# Microsegmenting: An approach for precise distance calculation for GPS based ITS applications

Aradhya Biswas, Goutham Pilla, and Bheemarjuna Reddy Tamma
Department of Computer Science and Engineering
Indian Institute of Technology Hyderabad, India
Email: {cs11b003, gouthamp, tbr}@iith.ac.in

***Abstract*—Onroad distance calculation between two geographical points is an integral part of various Global Positioning System (GPS) based Intelligent Transportation Systems (ITS) applications. We have found that mere calculating the distance between two geographical points without giving importance to geographical information of the road, such as curves can lead to under estimation of the distance calculated, cause of which we refer to as the “ *Displacement problem* ”.**

**In this paper, we propose the methodology of *Microsegmenting* to overcome the *Displacement Problem*. To validate the proposed method and to quantify improvement over the existing technique of distance calculation we conduct experiments using real-world GPS traces from cities: Hyderabad, India and Chicago, USA. The experimental results show a significant improvement in distance estimation over existing technique. The significance of the improvement can be visualized by the fact that, theoretically this improvement in distance calculation can improve the travel time prediction, an important ITS applications, by an average of 22 seconds (approx.) between each pair of traces.**

## I. Introduction

Rapid advent in the field of wireless communication and positioning technology, in the recent past, led to the deployment of wireless devices equipped with GPS sensor on numerous public as well as private vehicles. This development combined with the boom in the number of vehicles resulted in generation of massive amount of vehicular positioning data.

The core of any Intelligent Transportation system as well as any study relating to this field, is the distance and the speed calculation and ubiquitous GPS data is the general source of these calculations. When using the GPS positional data for speed calculations, the problem again boils down to the distance calculation.

The currently used method for distance calculation[1] involves finding the great circle distance [6] between the points of interest, which are generally the data points or GPS trace points. This gives rise to what we term as the *Displacement Problem* i.e., by the distance method we are actually calculating the displacement and not the distance between the points, as the path the vehicle is traveling need not be linear or a straight path, which is the only case when displacement is equal to the distance. For example in Fig. 1, the actual distance between points A and B is the distance marked in blue but distance method would provide $|\overline{AB}|$ as the distance. So one can easily see that, more the number of turnings or curvature in the road segment more erroneous will be the results. Theoretically, the deviation from the actual distance covered and the calculated distance by the distance method would be affected by following two factors:

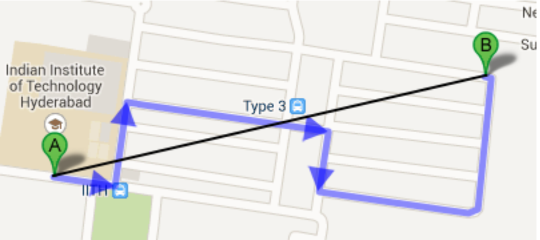

Fig. 1. Distance calculated by Distance Method

1) Frequency of GPS trace updation i.e., length of the road segment between each pair of GPS traces.
2) The curvature or the number of turns in the road segment between points of interest.

The first factor can be explained as, the decrease in the frequency of GPS traces results in an increase in the length of the road segment between two traces (holding other factors constant) which in turn increases the probability of inclusion of turnings and curves in the road segment. The second one can be explained as, with the increase in density of turns and curved roads in an area the probability of finding a curve or turning in any road segment also increases.

Any study or system when deployed on field confronts many constraints like transmission delays, network problems, and/or low quality GPS equipments which may result in highly inconsistent frequency of traces. These constraints are quite common on the field and hence contributes significantly in the error generation. Generally these can be overlooked easily, as generally any new system is tested in small scale controlled environment with high quality devices. In general the data on which the systems are deployed have a frequency of “a trace every 60-180 seconds” [2] [3], [4], [7]. Also the roads that are confronted in the urban areas are not quite straight.

In this research work we propose a novel method of *microsegmenting* for overcoming the *Displacement Problem* and hence for computing precise *onroad* distances. We also explain the working of a simple application that can be used for *microsegmenting* and finally we present a comprehensive empirical experimentation using data from real bus route(APSRTC route 502) in Hyderabad, Andhra Pradesh,

[1] This method of distance calculation is here on referred as *Distance Method*

[2] At some instants the interval was observed to be more than 750 seconds.

978-1-4799-2178-2/13/$31.00 ©2013 IEEE

India to validate our proposal and compare the results obtained by the distance method.

