The Utilization of Image Processing for Parking Spot Detection System

Author : Nezar Al-Akwaa

***Abstract* – This project focuses on the utilization of image processing technique to present a smart system for parking spot detection. The system will be able to analyze and apply multiple filters to the image to create information about the number of available and occupied parking spots in the lot. This system has been developed and tested using MATLAB and we are proposing this system as an economic alternative option for parking detection systems based on electronic sensors. The system will need the camera to be positioned in a fixed location in the parking lot to ensure image collection shares the same properties.**

Index Terms – Smart Parking, Parking Spots Detection, Image Processing, Image Segmentation, Image Collection

I. INTRODUCTION

In urban cities traffic jams are usual. To find a parking spot is one of the common problems in hospitals, malls and office buildings. The number of cars on the road is increasing which creates more time and effort for drivers to find a spot. In this project we're proposing a system that can analyze the parking lot based on image processing technique and provide feedback on the number of available and occupied spots in the parking lot using MATLAB image processing tools. The images of the empty and occupied parking will be taken from a fixed position, angle and location [5][7].

There are many parking cars detection methods such as ticketing system where it requires a ticket scanned before entering and after exiting the lot. Also, electronically, new systems based on sensors are deployed to monitor parking spots availability [5]. In this report, we're proposing a system based on image processing where a camera is used as a sensor for image collection. This system will help drivers find an available parking spot by counting the available spots that can be used. The development of this system will use image processing technique which should be a cheaper method compare to existing systems [3].

In the project, the parking slot is detected based on object identification, when a car is present or not in the area and then the calculated result is displayed on the window [3].

2- METHODOLOGY AND FRAMEWORK

Our system methodology will consist of two image acquisition, then images will go through segmentation process, where different filters will be applied including image subtraction to reduce noise, increase object accuracy for the purpose of car count and result calculation.

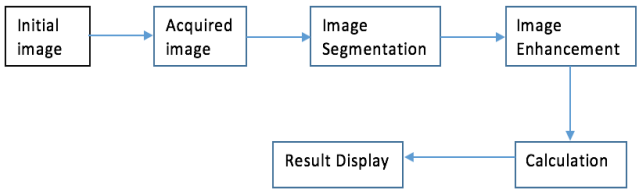

Fig 1: Block diagram of the system:

(A) Initial Image is a one-time image collection for the parking lot empty, as show in figure 3. The purpose of this procedure is to have a reference image that doesn't contain cars. The reference image will be used to subtract from the image that contains cars parked, which will identify objects in the parking lot. This image will be loaded and used during image processing in MATLAB. Also, in this step the user will be asked to enter the total number of the car spots in this lot, as shown in Fig 2.

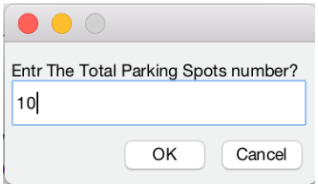

Fig 2. Request user to enter the number of spots in the parking lot

```
Num_Spots_Lot = inputdlg('Entr The Total
Parking Spots number?')
Save_entered_value =
str2num(Num_Spots_Lot{1})
```

Code 1. Request user input, convert entry to number and display in dialog box

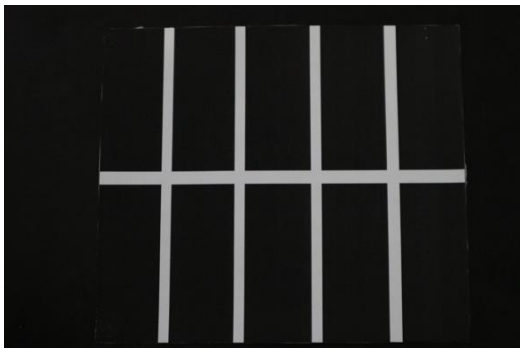

Fig 3. Initial image with empty parking lot

```
Image_2 =imread('NoCarParked.jpg');
Image_2 = imresize(Image_2,[400 NaN]);
figure(2), imshow(Image_2);
```

Code 2: Retrieve image for parking lot unoccupied

(B) Acquire Image of the parking lot occupied that shows the different number of cars parked. The camera must be position at the same location when the initial image was taken as shown in figure 4.

