VLOCI: Using Distance Measurements to Improve the Accuracy of Location Coordinates in GPS-Equipped VANETs

Farhan Ahammed[1], Javid Taheri[1], Albert Y. Zomaya[1], and Max Ott[2]

[1] School of Information Technologies,
The University of Sydney, NSW 2006, Australia
faha3615@it.usyd.edu.au
[2] NICTA, Australia, Australian Technology Park,
Level 5, 13 Garden Street, Eveleigh NSW 2015, Australia
max.ott@nicta.com.au

Abstract. Many vehicles rely on the Global Positioning System (GPS) to compute their locations. The inaccuracy of GPS devices means sometimes vehicles believe they are located in different lanes or roads altogether. Vehicular Ad Hoc Networks (VANETs) allow vehicles to communicate with each other using wireless means and thus connect them in a very dynamic wireless network. The algorithm **V**ANET **LOC**ation **I**mprove (VLOCI), proposed in this work, uses VANETs and distance measurements taken by each vehicle to improve the location estimates provided by all GPS devices. VLOCI is shown to perform efficient when erroneous distance measurements are present in the environment/computations.

Keywords: vehicular ad hoc networks, localization, GPS, distance measurements, location improve/refinement.

1 Introduction

VANETs are types of mobile networks—where the nodes are vehicles. The vehicles are equipped with wireless communication devices allowing them to transmit and share real-time information. With this information vehicles and drivers will have up-to-date information regarding the state of traffic, allowing them to avoid congested and other abnormally affected areas. VANETs are dynamic with vehicles travelling at speeds up to, and in excess of, 100 km/h. This leads to ever-changing wireless connections between vehicles resulting in some dense (on some city roads) and sparse (on country roads) areas which change over time (some city roads are dense only during certain hours of the day).

Many vehicles are nowadays equipped with GPS devices and it is quite possible that most, if not all, vehicles will have these devices as well in the future. GPS devices are accurate to within 10 metres [1]—more than the length of most family cars—resulting in situations where the GPS device incorrectly places its vehicles on the wrong road. Obtaining more accurate coordinates (position

P. Sénac, M. Ott, and A. Seneviratne (Eds.): MobiQuitous 2010, LNICST 73, pp. 149–161, 2012.

estimates) allows the vehicles to construct more precise models of their local traffic conditions.

With increased accuracy and better models, accidents can be prevented. Multi-car ‘pile-ups’ can be avoided if vehicles know immediately that other vehicles further in front are stopping suddenly or skidding. Although some sensors are already providing information about the vehicles directly in front and around the vehicle—VANETs can be used to provide information about vehicles further away. For example, when a vehicle detects a dangerous pot hole or other situation on the road, the exact co-ordinates of the problematic area can be immediately passed on to nearby vehicles. VANETs can be used to increase driver safety on the roads, but accurate coordinates is required for all vehicles—some drivers may incorrectly assume an accident or other incident is occurring on the wrong road.

This paper will look at using VANETs to improve on the position estimates provided by the GPS devices. Every vehicle can provide their position estimate to all vehicles within broadcasting range. It is also assumed every vehicle can measure the distance between them and other vehicles using already existing sensors/equipments [2,3]. When all vehicles combine the collected information, the algorithm LOCI can be used to adjust the GPS estimated position into a more accurate one.

An overview of previous work found in literature is presented in Section 2. Section 3 introduces the notation and defines the problem addressed in this paper. The method used to solve the problem is described and the VLOCI algorithm is presented in Section 4. Section 5 describes the simulations performed to test the devised algorithm. An discussion of the simulation results and concluding remarks are presented in Section 6 and 7, respectively.

2 Related Work

There does not seem to be much work in literature with the idea of improving location estimates in VANETs. There are algorithms designed to take advantage of some nodes that have GPS, or some other positioning, functionality to allow all nodes to compute their location.

Priyantha et al. proposed a technique called *anchor-free localization* (AFL) of providing localization to wireless sensor networks [4]. Their algorithm is decentralised where each node starts with a random initial coordinate assignment, and modifies its location estimates based on local distance measurements. The only information each node collects is the relative distance to its neighbouring nodes. With this information, the nodes construct a graph with the edges at the measured length/weight. A *mass-spring* based optimization is used to adjust the edge lengths of the graph. The edge lengths are adjusted based on the difference between the measured distances between neighbouring nodes and the corresponding computed distances in the constructed graphs.

