International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 16 (2017) pp. 5940-5945

Autonomous and Intelligent Park Detection and Reservation System Using Image Processing and Mobile Application System

Nayel Al-Zubi, [1]Moh'd Rassoul al-Hadidi, [1]Ayat Khraisat, [2]Hamzah Alzubi and [3]Saed Khawaldeh

[1]Computer Engineering Department, Al-Balqa' Applied University, Jordan.
[2]Department of Electrical Engineering, University of Liverpool, UK.
[3]Erasmus+ Joint Master Program in Medical Imaging and Applications; University of Burgundy (France), University of Cassino (Italy) and University of Girona (Spain).

[1]Orcid: 0000-0000-6251-9404

Abstract

This work presents an autonomous and intelligent system for detecting the parking spaces. It uses the already installed cameras without any reconfiguration, to capture a real-time image of the parking area and detect the parking spaces. The algorithm uses image processing to detect the available space of the park slices and guide the drivers to it. The complete system includes mobile application to reserve and book a park space.

The algorithm detects the spaces either indoor or outdoor parks irrespective of the daytime and in any angle the cameras were installed. This system could save time for the drivers and use the available camera with minimum cost.

Keywords: Image Processing, Park Spaces Detection, Mobile Application.

INTRODUCTION

The substantial increase in the motor vehicles in the past few years is clearly visible in the form of traffic and the cost of parking spaces in the societies. Currently, most of the existing car parks do not have an autonomous system. Most of them are manually managed and subjective to inefficient managements. Locating a car parking space is always inconvenience for car drivers because of the wasted time in searching for the available parking spaces. It is very crucial to design and establish a system that can track and manage parking space efficiently. The proposed system detects parking space using image processing and cloud based tracking application. Using the existing CCTV cameras and using image processing algorithms parking space can be tracked for vacancy and the live feed can be made available to the public using cloud and a dedicated application.

SYSTEM DESIGN

Identification of Parking Lots Spot

The system flow chart is shown in Figure 1.

Mark the Parking Lots

The first step to find empty parking lots is to identify parking lots spots by marking all of them with specific colored ellipses (in this project red has been chosen) in the center of each one. This is shown Figure 2.