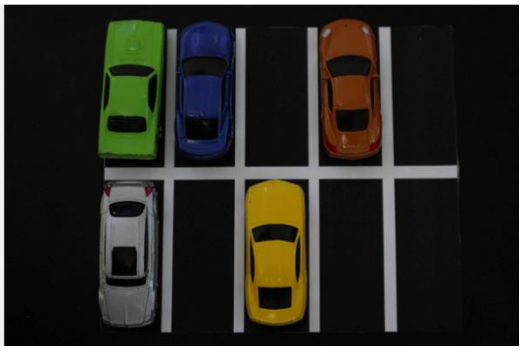

Fig 4. Image of the same lot occupied

```
Image = imread('FiveCarsParked.jpg');
Image = imresize(Image,[400 NaN]);
figure(1), imshow(Image);
```

Code 3: Retrieve image for cars parked and resize image

(C) An image subtraction operation was chosen to separate objects from the background [2]. The image of the car occupied was subtracted from the initial image. The results showing in figure 5, only the cars still present in the image while other details in initial image was removed such as parking's white lines separator.

Fig 5. Subtraction results image

```
sub = imsubtract(Image,Image_2);
figure(3), imshow(sub);
```

Code 2: Image subtraction operation

The results of the subtracted image will then go through the image processing steps. The five modules involved in parking spot detection are as followed in the image segmentation section.

(D) Image Segmentation the RGB image will be converted to gray scale then creates binary image. The following equation is used to covert RGB image to gray scale image [5].

$$gray = 0.229R+0.587G+0.114B$$

Equation 1: Convert to grayscale equation

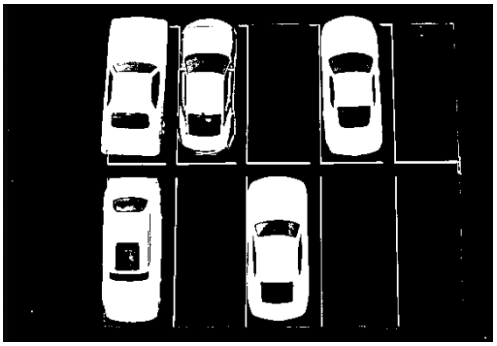

Fig 6. Result of subtraction image covered to gray

```
Igray = rgb2gray(sub);
figure(4), imshow(Igray);
```

Code 3: Convert the TrueColor RGB to the grayscale image

The binary image has the details about the essential information such as the shape and position of the object. The advantage, it will reduce the complexity of the data and simplify the process of object recognition and classification. There're different types of thresholding, basic, two-band-tile, optimal and adaptive.

In this project to separate the background and the object, basic method was selected which can be defined as the following [5].

$$g(x,y) = \{ 1 \text{ if } x > T \text{ and } 0 \text{ if } x \leq T \}$$

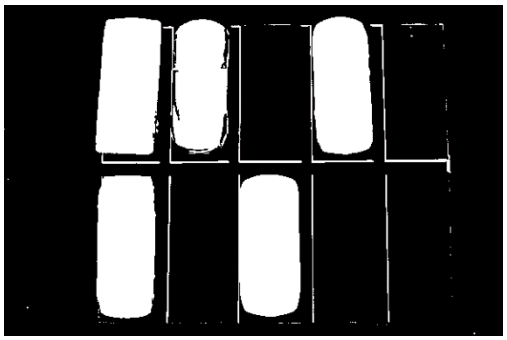

Fig 7: Gray image covered to binary

```
level=0.02;
Imgthresh = im2bw(Igray,level);
```

Code 4: Convert the grayscale image to binary

(E) During image enhancement process, the background pixels were filled using "imfill" operation to fill holes in the binary image BW, where conn specifies the connectivity.

```
Imgfilled = imfill(Imgthresh,'holes');
figure(6), imshow(Imgfilled);
```

Code 5: Fill background pixels of the input binary image

Also, Morphological Disk-shaped element is being used to remove imperfection and remove any noise created by image segmentation [2]. There are four operation types dilation, erosion, opening and closing. In our project we used dilation and opening operators. Opening removes small holes. This operator is widely used in image processing.

As far as the use of erosion, a strel object reflects flat morphological structuring element which is a major part of morphological dilation because it adds pixels to the boundaries of object in image. As shown in figure 8, the objects visibility improved, and image noise were eliminated.

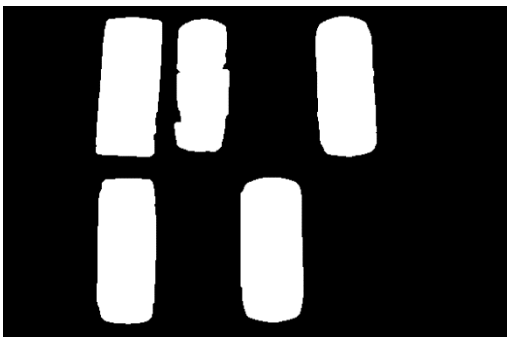

Fig 8: Image enhancement

```
structDisk = strel ('disk',5);
Imgopen = imopen(Imgfilled,structDisk);
figure(7), imshow(Imgopen);
RmvSml=bwareaopen(Imgopen,4,4);
figure(8), imshow(RmvSml)
```

Code 6: Morphological disk-shaped element with open on binary and remove small objects

(F) Bounding box for detected cars, in this step the focus was on exterior boundaries of the object detected [6]. Using "regionprops" it returns measurements for the set of properties which is specified by properties for each component (object) in the binary image. Where it can be used on contiguous and dis-contiguous regions.