Barani and Fathy [5] looked at the problem where not all vehicles are equipped with GPS devices or cannot receive signals from the GPS satellites. In their approach, the vehicles first attempt to find three neighbours within one-hop distance. If there is only one or two neighbours within one-hop, then distance information from neighbours within two-hop distance is used. They found when more than 40% of vehicles are equipped with GPS, most of the vehicles in the network will have at least three GPS-equipped neighbours. From these neighbours, methods such as trilateration [6] can be used to compute their location. Here the problem is not to improve GPS coordinates, but to provide localization to those unable to use the GPS.

Liu and Lin [7] improved location accuracy using the least-squares technique. Their technique is applied to cellular networks, where base transceiver stations are established and within communication range. Results obtained in field trials are compared to the co-ordinates computed by GPS devices. Consequently, these tests assume the GPS devices provide sufficient accuracy. The mean error produced, using their technique, is within the range of 200 metres: extremely large for use in VANETs.

Xu et al. [8,9] have developed an algorithm for use in wireless sensor networks where some of the sensors are located indoors—preventing detection of the signals from the GPS satellites. In their approach, only a subset of nodes are GPS-equipped, while other anchor nodes also know their true locations. The remaining nodes are able to compute their location based on distances between the GPS nodes and the GPS satellites, and distances between the GPS nodes and anchor nodes; as a result, not every node uses GPS to estimate their location. The DV-Hop algorithm is used by the remaining nodes for localization. Other techniques also exist that utilize the DV-Hop algorithm, where its accuracy is significantly proportional to the network density. Therefore, this cannot be used in VANETs where the network density continuously changes. Their tests results show the location error for the non GPS nodes achieving the same range as the nodes which use the GPS. However their algorithm does not improve further on this location error.

Some alternative solutions [10,11,12] are applied in situations where a subset of nodes, usually termed anchor nodes or base stations, have knowledge of their own positions. The remaining nodes then communicate with the anchor nodes to determine their locations as they have no other method to estimate their locations. Similarly, Benslimane [13] addresses the situation where not all vehicles are equipped with GPS devices, or some cannot obtain data from their GPS devices and need to collaborate with the GPS-equipped vehicles to determine their locations.

No work have so far been found in literature where all nodes are equipped with GPS devices and the problem addressed is finding ways to improve these estimates.

3 Problem Statement

The network in question is a VANET, thus the nodes are in fact vehicles. The set of all vehicles will be denoted V. The number of vehicles in the network is

$N = |V|$. Given any particular vehicle $n_i \in V$, any other vehicle that n_i can send and receive messages from are deemed its neighbour. The set of all neighbours of n_i is denoted $nbrs(n_i)$ and the number of neighbours of n_i is $m_i = |nbrs(n_i)|$.

Each vehicle n_i is located at $p_i = (x_i, y_i)$ (true location) and, due to inaccurate measurements, believes it is located at $\widehat{p}_i = (\widehat{x}_i, \widehat{y}_i)$ (computed/estimated location). The location error for n_i is the distance between its true and computed location $\delta_i = \|p_i - \widehat{p}_i\|$. It is assumed that every vehicle is able to take distance measurements between them and other vehicles. The true distance between vehicles n_i and n_j is denoted $d_{i,j}$, while the distance measured by n_i between itself and vehicle n_j is given as $\widehat{d}_{i,j} = \varepsilon \cdot d_{i,j}$, for some $\varepsilon \in \mathbb{R}$. Since each vehicle is assumed to have their own measuring device, it is also assumed that $\widehat{d}_{i,j} \neq \widehat{d}_{j,i}$ because $\widehat{d}_{i,j}$ is the distance measured by n_i while $\widehat{d}_{j,i}$ is the distance measured by n_j.

A metric used to gauge the performance of the localisation algorithm of a network is the *network location error* (Definition 1). For each vehicle n_i, its location error is already defined as δ_i. The network location error is the average of every vehicle's location error.

