Intelligence Routing Plan using K-d tree and Smart-A* with Google Map Data

Nyein Chan Soe, Thin Lai Lai Thein
University of Computer Studies, Yangon Myanmar
nyeinchansoe@ucsy.edu.mm, tllthein@ucsy.edu.mm

Abstract

Road traffic congestion or jam is main problem in urban area of both developing and developed countries. In order to solve this problem, traffic congestion states of road networks are estimated so that congested road can be avoided and set the best optimal route. In this system, the real time traffic congestion states of users` desired between source and destination are estimated and presented the results in Google Map. GPS data from mobile phones on vehicles are used to get the real time data while using the system. The historical traffic condition data of each road network on each time using the collected data are utilized in this research. In this paper the system is to analyze and avoid traffic congestions and get optimal route with modified A algorithm, also called Smart-A*. This includes the run-time traffic data from GPS enabled mobile and historical data in the database. In the system firstly, we take the GPS data (current location) and searching area and extracting geo-data between source and destination points of user by K-d tree. Second, search the traffic congestion data in that area by Google Traffic layer. Finally, calculate optimal route by Smart-A* and then show the result that best optimal route to the user.*

Keywords- A*; Smart-A*; Traffic Jams; Geo Database; K-d tree

1. Introduction

Yangon city is the main central point, the old capital and economic zone of Myanmar country. There are almost six million populations in the city and 14 percent of entire country. As a city with more 6 million people, the traffic conditions in Yangon are very similar to other major cities such as developing countries around the world: crowded and congested. In Yangon, the most common forms of transport for citizens are buses. However, due to severe traffic congestion, most of the buses travel very slowly during rush hours.

In this paper we use GPS data from mobile phones on vehicles to get the real time traffic status. This system uses proposed K-d tree and modified A* algorithm to reduce the data complexing or delaying and estimate the current traffic congestion states and solving optimal route of user's desired in the crowded city. This paper intends following factors. There are to show traffic congestion condition area of a route, to get accurate map result with short time and the last fact is to present the optimal route with accurate data result to mobile users.

This paper organizes as following. After introducing, section 2 review the related papers. In section 3, describes system overview and research area of the work. Section 4 presents the intelligence route planning approaches with K-d tree, Google Traffic Layer and modified A* algorithm. And section 5 shows the experimental result then finally, section 6 concludes this paper.

2. Reviews of Related Works

Some reports presented concerning the research work of the traffic monitoring and routing systems. In Geographic Information Systems for Spatial Analysis of Traffic Collision Locations in La Crosse, Wisconsin [1], the system and patterns defined in the study through the use of Geographic Information System using maximize efficiency of efforts to improve traffic and routing safety.

ESRI, California 92373-8100 USA [3] reported "GIS Solutions for Highway and Roadway Management". In this document it views the various methods for traffic problem addresses such as highway access management, maintenance, work order management and transport planning. The following data have some supports in this document: first is construction that integrate projects and financial manage software with GIS to better manage infra projects. And operation is that incorporated GIS

into business processes to improve operational performance.

In [4] "Minimization the traffic congestion using GIS, studying the effect of Transportation System Management (TSM)" measured the needs to have a clear view of flow patterns, locations, and existing road network. Geographic Information System can be effectively used analyses that's problems concerned with transportation.

According the related paper [9], Dr. Mohammed Otair showed the 3 major contribution is achieved by this research is the use of the k-d tree data structure for spatial clustering and comparing its performance to the brute-force approach. In "Approximate K-Nearest Neighbor Based Spatial Clustering Using K-D Tre", the results of the work are not performed revealed better performance using the k-d tree, compared to the traditional brute-force approach. In this paper, data set within the same cluster are shared common features that give each cluster its characteristics. And the implementation of Approximate kNN-based spatial clustering algorithm using the K-d tree is little complexed.