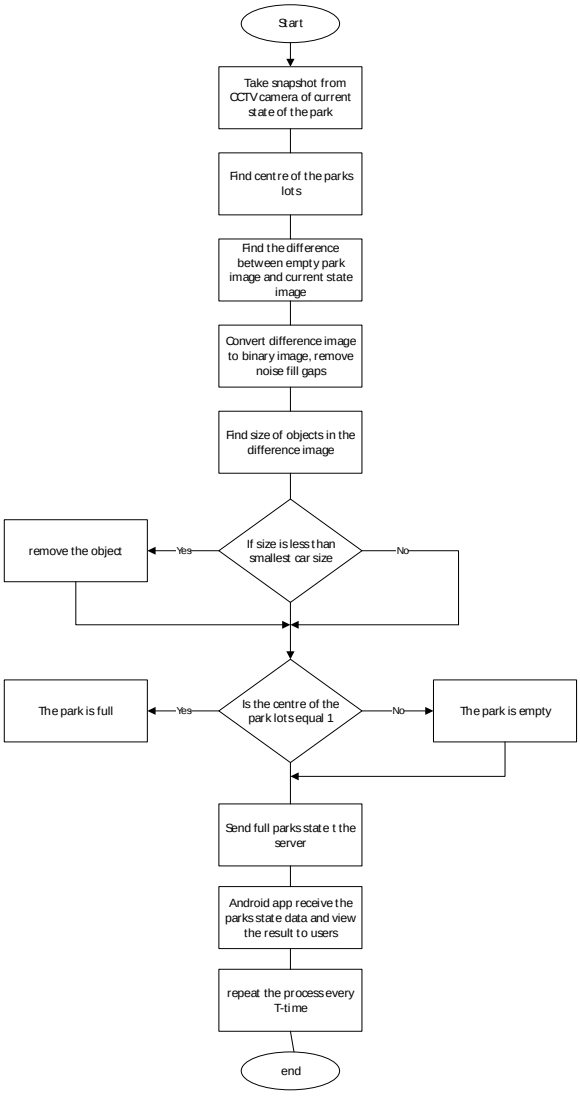

Figure 1: System Flow Chart

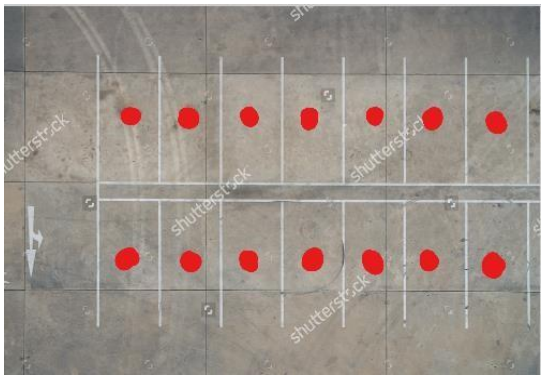

Figure 2: Marking parking lots

Extract Ellipses by Color

The red color extracting done by Initialize zero matrix of the same size of the original image then loop through original image to find I the current pixel in the iteration is red one, if it is it will be added to initialized matrix. After the loop was finished Initialized matrix was contain only the red objects from original image, as shown in Figure 3.

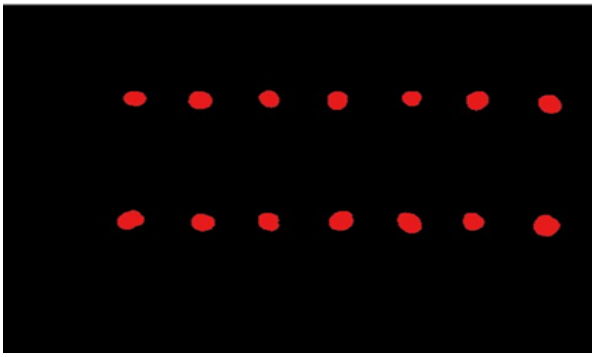

Figure 3: Extract Ellipses by Colour

Conversion Red Objects into Binary Image

The next process is to convert red objects image into binary image with thresholding of the first image, Figure 4.

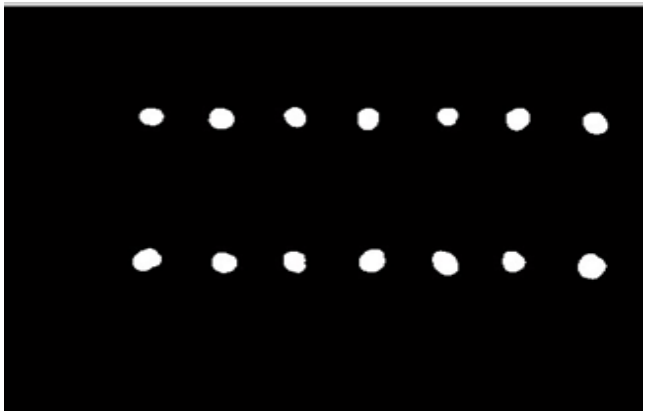

Figure 4: Convert Red Object into Binary Image

Removing Noise and Filling Gaps

Then gaps in the binary image will be filled and noise will be removed if it exists, Figure 5.

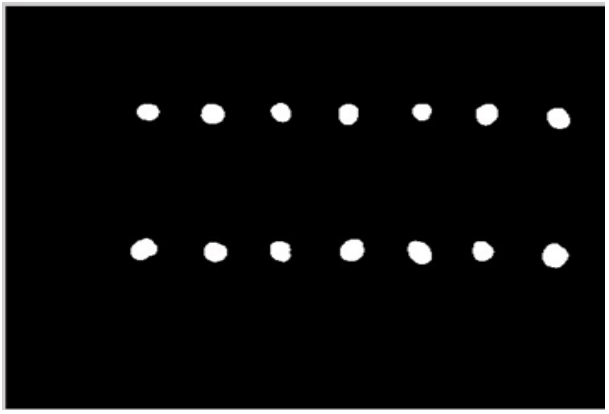

Figure 5: Fill the Gaps in the Binary Image

Determining the Center of each object

Finally, the center of each object in the binary image is determined which is also the centers of the parking lots and is saved for next steps in struct, and each center is numbered, as shown in Figure 6 and Figure 7.

