

Crime Prevention through Alternate Route Finding In Traffic Surveillance Using CCTV Cameras

Gupta, P., Purohit, G. N., Dadhich, A.

Abstract— Road Traffic Network Surveillance through CCTV Cameras finds its applications in various issues of social and administrative importance including Crime Prevention. Crimes associated with theft, kidnapping and missing cases demand tracking of paths of a suspect vehicle, which that vehicle might have followed. Since it is difficult to place CCTV Cameras at each and every traffic intersection point, information collected from CCTV Cameras placed at some important junctions can be exploited to generate the required paths. Algorithm for finding the routes of the suspect vehicle in given road traffic network with certain active intersection points, which are under CCTV surveillance, has been suggested. Simulation of the algorithm has been conducted in SUMO and results in the form of alternate paths between two given intersection points, where the target vehicle has been tracked through CCTV Cameras are presented.

Index Terms— Alternate Routes, CCTV Camera, Crime Prevention, Python, Simulation, Urban Mobility, Traffic Surveillance.

I. INTRODUCTION

This document simulates traffic controlled by CCTV Cameras and analyzes its application in crime prevention. Data collected by CCTV cameras installed at various traffic junctions for Real Time Traffic Signal Control can be used for various other applications. Crime Prevention is one such application. Suspect vehicles which have been caught at various CCTV Camera enabled junctions, can be traced for various in-doubt paths, which that vehicle might have followed. The information can be used to access the final destination of that suspect vehicle.

This paper is divided into six sections. The second section presents the related work. The third section discusses the technologies used in the experimental study. The fourth section describes the procedure of acquiring the required information. The fifth section presents the simulation experiment and results. The sixth section draws the conclusion.

II. RELATED WORK

Traffic Simulation is a study area that is being explored a lot. Traffic simulation is of much importance. But the work of CCTV cameras in Traffic Simulation is not much explored. This section shows the related work with respect to traffic simulation, computing traffic load, crime prevention etc. The current paper studies the use of CCTV cameras in Crime Prevention and finding the alternative paths between points in a network.

Road Traffic Networks is one of the important research areas. [1] presents the study of UK road traffic data. The paper studies the nature of congestion on road network and look at methods to improve our ability to predict journey times. [2] argues that calculating the best driving path (shorter in distance or faster in time) between two addresses is subject to many factors. Some of these factors are distances, road situation, road traffic and others. [3] introduces an alternative Graph that can compactly encode many alternative routes between two points in a road network. [4] provides a view on urban road networks, freeway networks, and route guidance, a comprehensive overview of proposed and implemented control strategies.

Jain V. et al. [5] argued that there are several hotspots in a city that are prone to congestion. They devised a way to solve this congestion problem using CCTV captured images and analyzing them by image processing. Dailey et al. [6] have also considered the use of CCTV cameras to estimate mean traffic speed.

Celil Ozkurt et al. in [7] have mentioned three main approaches to detect moving objects in a video viz. temporal difference, optical flow and background subtraction. They stress upon knowing the traffic density for traffic surveillance systems and use a neural network approach for the same. R. Blake et al. in [8] developed a CCTV camera based application and utilized image processing and pattern recognition methods and functional capabilities of a system to monitor the road, initiate automated vehicle tracking, measure speed and recognize number plates. Pallavi Choudekar et al. [9] presented approach for detecting vehicles through images instead of using electronic sensors embedded in the pavement. The authors worked with a camera installed alongside the traffic light, capture image sequences and then image sequences analyzed using digital image processing for vehicle detection, and according to traffic conditions on the road traffic light can be controlled.

Manuscript published on 30 June 2013.

* Correspondence Author (s)

Pratishtha Gupta*, Computer Science, Banasthali Vidyapeeth, Jaipur, India.

Prof. G. N. Purohit, His Department Name, Banasthali Vidyapeeth, Niwai/Tonk/Rajasthan, India.

Amrita Dadhich, Computer Science, Banasthali University, Jaipur, India.