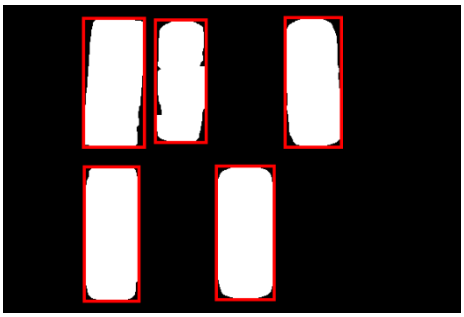

Fig 9: Cars were detected, and boundaries applied

```
Iprops=regionprops(RmvSml,'BoundingBox','Image');
hold on
text(8,785,strcat
('\color{green}Cars
Detected:',num2str(length(Iprops))))
hold on
for n=1:size(Iprops,1)
rectangle('Position',Iprops(n).BoundingBox,
'EdgeColor','r','LineWidth',2);
end
```

Code 7: Bounding box for detected cars measuring image regions

(G) Counting the number of occupied cars [4]. The MATLAB operator used is "sprintf" which formats the data in arrays using the format '%d' which converts the double-precision values to integers. Then implemented operation that will count available spots and create dialog box to display the results. Using the minus operator to subtract the total number of parking spots in the lot and the number of objects detected. The result will be converted from number array to a character so it can be displayed in a dialog box. As shown in figure 10.

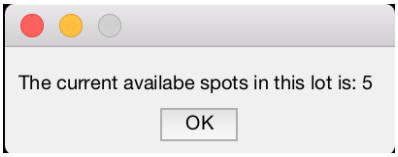

Fig 10: GUI display for the real-time available spots

```
result = sprintf('%d',n-0);
disp(result);
ConvToNumb= str2num(result)
AvaSpots = minus(Save_entered_value, ConvToNumb);
disp(AvaSpots)
CovFinalToStr= num2str(AvaSpots)
disp(CovFinalToStr)
f = msgbox(['The current availabe spots in this
lot is: ',CovFinalToStr]);
hold off
```

Code 8: Count the number of occupied and available spots and display the results on dialog box

3- SYSTEM ALGROTHEM

The main steps of the proposed algorithm for parking lot detection using image processing.

I. Request user to input the number of spaces in the lot using dialog box
II. Collect the image of the parking unoccupied this is called initial image
III. Collect the image of the parking occupied with cars
IV. Subtract the parking image of occupied and unoccupied. The image result will be processed by image segmentation as follows:

a. Convert the TrueColor RGB to the grayscale image
b. Convert grayscale to binary image with level specified
c. Apply filters that will reduce noise such as morphological and pixel background fill

V. Bounding box for detected cars measuring image regions
VI. Calculate number of objects in the image will be subtracted from number of parking spots available in the lot.
VII. Display the real-time available spots for parking

4- IMAGE ANALYSES

While the objective of the project was to provide real-time parking spot detection. The project authors were interested on exploring the image details.

An image histogram was performed on the image with cars occupied, first on the original image figure 11 and then after converting the image to gray scale, figure 12. The histogram counts the number of gray value of pixels, each pixel in the image has gray value between 0 and 250. To get more details about the image such as objects, counting pixels would be the method [8] [6].

When compared graphs, we observed that the gray image histogram shows the number of gray count value of pixel in the

image has significantly reduced, for example in figure 11 the highest number of gray value is 12 with counting pixel number = $13.8X10^4$. While in figure 12, the gray value count is 1 with counting pixel number = $1.8 X 10^5$

