Journal of Engineering and Applied Sciences 13 (13): 4981-4986, 2018
ISSN: 1816-949X

Intelligent Parking System using Circle Hough Transform

[1]Ajay Kamath and [2]R. Swarnalatha
[1]Department of EIE,
[2]Department of EEE, BITS-Pilani, Dubai Campus, Dubai, UAE

Abstract: Image processing is the science of extracting vital information from an image like edges, color distribution, etc. This information can be used as for analysis and also as a feedback to control many processes. This study gives an insight of how image processing can be used to control crowding of vehicles in a parking lot and compares this system to current smart systems already in place in terms of efficiency and feasibility. A new method to achieve the aforementioned goals involves a slight alteration to the appearance of parking spaces, thus, streamlining the algorithm for a faster detection without a heavy load on computing power. The different algorithms that are the focal point for this study are: canny edge detection and circle hough transform.

Key words: Image processing, manipulation, hough transform, circle, parking system, computing

INTRODUCTION

Let us begin with comparing the application at hand as an analogy of a control system with a feedback. This study starts with this basic idea as a foundation and then builds up on this idea to solve a practical challenge. (Davies, 2000).

The basic elements of process control involves a desired set point, a controller to correct errors in result and a feedback loop that provides the current output to the controller. The feedback is for comparison of the current output with the desired set point. The controller then gives the control signal to an actuator, so as to influence the output and correct the error (Davies, 2000; Arora *et al*., 2008).

Image processing is the science of manipulation of digital pictures and analysis of these digital pictures to obtain numeric results for further computations. Image processing is a field with vast number of applications ranging from artificial intelligence, robotics to security systems.

A feedback loop comprises of a sensor to sense the variable in question and a transducer converts it to a usable signal which is understandable by the controller. These sensors are usually designed with a specific goal in mind and often their use cannot be expanded to other purposes. This is where image processing comes into the picture. Modern day digital cameras are equipments designed to capture digital images which are an array of numeric data. This data can be processed to provide information for vital feedbacks. Here, cameras are acting as a sensor for the required feedback with the added advantage of expansion to other purposes.

Literature review

Data and information from images: An image initially begins as a capture of an array of data from the environment. Digital images are stored as a two dimensional array of data. Each element of the array is called pixel. Displaying these pixels in the intended two dimensional order is what gives us a final image we can see and understand. The size of the aforementioned array is what gives the size of the image. For example, an image with a resolution of 1920×1080 is basically a two dimensional array of pixels with 1920 rows and 1080 columns (Lew, 2001).

Now this image that has been captured, doesn't have any information that the computer can understand. Thus, this image undergoes image processing and then the algorithms return values that can be used by the computer for further calculations. These values returned that can be used is termed as the information extracted from the array of data (Lew, 2001). This is the basic theory that comes in handy for cameras being used for artificial intelligence and automation.

Edge detection algorithms: Edge detection is a major section of innovation in the field of image processing. As the name suggests, edge detection involves detection of the boundaries in an image, so as to distinguish and analyze the objects that can be seen in an image. These algorithms work on the principle of marking discontinuities in the colors and output levels of the image while the algorithm does an overall scan of the image (Kaur and Kaur, 2016).

Edge detection is widely used to feed information extracted from processed images to neural networks

Corresponding Author: Ajay Kamath, Department of EIE, BITS-Pilani, Dubai Campus, P.O. Box 3112, Deira, Dubai, UAE

engines to help develop artificial intelligence and to develop an artificial understanding of the world around us to computers. It works by marking the areas of discontinuities of in brightness and colors in the image.

The most common edge detection algorithm in use is canny edge detection (Kaur and Kaur, 2016). It is a multi-stage algorithm which involves the following steps.

Application of a Gaussian filter: This step serves two purposes. Firstly, it reduces the overall noise present in the image. Edge detection algorithms are very sensitive to noise in the image. There is a possibility that the small specks that appear as noise can be detected as tiny objects resulting in erroneous edge detection. Secondly, it gives the sharp object edges a gradient fading towards the background which is the basic principle on which canny detection works (Cope and Rockett, 2000).

Finding intensity gradient of the image: This step passes the image through a sobel kernel which is a preliminary detection of the edges. The basic logic behind this is that the gradients caused by the gaussian filter is always literally perpendicular to the edge of the object.