3. System Overview and Process flow of the Work

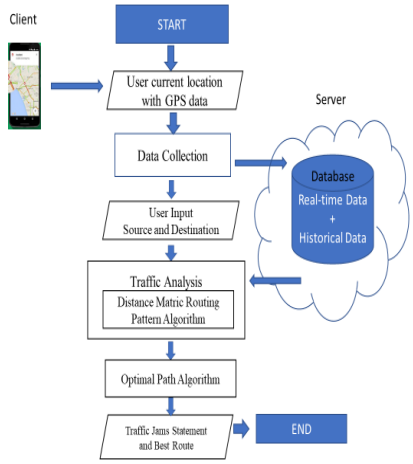


Figure 1. Flow diagram of the proposed system

This overview is the client-server system as figure 1. The client-side application captures the user location with GPS receiver. It is designed for android platform. The server operates these captured data and analysis the traffic routes in cloud database. There are operations steps are performed in this approach:

Step (1): Collecting the GPS data for the transportation network by client GPS receiver captures user current location with latitude and longitude.

Step (2): Storing the location GPS data and historical traffic data in the spatial database.

Step (3): Extracting latitude and longitude, speed, distance, time and direction with K-d Tree method.

Step (4): Estimating the most traffic jams places through Google Traffic Layer.

Step (5): Operating the traffic analysis and optimal path by using Smart-A* algorithm. After that, end user gets the result of traffic routes information via mobile as client.

3.1. Research Area and Data Collection

There are three townships where research areas in this approach: Insein township, Mingalardon township and Shwepyithar township. See as figure 2.

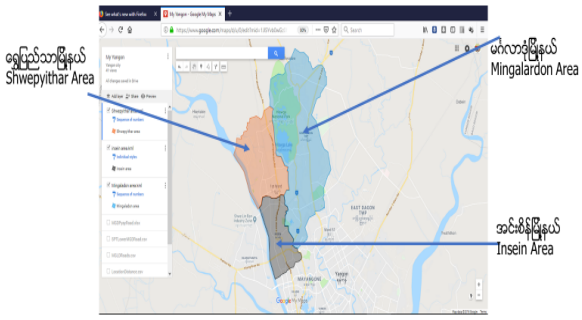


Figure 2. Three Research Areas of the Yangon

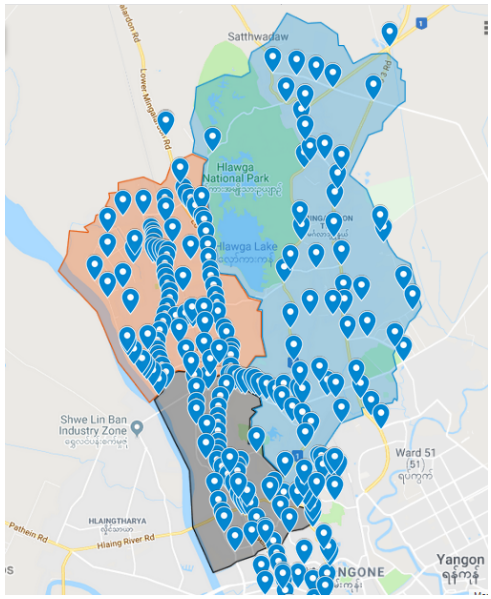


Figure 3. Bus Stops, Traffic Points and Intersections Datasets in the Research Areas

By the above figure 3, we collected and marked the location data on the customized map. These points are bus stops, traffic points, intersections points and stations. In each point includes their data attributes such as ID, Latitude, Longitude, Point name and Remark. These points are different on their attributes as following figure 4.