The rest of the paper is organized as follows. In Section II, we define the terminologies that have been used in the paper and also formally define the problem statement. In Section III, we provide an overview of our approach. In Section IV, we describe the procedure used for data collection. In Section V, we provide an overview of the evaluation procedure used. In Section VI, we present the results obtained and in Section VII, we finally conclude.

## II. Preliminaries

In this section we define the terminologies that are used in rest of the paper and formally define the problem statement targeted.

### A. Terminologies

*Definition 1.* **Segment** *A portion of a route, represented as a tuple of its starting and ending GPS coordinates or simply $(s, e)$, is referred to as segment $S$. The set of segments $\langle S_1, S_2, \cdots, S_n \rangle$ of a route should satisfy the condition, $e_i = s_{i+1}$ ,∀ integers $i \ \epsilon \ [1, n)$, i.e., the segments are contiguous parts of a route.*
*Hence, a route in turn can be defined as a set of segments as, $R = \langle S_1, S_2, \cdots, S_n \rangle$.*

*Definition 2.* **Microsegments** *A route, R, when partitioned in a manner that the segments arising due to the partition satisfy following properties:*

1) *Length, $l$, of all segments is equal.*
2) *Length, $l$ (in meters), lies in the range $(0, 10]$.*

*Such segments are referred to as* Microsegments.

*Definition 3.* **Probe vehicle** *Any physical system transmitting its GPS traces has been referred to as probe vehicle $V$.*

*Definition 4.* **Trace** *The tuple of latitude $L_\alpha$, longitude $L_\beta$, timestamp $t$, and optionally speed $v$, received from the GPS module deployed on a probe vehicle, is referred to as trace $T$.*

*Definition 5.* **Onroad distance** *If a probe vehicle traveling on a route $R$, at a constant speed $v$, takes time $t$, to reach point $B$ from point $A$. Then the onroad distance between point $A$ and $B$ on route $R$, is $(v \times t)$ .*

The *onroad* distance has been assumed to be the actual distance between any two points on a route and has been treated as base for all error calculations in this paper.

### B. Problem Formulation

Given a route $R$, specified in the form of a scaled map or a scaled aerial image and GPS traces $T = \langle T_1, T_2, \ldots, T_n \rangle$, each trace containing latitude $L_\alpha$, longitude $L_\beta$, and timestamp $t$, received from any vehicle $V$, that is traveling or traveled on $R$ ( i.e., both real time and history). The problem is to accurately calculate the *onroad* distance between any two traces $T_i$ and $T_{i+1}$, ∀ integers $i \ \epsilon \ [1, n)$.

**Problem Statement**: Given a route and a repository of real-time or historical GPS traces, of a vehicle on the given route we aim to develop a methodology that accurately calculates the distance traveled by the vehicle between any two traces using the simple and novel approach of *microsegmenting*.

## III. Our Approach

In response to the above stated problem, we propose a simple and novel method of *Microsegmenting*. The first step in this method is to divide the route in small segments of equal length, say $l$, i.e., the *microsegments* (Fig. 2) and each *microsegment* is given a unique ID, for example the ID can be the serial number of the *microsegment* from the start of the route.

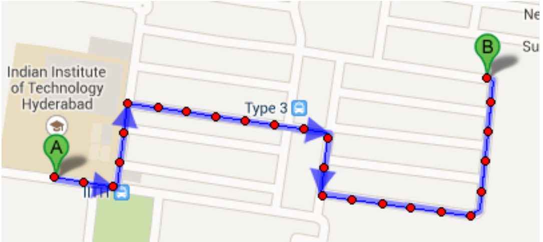

Fig. 2. Microsegmenting

The second step is matching the received GPS traces to *microsegments* (i.e., map matching) and each trace, say $T_i$, is then associated with a *microsegment* ID, say $I_i$, where $i$ refers to the serial number of the trace. The *onroad* distance and/or the average speed of the vehicle is then calculated in the third step of the method by the following formula :

$$D_i = [(I_i - I_{i-1}) \times l] \tag{1}$$

$$S_i = \frac{[(I_i - I_{i-1}) \times l]}{(t_i - t_{i-1})} \tag{2}$$

where, $D_i$ is calculated *onroad* distance traveled by the vehicle between the traces $T_{i-1}$ and $T_i$, $S_i$ is calculated average speed of the vehicle between the traces $T_{i-1}$ and $T_i$, $t_i$ is time stamp associated with the trace $T_i$.