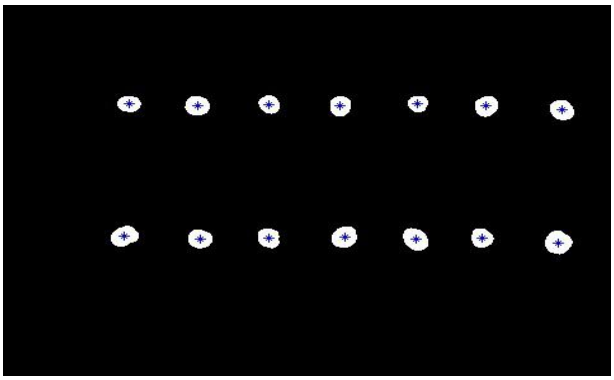

Figure 6: Determine the Centre of Each Object

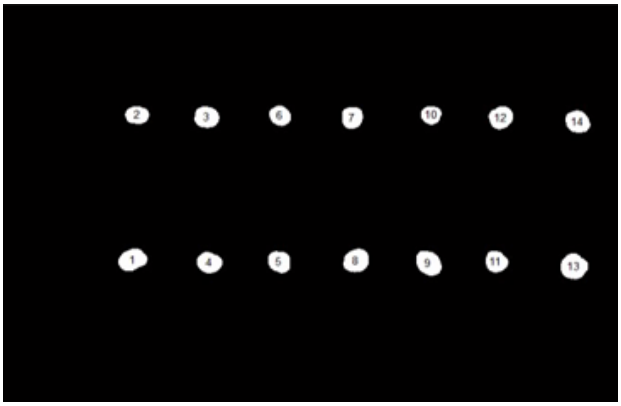

Figure 7: Number the Centre of Each Object

Determination of the available parking lots

The process flow is shown in Figure 8.

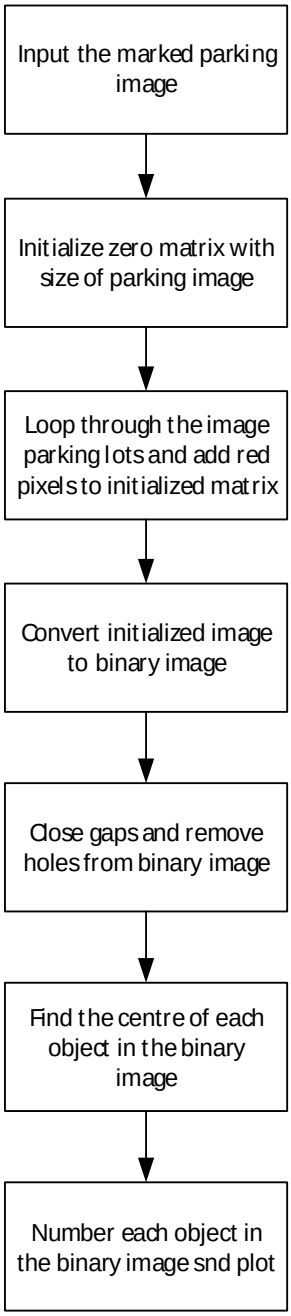

Figure 8: Process Flow

Subtract current status image to find difference in the car park status

Initial park status (when the park is empty) has been subtracted from current status image to extract cars in the park and remove unwanted objects in the park like lights, sidewalk…etc [2].

Images subtracting is done by subtracts each element in array of current status from the corresponding element in array of empty park and returns the absolute difference in the corresponding element of the output array, Figure 9.

Figure 9: Subtraction of initial park status from current image status

Convert the diff image into binary image

The result image from step one has been converted into binary image, binary image is important for the next steps, where binary image is a digital image that has only two possible values for each pixel. Typically, the two colors used for a binary image are black and white. The output image BW replaced all pixels in the diff image with luminance greater than level with the value 1 (white) and replaces all other pixels with the value 0 (black).

The level has been chosen which computes a global threshold (level). The level is a normalized intensity value that lies in the range [0, 1], Figure 10.