SHWEPYITHAR, INSEIN, MINGALADON AREA ROADS AND POINTS

ID	Latitude	Longitude	Point Name	Remark
1	17.03181	96.07647	SPT Bond	
2	17.0131	96.0822	UCSY Near	
3	17.0056	96.08352	TarSone Bus	
4	17.00371	96.08483	KwatThit Bus	
5	17.00168	96.08628	PaDoneMar Bus	
6	16.99871	96.08842	UCSY Bus	
7	17.0023	96.0908	UCSY	
8	16.99476	96.0912	NwarChan Bus	
9	16.99353	96.09184	Near	
10	16.98876	96.09277	ThabarWa Bus	
11	16.98382	96.09334	TharDuKan Bus	
12	16.97906	96.09366	PoneNya Bus	
13	16.97414	96.09435	OkePho Bus	
14	16.97176	96.09477	ThanPhyu Bus	
15	16.9682069	96.0955544	Ye6 Bus	
16	16.96432	96.09633	L_GateHaung bus	
17	16.96119	96.0969	ShwePyiTharKwe Bus	Traffic Light
18	16.95808	96.09751	SitTad Bus	

LocationName

Figure 4. Examples data of bus stops and attributes

Another collected data file is the facts of these points and their distances as the figure 5. It includes Source ID, Destination ID and Distance. We defined the distance unit between each points by km(Kilometer).

ROADS AND POINTS (Distance)

SOURCE					DESTINATION				DISTANCE
No	ID	Lat	Long	Point Name	ID	Latitude	Longitude	Point Name	Distance(Kilometer)
1	3	17.0056	96.08352	TarSone Bus	6	16.99871	96.08842	UCSY Bus	0.95
2	6	16.99871	96.08842	UCSY Bus	8	16.99476	96.0912	NwarChan Bus	0.513
3	8	16.99476	96.0912	NwarChan Bus	11	16.98382	96.09334	TharDuKan Bus	1.24
4	11	16.98382	96.09334	TharDuKan Bus	13	16.97414	96.09435	OkePho Bus	1.09
5	13	16.97414	96.09435	OkePho Bus	16	16.96432	96.09633	L_GateHaung bus	1.13
6	16	16.96432	96.09633	L_GateHaung bus	17	16.96119	96.0969	ShwePyiTharKwe Bus	0.386
7	17	16.96119	96.0969	ShwePyiTharKwe Bus	18	16.95808	96.09751	SitTad Bus	0.402
8	18	16.95808	96.09751	SitTad Bus	20	16.95011	96.10072	KanTharYar Bus	0.899
9	20	16.95011	96.10072	KanTharYar Bus	21	16.94435	96.10274	MyinHleGate Bus	0.676
10	21	16.94435	96.10274	MyinHleGate Bus	25	16.934	96.10199	DaNyinGone Point	1.16
11	25	16.934	96.10199	DaNyinGone Point	43	16.92542	96.09856	YwarThit Bus	1.06
12	43	16.92542	96.09856	YwarThit Bus	44	16.92142	96.09763	AungSanZay Bus	0.457
13	44	16.92142	96.09763	AungSanZay Bus	46	16.91533	96.09725	No1Gate Bus	0.687
14	46	16.91533	96.09725	No1Gate Bus	48	16.90767	96.09764	FawtKanZay Bus	0.854
15	48	16.90767	96.09764	FawtKanZay Bus	49	16.90504	96.09772	ThirMingalar Bus	0.283
16	49	16.90504	96.09772	ThirMingalar Bus	51	16.90066	96.09716	PyiTawThar Point	0.503
17	51	16.90066	96.09716	PyiTawThar Point	52	16.8882	96.09604	L YwarMa Bus	0.343

LocationDistance

Figure 5. Examples data of bus stops and Distances

4. Approaches of the Intelligence Route Planning

4.1. Data Storing and Extracting by K-d Tree

In the research paper we choice K-d tree structure for data storing and extracting because of this tree is most suitable for spatial database. A K-d tree (short for k-dimensional tree) is a space-partitioning data structure for organizing points in a k-dimensional space. K-d tree is also a special case of binary space partitioning trees. The idea of K-d is divided point between x-coordinate and y-coordinate: (1) by x-coordinate: split half the points left or on, and half right by a vertical line, (2) y-coordinate: split with a horizontal line that has half the points below or on, and half above. For data balancing the system requires K-d tree because It is sorted in any multiple dimensions.