Here it should be noted that, the *microsegments* can be of unequal length but as can be clearly noted that this would complicate the process of calculation and would also increase unnecessary overheads like maintaining the lengths of *microsegments*. Also, it should be noted that the length of *microsegments* directly affects the precision of calculations and computational overhead (e.g. map matching). So, smaller the length of *microsegments*, higher the precision and more will be the computational overhead.

### A. Implementation details and Tools Developed

As mentioned above, the proposed system involves three main modules: first is the *microsegmenting* module, second is the map matching module and final and the most simple is the calculating module.

In the map matching module many map matching algorithms are available [8]. The algorithm chosen in this module is of great importance as this is the only major overhead of the proposed method over the present method and is also the source of error if not performed correctly. For validation purpose we chose the most basic $O(n^2)$ algorithm to phase out any possibility of mistake in this stage of calculation.

The *microsegmenting* module is the preparatory module, as it is performed only once before the actual beginning of the process of the distance calculations. This module can be executed by physically going out on the route and gathering the GPS data but this method has some shortcomings. Firstly this method is not scalable, as the number of routes or the length of the route increases and secondly there is a huge scope for error as all the points are estimated individually, the error can creep in either via *onroad* distance calculation (for maintaining equal length) or via GPS coordinate estimation.

One more method for executing this module is by segmenting the route using the data available from Open Street Maps database(OSM). Uncontrolled and uneven length of route segments, difficulty in extracting the route data and unavailability of data for all routes are some of the shortcomings of this approach.

*1)* **The Microsegmenting App:** This application was primarily aimed at overcoming the above mentioned problems faced while implementing the *microsegmenting* module.

We developed a *microsegmenting* application (we call it as *microsegmentor*) for the purpose of evaluation of the proposed method. The application was developed in Python 2.7 and had a rudimentary GUI developed with the Tkinter Python library.

The application had three major components. The first part was aimed at defining the physical entity *route*, mathematically. This was achieved by approximating the route by a polyline (i.e., a set of lines formed by linearly joining consecutive points, $(p_i, p_{i+1})$, where, $i\epsilon[1, n)$, in a set of points, $P = \langle p_1, \cdots, p_n \rangle$, where $p_i = (x_i, y_i)$). In our implementation the set of points $P$, was obtained by tracing the route by clicking on it and with each click the canvas ( on which the image was loaded) coordinates were stored. Each consecutive pair of points when considered together form one of the line in the polyline.

The aim of the second part was to get segments of length $l$ (a parameter set by the user) on the polyline. Given a line segment we can easily partition it into smaller line segments of equal length $l$, by finding equidistant points, $(\langle q_1, \cdots, q_n \rangle$, where $q_i$ is canvas coordinate) on it (high school level coordinate geometry) and this can be repeated over all the line segments in the polyline. But if the length of a line segment is not an integral multiple of $l$, then the length of the last part would be less than $l$. As in Case 1 of Fig.3, this minuscule error when accumulated over all such line segments of the polyline, which would be many as possibility of getting line segment of length that is an integral multiple of $l$ is almost nil, would result in a noticeable error in the final calculations.

So let us consider, without loss of generality, line segment 1 $(p_i, p_{i+1})$ and line segment 2 $(p_{i+1}, p_{i+2})$. Let the set $\langle q_{11}, q_{12}, \cdots, q_{1n} \rangle$ be the equidistant points on line segment 1, such that $|\overline{q_{11}q_{1n}}| < |\overline{p_ip_{i+1}}|$ and $|\overline{q_{1i}q_{1i+1}}| = l$, for all integer $i$ in range $[1, n)$. Then as explained earlier if $|\overline{p_ip_{i+1}}| \neq ml$, for any integer $m$, then $|\overline{q_{1n}p_{i+1}}| < l$, say $l' = |\overline{q_{1n}p_{i+1}}|$, then when calculating the first point $q_2 1$ on line segment 2, find the point such that $|\overline{p_{i+1}q_{21}}| = l - l'$ and then repeat as usual. By iterating in this way, the error would occur only at the last part of the line segment 2 and the error would be less than $l$ due to the same reason explained earlier, as in Case 2 of Fig.3.