Non-maximum suppression: This stage removes the unwanted pixels that may have a gradient and was marked as an object.

Hysteresis thresholding: This passes the image through two threshold values that gives sharper edges resulting in a high clarity result.

Hough transform: Hough transform is a process for extracting features which is used in digital image processing and image analysis. This process is used to find instances of objects within a specific class of shapes using a voting procedure (Wang and Chen, 2009).

The classical hough transform was used to mark and position lines in the image. But now it is being used to mark shapes like circles and ellipses. This brings us to Circle Hough Transform (CHT). CHT is a variation of hough transform used to detect circular objects within a digital image. The circles are detected by the aforementioned “Voting” procedure in hough parameter space and then picking out the local maxima in an accumulator matrix. On a cartesian plane, a circle is represented by the following equation:

$$(x-a)^2+(y-b)^2 = r^2$$

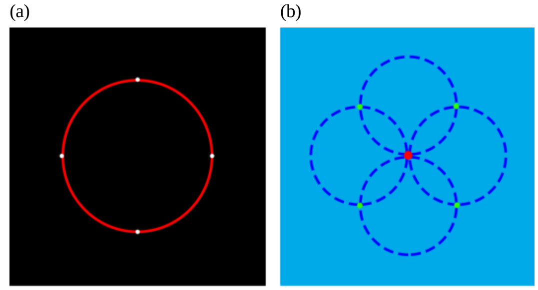

Fig. 1: Example of hough transform being performed on an image: a) Original and b) Hough parameter space

Where:
x, y = Points on the cartesian plane
a, b = Coordinates of the center of the circle
‘r’ = The radius of the circle

The concept of circle hough transform revolves around the above mentioned equation. An additional feature used in circle hough transform is edge detection (Chai *et al*., 2014).

Take a sample image with a circle in Fig. 1a. Consider 4 points, marked on the circle in white as shown on the circle. Note that the radius of the circle is assumed to be known for this example. For each of the marked points (x, y) on the shown circle, define another circle with the marked points as the centers and radius r. This forms the hough parameter space as Fig. 1b.

This is where the accumulator matrix comes into play. The accumulator matrix is used to track all the points in the Hough parameter space where an intersection occurs. From here, the local maxima point can be found (marked in red) in the Hough parameter space. The position (a, b) of this marked maxima is the center of the original circle (Chai *et al*., 2014).

This algorithm can then be extrapolated and used to find multiple circles within a radius range. This extrapolation of hough transform is what will be used in the application of image processing to control crowding in a parking lot.

MATERIALS AND METHODS

Overview: One everyday example of automation where image processing can be applied is to control crowding of vehicles in a parking lot (Banerjee *et al*., 2011).

Currently, every mall in Dubai and many around the world have systematic parking monitoring systems in place. These systems help the motorists entering those respective parkings to be able to find parking spots for their vehicles with ease. It helps them to avoid areas if the system returns that the area has no current free spaces.

In more advanced systems such as the ones in place in the city center malls and also in Dubai Mall, an ultrasonic sensor above each parking space which detects the presence of a sizable object in that respective space, in this case, a car. Now in the event of a car already present, it will signify this using an LED which turns from green (when empty) to red (when occupied). Before entering the row of parkings, a motorists can take a quick scan of the row and try to spot a green light before advancing into said row. An added feature is that the simple boolean data from all sensors are sent to a central system for monitoring and thus that will display the number of parkings open in that parking lot, thus, giving an overall density of the crowd in the lot (Sun *et al*., 2015).

The disadvantages of the above-mentioned system is that there is a large amount of complexities in the installation of such a system. Considering the capacities of a large mall like Dubai Mall with over thousands of parking spaces, installing a sensor over each parking space is a huge and time intensive process.

The solution for this disadvantage is to use image processing and cameras to monitor the parking lot. This reduces the complexity of installation as one camera can serve to monitor a whole area of the parking lot. The salient feature here being that most parking lots of commercial complexes have cameras in place for security and surveillance. It will be just a matter of calibrating these cameras, acquiring these images and posting them to the program which will process the image and return the number of spots available to incoming motorists.

The advantage of this system is that it serves 2 purposes: one, as a convenience for incoming motorists to find parking spots easily and also, it doubles as a security system for surveillance. Both of which are necessary for any commercial complex in todays day and age.