The k-d tree is a modification to the BST that allows for efficient processing of multidimensional keys. We define the discriminator of K-d tree that unknow level is i and it to be i mod k for k dimensions. In sample calculation, storing data organized by xy coordinates, k is 2 (coordinates), with the x-coordinate field predicted designated key 0, and key 1 for y. At each level, the discriminator alternates in each x and y. Figure 6 shows an example of collection of k-d tree decomposition containing six data points. The objective of nearest neighbor search (NN) algorithm is to find the point in the tree that is nearest with user input point. This search can be done the process accurately by using the tree properties to quickly eliminate large some parts of the search area.

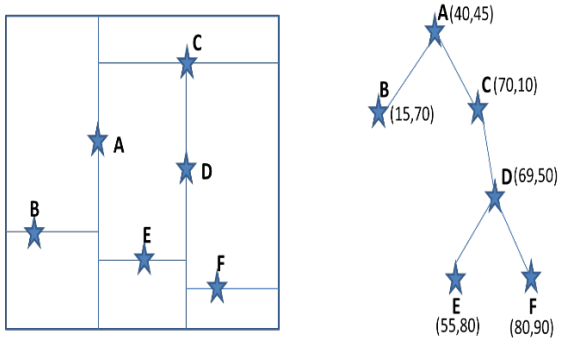


Figure 6. The k-d tree decomposition for a 100 x 100-unit region containing six data points.

The following algorithm is the procedure for constructing the kd-tree.

Algorithm BuildingKdTree(A,weight)

1. If A includes only one point
2. then storing this point and return a leaf
3. else if weight value is equal to even

4. then Split A with the median x-coordinate on vertical line l //discriminate A1 (left of or on l) and A2 (right l)
5. else Split A with the median y-coordinate on horizontal line l //discriminate A1 (below or on l) and A2 (above l)
6. Vleft ← BuildingKdTree(A1, weight+1), A1++
7. Vright ← BuildingKdTree(A2, weight+1),A2++
8. Create a node V storing l, make Vleft with the left child of V, and make Vright with the right child of V.
9. return V

4.2 Estimating the Most Traffic Jams Places through Google Traffic Layer

After getting the facts that traffic routing data in short time by extracting K-d tree structure, next step is to get traffic jams information. According to the step (4), the Maps JavaScript API allows us to add real-time traffic information (where supported) to our maps using the Traffic Layer object. Traffic information is refreshed frequently, but not instantly. Google platform supports us traffic data by applying the following function codes (Figure 7).

```
<script>
  function initMap() {
    var map = new google.maps.Map(document.getElementById('map'), {
      zoom: 13,
      center: {lat: 34.04924594193164, lng: -118.24104309082031}
    });

    var trafficLayer = new google.maps.TrafficLayer();
    trafficLayer.setMap(map);
  }
</script>
<script async defer
src="https://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY&callback=initM
</script>
```

Figure 7. Sample code to get the traffic data by Traffic Layer object

And then we take the traffic jams conditions as shown in figure 8 through the Google Traffic Layer Object such that congestions, Bus-stop, road circumstances and other. We save these traffic layer (jams areas) in the database and set the output while finding the optimal route. Now we do analysis for the optimal route by using Smart-A* algorithm.