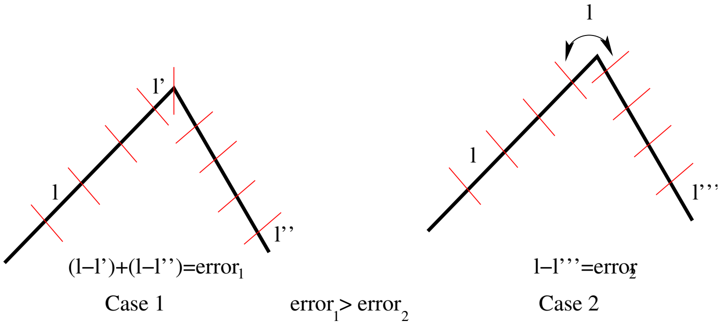

Fig. 3. Possible source of error in Microsegmenting

The last part of the application was to convert the canvas coordinates to GPS coordinates. The bearing calculated between any two canvas points will be same if calculated between their GPS coordinates and as the maps used were scaled, Euclidean distance between the points on the canvas could be easily calculated by scaling the distance to original Euclidean distance by multiplying it with ratio of scale value to scale length. Then by Eqn. 3 and Eqn. 4 [5] the canvas coordinates can be converted to GPS coordinates, $(\varphi_2, \lambda_2)$ :

$$\varphi_2 = asin(sin(\varphi_1) * cos(d/R) + cos(\varphi_1) * sin(d/R) * cos(\theta)) \tag{3}$$

$$\begin{aligned} \lambda_2 = \lambda_1 + atan2(&sin(\theta) * sin(d/R) * cos(\varphi_1), \\ &cos(d/R) - sin(\varphi_1) * sin(\varphi_2)) \end{aligned} \tag{4}$$

where, $\varphi$ is latitude, $\lambda$ is longitude, $\theta$ is the bearing (in radians, clockwise from north), $d$ is the distance traveled, $R$ is the earth's radius ($d/R$ is the angular distance, in $radians$)

It can be noted, the application of Eqn. 4, requires GPS coordinates of a reference point. This point can be any point on the canvas, we considered the starting point of the route as the reference point. The bearing and the distance are calculated for the starting point and the point of interest with respect to the canvas coordinates. The coordinates so obtained are then transformed to the GPS coordinates of the point of interest.

So applying the last step to the set of points obtained in second step gives the set of GPS coordinates of the microsegments.

## IV. Data Collection

The evaluation procedure, as explained in the next section, was designed as a two step procedure and hence the data collection was also divided in two stages. In the first stage the data was collected using high precision USB GPS Module connected to a Lenovo Thinkpad, which was deployed on a probe vehicle (Maruti Omni van). The GPS module was used at an update rate of 1 Hz. The data was collected in the form of a text file. Each trace of the data consisted latitude, longitude, time stamp and speed.

In the first stage the data was collected over two routes.