Definition 1 (Location Error). *Given a network of vehicles V, where each vehicle n_i believes it is located at $\widehat{p}_i$, while its true location is p_i. The* location error *of the network (E_V) is defined as*

$$E_V = \frac{1}{n} \sum_{n_i \in V} \delta_i = \frac{1}{n} \sum_{n_i \in V} \|p_i - \widehat{p}_i\|$$

Using the definition of network location error, the problem of localization can be formulated as shown in Definition 2.

Definition 2 (Localization). *Given a network of vehicles V, the goal of* localization *is for every vehicle $n_i \in V$, located at $p_i = (x_i, y_i)$, to compute its position $\widehat{p}_i = (x'_i, y'_i)$ such that the network location error is minimised.*

The problem of *Location Improvement* (Definition 3) is addressed in this paper. The aim is to find a method of adjusting every vehicle's current estimated position $\widehat{p}_i$ such that the location error is reduced.

Definition 3 (Location Improvement). *Let V be a network of vehicles. Assume that every vehicle n_i has computed its own estimated position $\widehat{p}_i$. The problem of* Location Improvement *is to find a function f which modifies some or all of the vehicle's estimated position such that the location error of the modified network has minimised.*

4 Method and Algorithm

4.1 Network Topology

In many situations, also addressed in literature, the topology of the network in question is modelled as a random topology with little or no restrictions on

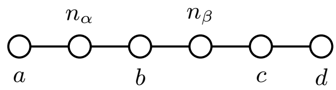


Fig. 1. The network topology within a single lane of a road. Node n_α has three neighbours $\{a, b, n_\beta\}$ and node n_β has four neighbours $\{n_\alpha, b, c, d\}$.

the nodes' locations: a non-realistic issue in VANETs sometimes. For example, within a lane of a road, the local network topology is already known and all vehicles in the same lane are lined up one behind another. Figure 1 depicts the 'lane topology.'

In this paper the vehicles are assumed to be travelling in one lane and in the same direction. In this model a co-ordinate scheme is used where $y_i = y_j$ for all $n_i, n_j \in V$. Each vehicle $n_i \in V$ has an estimate of their position $\hat{p}_i$ and this information is periodically sent to its neighbours. Using their distance-measuring device, each node measures the relative distance $\widehat{d_{i,j}}$.

Road-side infrastructures (permanent, static nodes) can be used to provide vehicles the necessary linear transformation required to convert the local co-ordinates into their global counterparts.

4.2 Accuracy of Distance Measurements

The accuracy of the distance measurement devices refers to the values of $\widehat{d_{i,j}}$ in the simulation and how far they differ from true distances $d_{i,j}$. Two techniques used to measure distance are time-of-arrival and received signal strength [14]. The accuracy of these techniques can be modelled such that as the distance to be measured increases, the accuracy of the measurements taken also decreases [15,16]. This is how distance measuring is modelled in this paper. The statement 'the accuracy is set to α metres' means that $\widehat{d_{i,j}} = E \cdot d_{i,j}$ for all $n_i, n_j \in V$, where $E \sim N(1, \alpha^2)$; $N(1, \alpha^2)$ refers to the Gaussian distribution with mean 1 and standard deviation α. Thus, if a vehicle is $d_{i,j}$ meters away, then roughly 68% of the distance measurements $\widehat{d_{i,j}}$ lie within the range $(1-\alpha)d_{i,j} < \widehat{d_{i,j}} < (1+\alpha)d_{i,j}$.

Using this model, vehicles further away have a larger probability of producing more erroneous data. The further away a neighbouring vehicle is (i.e. the larger the value of $d_{i,j}$), the larger the error range of the measured distance. Measuring the distance to vehicles within close range is more accurate than measuring the distance to those further away.

4.3 Neighbourhood Size

Since the network is modelled as a single lane, the size of the network is straightforward. There are N vehicles, two are at the end points while the rest form the line. This leads to the definition of a vehicle's set of neighbours. Every vehicle has at most $2M$ neighbours (for some $M \in \mathbb{Z}$) they can communicate with; the M closest vehicles in front and the M vehicles in behind. That is, M defines the maximum number of neighbours in either direction. Figure 1 gives an example when $M = 2$. Here, every vehicle has at most 4 neighbours. Vehicle n_β has the maximum of four neighbours—two in front and two behind—while vehicle n_α only has three.