Figure 8. Traffic Status of Desired Place

4.3 Finding the Optimal Path by Smart-A*

This is ready to define the Smart-A* algorithm in this paper. In experiments, if we have a consistent heuristic, then A* can be much faster than Dijkstra's algorithm. Example: Consider cities (points on the plane), with roads (edges) connecting them. Then the straight-line distance is a consistent heuristic. First, we define the cost function. The cost of a node, f, is given by the following partial differential equation:

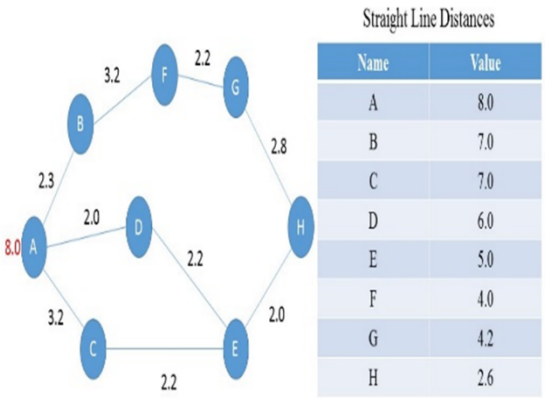
$$f = g + h \quad (1)$$

where, g is distance from source to current node and h is estimate distance of current node to destination. The approach of the best first search considers the shortest path by the above equation with each node.

Example: Find the best optimal route in consider township, with roads such as edges connecting them.

First, we define the source= A and the destination= H and the cost of function.

By the cost of node, f, is given with the sectional differential equation: f = g + h



Straight Line Distances

Name	Value
A	8.0
B	7.0
C	7.0
D	6.0
E	5.0
F	4.0
G	4.2
H	2.6

Figure 9. Finding optimal path by Smart-A*

According the equation f(B)=9.3, f(C)=10.2, f(D)=8.0 and the minimum is D.
With Calculating, three possible routes is:

A,B,F,G,H=28.3

A,C,E,H=22

A,D,E,H=19.8

First, we have A value is 8.0 and continuously search the values of each B, C and D. According the equation f(B)=9.3, f(C)=10.2, f(D)=8.0 and the minimum is D. And calculate the D to destination through E. Finally, we can track the shortest path for the routing is:

A→D→ E→ H

So, this is simple sample calculation for the optimal routing planning. This means the optimal route among possible routes is Current location (A) to UCSY (H) through Point (D) and Point (E). In the system we use Smart-A* algorithm that modified A* algorithm. To complete the searching optimal path with reducing time and memory, the algorithm repeats the steps as the following.

1. Start the program and open Lists.
2. Set the traffic layer with user location (google.map).
3. Insert start point node into empty set of First List.
4. Record minimum cost from First list and carry it to Last List.
5. Check all expanding nodes of current for adjacency.
6. If no adjacency node in First List, calculate cost of nodes and record as Equation (1).
7. Insert the cost into First List.
8. If adjacency node is already in First List, record cost of current node than previous cost. If not, change previous node and recalculate the cost.
9. Add costs of start and end points nodes into Last List.
10. End the searching and close Lists.

5. Experiments and Evaluation Result

The user can apply the system ease and simply by client. The GUI interface view of this system is simple and user will detect the results quickly. Figure 10 is first page of the system.

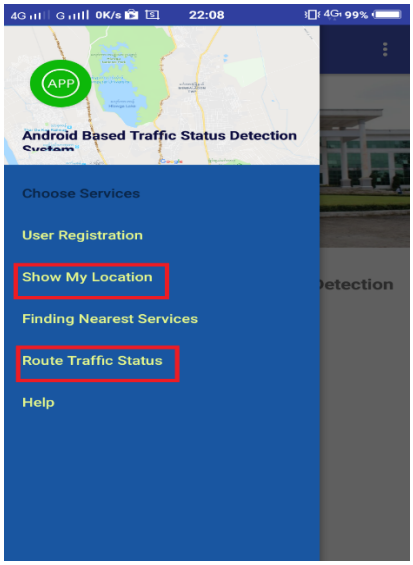


Figure 10. User Interface View in Client

In the experiments, it collects the current location of user and then it detects the most traffic congestion areas and find the optimal path. In the paper, the most traffic areas where Sawbwargyigone intersection, Khayaypin Point, Danyingone Point and University of Computer Studies in Insein, Mingalardon and Shwepyithar townships, Yangon. We show the route information from GPS data by calculating the optimal route after extracting and estimating processing. The following figure is sample user interface of the application showing the status that user takes current location with coordinates as shown in figure 11.

