OPTIMUM MAP MATCHING AND ROUTING SELECTION FOR REAL TIME VEHICLE NAVIGATION SYSTEM

Youssef Ibrahim Abd-Elaziz
Cairo University , Faculty of Engineering
Department of Public Works

Hazem Ahmed Fathy Barakat
Cairo University , Faculty of Engineering
Department of Public Works

Amr Hanafi Ahmed Ali
Benha University , Faculty of Engineering-Shoubra
Surveying Engineering Department

Heba M. M. Ahmed El-Qadi
Benha University , Faculty of Engineering-Shoubra
Surveying Engineering Department

ABSTRACT

Most transport telematics applications and services such as Vehicle Navigation Systems “VNS”, dynamic route guidance systems, fleet management, collision avoidance systems, advanced traveler information and on-board emission monitoring require continuous and accurate positioning information of vehicles traveling on the road network. Many services also require real-time display of the vehicle location on a map in an error-free fashion. Two essential components commonly used for such applications and services are, sensors to determine the geometric position of the vehicles, and Geographic Information Systems (GIS) based digital maps for the identification of the physical location of the vehicles. Even with very good sensor calibration and sensor fusion technologies, inaccuracies in the positioning sensors are often inevitable. There are also errors associated with spatial road network data. An interesting but challenging problem is to integrate positioning sensor data with digital map data to improve the accuracy with which the vehicle location on a road link is determined.

The main objectives of this research are to define reliable real time and automatic functions, where an improved Map Matching (MM) algorithm to reconcile inaccurate location data with inaccurate digital road network data was developed. The basic characteristics of the algorithm take into account the historical trajectory of the vehicle, and topological information on the road network (e.g., connectivity and orientation of links). This then enables a precise identification of the correct link on which the vehicle is traveling. Also, determination of the shortest path distance between source point and destination point was implemented.

KEY WORDS:

Mobility, Vehicle Navigation System, Map Matching, Routing, Shortest Path Distance

1. INTRODUCTION

In today's world, mobility is considered a very important factor in every field. This has caused the number of vehicles to increase on a daily basis. However, the capacity of road networks is limited and the route planning is not optimized. Naturally, this has led to the well known problems of urban traffic congestion [1].

There are two ways to deal with these problems; either by optimizing the road networks' capacity or by optimizing the road networks' efficiency through the Vehicle Navigation System (VNS). VNS shares a much common element where all the navigation components are covering a wide range of Geo-scientific disciplines such as geoinformatics, optimization theory and digital cartography.

The three common modules at most of the VNS are the positioning module, the communication module, and the location (map database) module. In this research we will focus mainly on the third module.

The first is the positioning module, within which; the system is going to define the current mobile vehicle position. Different positioning sensors that can be applied in these systems are either infrastructure or autonomous systems. The GPS represents the major applied Satellite navigation systems that is used in this direction, on the other hand, Dead reckoning principles define the major useable theory for many sensors "Inertial navigation sensors, odometrs and compass and differential odometry" [2].

The second is an essential aspect for most of the VNS which is the communication module where it is used for backhauling the determined vehicle position, the necessary information to the vehicle on call. In addition, it can be used for forward communication of range corrections for the Differential GPS (DGPS) when it is applied. The conventional radio, mobile satellite systems, beacons and cellular technology are some of these affordable communication systems [3].

The third module the paper is focusing on is the Location Module. Herein, the system starts gaining the data gathered from the first two modules and converts them into useful information that can be applied in different forms, where the location module defines the vehicle's position relative to land features such as road networks, road intersections and names etc… [4]

The two major functions of this module we are going to focus on are:

- **Map Matching function**

The assumption behind map matching is that the vehicle is always tracking on a road. Map matching function is to match the vehicle position into the road network; in order to improve the accuracy of the position [5].

- **Routing Function**

This function is used to generate the optimum route to the destination. Route guidance is used to indicate the path to the driver by means of a series of visual or audio instructions. It is very useful to display the current location of the vehicle, and the planned route on the background of a road map [6].

From the above functions, we can see that the applied data should be prepared in a way that fits the needed results. In other words, spatial data should be arranged digitally, where the road network layer should be prepared as follows.

1. Road network, the complete set of road segments is defined in a vector format and updated by the topological data.
2. Getting the points of intersections is a major requirement for the route planning.
3. Additional information about road classification, traffic flow and traffic restrictions will enhance the performance of the system.

Other requirements might be needed for other functions, but here, we will focus on the previous functions and the related spatial data.

2. METHODOLOGY

Recalling, that in this research we will focus on the essential functions of location module, these functions are reliable, real time and automatic functions. The First is developing an improved Map Matching (MM) algorithm. The second is the generation of the shortest path distance between source point and destination point.

2.1. Map Matching (MM)

The principle of map matching method ensures that the position is snapped or matched with the nearest street. However, a street network can be quite complicated [7].

It is done by correlating two sets of geographical data (e.g. GPS records of object positioning versus road network) [8]. This technique is used to reduce the errors due to both digital map and positioning data inaccuracy. These errors are systematic as well as random in nature.

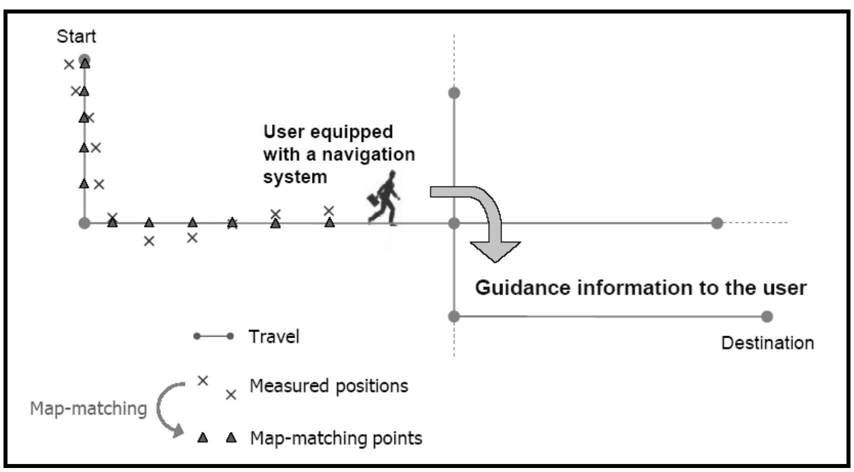

Fig.(1): Map Matching Process.

The map matching process includes finding the path that is the best estimation of the route which was actually taken by the user as in Figure (1) [9]. Different techniques are performed to fit this requirement.

2.1.1 Map matching techniques

As mentioned before, Map Matching (MM) techniques are often used to determine the accurate position of a vehicle on a road map. Most of the formulated algorithms utilize navigation data and the digital road network data. One of the common assumptions made about implementing MM is that the vehicle is essentially constrained to a finite network of roads. Most of the cases also proved that the digital map used for MM must be quite robust in order to generate the position outputs in an error-free manner. Procedures for MM vary from those using simple searching techniques to others applying more complex mathematical techniques.