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](http://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC-BY-NC-ND license <http://creativecommons.org/licenses/by-nc-nd/4.0/>

[10] presents a solution technique for large scale road network equilibrium assignment and related flow problems with nonlinear costs. It is shown that this nonlinear network problem can be solved without explicitly considering any of the constraints—they are satisfied automatically in the procedure developed—and without storing all of the individual decision variables. In [11] an approach is proposed for real-time coordination of signal phase timings for a network. This study discusses how some network conflicts are resolved and the corresponding algorithmic procedure of REALBAND.

III. TECHNOLOGY USED

For simulation, Simulation of Urban Mobility (SUMO) and Python Programming language have been used.

A. Simulation of Urban Mobility (SUMO)

It is an open source traffic simulation package including the simulation package itself as well as supporting tools, mainly for network import and demand modeling. SUMO helps to investigate a large variety of research topics, mainly in the context of traffic management and vehicular communications. SUMO is used to make network, and this is done with help of XML.

XML files are made for traffic simulation in SUMO. SUMO defines the format for making XML files for the purpose of defining the network. The network is depicted as a combination of edges and nodes. Nodes represent the junctions in a traffic network and the edges represent the connections between them. Files are in XML format. Network as needed can be made in SUMO. Also, vehicles can be simulated on the network and other facilities like including traffic lights are included in the network.

SUMO also provides various modules like NETCONVERT, NETGENERATE, DUAROUTER, JTRROUTER, DFROUTER, POLYCONVERT etc. These modules are available and provide functionalities useful in traffic simulation.

B. Python

SUMO does not provide the facility of implementing any algorithm and so python programming language is used in the process of analysis of crime prevention and calculating the alternative paths between the junctions (nodes) of a network. Python is a highly structured language and very user-friendly also.

Python version 3.3 is used. Python supports the XML file format compatibility and thus provides a way through which files from SUMO can be imported in Python, processed and the results are presented in SUMO again for viewing the results in simulation.

Python has modules like minidom, lxml, elementtree that support XML files in Python. These are imported in Python for working with XML files. Because of this support of XML in Python, this programming language is used.

IV. SIMULATION EXPERIMENT AND RESULTS

Using Simulation of Urban Mobility, a network is simulated and an algorithm to find out the alternate paths between two nodes of the network is implemented in python.

A. Making network using SUMO

To work on the simulation software, XML files that contain the information about the nodes, edges, routes etc are created. The main component of Simulation software is NETCONVERTOR that has been used. The steps to make any network are as described in the following points:

1) Making the Node file

When the network is represented with the help of a graph like entity, the junctions of the network are represented by the nodes of the graph. To make a network in the simulator, the node file constructed is specified with the help of the following parameters and an XML file is created using the data.

- **Id**: the ID name of the node, defined by users with numbers, word strings or both.
- **X**: the x-coordinate location of the defined node (in meters)
- **Y**: the y-coordinate location of the defined node (in meters)
- **Type**: the signal control type of the defined node. It is an optional attribute and defined with *priority* and *traffic_light* for unsignalized and signalized intersections respectively.

This XML file is saved with the extension ‘.nod.xml’ for further use.

2) Making the Edge file

Edge file shows the links by which the junctions in a network are connected. In the simulator, the edge files have been constructed using the following parameters:

- **Id**: link ID, defined by users with numbers, word strings or both.
- **From**: ID of the upstream node of the respective link.
- **To**: ID of the downstream node of the respective link.
- **Type**: ID of the link type, defined in the link type file.
- **Allow/disallow**: ID of the vehicle group allowed to enter a particular lane

The XML file is saved using the extension ‘.edg.xml’.

3) File specifying the type specification of the network

The information about each link type can be defined in an additional file with extension *.typ.xml*. The following shows the attributes that are used to define the type specification of the network.

- **Id**: defined by users with numbers, word strings or both.
- **Priority**: driving priority based on traffic regulations and is defined with numbers. The higher the number, the higher the priority for the respective road. The priority information will override information from the node file, if both of them exist.

- **NumLanes:** number of lanes on the respective road.
 - **Speed:** maximum allowed link speed.
- The type definition file is stored as .typ.xml.

4) File to depict the connections in the network

The default in SUMO based on the given geometric design of all possible and logical traffic movements. U-turns, are allowed, if there is no corresponding specification. To specify traffic movements and lane connections an additional file with extension .con.xml is made. The following parameters are used to define the traffic movements in the network.