The solution to the above mentioned drawback is the usage of cameras mounted and calibrated in a way to recognize the difference between empty and occupied parking spaces (Banerjee *et al*., 2011).

Setup and calibration: A rig of a parking lot is prepared as shown in Fig. 2. A green circle is placed within the central area in a way such that a vehicle occupying the spot in mention will eclipse the circle from the view of the camera monitoring the lot (Jung, 2014). The green circle thus, acts as a marker as to whether the spot is occupied or vacant. The parking lot rig that has been designed has a total of 6 parking space as shown in Fig. 2. Now, once it passes through the circle hough transform algorithm, the output image marks all the green circles visible, counts the number of green circles, thus, returning the number of parkings available. This is where circle hough transform comes into the picture.

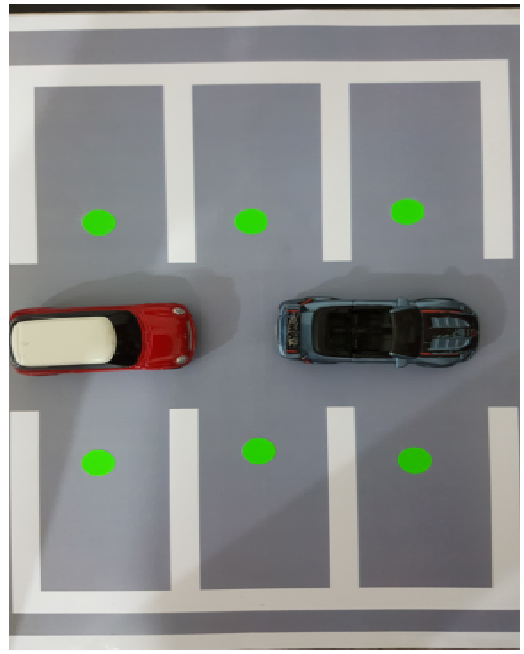

Fig. 2: A miniature rig of a parking lot

Currently, for the algorithm to work at full potential, the camera has to be placed at an almost vertical, i.e., 90° with respect to the plane of the parking lot. This vertical angle helps in easy detection of the circle. At an oblique point of view for the camera, the circle will appear as an ellipse which may cause the algorithm to not be able to detect the green marker circles on the parking spaces.

A further extension of this algorithm could be to expand the circle hough transform algorithm to detect ellipses to allow for an easier calibration of the camera as a sensor in this environment.

Application of circle hough transform: The green circle acts as a marker as to whether the spot is occupied or vacant. Simply enough, if a parking space is occupied, the green circle is not visible.

The parking lot in mention has a total of 6 parking spaces. Now, once it passes through the algorithm, the output image marks all the green circles visible, counts the number of green circles, thus, returning the number of parkings available.

This code marks the green circles and stores a matrix of the respective circle center coordinates in the variable 'Centers'. Also it stores their respective radii in the variable 'radii'. Figure 3 shows the code window in MATLAB.

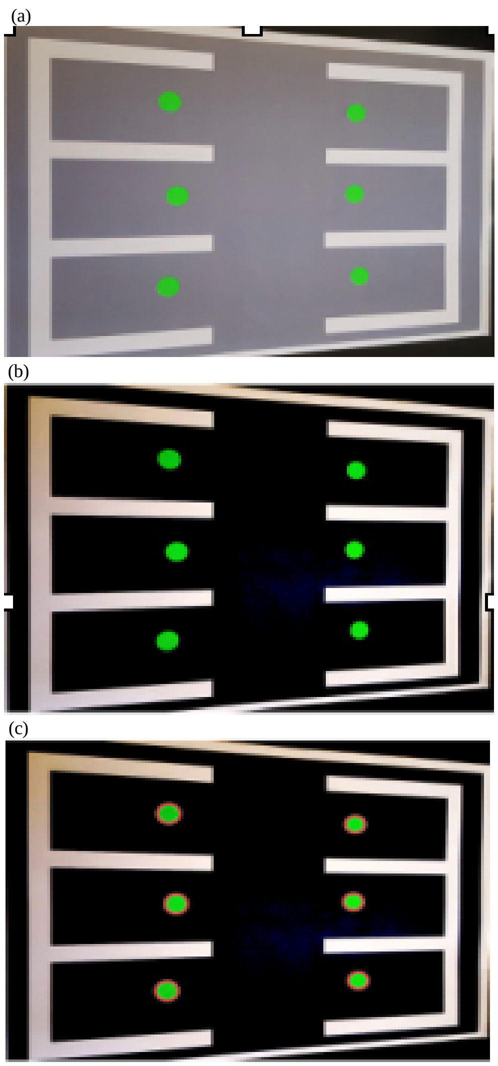

Fig. 3: a) The initial image of empty parking lot; b) The image with a higher contrast and c) The green circles marked in red

