

# IMPLEMENTATION OF IMAGE PROCESSING IN REAL TIME CAR PARKING SYSTEM

Ms.SAYANTI BANERJEE

*Ajay Kumar Garg Engineering College, Department of electrical and electronics  
Ghaziabad,UP ,India*

Ms.PALLAVI CHOUDEKAR

*Ajay Kumar Garg Engineering College, Department of electrical and electronics  
Ghaziabad,UP ,India*

Prof.M.K.MUJU

*Ajay Kumar Garg Engineering College, Department of Mechanical  
Ghaziabad,UP ,India*

## Abstract

Car parking lots are an important object class in many traffic and civilian applications. With the problems of increasing urban traffic congestion and the ever increasing shortage of space, these car parking lots are needed to be well equipped with automatic parking Information and Guidance systems. Goals of intelligent parking lot management include counting the number of parked cars, and identifying the available location. This work proposes a new system for providing parking information and guidance using image processing. The proposed system includes counting the number of parked vehicles, and identifying the stalls available. The system detects cars through images instead of using electronic sensors embedded on the floor. A camera is installed at the entry point of the parking lot. It captures image sequences. The image sequences are then analyzed using digital image processing for vehicle detection and according to the status of vehicle occupancy inside, real time guidance and information is provided to the incoming driver.

**Keywords:** Car parking, Image Processing, Edge detection, image matching, Prewitt operator

## 1. Introduction

At the parking lots located in resting facilities of the expressway, information concerning the state of congestion is offered to drivers as previous notice by the traffic control system. To realize this state of congestion, the sensors for detecting cars have already been set up at the entrance and the exit or under the road surface of the parking section. On the other hand, as the hardware for image processing has been developing, the image processing has been applied to many kinds of purpose recently. The following is a brief survey of the papers that have emphasized the detection of parking cars in the parking lot. [2] used the change of the variance of brightness on the road surface in the stationary image (difference between consecutive frames). [4] Showed the method to detect the car moving by the subtraction between consecutive images. In [6], the authors proposed the method to count cars by tracking the moving objects for the whole area of the outdoor parking lot as compared with for every parking division in [4]. [5] Showed that it is effective to use time differential images to extract moving objects from stationary objects. However, a moving object can often be taken as many regions (called moving regions) in the differential image.

In the present work the designed system aims to achieve the following.

- Images of the incoming cars are captured in real time
- Depending upon the status of car occupancy inside they are allowed to enter the parking lot
- Once the parking in the left side is full, car is directed towards right.
- Once both sides of the parking lot are full, no car is allowed to enter the parking lot.

Components of the current project

- Hardware module
- Software module
- Interfacing

Hardware Module

Image sensors: In this project a USB based web camera has been used.

Computer: A general purpose PC as a central unit for various image processing tasks has been used.

Platform: consisting of a few toy cars and LEDs (prototype of the real world car parking system).

Software Module

MATLAB version 7.8 as image processing software comprising of specialized modules that perform specific tasks has been used.

#### Interfacing

The interfacing between the hardware prototype and software module is done using parallel port of the personal computer. Parallel port driver has been installed in the PC for this purpose.

## 2. Methodology

Following are the steps involved

- Image acquisition
- RGB to gray conversion
- Image enhancement
- Image matching using edge detection

#### Procedure

##### Phase1:

- Initially image acquisition is done with the help of web camera
- First image of a car is captured
- This image is saved as reference image at a particular location specified in the program
- RGB to gray conversion is done on the reference image
- Now gamma correction is done on the reference gray image to achieve image enhancement
- Edge detection of this gamma corrected reference image is done thereafter with the help of prewitt edge detection operator

##### Phase2:

- Images of the cars entering the parking lot are captured at an interval of 2 seconds
- RGB to gray conversion is done on the sequence of captured images
- Now gamma correction is done on each of the captured gray image to achieve image enhancement
- Edge detection of these real time captured images of the cars is now done with the help of Prewitt edge detection operator

##### Phase3:

- After edge detection procedure both reference and real time images are matched and if they match each other by more than 90% the incoming car is allowed to enter the parking lot.
- In this project the designed hardware has been considered to have a maximum capacity of 20 cars which is divided into two parts-right side and left side. First the cars are guided towards left. Once the left side is full, the cars are directed towards the right side of the parking lot.