In this paper, the statement 'the half-neighbourhood size is M' means every vehicle can communicate with up to $2M$ vehicles.

4.4 Computing Position Estimates

Each vehicle receives messages from its neighbours containing their estimated positions. Additionally, each vehicle can measure the distance between itself and its neighbours. To counter the variance in erroneous distance measurements, multiple measurements can be taken for which the average can be used as the final distance measurement.

If vehicles n_j and n_i are neighbours, then n_i can obtain n_j's estimated position $\widehat{p}_j$ and the measured distance $\widehat{d}_{i,j}$. Using these two pieces of information, vehicle n_i can calculate its position assuming that n_j's estimated position is correct, i.e.

$$\widehat{p}_i^j = (\widehat{x}_i^j, \widehat{y}_i) = (\widehat{x}_j \pm \widehat{d}_{i,j}, \widehat{y}_j) \tag{1}$$

remembering that since in this model $y_i = y_j$, the vehicles can assume $\widehat{y}_i = \widehat{y}_j = 0$. Converting these co-ordinates into global co-ordinates requires a simple linear transformation. For generality define $\widehat{p}_i^i = \widehat{p}_i$ and $\widehat{d}_{i,i} = 0$. The reason for the plus/minus term in Equation 1 is that vehicle n_j may be relatively in front or behind vehicle n_i and so the value of $\widehat{x}_i^j$ should be adjusted accordingly.

Now given the set of neighbours $nbrs(n_i)$, vehicle n_i can construct the set $\{\widehat{p}_i^j \mid n_j \in \{n_i\} \cup nbrs(n_i)\}$. With this set of co-ordinates, an average can be computed to calculate a new estimate for $\widehat{p}_i$. A weighted average function $w : V \to \mathbb{R}$ is defined in this work to estimate $\widehat{x}'_i$ as follows:

$$\widehat{x}'_i = \frac{\sum_{\{n_i\} \cup nbrs(n_i)} w(n_k) \cdot \widehat{x}_i^k}{\sum_{\{n_i\} \cup nbrs(n_i)} w(n_k)} \tag{2}$$

This value becomes the new estimated x co-ordinate for vehicle n_i. Figure 2 shows graphically how an average estimate is used to compute the new estimated position from the cluster of points.

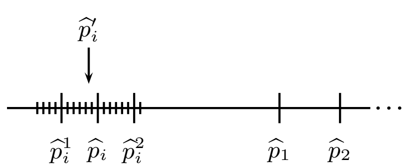


Fig. 2. The cluster of points from the computed set $\{\widehat{p}_i^i, \widehat{p}_i^1, \widehat{p}_i^2, \ldots, \widehat{p}_i^{m_i}\}$

This weighted average position becomes the new estimated position for vehicle n_i, and this is the coordinate transmitted to its neighbours in the next turn/iteration.

4.5 The Weight Function

To compute the weighted average, a weight function w needs to be defined. As the measured distances become more inaccurate the further away neighbouring vehicles are, it was decided the weights should be influenced by the measured distance $\widehat{d}_{i,j}$. For example, vehicles should put more weight to the estimated position constructed from data obtained from a vehicle that is 20 metres away, compared to a vehicle that is 400 metres away since there is more error involved in the calculations involving the latter case.

The weight function must therefore be inversely proportional to the measured distance. Two functions were considered: (1) inverse functions of the form $g(x) = A/x^B$ and (2) inverse exponential functions of the form $f(x) = Ae^{-x^2/B}$. In both cases, $x = \widehat{d}_{i,j}$. Thus the weight function takes the distance $\widehat{d}_{i,j}$ as the parameter and gives more weight to vehicles closer to n_i. The latter form f was chosen over the alternate inverse function because results have shown g generally decreases too quickly and better results were achieved using a function of the form of f.

