtering of Binary Images,"* in IEEE Transactions on Information Forensics and Security, May 2017, vol. 12, no. 5, pp. 1207-1217.
5. P. Arbelaez, M. Maire, C. Fowlkes and J. Malik, ´ *"Contour Detection and Hierarchical Image Segmentation,"* in IEEE Transactions on Pattern Analysis and Machine Intelligence, May 2011, vol. 33, no. 5, pp. 898- 916.
6. Z. Zhang, X. Ai, C. K. Chan, and N. Dahnoun, *"An efficient algorithm for pothole detection using stereo vision,"* in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), May 2014, pp. 564–568.

7. R. Fan, U. Ozgunalp, B. Hosking, M. Liu and I. Pitas, *"Pothole Detec tion Based on Disparity Transformation and Road Surface Modeling,"* in IEEE Transactions on Image Processing, 2020, vol. 29, pp. 897-908.
8. A Fox, BV Kumar, Chen ,F Bai. *Multi-lane pothole detection from crowdsourced undersampled vehicle sensor data*. IEEE Transactions on Mobile Computing. 2017 Apr 4;16(12):3417-30
9. H. Han, et al.,*"SenSpeed: Sensing driving conditions to estimate vehicle speed in urban environments,"* in Proc. IEEE INFOCOM, Apr. 2014, pp. 727–735