Asian Journal of Convergence in Technology Volume IV Issue I
ISSN NO: 2350-1146 I.F-5.11

Smart Parking System

Suprit Atul Gandhi[1], Hasan Mohammad Shahid[2]
Department of Computer and IT
College of Engineering, Pune
[1]*supritgandhi@gmail.com*, [2]*hasan.mshahid2522@gmail.com*

***Abstract*—Traffic Congestion nowadays has increased to a large extent, it has been growing exponentially. Car parking is one of the major contributors to this issue. A large amount of fuel is burned everyday in search of parking slot everyday. Here, we have proposed a solution that can ease the driver's job for getting vacant available parking slot. Underlying idea is to direct driver to vacant parking spot using sensors and LED's.**

Keywords—Internet of Things, Smart Parking, LED, Sensors, Database.

I. INTRODUCTION

Parking nowadays is a great pain especially when the parking space is huge. Due to extreme traffic conditions and over-crowding of vehicles, searching for a vacant slot is very hectic. There is a direct need to tackle this issue. The required solution should not only ease the parking system but also it should bring the new revolution in transportation in the era of Smart World.

Fig 1 gives the detailed explanation of fuel that can be saved if we introduce Smart Parking Technology worldwide.

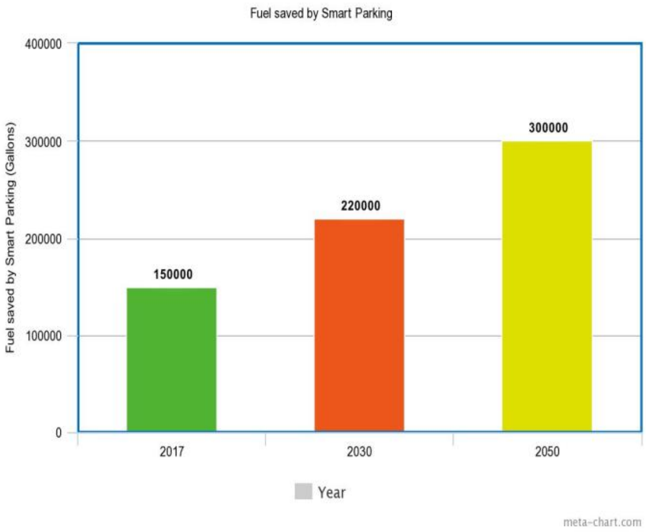

Fig 1: Fuel saved by smart parking systems (in Gallons)

Reliability in the parking system will be increased when the driver gets the correct indications i.e. the exact path which leads to the vacant parking slot. The solution proposed in this paper tries to utilize the path/road in between the parking slots used for commuting will be installed with the sensors and indicators which will be programmed to give the accurate path to the driver. At the entry, the guard will ask for the vehicle number and store it in the database and then he/she will assign a vacant parking slot to that number. The guard through his computer at the entrance will assign the particular slot to the driver and hence the path with which driver has to drive the car will be highlighted which will direct the car to the assigned parking slot. The path will be made using various indicators, LEDs, sensors, etc.

II. EXISTING SOLUTIONS

At present some countries have portals through which users can gain information about parking areas via internet. This system can give user the information about parking space but it won't be able to give which parking slot is vacant and occupied. Hence, such system cannot smartly handle the issue.

Car lifts along with automated robotic system, which automatically takes car to a particular parking spot as soon as car enters on a platform. This system cannot be installed by medium scale shopping malls, movie theatres as it can cost them a huge amount.

At many public places, the system only shows the availability but it cannot show the exact slot and path to the slot available.

Hence, there is the need to smartly find the path to the vacant spot.

III. OVERVIEW OF THE SOLTUION

A smart solution should not only indicate the availability of the vacant slot, but also it should direct the driver to that path. Each space should be tracked by the solution. For this, multiple types of sensors can be used. When the status of the sensors changes, the device at the entrance of the parking arena i.e. guard person will be notified about the same.

The basic working is every parking arena has white or yellow colored lining on the road so after installing LEDs at a particular interval on the lining it can be used to direct the driver to the vacant slot. Only those LEDs will glow which lead to some vacant place, others will be either off or will glow in some other light as per the requirement.

Following figure illustrates the basic working of the solution:

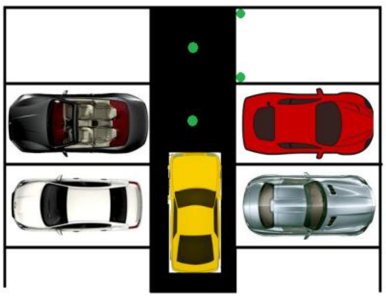

Fig 2: Basic layout of the solution

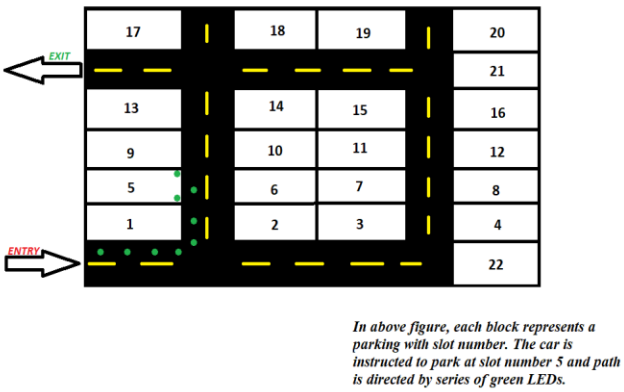

Fig 3a: Shows the path directed by green LEDs

In figure 3a, the car is instructed to park at slot number 5, so LEDs which extend up to spot number 5 will glow and hence, it will direct the driver of the car accordingly.