- **From:** ID of the link which the traffic movements will be specified.
- **To:** ID of the link which is the downstream link of the above defined link.
- **FromLane/toLane:** lane number of the defined link in 'From' and the lane number of the link in 'To', which are connected.

5) Network generation

Network file in SUMO is named with extension .net.xml. With the above defined files, .nod.xml, .edg.xml, .con.xml and .typ.xml, the network file quickstart.net.xml will be generated by apply the module NETCONVERT. For efficient execution, a configuration file that includes the names of the input files, the name of the output network file and other required actions are created.

The file is stored with the extension .netc.cfg. The network x.net.xml is created by executing the following command in the command line interpreter. netconvert -c x.netc.cfg.

6) Adding the traffic demand file

Traffic demand and route data is defined together with vehicle type data in a file with the extension name .rou.xml. Following listing shows the related attributes included in the traffic demand file:

- **id:** ID of the vehicle type, defined with numbers, word strings or both;
- **accel:** maximum acceleration of the respective vehicle type (in m/s²);
- **decal:** maximum deceleration of the respective vehicle type (in m/s²);
- **sigma:** drivers' imperfection in driving (between 0 and 1);
- **length:** vehicle length (in meters);
- **maxSpeed:** maximum vehicular velocity (in m/s);
- **Color:** color of the vehicle type. It is defined with 3 numbers (between 0 and 1) for red, green and blue respectively. Values are separated by comma and in quotes with no space between the values. For example, 1,0,0 represents the red color, 0,1,0 represents green color and 0,0,1 represents blue color.

The sequence of the attributes can be changed. The attribute *sigma* is assigned as 0.5 for all vehicle types.

7) Configuration file for the traffic simulation of the example network

An efficient traffic simulation execution can be achieved with the use of command line, especially when dealing with

large and sophisticated traffic networks. The configuration file is generated and is saved with the extension .sumo.cfg.

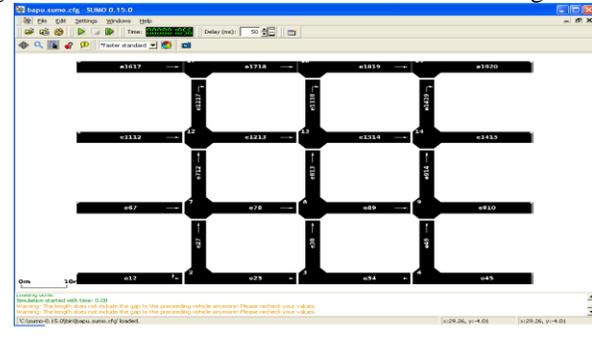


Fig. 1

After all the files are constructed and the network is created, the simulation is carried out as the following CLI command.

SUMO-GUI -c x.sumo.cfg.

A network with 20 nodes has been constructed with unidirectional roads as shown in Fig. 1.

B. Working with python

The Elementtree library is used to import the XML files of simulator. This is for the purpose of applying the algorithm to find alternate paths between the nodes of the network. The following flowchart shows the algorithm that is used to find the alternate paths between any two nodes of the network. The pseudo code shows that the source node and the destination node are taken as input from the user, and all the paths between them are determined.

Input Source node and Destination node

Current node= Source node

Push onto 2-D stack

Repeat while stack is empty:

Pop the top

Compare with the destination node

If (top > destination)

Break

If (top < destination)

Append into output path array

Update current node (= popped node)

If (top= destination)

Copy path array into another variable and print as output.

Pop the elements of the path array till top of the array= source of that element

C. Constructing alternate routes back on the simulator

The nodes that are obtained as output from the algorithm implemented in Python, are used back in the simulator to have a visual appearance of the alternate paths.

Using the algorithm as explained in the above section, alternate paths between different nodes are determined. This is done by the nodes that come as output from the algorithm implemented in python. The output is in form of nodes that specify the routes between the source node and the destination node.



The routes using this output are constructed again in the simulator, to have a visual network with alternate routes between the nodes.

The algorithm calculates the nodes that will be encountered between the different paths. These nodes and their corresponding edges are used to construct XML file back from the output using python language itself. To specify, the root of the tree, the elements of the same to construct the XML file back from python results are specified. The different fields and their values are set according to the results that are obtained. The node files and the edge files are constructed using python for use by the simulator. Further other files are modified if needed and the output is the path between the desired nodes.