The most commonly used geometric MM approach is based on an assumption that each positioning point matches with the closest 'node' in the network, it is known as point-to-point matching. A number of data structures and algorithms exist to identify the closest node from a given point in a network. These methods are easy to implement, although they are very sensitive to the way in which the network was digitized, hence leading to errors. One natural way to proceed is to match the vehicle position with the "closest" node in the network. [10].

In particular, recall that the Euclidean distance between two points a and b in

R 2 is given by:

$$\|\text{b-a}\| = \sqrt{(\text{y}_\text{b}\text{-y}_\text{a})^2 + (x_b - x_a)^2} \qquad (1)$$

Where x_a,y_a are coordinates of point a and x_b,y_b are coordinates of point b

Another geometric MM approach is point-to-curve. In this approach, the positioning point from the navigation system is matched with the closest curve in the network. Each curve comprises line segments which are piece-wise linear. Distance is calculated from the positioning point to each of the line segments. The line segment which gives the smallest distance is selected as the one on which the vehicle is assumed to be traveling. Although this approach gives better results than point-to-point matching, it provides, in some cases, several shortcomings that make it inappropriate in practice as it generates very unstable results in dense urban networks.

The next natural way to proceed is to attempt to identify the arc in N that is "closest" to positioning sensor point (Pt), rather than the point that is closest to Pt.

So, the most common approach is to use the minimum distance from the point x to the curve. Since we are dealing with piece-wise linear curves, to find the minimum distance from a point to a curve, we must find the minimum distance from x to each of the line segments that comprise the line 'A' and then select the smallest one.

Obviously, it is fairly simple to find the minimum distance between a point and a line.

In particular, suppose that we let {ax+by=c} denote the line, 'A' through 'a' and 'b'. Then, the minimum distance between some point c and this line is given by:

$$D(c, A) = \frac{(y_a - y_b)\, x_c + (x_b - x_a)\, y_c + (x_a y_b - x_b y_a)}{\sqrt{(y_b - y_a)^2 + (x_b - x_a)^2}} \quad (2)$$

Where D(c,A) is the distance along the “perpendicular” from c to the line 'A' , xa,ya,xb,yb coordinates of point a and b passing through line A and xc,yc is the coordinates of point c.

To match Pt to an arc, one must calculate the minimum distance between Pt and all “reasonable” arcs in N and choose the one that is closest. While this approach may seem perfectly appropriate at first glance, it does have several shortcomings that make it practically in appropriate.

Since, point-to-curve matching does not make use of “historical” information; some mismatching might arise as illustrated in Figure (2). The estimated position P2 is closer to arcs B than A, where the history of the route provide that P0, P1 are matched to Arc A and P2 should also be matched to arc A. The instability is another problem with point-to-curve matching. This is illustrated in Figure (3). The points P0, P1, and P2 are all equidistant from arcs A and B. But, it turns out that P0 and P2 are slightly closer to A and P1is slightly closer to B. Hence, the matching oscillates back and forth between both [11].

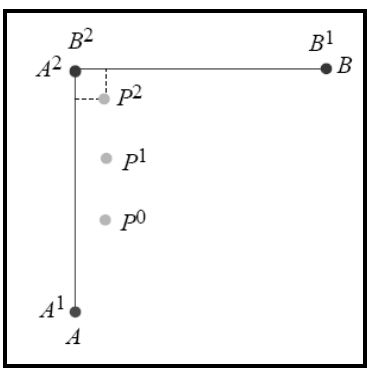

Fig. (2): One problem with Point-to-Curve matching.

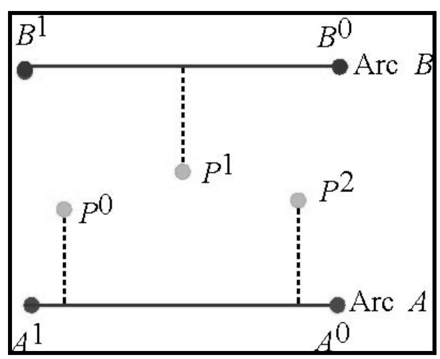

Fig. (3): Another Problem with Point-to-Curve Matching.

Another geometric approach is to compare the vehicle’s trajectory with known roads. This is also known as curve-to-curve matching [10].

In this approach, first, the candidate node using point-to-point matching is identified. Then, given a candidate node, piece-wise linear curves are constructed from the set of paths that created from that node. Second, piece-wise linear curves are then constructed using the vehicle's trajectory. The distance between the curve along the vehicle trajectory and the curve corresponding to the road network is then determined. The road arc which is closest to the curve formed from the vehicle trajectory is considered as the apparent traveling route. This approach is quite sensitive to outliers and depends on point-to-point matching, sometimes giving unexpected results.

Map matching algorithms can also be classified into semi-deterministic map matching and probabilistic map matching algorithm. The semi-deterministic concept is based on the presumption that the vehicle is always on a pre-defined route on road network.

The algorithm repeatedly asks "Is the vehicle still following the route?" and "What is the vehicle's location along the route?" The vehicle is snapped on the route if some tests are satisfied (i.e. when it executes turns consistent with differences in direction emanating from the nodes, or when it confirms to the geometry of given individual links). The location along the roads is estimated and errors in the estimation are automatically removed at each node, where an expected change in the vehicle heading really occurs [12].

On the other hand, the probabilistic map matching algorithms maintain a running estimate of uncertainty in location, which is taken into consideration in determining whether the vehicle is on a road or not. The algorithm repeatedly asks "where is the vehicle?" with no a prior presumption that it is on a road. In this approach the vehicle position is assumed to be estimated by the computed coordinates and associated confidence ellipse. If the ellipse contains one intersection, a map match is made easily. For the case that more than one intersection is contained in the confidence ellipse, then connectivity checks are made to determine the most probable location of the vehicle. The estimate of location uncertainty is reduced each time until the next match occurs [5].

Furthermore, Map matching algorithms can be classified into on line map matching algorithm and off line map matching:

On line map matching methods snap the device captured geo-spatial feature position on the base reference in real time. On line snapping determines the road segment on which the vehicle currently is located during a trip, in real time. On line map matching typically can not use the global information contained in the data, it means that only the current and the previous data points are available.

Off line map matching counterparts post snap the data after the whole set of data is collected. Off line snapping, which finds the overall (i.e., a sequence of arcs in the map) of a vehicle after the trip is over. Off line map matching typically can use the global information contained in the tracked data points; it means that the current, previous and the next data points are available, [8] [13]. Figure 4 represents the on line/off line approaches.