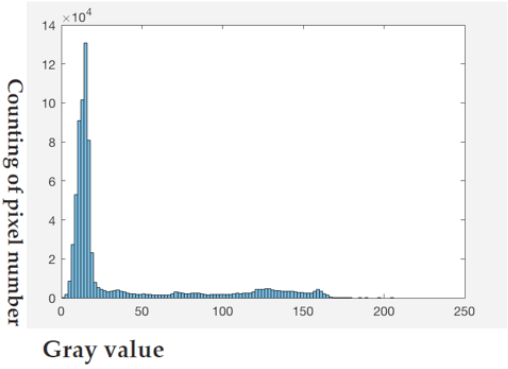

Fig 11: Original image histogram

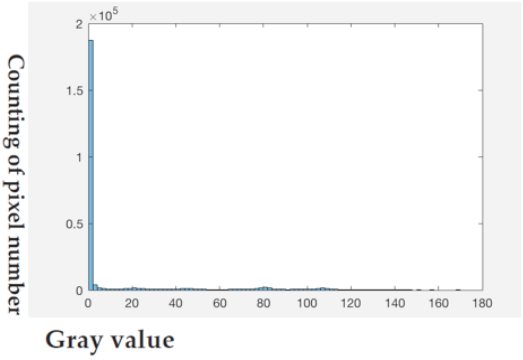

Fig 12: Gray image histogram

This shows the original image losses a lot of details during image segmentation. This should not be a concern as we are trying to simplify the image by removing all unwanted details to allow the system detect objects accurately.

5- EXPERMINETAL RESULTS

We have tested our system using artificial cars and parking lot. As shown in the figures used above, we were able to prove that the system was able to detect the objects "Cars" and come up with the number of available spots. This was also tested against different images that has different number of cars parked, example below:

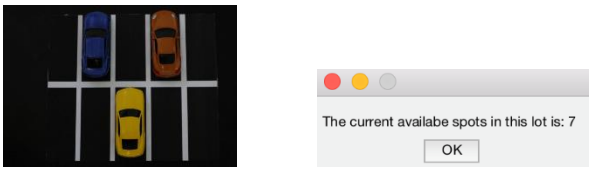

Fig 13: Test for parking lot with 3 cars occupied

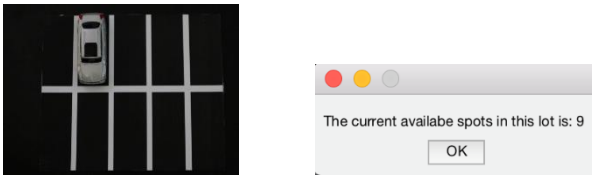

Fig 14: Test for parking lot with 1 cars occupied

6- CONCLUSION AND FUTURE WORK

Smart parking slot detection systems based on image processing was designed and tested using MATLAB. The process involved different type of image processing and segmentation preformed which allowed the system to learn and count objects. We believe this is an efficient system comparing to other systems that requires more electronic hardware involved such as sensors.

The system was tested using images that have artificial cars and parking lot. The results were successful as the system was able to detect the number of cars against differed testing scenarios. The main idea in this project is to present a system that will be able to count the available spots in a parking lot. This system will help drivers to find a real-time spot to park rather than driving around trying to find one while all occupied.

In future work, the second part of the project will focus on adding the ability for the system to detect the number of parking spots in an empty lot. This will eliminate the user intervention to enter the lot spots number utilizing the initial image acquisition for an empty parking lot.

7- ACKNOWLEDGEMENT

This project was built using MATLAB for (SYS5100) Systems Engineering course at University of Ottawa. The authors would like to thank the course professor and TA for all the help provided during this project.

8- REFERENCES

1- Suhr, J. K., & Jung, H. G. (2014). Sensor Fusion-Based Vacant Parking Slot Detection and Tracking. IEEE Transactions on

Intelligent Transportation Systems, 15(1), 21-36. doi:10.1109/tits.2013.2272100

2- Nikam, A., Patil, P., Shinde, S., & Toppo, S. (2017). Smart Parking System for Locating Vacant Parking Slots. International Journal of Advanced Research in Computer Engineering & Technology.

3- Z. Rajani, M. Viegas, "Smart Parking Space Detection Using MATLAB and Internet of Things" *International Journal of Scientific Development and Research*, vol. 2, no. 3, March 2017.

4- Rawat, A., Manisha, M., Ragashree, M., Mrudula, S., & Suhas, S. (2018, April). Parking Space Detection Using Image Processing in MATLAB. Paper presented at International Research Journal of Engineering and Technology, Volume: 05, Issue: 04.

5- Yusnita, R., Norbaya, F., & Basharuddin, N. (2012, June). Intelligent Parking Space Detection System Based on Image Processing. Paper presented at International Journal of Innovation, Management and Technology, Vol. 3, No. 3.

6- Lopez, M., Griffin, T., Ellis, K., Enem, A., & Duhan, C. (2019). Parking Lot Occupancy Tracking Through Image Processing. Paper presented at Proceedings of 34th International Conference on Computers and Their Applications.

7- Döge, K., & Krimmling, J. (2012, September). Video-based parking space analysis for traffic management. Paper presented at 13th IFAC Symposium on Control in Transportation Systems, Sofia, Bulgaria.

8- Sinecen, M. (2016). Digital Image Processing with MATLAB. NTECH Open Science.