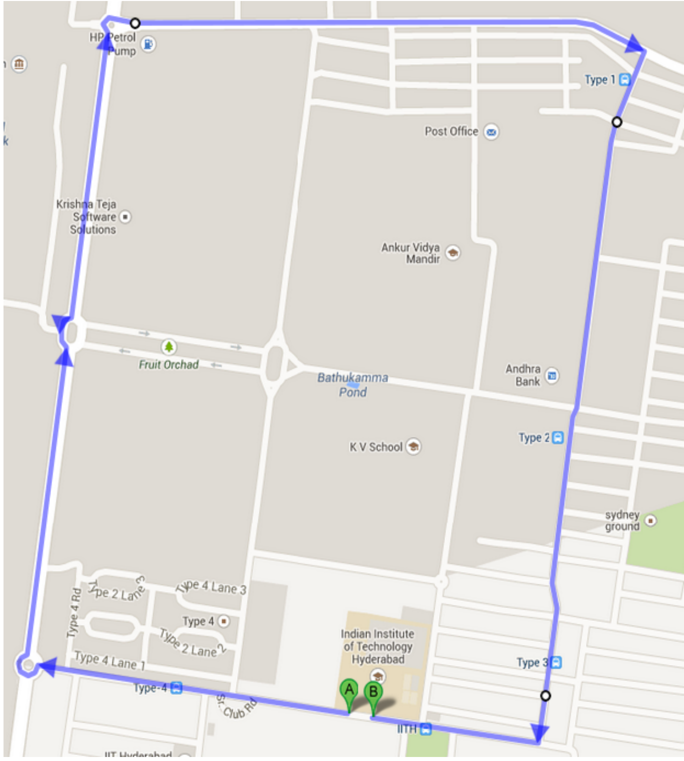

Fig. 4. Route 1

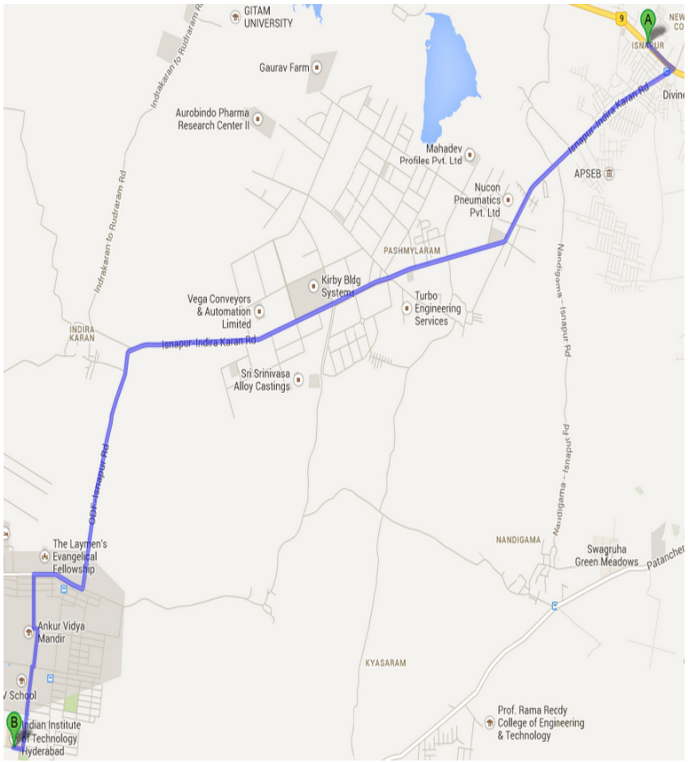
GITAM UNIVERSITY
Gaurav Farm
Aurobindo Pharma Research Center II
Mahadev Profiles Pvt. Ltd
Nucon Pneumatics Pvt. Ltd
APSEB
ISNAPUR
Isnapur-Indira Karan Rd
PASHMYLARAM
Kirby Bldg Systems
Vega Conveyors & Automation Limited
Turbo Engineering Services
INDIRA KARAN
Sri Srinivasa Alloy Castings
Indrakaran to Rudraram Rd
Nandigama - Isnapur Rd
ODF Isnapur Rd
The Laymen's Evangelical Fellowship
Ankur Vidya Mandir
School
Indian Institute of Technology Hyderabad
NANDIGAMA
Swagruha Green Meadows
KYASARAM
Prof. Rama Recdy College of Engineering & Technology
A
B

Fig. 5. Route 2

1) An auto-rickshaw route [3](Fig. 4)
(*onroad* length = 5.4 KM)
2) Andhra Pradesh State Road Transport Corporation (APSRTC) Bus route 502 [4](Fig. 5)
(*onroad* length = 11.7 KM)

For the second stage we collected data using Chicago Transit Authority (CTA) Bus Tracker API [3]. All buses under CTA are equipped with GPS devices. They use GPS data for real-time bus arrival information and for identifying buses on maps. A web interface and many third party mobile apps are available which uses their Bus tracker API.

For this particular experiment we used one trip data of CTA route 6 (Jackson Park Express). The data was only filtered for duplicate rows before using the data.

## V. Evaluation Procedure

To analyze the proposed method empirically and to quantify the improvements over existing method of distance calculation we devised a comprehensive evaluation strategy. The evaluation procedure was performed in two stages: simulated and real-world scenarios.

### *A.* Simulated Scenario

This section was aimed at analyzing the performance of proposed method of *microsegmenting* by comparing it with the *onroad* distance and also to identify the improvement, if any, over the distance method, in a simulated real-world scenario. This step can be further divided in five parts:

*1)* **Simulating real-world scenario:** The time difference between receiving of traces in real-world establishments and monitoring systems ranges generally between 60 to 120 seconds [3], [4]. Hence to simulate this scenario, the time difference between any two traces that were considered for calculation was a random number lying in the range [60,120]. Let say $T = \langle T_1, T_2, \cdots, T_n \rangle$ is the set of traces considered for the distance calculation and let $t_i$ be the timestamp associated with the trace $T_i$. Then $t_i$ and $t_{i+1}$ satisfy the relation,
$60 < t_{i+1} - t_i < 120$, $\forall$ integers $i \,\epsilon\, [1, n)$.