As above finding the optimal path by Smart-A*, the user takes the best optimal route (go around with orange arrow) but avoid other lanes. It also abstains the most traffic jams places through google map traffic data. And the following figure 12 is showing the result that user should march and go to his goal by Optimal path Smart-A* in experiment.

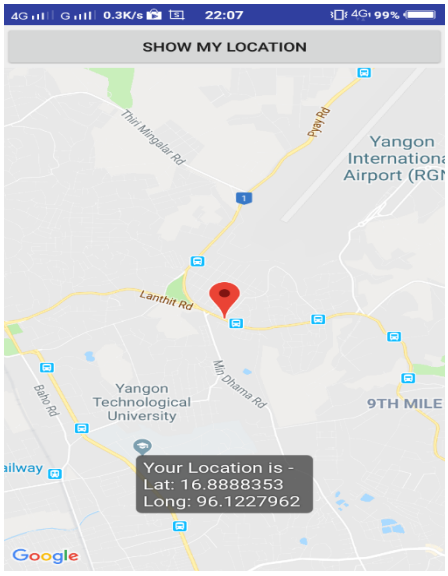


Figure 11. Collecting user current location on map

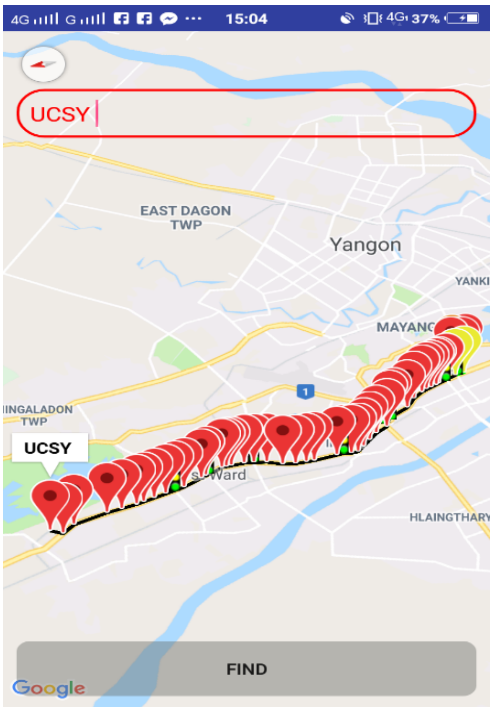


Figure 12. Experiment result by Smart-A*

To get the accuracy result of proposed algorithm, the processing time of these algorithm is depending on the expanded nodes. By comparing the shortest path algorithms, in Dijkstra algorithm, the time complexity is O((N+K) * log (N) = O (K* log (N)), where O(N) is runtime of loop, N is the number of node and K is the number of links between each of nodes. Cause of the time complexity log N is extracting minimum of the heap and correctness of algorithm is as number of nodes increases and then running time of this will become longer.

The modified algorithm reduces more processing time because of the shortest path search starts from source and expands node that towards destination and our algorithm uses the same approach as Dijkstra but not allowed accumulated cost of edge plus the time complexity. Therefore, we prove the proposed Smart-A* algorithm reduce the finding process than another shortest path algorithm with the result graph chart (Figure 13).

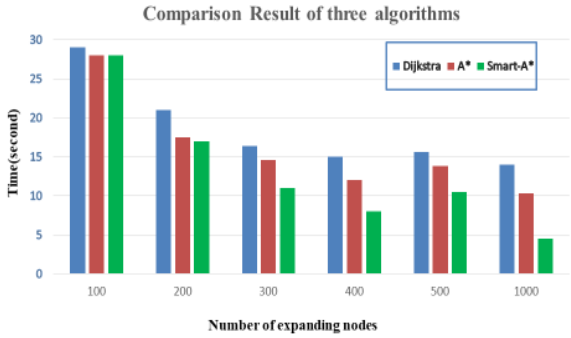


Figure 13. Evaluation result of the system

6. Conclusion

The system estimates the traffic congestion areas and choices the best route in current time. A* algorithm is a graph search algorithm that find an optimal path from a given initial node to a given goal node in a mapped area. Smart-A* is better when we know both starting point and destination point to check the optimal route. Moreover, it also provides the users the accurate map with more efficient estimation results and saving more time other cause design of K-d tree database structure.