4.6 VLOCI Algorithm Overview

The concepts explained in the previous section are used to design a location-improvement algorithm that is both (1) distributed as multiple vehicles collaborate to achieve the goal of improving every vehicle's position estimates, and (2) scalable as the number of vehicles able to participate is not restricted.

An iteration, performed by vehicle n_i is the process of receiving messages from its neighbours then updating its estimated position. The number of iterations hence determines the number of times every vehicle updates its estimated position as shown in Algorithm 3

1. Transmit a message containing the current estimated position ($\widehat{p}_i$).
2. Wait to receive messages from neighbouring vehicles. The received messages should contain the estimated position of the vehicle that sent the message.
3. For each vehicle a message was received from, measure the distance to it. The vehicle that is taking the measurements should take multiple measurements (D times) and use the average as the final measured distance. The value of D is based on the technique used to obtain distance measurements. Smaller values of D can be used with more accurate distance measuring devices.
4. Vehicle n_i now knows the values of $\{\widehat{p}_j, \widehat{d}_{i,j}\}$ for each of its neighbours. Equation 1 is then used to compute possible co-ordinates it could be located at. A set of these possible position estimates $\{\widehat{p}_i^i, \widehat{p}_i^1, \widehat{p}_i^2, \ldots, \widehat{p}_i^{m_i}\}$ is then constructed.
5. The weighted average of the set of possible position estimates (Equation 2) is calculated. This final co-ordinate becomes the new estimated position.

while $iterationCount < I$ **do**
    `transmitMessage(`$\widehat{p_i}$`)` M contains the received messages

    `// We have each `n_j`'s computed position. Now measure the distance from them.`

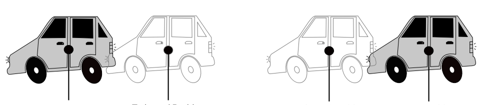
    `// Wait for `λ` messages`
    `wait(`M`)` **foreach** $M_j \in M$ **do**
        `// Use avg of `D` measurements.`
        $\widehat{d}_{i,j} = \frac{1}{D}\sum_{k=1}^{D}$`takeDistMeas(`$n_j$`)`

        if $\widehat{x}_i < \widehat{x}_j$ **then**
            $\widehat{p}_i^j = (\widehat{x}_i^j, \widehat{y}_i) = (\widehat{x}_j - \widehat{d}_{i,j}, 0)$
        else
            $\widehat{p}_i^j = (\widehat{x}_i^j, \widehat{y}_i) = (\widehat{x}_j + \widehat{d}_{i,j}, 0)$
        end
    end
    `// Now compute the weighted average of all the probable co-ordinates of `n_i`.`
    $\widehat{x}_i' = \frac{\sum w(n_k)\cdot \widehat{x}_i^k}{\sum w(n_k)}$ $\widehat{y}_i' = 0$
end

Fig. 3. Algorithm VLOCI: updating position estimates (vehicle n_i)

4.7 Skewed Positions

Additional scenarios were devised where the initial estimated positions were predetermined. A vehicle n_i is said to be *skewed to the left*, if $\widehat{x}_i < x_i$ and skewed to the right when $\widehat{x}_i > x_i$. Figure 4 gives a examples of this concept. The number of vehicles initially skewed to the left is γ. The remaining $N - \gamma$ vehicles are then positioned skewed to the right.



(a) Skewed to the right

(b) Skewed to the left

Fig. 4. Example of cars "skewed to one side"

4.8 Metric for Measuring Performance of VLOCI

The metric used to gauge the performance of VLOCI is the location error (Definition 1). The location error is the average distance between a vehicle's computed position and its actual position at the current point in time. The smaller the location error, the better the computed model of the vehicle's locations.

Assessing how the computed positions of all the vehicles would improve over time was the goal of the tests. The vehicles were simulated to receive GPS measurements only once and at the beginning. Afterwards, only VLOCI was used to further improve the position estimates.

5 Simulation Results

Simulations were performed to assess the effect of two variants in VLOCI: the accuracy of the distance measuring devices α (Section 4.2) and the neighbourhood size $2M$ (Section 4.3). The simulations were run with the parameters shown in Table 1. The network itself was static and the vehicles were set to be stationary. The weight function coefficients were chosen empirically. When vehicles were approximately 50m away the error in distance measurements were not detrimental to VLOCI's performance. The coefficients of the weight function were chosen to reflect this. It was the aim of the simulations to find how the neighbourhood size affects the rate of improvement of VLOCI and for what values of α does VLOCI still perform adequately.