*2)* **Calculating the *onroad* distance:** As described earlier in data collection section of the paper the GPS data was collected at an frequency of 1 Hz *i.e.,* the speed data of the probe vehicle was available for each second of the test journey. For calculation of the *onroad* distance between any two points on the test route, the speed parameter (in m/s) of all the traces between the two points were added up to get the distance in corresponding units. (because, $distance(m) = speed(m/s) \times time(s)$ and if $time = 1s$ then $|distance| = |speed|$ ). The underlying assumption in the *onroad* distance calculation of the route is that the speed at which the probe vehicle travels remains constant over the period of one second. The above mentioned algorithm was iteratively applied to all consecutive pairs of traces $(T_{i-1}, T_i)$ in the set $T'$ and the results were stored.

*3)* **Distance calculation (*microsegmenting* method):** The *microsegmenting* method was used to calculate distance and the algorithm was repeatedly applied for all consecutive pairs of traces $(T_{i-1}, T_i)$ in the set $T'$, for calculating the distance between $T_{i-1}$ and $T_i$ and the results were stored.

[3] Here on referred to as route 1
[4] Here on referred to as route 2

*4)* **Distance Calculation (Distance method):** The distance between all consecutive pairs of traces $(T_{i-1}, T_i)$ in the set $T'$ were calculated using the Haversine Formula [5] and stored.

*5)* **Error calculation:** The Root Mean Square Error (RMSE) values for distances calculated by *microsegmenting* method and Distance method were computed with onroad distance as the reference.

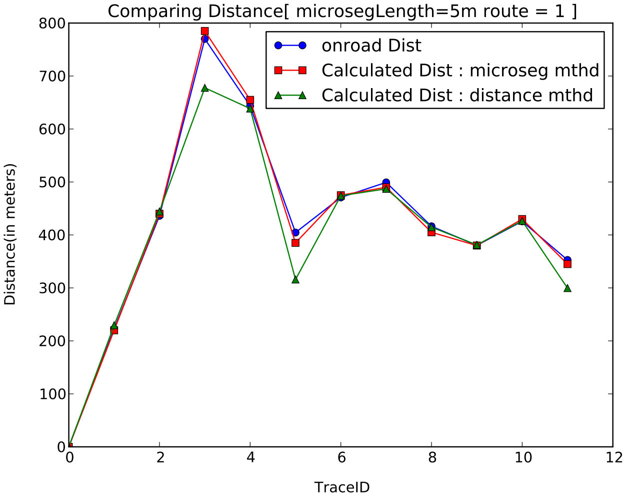

Fig. 6. Comparing of *onroad* and calculated distances of route 1

## B. Real-world Scenario

In this section, we aimed at quantifying the improvement in the distance calculation achieved by proposed method over the distance method. These methods were applied over a real world data set. CTA data set, $T''$, of route 6 was considered for this purpose. This step is further divided into two parts:

*1)* **Distance Calculation (*microsegmenting* method):** The *microsegmenting* method was used to calculate distance and the algorithm was repeatedly applied for all consecutive pairs of traces $(T_{i-1}, T_i)$ in the set $T''$, for calculating the distance between $T_{i-1}$ and $T_i$ and the results $R$ were stored.

*2)* **Distance Calculation (Distance method):** The distance between all the consecutive pairs of traces $(T_{i-1}, T_i)$ in the set $T''$ were calculated using the Haversine Formula [5] and the results $R'$ were stored.

*3)* **Distance Calculation (using OSM data):** In this approach the route was segmented using data points, $O$, obtained from the OSM database. The distance between all the consecutive pairs of traces $(T_{i-1}, T_i)$ in the set $T''$ were calculated and the results $R''$ were stored.