## 3. Image Enhancement

The acquired image in RGB is first converted into gray. Now we want to bring our image in contrast to background so that a proper threshold level may be selected while binary conversion is carried out. This calls for image enhancement techniques. The objective of enhancement is to process an image so that result is more suitable than the original image for the specific application. There are many techniques that may be used to play with the features in an image but may not be used in every case. Listed below are a few fundamental functions used frequently for image enhancement.

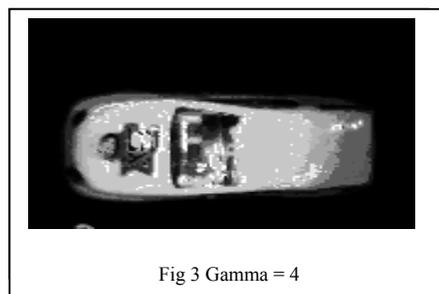
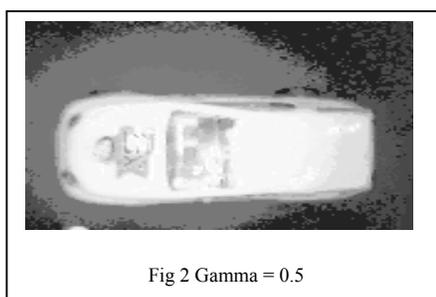
- Linear (negative and identity transformations)
- Logarithmic (log and inverse log transformations)
- Power law transformations(gamma correction)
- Piecewise linear transformation functions

The third method i.e., power law transformation has been used in this work. The power law transformations have the basic form

$$s = cr^\gamma$$

Where S is output gray level, r is input gray level, c and  $\gamma$  are positive constants. For various values of gamma applied on an acquired image we obtained the following graph shown in figure1.

From this figure it is evident that the power law curves with fractional values of  $\gamma$  map a narrow range of dark input values into a wide range of output values with the opposite being true for higher values of input levels. It depicts the effect of increasing values of  $\gamma > 1$ . The images are shown with  $\gamma = 1, 2, 3, 4, 5$  as may be seen, the figure with  $\gamma = 1$  gives the best results in terms of making fine details identifiable. As is evident the fractional values of  $\gamma$  can not be used since these attempts show a reverse effect of brightening the image still further which we may find as undesirable for the present case.



#### 4. Edge Detection and Image Matching

Step 1: Edge detection: Among the key features of an image i.e. edges, lines, and points, we have used edge in our present work which can be detected from the abrupt change in the gray level. An edge essentially demarcates between two distinctly different regions, which means that an edge is the border between two different regions.

Here we are using edge detection method for image matching:

- Edge detection methods locate the pixels in the image that correspond to the edges of the objects seen in the image.
- The result is a binary image with the detected edge pixels.
- Common algorithms used are Sobel, Prewitt and Laplacian operators.

We have used gradient based Edge Detection that detects the edges by looking for the maximum and minimum in the first derivative of the image.

- First derivative is used to detect the presence of an edge at a point in an image.
- Sign of the second derivative is used to determine whether an edge pixel lies on the dark or light side of an edge.

The change in intensity level is measured by the gradient of the image. Since an image  $f(z, y)$  is a two dimensional function, its gradient is a vector

$$\begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{df}{dx} \\ \frac{df}{dy} \end{bmatrix} \quad (1)$$

The magnitude of the gradient is given by

$$G[f(x,y)] = \sqrt{G_x^2 + G_y^2} \quad (2)$$

The direction of the gradient is

$$B(z, y) = \tan^{-1} (G_y/G_x) \quad (3)$$

where the angle  $B$  is measured with respect to the X-axis. Gradient operators compute the change in gray level intensities and also the direction in which the change occurs. This is calculated by the difference in values of the neighboring pixels, i.e., the derivatives along the X-axis and Y-axis. In a two-dimensional image the gradients are approximated by

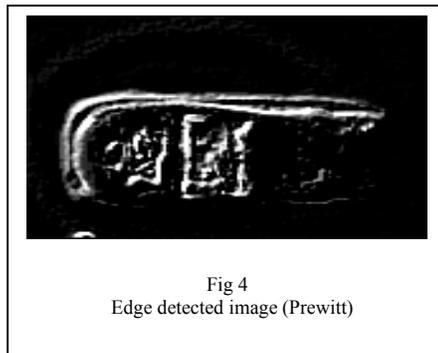
$$G_x = f(i+1,j) - f(i,j) \quad (4)$$

$$G_y = f(i,j+1) - f(i,j) \quad (5)$$

Gradient operators require two masks, one to obtain the X-direction gradient and the other to obtain the Y-direction gradient. These two gradients are combined to obtain a vector quantity whose magnitude represents the strength of the edge gradient at a point in the image and whose angle represents the gradient angle.