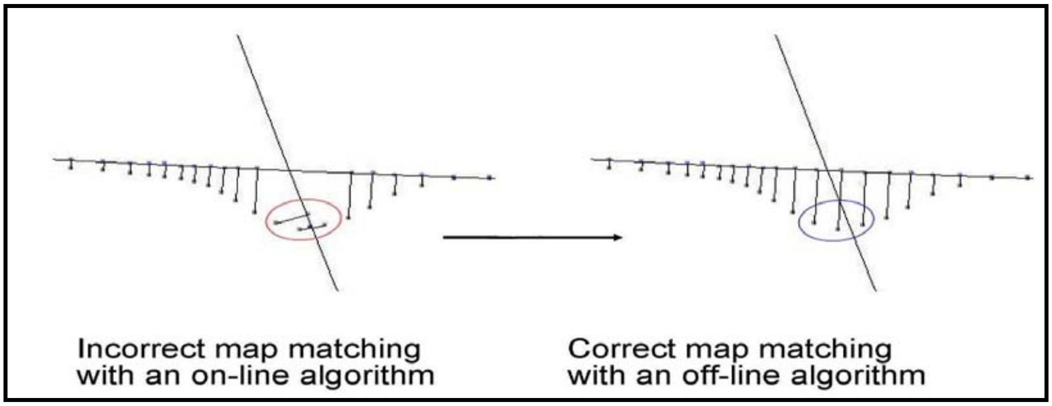

Fig. (4): On line – Off line Map Matching techniques.

For this research, a new algorithm that combines many well-known algorithms is developed. The major objective is to combine their advantages and avoiding the major shortcomings.

The details of the proposed algorithm and mathematical models are described below.

2.1.2. The proposed map matching algorithm

The proposed algorithm follows different criteria and techniques of map matching. It follows the semi-deterministic map matching technique, where both the start point and destination point are well known. Although, the route is pre-defined by the driver request, the driver is free to decide whether to follow it or not.

The algorithm is developed to recognize if the driver is following the generated route or not, as it will re-generate the route to the destination point from the current vehicle position in case of any sudden changes to the predefined route is occurred.

At the same time, the proposed algorithm allows matching the vehicle to any road network segments, not only the pre-defined segments; so, it follows the probabilistic map matching technique.

Meanwhile, the proposed algorithm satisfies on line map matching technique as navigation systems required real time feature.

Also, it satisfies the point–to–curve matching technique except at intersection where it applies point – to – point matching technique. These assumptions provide better and flexible system and enhance the map matching results.

2.1.2.1. Description of algorithm input:

The inputs of the developed MM algorithm are the positional data and topological information of the road network.

The positional data includes position (e.g., easting and northing), where the source for this work is simulated as a captured set of GPS observations in navigational mode "Single Point Position with accuracy ± 10 m".

Also, the digital road network in vector format is taken as an input. A road network is then updated by the road directions "topological information" [14].

2.1.2.2. The proposed MM algorithm criteria

The map matching step involves the implementation of several criteria that may be implemented in a certain sequence. The first criterion, proximity rule "The Shortest Distance between a Point and a Road Segment Criteria" involves assigning the measured location into the nearest road to the instantaneous measured data; the true solution cannot be achieved by applying this criterion only. The second criterion, bearing matching involves the assertion that the nearest road, to which the vehicle position is correlated, has similar bearing as the direction of travel. The third criterion, the access one, involves that the nearest and bearing matched road have access to the measured vehicle direction. The sequences of the map matching process can be implemented using the previously mentioned criteria and algorithm in a logical manner. The matching process will be based on the integration between these criteria as mentioned in research work[15].

Regarding the Shortest Distance between a Point and a Road Segment Criteria

Let us suppose that G (V, E) be the directed graph describing the road network, where V is the set of nodes and E is the set of segments. Let Qi be the set of points given by the GPS data stream i=1…T. Each GPS point consists of a pair of coordinates (Xi ,Yi).The orthogonal distance between a GPS point Q and the oriented link AB is the distance QQ` where Q` is the projection (snap) of a point Q on line AB (i.e. the intersection between the line AB and the perpendicular line passing through point Q), as in Fig. (5).

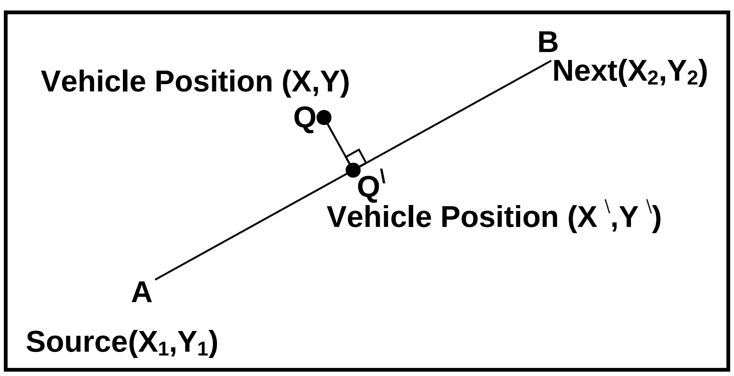

Fig. (5): Shortest Distance between Vehicle and Road Segment.

The orthogonal distance can be measured using the following equations

$$\mathbf{M=Y_2\text{-}Y_1/X_2\text{-}X_1} \quad (3)$$
$$\mathbf{M`=-1/M} \quad (4)$$
$$\mathbf{C`=Y\text{-}X*M`} \quad (5)$$
$$\mathbf{C=Y_1\text{-}X_1*M} \quad (6)$$
$$\mathbf{X^{\backslash}=C^{\backslash}\text{-}C/M\text{-}M^{\backslash}} \quad (7)$$
$$\mathbf{Y^{\backslash}=M*X^{\backslash}+C} \quad (8)$$

Where:

(X_1,Y_1) , (X_2,Y_2) , (X,Y) , $(X^{\backslash},Y^{\backslash})$: are the coordinate of A,B,Q,$Q^{\backslash}$ respectively.

M: Slope of the segment AB.

$M^{\backslash}$: Slope of the segment perpendicular to the segment AB.

C: Intercepts of the segment AB.

$C^{\backslash}$: Intercepts of the segment perpendicular to the segment AB.

$$\mathbf{d(Q,AB)=QQ^{\backslash}=\sqrt{((X\text{-}X^{\backslash})^2+(Y\text{-}Y^{\backslash})^2)}} \quad (9)$$

Where d(Q,AB): is the orthogonal distance between node Q and segment AB.

If d (Q, AB) is equal to zero, then Q is on the line. Depending only on the shortest orthogonal distance in matching procedure will often give incorrect results [8]. A better way to proceed is to integrate these criteria with another one which is based on the direction phenomena “bearing of the road segment and access rules”

At The Road Bearing Rule Criteria, a simple comparison between a vehicle’s direction of travel (its bearing) and attribute data of road segment (direction) is used for that purpose. The determination of the line between any two successive vehicle positions V1 and V2, as shown in fig. (6) is implemented. The bearing of the road segment is also computed while its direction “one way- two ways” is known from topological information.

$$\mathbf{Bearing\ V_1, V_2 = tan^{-1}(X_2\text{-}X_1/Y_2\text{-}Y_1)\quad if\ (Y_2\text{-}Y_1) = 0\ add\ \prod,\ if\ < 0\ add\ 2\prod} \quad (10)$$

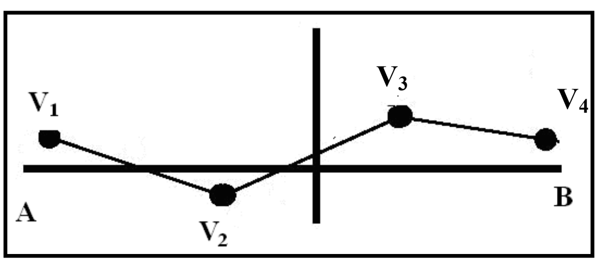

Fig. (6) Bearing of Position Line Segment.