For this calculation the traces were first matched with the segments obtained by considering the OSM data points on the route. Let $T_{i-1}$ match to the segment $(o_{j-1}, o_j)$ and $T_i$ match to the segment $(o_k, o_{k+1})$. If $T_{i-1}$ and $T_i$ are matched to same segment then the distance between them is calculated directly using the Haversine Formula. Otherwise $|\overline{T_{i-1}T_i}|$ is calculated in two steps, firstly the length of all the segments between the points $o_j$ and $o_k$ are added, length of each segment is calculated using Haversine Formula [5]. Then the distance of $T_{i-1}$ and $o_j$ and the distance of $o_k$ and $T_i$, using the Haversine Formula, the distances so obtained are added to the result obtained in the previous step.

The root mean square values of the sets obtained by computing $r_i - r'_i$, $r_i - r''_i$, $r''_i - r'_i \ \forall \ i\epsilon[1, n]$, where $r_i \epsilon R$, $r'_i \epsilon R_i$ and $r''_i \epsilon R''$, were computed.

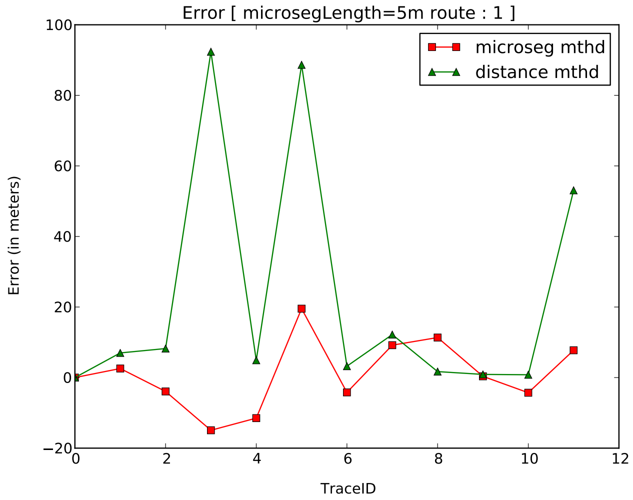

Fig. 7. Error in distance calculated for route 1

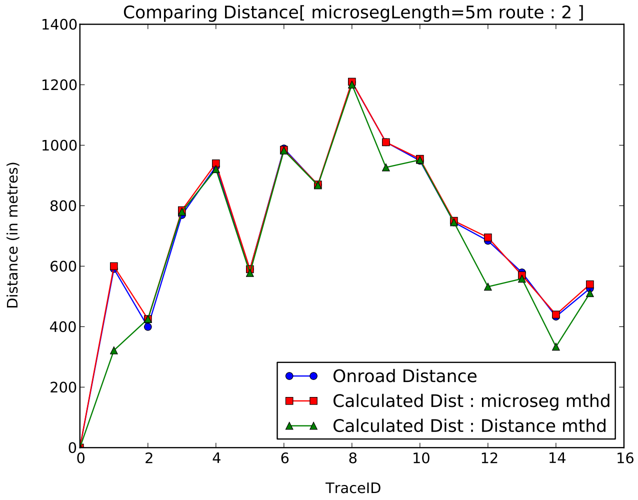

Fig. 8. Comparing of *onroad* and calculated distances of route 2

# VI. Performance Results

## A. Simulated Scenario Results

As the results were calculated with random inputs, so a range of RMSE values were observed. The range presented was calculated over hundred trials. The plots presented are the results of one of the trial. The RMSE value for the microsegmenting approach was found to lie in range (8m,15m). Whereas the RMSE value for the distance method was found to lie in range (57m,109m). The plots of the distance calculation