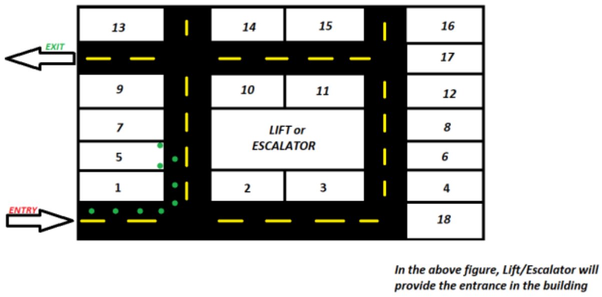

Fig 3b: Parking with entrance to the building

Figure 3b shows the entrance to the building by lift or escalator. Hence, figure 3b covers one of the possibilities of parking arena. Hence, almost all the possibilities are covered.

IV. COLLECTING VACANT SLOTS DATA

At the entry, the guard will ask the driver about the vehicle number. Then the guard will assign a vacant slot number to that particular vehicle. The slot number will be stored in a database server with the help of MySQL. The guard at the exit can access that database. At the exit, the guard will ask for the vehicle number. With the help of vehicle number, the guard will be able to access the parking slot number. In this way, that slot number can be made empty again. Both guards at the entry and exit will be able to access and change the database so that they know about the vacant and filled parking slot number. Entry and exit times can also be stored in the database.

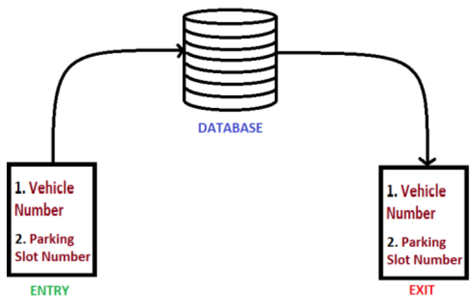

Fig 4: Database Layout.

V. Types Of Sensors

Proximity sensors are used to detect nearby objects without any physical contact. First the guard will assign a vacant parking slot, and as programmed the particular LEDs will glow. Types of sensors that can be used are as follows [1], [2], [3]:

Inductive:

Sensing range is between 4 – 40 mm. Detection of ferrous materials.

Example: Iron, Steel, Aluminum.

Capacitive:

Sensing range is between 3 – 60 mm. Detection of non-ferrous materials. Example: Liquids, Plastic, Wood.

Photoelectric:

Sensing range is 25m by photoelectric proximity sensors. Light is used to detect the objects. Example: Silicon, Paper, Metal.

Ultrasonic:

Sensing range is up to 2.5 m. Sound waves are used to detect objects.

Example: Cellophane, Foam, Glass.

In the case of Smart Parking, we need the average ground clearance of the vehicles. By observing, the average ground clearance of car is 150 mm, so we need the sensors that can sense up to 150 – 200 mm.

VI. Computing the shortest path

For computing the path, parking slots and LEDs along the way can be considered as vertices of a graph. Those slots which are vacant can be considered as active vertices along with LEDs as vertices of a graph. Following figure shows the graph representation of a parking slot used in Fig 3.

Fig 5 shows the graphical representation of parking slot and Fig 6 shows equivalent graph for parking slot as shown in Fig 5.

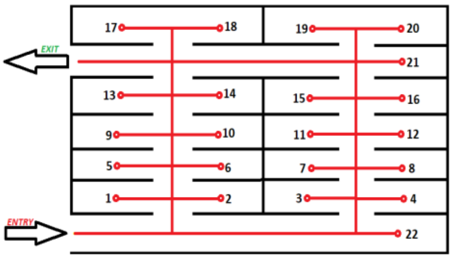

Fig 5: Graphical Representation of parking arena.

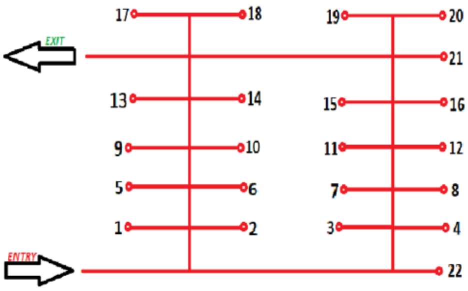

Fig 6: Graphical Equivalent of Fig 5

Here considering the vertices of the graph, by considering the end points various algorithms can be implemented. Some of the algorithms are mentioned below:

<u>Bellman's Ford Algorithm</u>:

This algorithm gives the shortest path from one node to all other nodes similar to Dijkstra's algorithm. This algorithm works on negative edge weights but it doesn't work on negative cycles. In negative cycles, shortest path doesn't exist. Time complexity is $O(E^3)$.