Through a simple comparison between bearing of position line segment and road segment, the road selection is determined [16].

Through The Road Access Rule Criteria, road network is updated with topological information such as access of road. By using this information, we can improve the proposed algorithm to reduce the set of road segment where the integration between the above mentioned criteria will lead us to take a better decision, where we can improve our mission by answering the repeated question "Which road segment is the correct one that the vehicle should match with respect to it?"

In such case, most of the time the result is expected to be the correct one.

The major steps that were followed to implement the MM algorithm are shown in Figure (7).

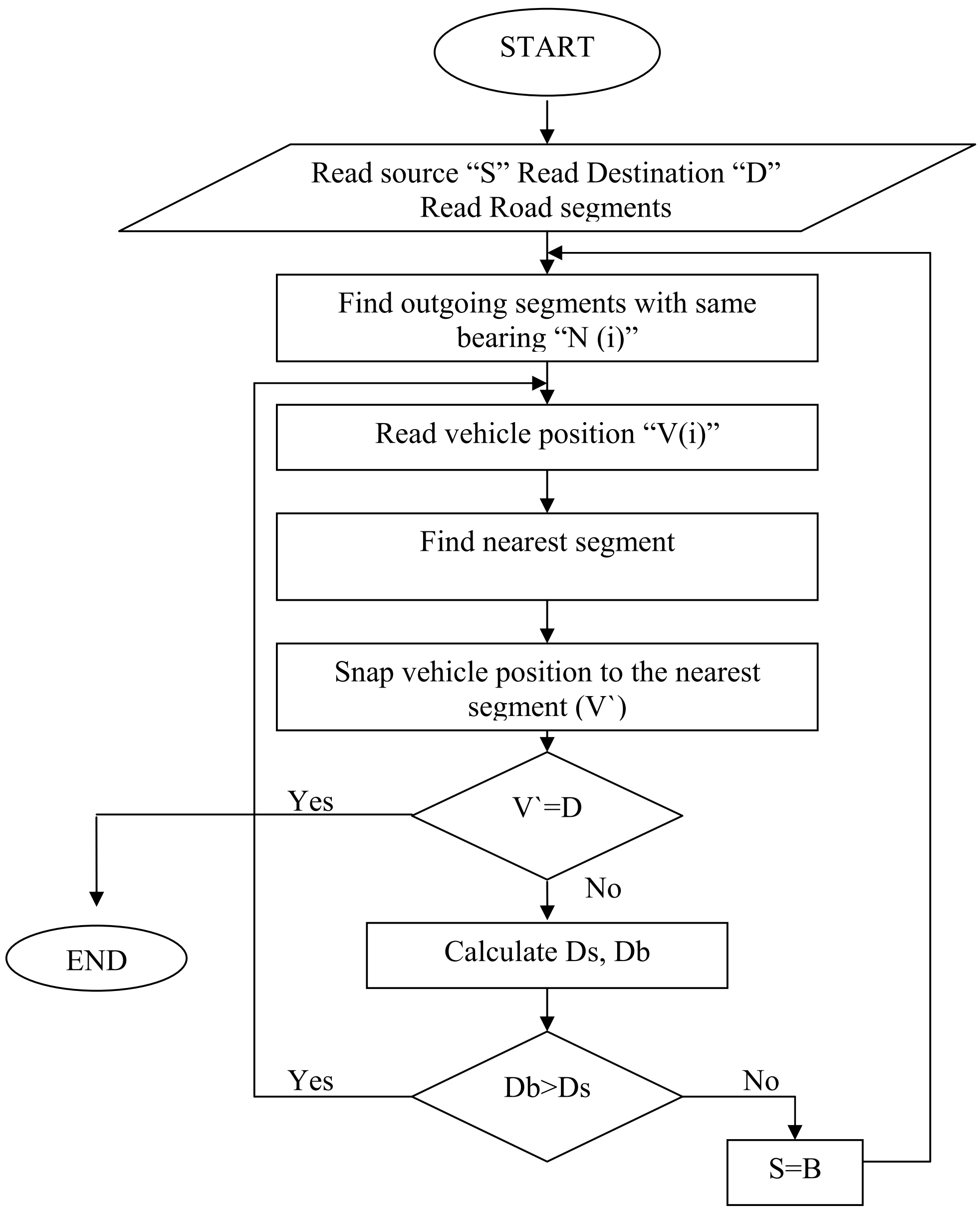

Fig. (7): Map Matching Flow Chart.

Where: Ds is the distance between the snapping point and the start point of the segment and Db is the distance between the snapping point and the next point of the segment.

After fulfilling the major needs of MM, another function is developed to increase the efficiency of the VNS. On time routing this function will define the optimal route between two points at the road network.

2.2. On Time Routing

A large number of optimization problems are mathematically equivalent to finding an optimum path in a graph. The algorithmic core problems that underlie the vehicle navigation application is a special case of finding optimum path from specified source node in road network “directed graph” to a specified destination node. This process here was developed using a novel algorithm. The algorithm is based on the concept of graph theory and algorithmic techniques.

Let G (V, E) be a directed graph whose segments are weighted by a function. We interpret the weights as the segment lengths, where the length of the path is the sum of weights of its edges. The problem lies in finding a path of minimum length from a given source S € V to a given destination D € V.

The proposed algorithm purely follows the object oriented programming concept and the combination between most of the algorithmic techniques is done, as applying dynamic programming, partitioning, tree searching, recursion, and geometric method (nearest neighbor problem).

2.2.1. Data description and solution achievement

A road network can be represented by a multi-graph which consists of nodes and segments “the intersection in a road networks form the nodes of the multi-graph”.

The road lines between intersections in a road network form the segments of the multi-graph because one way road exists, all segments are directed.

The actual solution can be divided into two distinct components; the graph processing and the route finding. For the graph processing component, the nodes of the graph are loaded in dynamic array list where the structure of each element of the list includes all nodes connected with that node and the length between them. Such a method occupies small storage area which improves the performance and speed of the algorithm.

For route finding algorithm, the source and destination nodes must be defined and they must be one of the graph nodes; if they are not, we will use the nearest neighbor technique in order to find the closest nodes to the source and destination nodes [14].

The implemented shortest path distance algorithm is represented in figure (8).