- Calculating alternate routes between the nodes 1 and 10

The routes obtained between the nodes 1 and 10 are:

1-2-3-4-9-10

1-2-3-8-9-10

1-2-7-8-9-10

The network snapshot using these output paths is depicted in the following figure. The figure 2 shows the snapshot taken from the simulator and saved as a bitmap image.

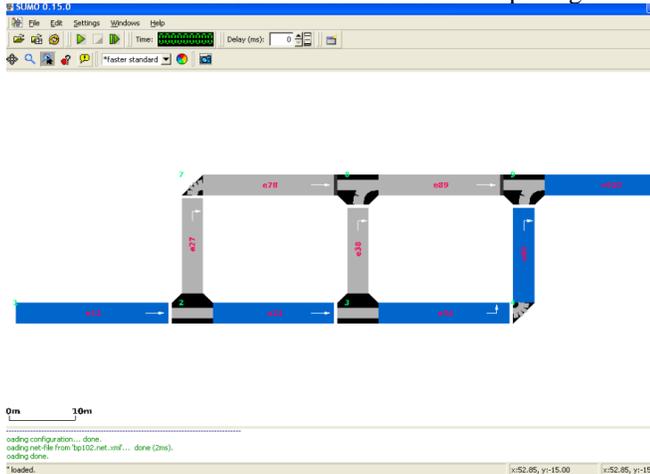


Fig. 2

- Calculating alternate routes between nodes 1 and 13

The paths obtained between nodes 1 and 13 are as follows:

1-2-7-12-13

1-2-3-8-13

1-2-7-8-13

The network snapshot taken using these paths is as shown in the figure 3.

- Calculating alternate routes between nodes 1 and 19

The different routes obtained between nodes 1 and 19 are as follows that are the output of the python processing of XML files.

1-2-7-12-17-18-19

1-2-3-8-13-18-19

1-2-3-4-9-14-19

1-2-7-8-13-18-19

1-2-7-8-9-14-19

1-2-3-8-13-14-19

1-2-7-12-13-14-19

1-2-7-12-13-18-19

The simulator snapshot obtained from these paths is as shown in fig. 4.

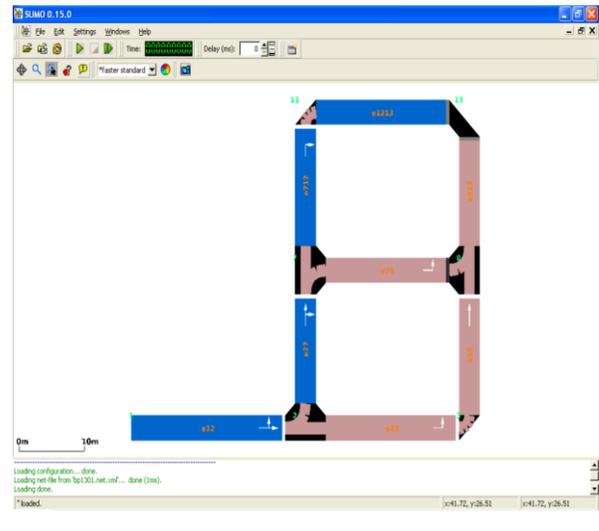


Fig. 3

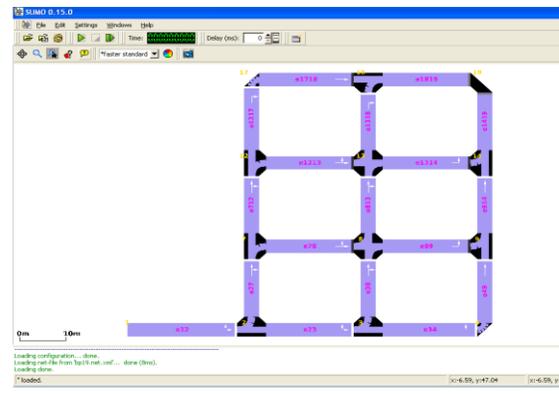


Fig. 4

These figures show the alternate paths between the nodes as calculated by the algorithm and the simulator.