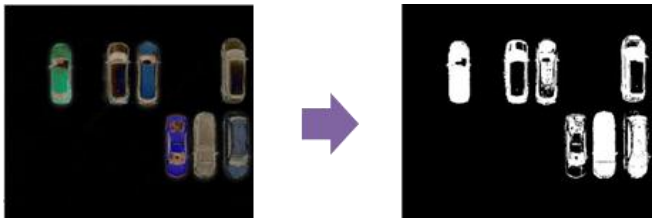

Figure 10: Convert image into binary image

Remove noise and unwanted small objects

The noise and unwanted objects are removed. Where the unwanted objects are all connected components (objects) that have fewer than P pixels from the binary image. This is producing another binary image. The default connectivity is 8, this operation is known as an area opening. P pixel has been changed from car park to another according to the park environment

Figure 11: Removal of unwanted objects

2.2.4. Close gaps and connect nearby blobs

function performs closing with the structuring element Create a disk-shaped structuring element. Use a disk structuring element to preserve the circular nature of the object. Specify a radius of x pixels so that the largest gap gets filled.in this project radius has the value of 10.

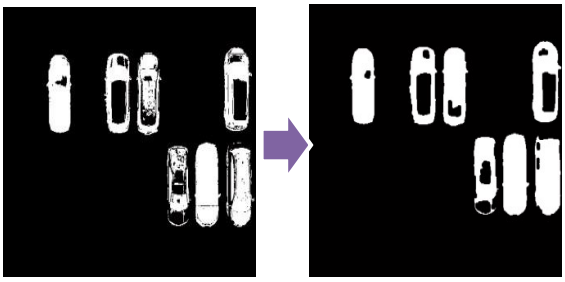

Figure 12: Closing gaps and connecting nearby objects

2.2.5. Generate convex hull image from binary image

To generate convex hull a function is used to compute the "convex hull of a binary image",'objects', which means to Compute the convex hull of each connected component of BW closed individually, **Error! Reference source not found.**.

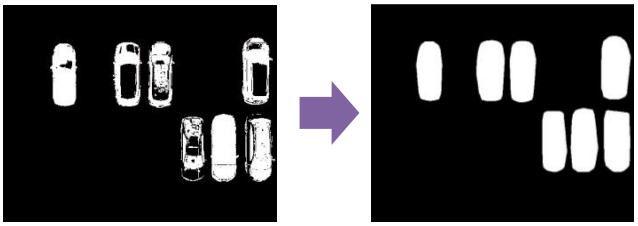

Figure 13: Generate the convex hall image

Label connected components in 2-D binary image

Each object in the image is labelled, and is used to be converted to a label matrix, L, into an RGB colour image for the purpose of visualizing the labels regions. The function determines the colour to assign to each object based on the number of objects in the label matrix and range of colours in the colormap. Another function picks colours from the entire range.

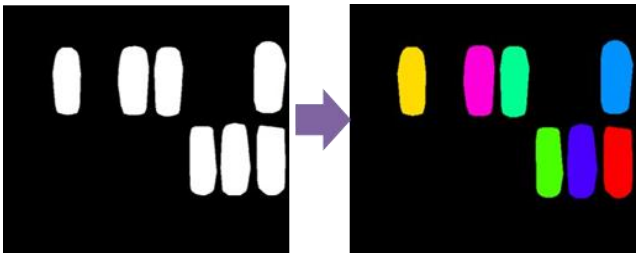

Figure 13: Label connected components in 2-D binary image

Calculate objects area

Region props function Returns a scalar that specifies the actual number of pixels in the region, the area size depends on the image size. The area of the objects in the labelled image.

areavalues <1x7 double>

	1	2	3	4	5	6	7	8
1	38952	43887	41276	39346	44125	47191	43540	
2								
3								

Figure 14: Calculation object area

Isolate unwanted objects

The area has been computed to help determine if the object can cover park lot or not, not every object in the image is a car it can be man, box, cat..etc., such objects have to be removed before proceeding to next steps. [4]. This has been done using comparing smallest object area that can cover the park with all objects area in image and isolate smaller object by applying find function in MATLAB, in this example there are no isolated objects.