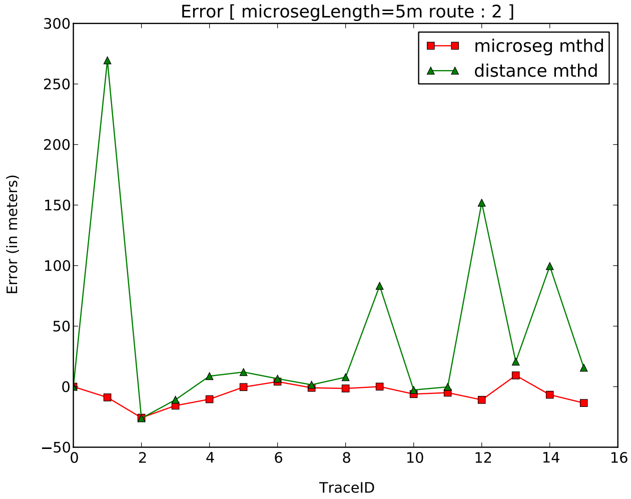

Fig. 9. Error in distance calculated for route 2

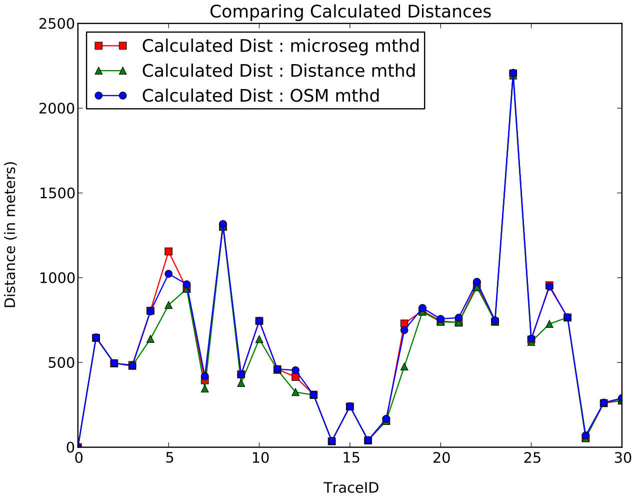

Fig. 10. The distances calculated on CTA route 6

results, shown in Figs. 6 and 8 show that the distance method in most cases underestimates the distance between traces and hence instantiates the hypothesis presented in the introduction. Figs. 7 and 9 compare the errors in distance calculations. Going with the trend of RMSE values these plots clearly show that the proposed method significantly out performs the distance method at most of the trace points.

### B. Real-world Scenario Results

The root mean square difference value of the distances for

- the microsegmenting method and general method is 93.142 m.
- the microsegmenting method and the method using OSM[5] data is 28.068 m.
- the method using OSM data and the general method is 79.298 m.

A closer inspection of the plot in Fig. 10 confirms the hypothesis that the distance method underestimates the distance between two traces.

The impact of the improvement in the distance calculation can be judged by the fact that, theoretically the travel time prediction improves by approximately 22 seconds between each pair of traces, by assuming the average traffic speed as 4.16 m/s (15 Kmph) [1], [2] and considering the root mean square difference value as an average improvement.

## VII. Conclusions and Future work

In this work, we proposed a novel method of *Microsegmenting* for addressing the *Displacement problem*. We conducted a two-staged experiment to empirically analyze the performance of the proposed method and determine the improvement over the existing methods. While the first stage served as the testing phase, the second stage provided a glimpse of improvement that can be achieved by replacing existing techniques with the proposed one. The results obtained were in accordance with the hypothesis and the proposed technique of *microsegmenting* showed a significant improvement over the distance method. The evaluation procedure was not based on live data but the proposed method can also be similarly applied in real-time setups.

In our present implementation, the vertices of the polyline used to approximate the route are obtained manually, in future we aim to automate this process.

## Acknowledgment

This work was supported by the Deity, Govt of India (Grant No. 13(6)/2010CC&BT).

## References

[1] Average CTA Bus Speed. *http://www.transitchicago.com/assets/1/brt/text_of_we_ash_inforgraphic.txt*

[2] Average Hyderabad Traffic Speed. *http://cseindia.org/node/1792*

[3] CTA Bus Tracker. *http://www.transitchicago.com/developers/bustracker.aspx*

[4] MBTA Bus Tracker. *http://www.nextbus.com/predictor/stopSelector.jsp?a=mbta*

[5] Haversine Formula and Destination point given distance and bearing from start point. *http://www.movable-type.co.uk/scripts/latlong.html*

[6] S. Sananmongkhonchai, P. Tangamchit and P. Pongpaibool. "Road Traffic Estimation from Multiple GPS Data Using Incremental Weighted Update", *in Proc. of ITS Telecommunications 2008*, pp. 62-66, October 2008.

[7] Arvind Thiagarajan, James Biagioni, Tomas Gerlich and Jakob Eriksson. "Cooperative Transit Tracking using Smart-phones*", *in Proc. of ACM SenSys 2010*, November 2010.

[8] Fernando Torre, David Pitchford, Phil Brown and Loren Terveen."Matching GPS Traces to (Possibly) Incomplete Map Data: Bridging Map Building and Map Matching", *in Proc. of ACM SIGSPATIAL GIS 2012*, November 2012.

[5] Due to the unavailability of OSM data for route 1 and route 2 similar comparisons were not possible in Simulated Scenario