And then this system not only shows the results of the traffic congestion states of source and destination of region of all interest places in Yangon but present also the optimal route with short time for user. This system shows intelligence method for Traffic Monitoring System of Yangon City in Myanmar. So, people can know not only the traffic jam information of desired point but also avoid the traffic point on their target in real-time condition.

References

[1] Cory A. Brose Saint Mary's University of Minnesota, Graduate Studies in Resource Analysis, 700 “Geographic Information Systems for Spatial Analysis of Traffic CollisionLocations in La Crosse, Wisconsin”,

Terrace Heights #10; Winona, Minnesota, 55987, USA.

[2] Celso Goncalo Dias Junior, GIS/RS Rresearcher, State Office of Public Safety, DETRAN- State Vehicles Trqaffic Department, "Vehicles Traffic Management and GIS Planning System- A Study if DETRAN Project-Parana State /Brazil", Av. Visitor Ferreia do Amaral, 2940-ZIP 82800 900, Curitiba, PR, Brazil, International Achieves of Photogrammetry and Remote Sensing. Vol. xxxi, Part B4, Vienna 1996.

[3] ESRI, 380 New York Street Redlands, California 92373-8100 usa, "GIS Solutions for Highway and Roadway Management", Copyright © 2011 Esri. All rights reserved. Esri.

[4] Anitha Selva Sofia S.D.1, Nithyaa.R2, Prince Arulraj.G3 Assistant Professor, Department of Civil Engineering, SNS College of Technology, Coimbatore, Tamilnadu, India, "Minimizing the Traffic Congestion Using GIS", IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 1, Issue 1, March, 2013, ISSN: 2320 – 8791 www.ijreat.org

[5] Shekhar K. Rahane, Prof. U. R. Saharkar , Pg student in Civil Engineering (Construction and Management), Dr. D. Y. Patil Institute of Engineering and Technology Ambi, University of Pune, Maharashtra, India, "Technique Identification For Road Traffic Congestion Solution In Talegaon Dabhade State Highway-55", Journal Of Information, Knowledge And Research In Civil Engineering ISSN: 0975 – 6744| NOV 13 TO OCT 14 | Volume 3, Issue1.

[6] Yangon Region Government, Yangon City Development Committee; YCDC, Japan International Cooperation Agency; JICA, JICA Study Team, "Yangon 2040 The Peaceful and Beloved Yangon ―A City of Green and Gold―", The Strategic Urban Development Plan of the Greater Yangon March, 2013..

[7] Ke Zhang, Guangtao Xue, Shanghai Jiao Tong University, Shanghai, China, "A Real-Time Urban Traffic Detection Algorithm Based on Spatio-Temporal OD Matrix in Vehicular Sensor Network*", doi:10.4236/wsn.2010.29080 Published Online September 2010.

[8] Jan Jannink, Stanford University, Computer Science Dept. Stanford, "Implementing Deletion in B+ Trees," CA 94305.

[9] Dr. Mohammed Otair, Department of Computer Information Systems, Amman Arab University, Amman, Jordan, "Approximate K-Nearest Neighbour Based Spatial Clustering Using K-D Tree", International Journal of Database Management Systems (IJDMS) Vol.5, No.1, February 2013.

[10] Ben pfaff, "an introduction to binary search tree and balanced tree" label binary search tree library, volume 1: source code, version 2.0.3, free software foundation, inc, (2004).