Determine full parking lots locations

From last section, the centre of each parking lot has been calculated and saved in struct centroids. If the centre is covered then the parking lot is full, actually this is not accurate so the if statement checks around the object also to get more accurate result.

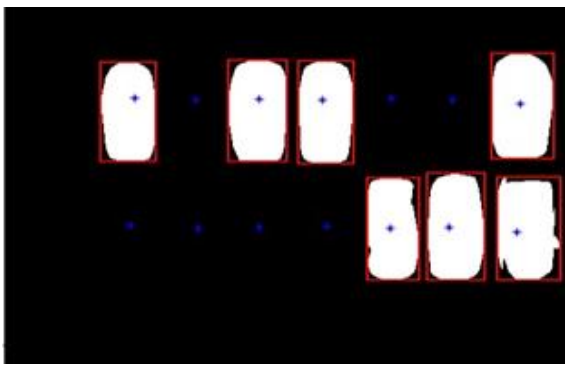

Figure 15: Check if the centre is covered

The full parking lots locations are saved in full struct, according to the numbering technique used after centers has been found in the last section

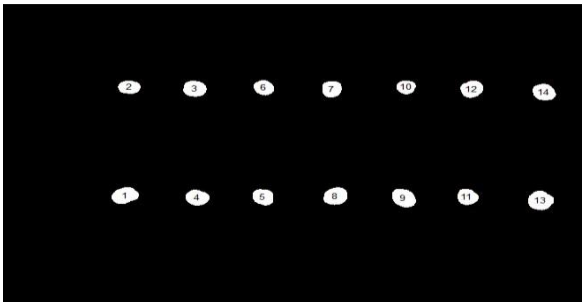

Figure 16: Numbered parking lots spots

The full parking lots spots are:

full <1x7 double>

	1	2	3	4	5	6	7	8
1	2	6	7	9	11	13	14	
2								
3								

Figure 17: Numbers of full parking lots

The empty parking lots spots are:

empty <1x7 double>

	1	2	3	4	5	6	7	8
1	1	3	4	5	8	10	12	
2								
3								

Figure 18: Numbers of not available parking lots

Figure 19: Result which appear to public

ANDROID APPLICATION

The android app user interface of this project is a very simple one with a few windows, the main target of this app is to receive data about current status of the parking lots from server then to display data in a familiar way to the users, then users can reserve a parking lot for some time, this order will be sent to the server through internet to parking lots administrator [3]. The first window of the app contains numbers of buttons for each park uses this app. For testing there is only three parks called park1, park2, park3.

Figure 20: Numbers of buttons for each park uses this app

When clicking on any park a picture of the park will be shown with numbers on each park, the picture of the park will help the user to pick the suitable park for him, in bottom of the screen there is ordering button to reserve park.

Figure 21: A picture of the park with numbers on each park

when user click on it, list of the available parks will be requested through http request GET, the parks that's not available are reserved or full.

Figure 22: A picture of the park with List of the available parks

When user click on the park there will http request POST will be sent to the server to add the park to reserved parks, If this process done successfully the toast massage will appear contain the number of park that has been reserved

Figure 23: Toast massage that contain the number of park that has been reserved

CONCLUSIONS

In this work, an algorithm is developed that able to display the current status of the parking using the least possible costs. Many different parking guidance systems have been developed and designed to shorten the searching time for vacant parking lots. This is especially crucial for drivers who need to search for available car parks during peak hours or when the car parks are almost full, this system will help car drivers to find an empty car parking lot thereby saving people's time and fuel. This software has been developed using image processing algorithms. Future work could be to extract vehicle's plate number which would be helpful for security purposes or as simple as finding a car, it could also be to deploy the same software using other platforms such as iOS.

REFERENCES

[1] [1] Fundamentals of Image Processing Ian T.Young Jan J. Gerbrands Lucas J. van Vliet

[2] [2] http://www.idc.com/prodserv/smartp hone-market-share.jsp

[3] Head First Android Development Book by Anthony J.F. Griffiths and David Griffiths

[4] MATLAB Programming Fundamentals

[5] Parking Space detection based on camera and image processing. https://www.youtube.com/ watch?v=8A7vfMP0r7s