RESULTS AND DISCUSSION

Completely empty parking lot: This uses an image with a completely empty parking lot and passed to the algorithm. This step was used as a calibration step, so that, given a certain camera distance from the surface and camera angle, we can find the correct radius range for which the algorithm can find the green circle markers.

Note: The image contrast had to be drastically increased, so that, when the image is converted to grayscale, the background tarmac is darker than the bright green circle. This causes a sharper edge for the circle which can be detected faster and more accurately. Figure 3 shows the different phases of the image being processed (Fig. 4).

Partially occupied parking lot: This iteration simulates a situation wherein 3 out of the 6 parking spaces are occupied. Once the image of the parking lot is passed to the algorithm, it returns an image where the green markers in the available spots are marked in red and also returns the correct number of parking spaces available to park in. Figure 5 shows the image of the partially occupied parking lot being processed to show a successful end result (Fig. 6).

Both the iterations of the experiment have shown successful results. This proves to be a novel idea with a minimum amount of software maintenance to control crowding in a parking lot.

In the first iteration, "Completely empty parking lot": The rig was made with 6 parking slots and none of them occupied. The script returned an accurate result of 6 parking spaces being empty as seen in the following image of the command window.

In the second iteration, "Parially occupied parking lot": 3 spaces out of the 6 on the rig were occupied by vehicles. The script once again returned the accurate no. of spaces being empty, i.e., 3 parking spaces empty.

Improvisation of the algorithm: During the initial few tests and trials of the experiment. The process would be to capture the image and directly run it through the

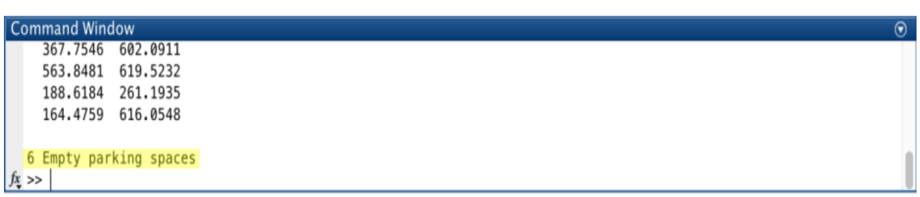

Fig. 4: MATLAB command window showing the number of parking spaces available in the empty parking lot

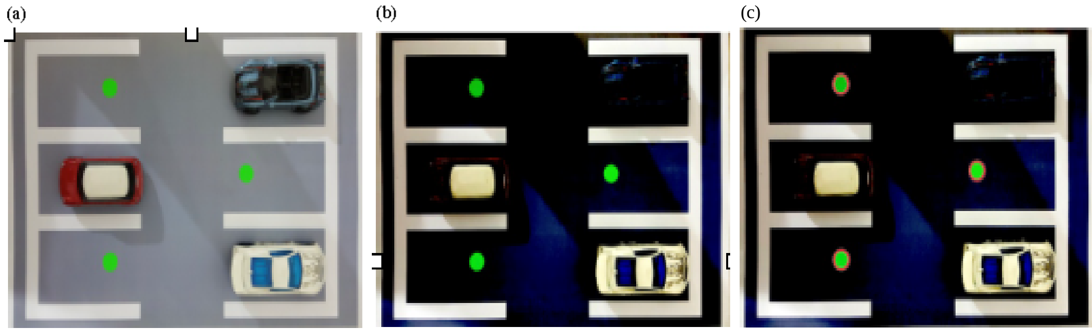

Fig. 5: a) The initial image; b) The image with a higher contrast and c) The green circles on the empty spaces are marked in red

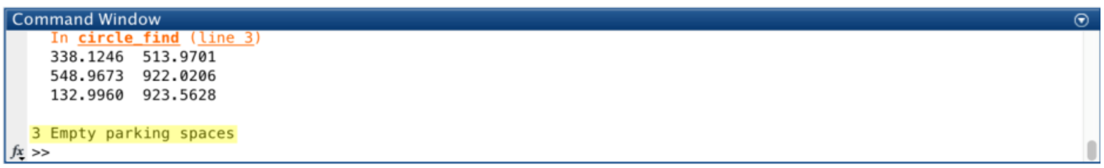

Fig. 6: MATLAB command window showing the correct number of parking spaces available in a partially occupied parking lot

algorithm. The problem with those trials was that in a grayscale image, there wasn't a clear distinction between the circular marker and the grey tarmac. This yielded results whose accuracy was very dependent on the ambient light. In the event of any error in angle of light and/or intensity of light the accuracy of the edge detection would be lowered significantly.