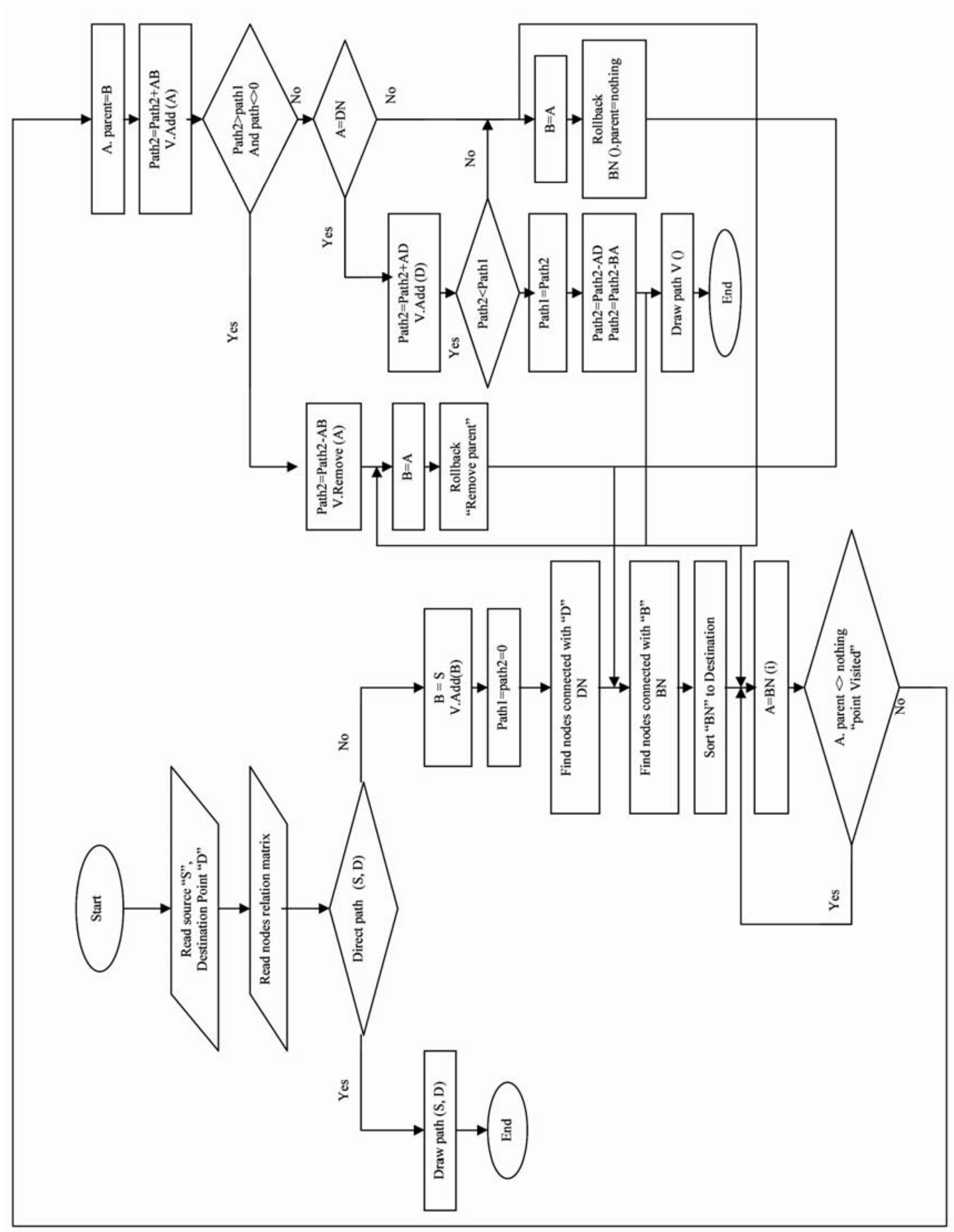

Fig. (8): Shortest Path Distance Flow Chart

Where the input data are **N** that is an Integer variable that represents the number of nodes.

Nodes is a Dynamic array list of dimension N, including all nodes structure elements,
S is a Specified source node € Nodes, and **D** is a Specified destination node € Nodes.
And the output data is **V** (a Vector includes the nodes of the optimum path in sequence from S to D).
The working storage includes, **Path1** that is a Double variable (the length of the previous path).
Length Double variable which is the length of the current path, **DN** is a dynamic array list includes node elements connected with D and **BN** which is a dynamic array list includes node elements connected with B "current node" which is equal S at beginning.

3. RESULTS AND ANALYSIS

3.1. On Road Map Matching

For the sake of analyzing this part, a simulation study was performed. Simulation for VNS data for urban environment can be undertaken considering a road network. The vehicle motion is considered to travel from point "A" to point "B" as shown in figure (9).

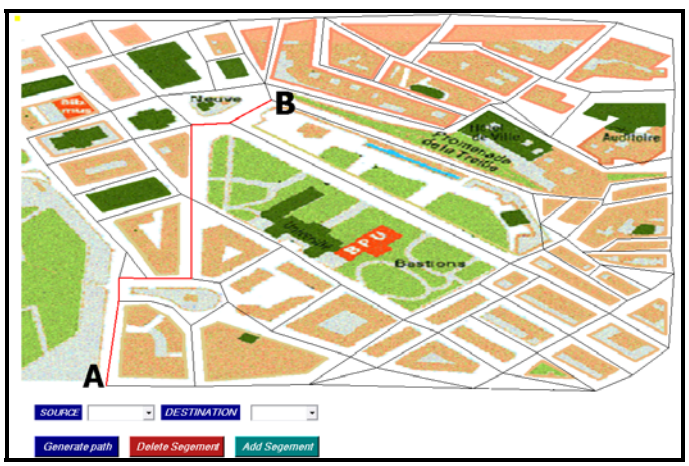

Fig. (9): Vehicle Test Route from Point "A" to Point "B"

The experimental positioning data is simulated as presented in case (1, 2, and 3) where pseudorandom errors were introduced into the assumed true vehicle location. The error assumed to have the behavior of GPS error in single point positioning mode with different magnitude to indicate the nature of the noise error that may affect the measurements. The errors are assumed to be $\Delta X = \Delta Y = \pm 10$ m.

In order to implement the simulation, the following steps were taken:

1. Defining the true position of the vehicle with fixed time rate.
2. Assuming the random noise range, as a result of using the GPS navigators as the applied positioning sensor.
3. Accumulating the true position with the random noise to get the noisy position.
4. Snapping the noisy position using the proposed algorithm.

- **Case (1)**

The true vehicle position specified for two successive epochs at time interval 2 seconds (with bout distance interval 10 m). After implementing the matching process, the vehicle position can be represented as true, noise and snapped locations as shown in figure (10).

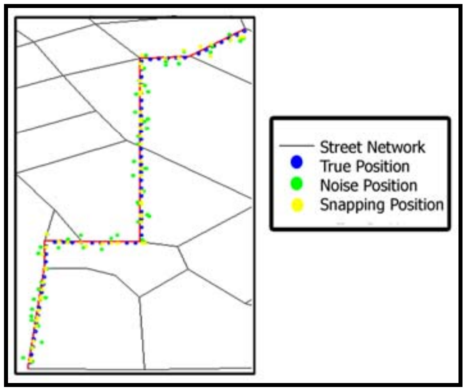

Fig. (10): True, Noise and Snapped Vehicle Position Data (Case 1)

The true, noise and snapped positions X-coordinates, Y-coordinates were defined.