1. Assign a distance to all the nodes. Set the initial node as zero and all the remaining node as infinity.
2. Distance from current node to the neighbour node can be negative.
3. It also consists of visited and unvisited sets. Initially, visited set consist of only initial node and unvisited set consist of nodes other than the initial node.
4. For a particular node, consider all its neighbour nodes. Calculate (distance of the current node) + (distance from the current node to the neighbour

node). If the calculated distance is less than their current tentative distance, then replace it with the new distance.

5. After considering all the neighbours of the current node, mark the current node as visited and remove it from unvisited set.
6. When the destination node is reached, then that node is marked as visited and we have reached our destination node.
7. Otherwise, set the unvisited node marked with the smallest tentative distance as the next current node and repeat the steps from Step 3.

Dijkstra's Algorithm:

This algorithm works on the weighted graph. It starts from the initial node and finds the shortest path to reach the destination node. Time complexity is $O(V^2)$.

1. Assign a distance to all the nodes. Set the initial node as zero and all the remaining node as infinity.
2. Let there be two sets as visited and unvisited sets. Initially, visited set consist of only the initial node and unvisited set consist of nodes other than the initial node.
3. For a particular node, consider all its neighbour nodes. Calculate (distance of the current node) + (distance from the current node to the neighbour node). If the calculated distance is less than their current tentative distance, then replace it with the new distance.
4. After considering all the neighbours of the current node, mark the current node as visited and remove it from the unvisited set.
5. If the destination node is marked as visited, then we have reached our destination and the algorithm has finished.
6. Otherwise, set the unvisited node marked with the smallest tentative distance as the next current node and repeat the steps from Step 3.

Floyd-Warshall's Algorithm:

This algorithm is used to find the shortest path between all pairs of vertices. Negative edges are allowed in this algorithm. Time complexity is $O(V^3)$.

1. Let V be the number of vertices in a weighted graph.
2. Create an array of minimum distance as dist of size V x V.
3. For each vertex v, assign dist[v][v] as zero.
4. For each edge(u, v), assign dist[u][v] as weight(u, v).
5. If dist[i][j] is less than (dist[i][k] + dist[k][j]), then assign dist[i][j] as (dist[i][k] + dist[k][j]).
6. Otherwise, repeat Step 4 for each vertices V.

A* Algorithm :

A^* algorithm is a combination of depth first search and best first search. This algorithm is given by the equation

$$\mathbf{f = g + h}$$

Where g is the cost it takes to reach a particular node and h is the heuristic function which informs the cost it will take to reach the goal from that node.

1. First, we create an open and a closed list.
2. Initialize both open and closed list.
3. We put the node from where we start(starting node) in the open list.
4. Find the node with the least value of **f** on the open list and call it as 'q' and pop q from the open list.
5. Now, generate all the successor of q and set their parents to q.
6. If a successor s is a goal, then stop the search.
7. s.g is (q.g + distance between successor and g)
   s.h is distance of goal from successor
   s.f is (s.g + s.h)
8. When a node with the same position as successor is in the open list or closed list or which has a lower f than successor, then skip this successor.
9. Otherwise, add the node to open list and repeat from Step 6.
10. Now, push q to the closed list and repeat Step 4.

VII. OTHER POSSIBLE SOLUTIONS

Other possible solution can be in the form of maps that can be accessed through cell phones but creating a map of indoor roads and sending it to user's cell phone can be difficult.

Another possibility can be that at the entrance gateway, we can instruct the driver the vacant parking slot assigned. For example, let's say slot assigned is 2-A-3 (2-A-3 = floor 2, row A, space 3rd from the entrance). Here no doubt driver knows the vacant space readily but this might work well in small parking arena, but if parking arena is large, say it can accommodate 200 – 300 cars, then finding the vacant spot (here 2-A-3) will still be difficult.

Hence, there might exist some other possibilities to ease the process but they cannot guarantee the ease of entire system. Hence, solution proposed in the paper tries to minimize all the problems mentioned in the paper.

VIII. CONCLUSION

By looking at the problems and set of possible solutions, we have proposed the solution that can accurately direct the users/drivers to the vacant parking slot and hence, it will save not only time but also, fuel. So, it will increase efficiency of parking at commercial and public places, reducing the hectic job of the drivers.

REFERENCES

[1] Micromachined Ultrasound-Based Proximity Sensors By Mark R. Hornung, Oliver Brand
[2] http://www.machinedesign.com/sensors/proximity-sensors-compared-inductive-capacitive-photoelectric-and-ultrasonic
[3] http://thegadgetsquare.com/what-are-proximity-sensors-types-and-how-it-works/
[4] Smart Mobility – Connecting Everyone: Trends, Concepts and Best Practice