As you have observed in Fig. 3 and 5, in stage b of both images, the image goes through a high contrast filter, so as to give a significant distinction between the marker and the tarmac once the image is passed on from the RGB to grayscale domain for the circle hough transform algorithm. This produced very accurate results as observed in the iterations of the experiment and these results were independent of the ambient light as the high contrast filter created a distinction in itself and brought only bright circular objects into focus.

CONCLUSION

The most optimum method for controlling crowding in a parking lot is with circle hough transform. This is where the parking lot is designed with a bright green circle marker on each parking space which gets covered by a vehicle in that respective parking space. Circle hough transform method is the most optimum method because the logic is easy to parse and it is the least complex to implement. There is a much thinner margin of error with this method because of the fact that the algorithm is very modular.

For further development of this system, it can be extended to a system where multiple cameras can monitor multiple areas of a larger parking lot and the data from all these cameras can be consolidated at one central computing system. This computing system can then deploy numeric results to LED displays at parking lot entrances to help incoming motorists to find parking spaces faster. From a practical aspect, the cameras capturing images should be placed at an angle inside indoor parking lots which do not provide much vertical space for a camera to be completely vertical above the plane of the parking lot. In such a scenario, from the view point of the camera, the circular marker will appear as an ellipse. Thus, circle hough transform can be modified to detect these ellipses.

ACKNOWLEDGEMENTS

On the outset, we would like to express our gratitude to Prof. R.N. Saha, Director of BITS-Pilani, Dubai Campus and Dr. Abdul Rajak, HOD of EEE Department. We would also like to thank Mr. Avinash Kamath who helped to refresh web development skills which has been used extensively in this project to give a realization to theoretical algorithms on edge detection using Javascript.

REFERENCES

Arora, S., J. Acharya, A. Verma and P.K. Panigrahi, 2008. Multilevel thresholding for image segmentation through a fast statistical recursive algorithm. Pattern Recognit. Lett., 29: 119-125.

Banerjee, S., P. Choudekar and M.K. Muju, 2011. Real time car parking system using image processing. Proceedings of the 3rd International Conference on Electronics Computer Technology (ICECT'11) Vol. 2, April 8-10, 2011, IEEE, Kanyakumari, India, ISBN:978-1-4244-8678-6, pp: 99-103.

Chai, Y., S.J. Wei and X.C. Li, 2014. The multi-scale hough transform lane detection method based on the algorithm of Otsu and Canny. Adv. Mater. Res., 1042: 126-130.

Cope, R.K. and P.I. Rockett, 2000. Efficacy of gaussian smoothing in canny edge detector. Electron. Lett., 36: 1615-1617.

Davies, E.R., 2000. Image processing and its applications. Electron. Commun. Eng. J., 12: 194-194.

Jung, H.G., 2014. Semi-automatic parking slot marking recognition for intelligent parking assist systems. J. Eng., 1: 1-8.

Kaur, P. and B. Kaur, 2016. 2-D geometric shape recognition using canny edge detection technique. Proceedings of the 3rd International Conference on Computing for Sustainable Global Development (INDIACom'16), March 16-18, 2016, IEEE, New Delhi, India, ISBN:978-1-4673-9417-8, pp: 3161-3164.

Lew, M.S., 2001. Principles of Visual Information Retrieval. Springer, London, UK., ISBN:1-85233-381-2, Pages: 359.

Sun, W., N. Shibata, M. Kenmotsu, K. Yasumoto and M. Ito, 2015. A method for navigating cars in multilevel parking facility. J. Inf. Process., 23: 488-496.

Wang, L. and L.Q. Chen, 2009. Concentric circle detection based on chord mid-point hough transform. J. Comput. Appl., 7: 1937-1939.