V. CONCLUSION

Increasing traffic congestion coupled with improved technology, cost constraints, and increasing environmental consciousness has provided an impetus to develop cost effective systems, aimed for improving the efficiency and effectiveness of the traffic system.

The simulator, Simulation of Urban Mobility is used to simulate a network. Using the simulated network, the alternate paths between some nodes are computed. This is done by applying the algorithm proposed. The algorithm is implemented in Python programming language and as a result, the alternate paths are calculated.

The vehicle caught at some site need to be traced on those paths only and the complete network need not to be searched, which is practically very difficult and inefficient approach. As the complete network need not be traced for finding the vehicle, this results in an efficient crime detection system.



It is assumed that CCTV cameras are placed at some site in road traffic network. Any suspected vehicle is detected in the network and is recorded by these placed CCTV cameras. Using the nodes or the junctions where the camera is installed, the computation for finding the alternate paths between those junctions is done and so the vehicle will be detected through these paths only, the complete network need not be traced completely and so the suspected vehicle can be detected in lesser time.

REFERENCES

- [1] Gibbens R.J, Saatci Y. *Data Modelling and Inference in Road Traffic Networks*. <http://mlg.eng.cam.ac.uk/yunus/ra-paper.pdf>
- [2] Halaoui H.F. *Intelligent Traffic System: Road Traffic Networks with Time-Weighted Graphs*. International Journal on Infonomics. Volume 3, Issue 4, December 2010.
- [3] Bader R., Dees J., Geisberger R., Sanders P. *Alternative Route Graphs in Road Networks*.
- [4] [Papageorgiou, M.](#) *Review of Road Traffic Control Strategies*. Proceedings of IEEE. Volume 91, Issue 12. Page Number 2043-2067.
- [5] Jain Vipin, Sharma Ashlesh, Lakshminarayanan (2012). *Road traffic congestion in developing the world*. ACM 978-1-4503-1262-2/12/03.
- [6] Dailey Daniel J., Cathey Frederick W., Pumrin Suree(2001). *The use of uncalibrated roadside CCTV camers to estimate mean traffic speed*.
- [7] Ozkurt Celil, Camci Fatih, (2009) Automatic Traffic density Estimation and Vehicle Classification for Traffic Surveillance Systems using Neural Network. Mathematics and Computational Applications, Vol 14, No3, pp 187-196,2009.
- [8] E. Atko'ci _unas, R. Blake, A. Juozapavi'cius, M. Kazimianec (2005) *Image Processing in Road Traffic Analysis*. www.lana.lt/journal/19/Atkociunas.pdf.
- [9] Choudekar Pallavi, Banarjee Sayanti, Muju M.K. *Real Time Traffic Control Using Image Processing*. Indian Journal of Computer Science and Engineering. http://www.researchgate.net/publication/50417981_REAL_TIME_TRAFFIC_LIGHT_CONTROL_USING_IMAGE_PROCESSING.
- [10] LeBlanc Larry J., Edward Morlok K., Pierskalla William P.; *An efficient approach to solving the road network equilibrium traffic assignment problem*. Volume 9, Issue 5 October 1975, Pages 309-318.
- [11] Dell'olmo P., Mirchandani P. B.; *REALBAND: an approach for real time coordination of traffic flows on networks*. Transportation Research Board TRB. Annual meeting (1995), pp 106-116. ISSN- 0361-1981.



Prof. G. N. Purohit is a Professor in Department of Mathematics & Statistics at Banasthali University (Rajasthan). Before joining Banasthali University, he was Professor and Head of the Department of Mathematics, University of Rajasthan, Jaipur. He had been Chief-editor of a research journal and regular reviewer of many journals. His present interest is in O.R., Discrete Mathematics and

Communication networks. He has published around 40 research papers in various journals.



Pratishtha Gupta is an active researcher in the field of image processing, currently working as Assistant Professor in Department of Computer Science at Banasthali University(Rajasthan), India. She has done M.Sc (CS) and B-Level and undergoing PhD from Banasthali University (Rajasthan), India.



Amrita Dadhich is an active researcher in the field of Geographical Information System and has done M. Tech (CS) from Banasthali University.