Then, the related true and snapped drift is calculated according to equations 11& 12. These true and snapped drift for case 1 is represented in figure (11)

Where:

True Drift $=\sqrt{[(X_{Noise}-X_{True})^2+(Y_{Noise}-Y_{True})^2]}$ (11)

Snap Drift $=\sqrt{[(X_{Snapped}-X_{True})^2+(Y_{Snapped}-Y_{True})^2]}$ (12)

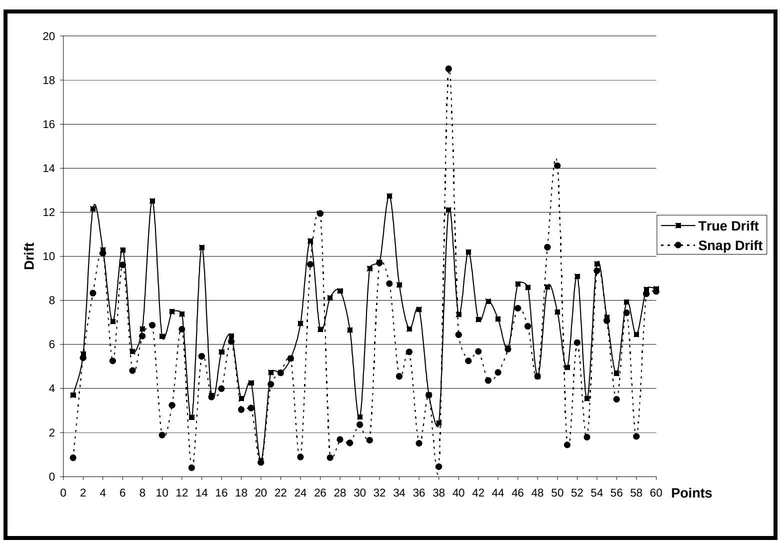

Fig. (11): True and Snapped Drift (Case 1)

The mean value of the assumed positional drift was reduced from being 7.00 m to 5.4 m after matching process. Such a drift is adequate for many tracking navigation applications, the major short come, appears when the vehicle reaches an intersection. More care, during junctions, should be taken into consideration.

- **Case (2)**

The true vehicle position specified for two successive epochs at time interval 5 seconds (with distance interval 25 m). After implementing the matching process, the vehicle position can be represented as true, noise and snapped locations as shown in figure (12).

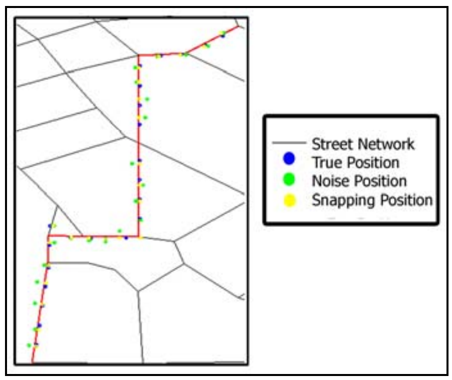

Fig. (12): True, Noise and Snapped Vehicle Position Data (Case 2)

The results for case (2) are determined as true, noise and snapped positions. Also, the related true and snapped drift is calculated and represented in figure (13)

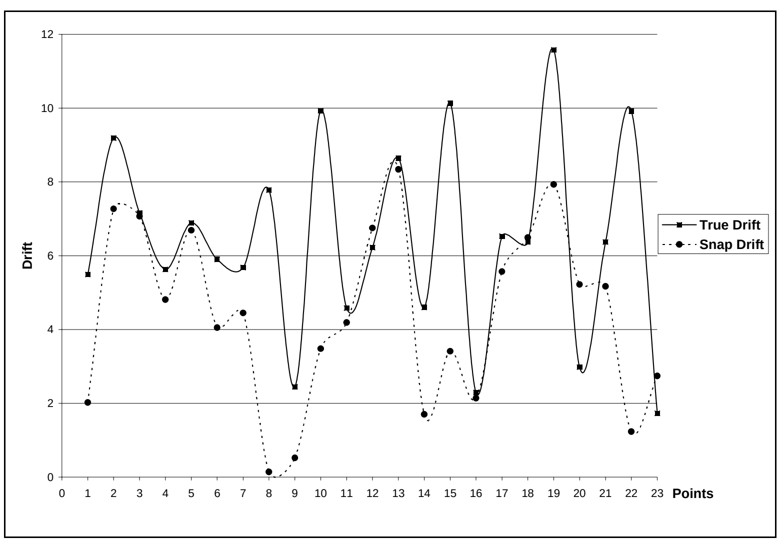

Fig. (13): True and Snapped Drift (Case 2)

The mean value of the assumed positional drift was reduced from being 6.44 m to 4.41 m after matching process.

- **Case (3)**

The true vehicle position specified for two successive epochs at time interval 5 seconds (with distance interval varied from apart to other along the route from point "A" to point "B" which is most likely the reality). After implementing the matching process, the vehicle position can be represented as true, noise and snapped locations as shown in figure (14).

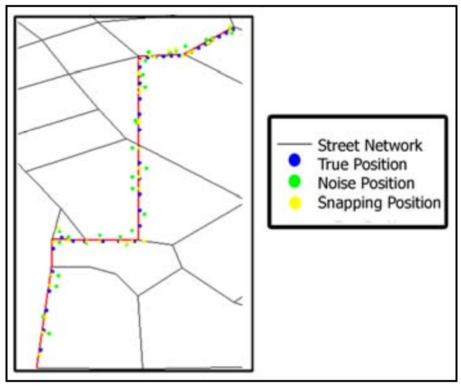

Fig. (14): True, Noise and Snapped Vehicle Position Data (Case 3)

The results for case (3) are determined as true, noise and snapped positions. Also, the related true and snapped drift is calculated and represented in figure (15)

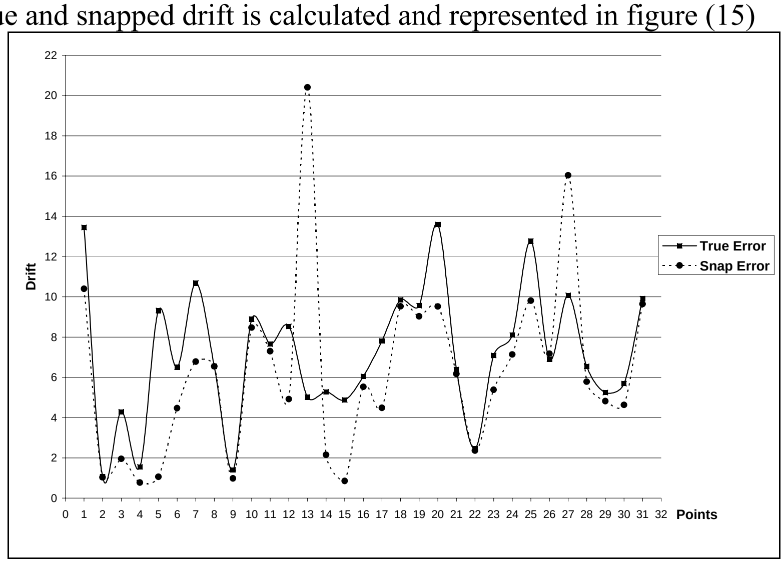

Fig. (15): True and Snapped Drift (Case 3)

The mean value of the assumed positional drift was reduced from being 7.26 m to 6.10 m after matching process.