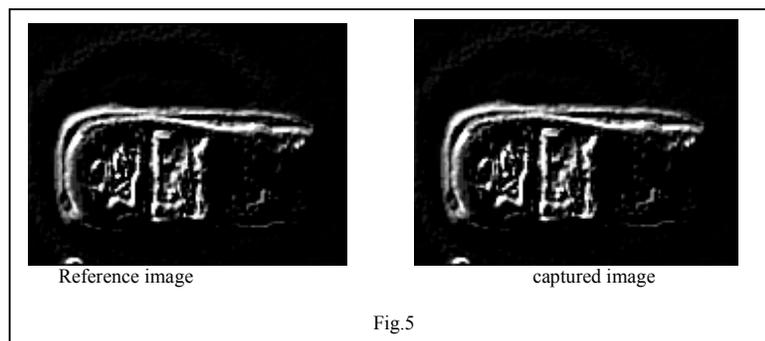
The edge detection operator we have used in the present work is Prewitt. Mathematically, the operator uses 3X3 kernels which are convolved with the original image to calculate approximations of the derivatives – one for horizontal changes, and one for vertical. If we define  $A$  as the source image, and  $G_x$  and  $G_y$  are two images which at each point contain the horizontal and vertical derivative approximations, the later are computed as

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} * A \quad \& \quad G_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} * A \quad (6)$$



Step 2: Image matching: Edge based matching is the process in which two representatives (edge) of the same objects are pared together. Any edge or its representation on one image is compared and evaluated against all the edges on the other image.

Edge detection of reference and the real time images has been done using Prewitt operator. Then these edge detected images are matched and accordingly the cars are allowed to enter the parking lot.



## 5. Experimental Results

Experiments are carried out and we get the following results.

Result 1: Left side parking

At first the parking lot is vacant with maximum capacity of 10 cars in the left side. Therefore first 10 images of the incoming cars are matched with the reference image and cars are allowed to enter according to the percentage of matching.

Result 2: Right side parking

After the entry of 10 vehicles on the left side, cars are guided towards the right side parking.

Result 3: Similarly, after the entry of 10 vehicles on the right side, cars are not allowed to enter the parking lot.

## 6. Summary and Conclusions

A vision-based car parking lot management system is proposed in this paper. The vision-based method makes it possible to manage large area by just using several cameras. It is consistent in detecting incoming cars because it uses actual car images. It is cheap and easy-installed because of the simple equipments. Drivers can get useful real-time parking lot information from this system by the guidance information display.

## References

- [1] Christopher M. Bishop, "Pattern recognition and machine learning," Springer, 2006.
- [2] E.Maeda, and K.Ishii: "Evaluation of Normalized Principal Component Features in Object Detection", IEICE Trans. Information and Systems, Vol.J75-D-11, 3, pp.520-529 (1992)
- [3] I. Masaki, "Machine-vision systems for intelligent transportation systems," IEEE Conference on Intelligent Transportation System, Vol.13 (6), pp.24-31, 1998.
- [4] K.Ueda, I.Horiba, K.Ikeda, H.Onodera, and S.Ozawa : An Algorithm for Detecting Parking Cars by the Use of Picture Processing ", IEICE Trans. Information and Systems, Vol.J74-D-11. IO. pp.1379-1389 (1991).
- [5] M.Yachida, M.Asada, and S.Tsuji : " Automatic Analysis of Moving Images", IEEE Trans. Pattern Anal. & Mach. Intell, Vol.PAM1-3, I . pp.12-19 (1981)
- [6] T.Hasegawa, and S.Ozawa : " Counting Cars by Tracking of Moving Object in the Outdoor Parking Lot ", IEICE Trans. Information and SystemsVol.J76-D-II, 7, pp.1390-139A (1993)