As a result of the investigated cases, the algorithm provides positive response. Tests show that the procedure computed correct matches every where. It is observed that some points were snapped incorrectly at the intersections as shown in figures (10), (12) and (14).

3.2. On Time Routing

The function is to generate online optimum path between a starting point and a destination point. This procedure was developed using algorithm based on the concept of graph theory and algorithmic techniques. The developed algorithm is implemented using VB.NET programming language.

The prerequisite data to this function is the vector road network layer which can be represented by a multi-graph (m-graph). An m-graph consists of nodes and segments. The intersection of road networks forms the nodes of the m-graph. The road lines between intersections in a road network form the segment of the m-graph; all segments are updated by the direction.

Figure (16) shows the road network layer with background of the raster image and table (1) shows sample to the relations between network segments and the length of these segments.

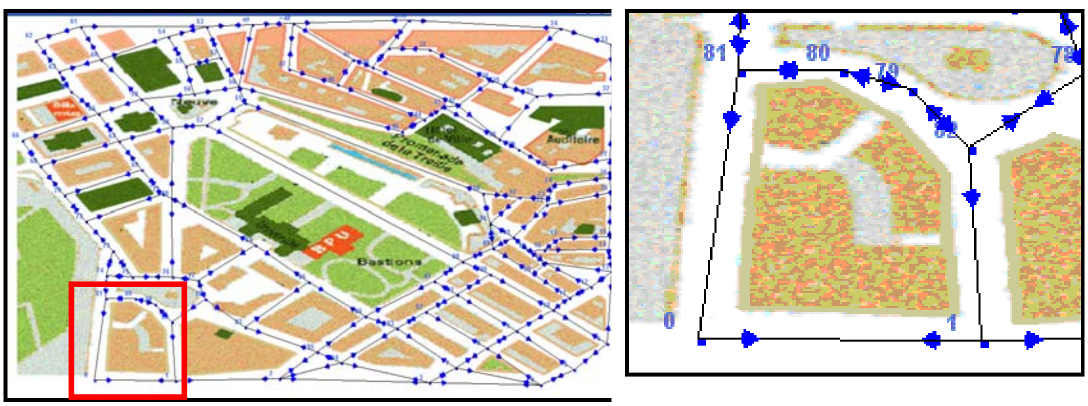

Fig. (16): Road Network Layer "with directions"

Table (1): Network Segment Data.

From	X(From)	Y(From)	To	X(To)	Y(To)	Distance(m)
0	5120016	280364	P1	5120151	280365	135
0	5120016	280364	P81	5120035	280236	129
1	5120151	280365	P0	5120016	280364	135
1	5120151	280365	P2	5120324	280362	173
1	5120151	280365	P82	5120145	280273	92
81	5120035	280236	P74	5120036	280200	36
81	5120035	280236	P80	5120084	280236	49

Table (1) represents the nodes of the street network and the relation between each of the connected nodes, where each node is represented by its coordinates form (Node number, X and Y coordinates) To the connected nodes that are identified also by their (node numbers X and Y coordinates). For example, node 0 is connected with nodes 1, 81

As mentioned before, the choice of the navigation route can depend on the distance, time, velocity, etc…...

Herein, the assumption is to generate an optimum path based on shortest distance, so it can be called as Shortest Path Distance "SPD".

Through this function and according to the designed algorithm, the next result is obtained, where the SPD from point "K" to point "Z" is represented in figure (17).

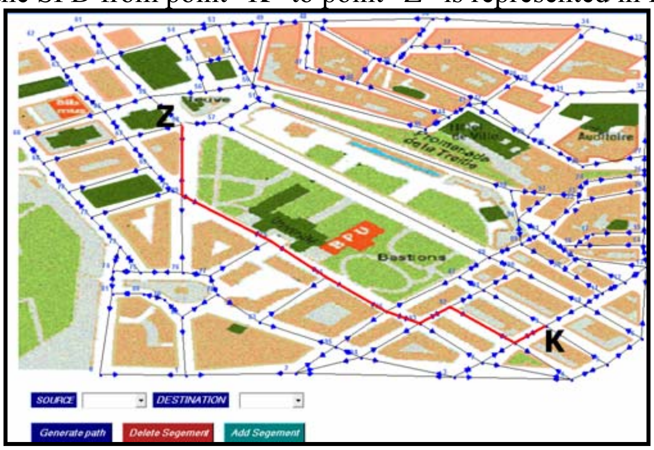

Fig. (17) : Shortest Path from Point "K" to Point "Z" 'Red Route'

As mentioned before, the road network is represented as an m-directed graph which defines the direction of the segments over the road network. So, the developed package is taken this matter into consideration. It can be observed that the shortest path from point "K" to point "Z" (Route 1) differ from the shortest path from point "Z" to point "K" (Route 2) as shown in figure (18).

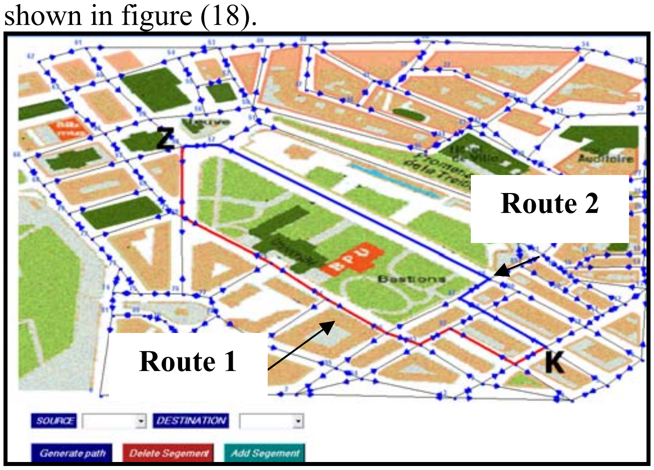

Fig. (18) : Shortest Path from Point "Z" to Point "K" 'Blue Route'

In case of a sudden appearing constraint during the navigation process, if any part of a road network had an unexpected situation that prevents the tracking process to be run on that direction, the algorithm is designed to deal with such a situation and recalculate the modified optimum path

In figure (19), it is assumed that the segment from point "R" to point "T" is neglected due to a problem occurring in that road. The path from point "K" to point "Z" is recalculated in real time and the new route is generated.

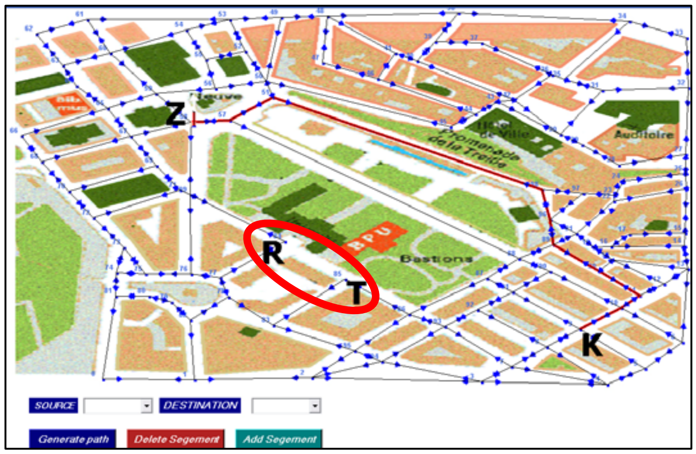

Fig. (19): Recalculation of Shortest Path from Point "K" to Point "Z" due to Deleting Segment from point "R" to point "T"

Finally, the "On Time Routing Process" meets the standard requirements for the routing function. It offers the SPD between nodes. It taking into consideration the direction of the road network. It can handle sudden situations like occurrence of a problem to specific segments and regenerate the optimum solution in real time.

4. CONCLUSIONS

Vehicle Navigation System as part of transport telematics applications can be proposed as the alternative for optimizing the traffic congestion. It is assumed to reduce drivers' stress all over the road network and to increase road safety. Also, it can be used for solving problems such as: non optimized road planning and limited capacity of road network. Urban emergency response and disaster management require quick and reliable vehicle navigation where optimal route to the destination is required.

The main purpose of VNS is to guide vehicles from any position to a chosen destination point in a safe and optimal way. Any efficient VNS should have three main modules: the first is the positioning module used for the determination of the current mobile vehicle position; the second is the communication module that is used for backhauling the determined vehicle position, and the third one is the location module that converts the determined position into useful information (location, direction…).

This module has different processes, where the most applicable ones are map matching and routing functions. Map matching process is to snap the incoming vehicle position, which is approximated by the navigational positioning sensors such as GPS to the defined digital road network. Another function related to the location module is the routing function. This function is used to generate the optimum route from the source point to the destination point based on certain conditions as shortest path or less time.

For the map matching process, the process acts upon matching the incoming vehicle position to the extracted street network using developed algorithm based on different criteria and techniques of map matching. The proposed algorithm is implemented with VB.NET programming language. A simulation was done in order to test this process. Three cases were simulated where the assumed positional drifts were reduced in the three cases. And, in case (3) which is very close to the reality the positional drifts were reduced from 8 m to 7 m after matching process.

Finally, merging different techniques improved the accuracy of the map matching process. The algorithm is functioning well. Tests show that the procedure computed the correct matching every where.

Despite the fact that the results are performing well, additional researches is still need to be done in order to verify the accurate performance of the algorithm.

Moreover, it is recommended to use Artificial Neural Networks techniques and fuzzy logic algorithm in order to improve the performance and accuracy of matching techniques.

With respect to routing function, the function is to generate the optimum path "shortest path distance" between a start location to a destination point using a developed algorithm based on graph and networks theories. The proposed algorithm was implemented with VB.NET programming language.

During the implementation of this procedure, the use of dynamic programming speeds up the search method in optimum path finding problem. Also, applying object oriented concept facilitates the stage of developing, implementation, and modification until the proposed algorithm is reached.

At the end, the function defines the shortest path distance while taking into consideration the direction of the road. Also, during navigation process if any part of a road network had unexpected situation that prevents the tracking process to be run on that direction, the package has the capability to regenerate the path and define the new optimum solution in real time.

5. REFERENCES

[1] Flinsenberg, I. C. M. (2004): “Route Planning Algorithms for Car Navigation”, Ph.D. Dissertation, Institute for Programming Research and Algorithmic, Ingrid C.M., September 2004.

[2] Krakiwsky, E. J. (1994): “Comparison of IVHS Navigation Systems”, Third International Conference on Land Vehicle Navigation, Dresden, Germany, 14-16 June 1994.

[3] May, M. B. (1993): “International Navigation and GPS”, GPS world, Sep. 1993.

[4] Krakiwsky, E. J., and Bullock, J. (1994): “Digital Road Data Putting GPS on the Map”, GPS world, May 1994.

[5] French, R. L. (1989): “Map Matching Origins, Approaches and Applications”, Second Symposium on Land Vehicle Navigation, Munster, Germany.

[6] Claussen, H., Lichtner, W., Heres, L., Lahaije, P., and Siebold, J. (1989): “Proposed Standard for Digital Road Maps to be Used in Car Navigation Systems”, Second Symposium on Land Vehicle Navigation, Munster, Germany.

[7] Guo, F., Ji Y., and Hu, G. (2002): “Methods for Improving the Accuracy and Reliability of Vehicle Borne GPS Intelligence Navigation”, www.gisdevelopment.net, 14 August 2002.

[8] Zhou, J. (2005): “A Three- Step General Map Matching Method in the GIS Environment: Travel / Transportation Study Perspective”, 2005, www.ucgis.org.

[9] Marchal, F., Huckney, J., and Axhausen, K. W.J. (2004): “Efficient Map Matching of Large GPS Data Sets- Tests on a Speed Monitoring Experiment in Zurich”, Arbeitsbericht Verkehrs-und Raumplanuny Transport Systeme, ETH , Zurich July 2004.

[10] Ocheing, W.Y., Quddus, M., and Noland, R. B. (2003): “Map Matching in Complex Urban Road Networks”, Center for Transport Studies, Department of Civil and Environmental Engineering, Imperial Collage London, W.ochieng@imperial.ac.uk.

[11] Bernstein, D., and Kornhauser, A. (1996): “An introduction to Map Matching for Personal Navigation Assistants”, Princeton University, New Jersey Tide Center, August 1996.

[12] Taylor, G. and Blewitt, G. (2000): “Virtual differential GPS & road reduction filtering by map matching”, 2000, geodesy.unr.edu/publications.html.

[13] Yin, H., and Wolfson, O. (2004): “A weighted-based Map Matching Method in moving objects Database”, 16th International Conference on Scientific and Statistical Database Management, June 2004, Greece.

[14] El-Qadi Heba. (2007): “Towards Real Time Vehicle Navigation System”, M.Sc. Thesis, Cairo University, Public works department, 2007.

[15] Abdel-Maguid, R. H. (2004): “Development of Knowledge Based Map Matching Intelligent Land Vehicle Navigation”, Civil Engineering Research Magazine CERM, Faculty of Engineering, Al-Azhar University Vol. (26).No. (1), January 2004.

[16] Greenfeld, J. S. (2001): “Matching GPS Observations to Locations on a Digital Map”, NJ TIDE Center, Newark, 2001, www.njtide.org.