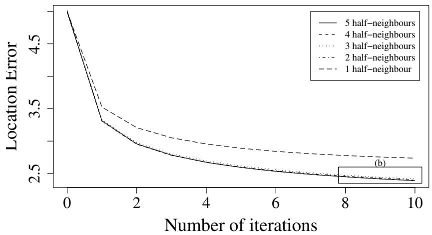
Table 1. The simulation parameters used

Parameter	Value
Network size (N)	10
Half-neighbourhood size (M)	1–10
Distance measurement accuracy (α)	0–30%
No. distance measurements taken (D)	5
Distance between vehicles	20 metres
Weight function f coefficients	$A = 100$, $B = 550$
No. iterations (I)	10
No. tests per scenario	10 000
No. Vehicles skewed to left (γ)	0,1,2,3,4,5

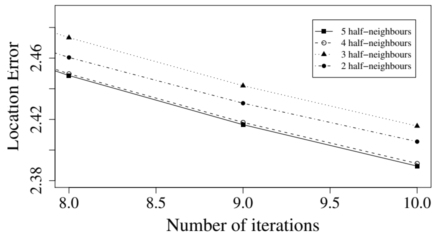
5.1 Results

Figure 5a shows how the neighbourhood size effects VLOCI's performance, for half-neighbourhood sizes (M) of 1–5. To give a clearer picture of how the different curves compare after 7 iterations, Figure 5b shows the same graph for iterations 8–10 inclusive. The distance measurement error (α) was set to zero for these simulations to model errors only arising from the initial GPS measurements.

When the half-neighbourhood size ranges from 5–10, Figure 6 shows how the location error reduces, for iterations 8–10 inclusive. Again $\alpha = 0$. The best results occurred when the half-neighbourhood size is set to $M = 7$. This is the value M was fixed at when testing VLOCI's ability to handle errors in distance measurements.



(a) Iterations 1–10.



(b) Iterations 8–10.

Fig. 5. The effects of local neighbourhood size on location error. Half-neighbourhood size 1–5. $\alpha = 0$.

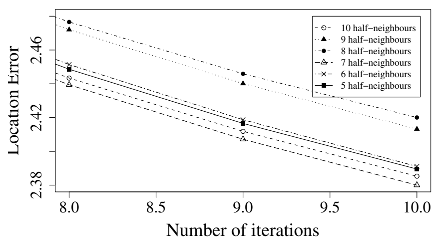


Fig. 6. The effects of local neighbourhood size on location error. Half-neighbourhood size 5–10. $\alpha = 0$. Magnified for iterations 8–10.

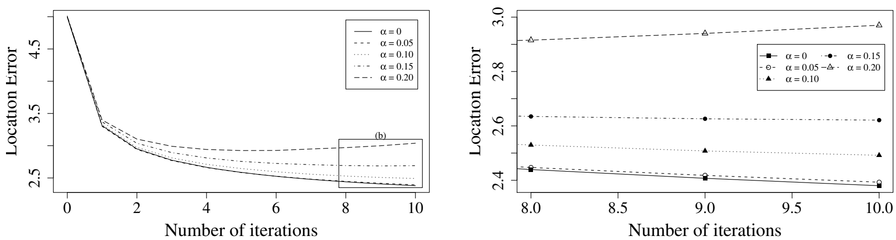
The graphs in Figure 7 show how VLOCI performs when errors are incorporated in distance measurements. For the value of $\alpha > 0.20$ the location error began to increase after the fifth iteration. When $\alpha < 0.20$ the location error still decreases during the first 10 iterations.

One of the reasons explaining why the location error begins to increase after some time is due to how the vehicles are skewed over time. Eventually too many cars believe they are located on the same side of their actual position. That is, too many cars are skewed on the same side. Figure 8 shows that once all the vehicles are on the same side (skewed to the right) the location error no longer improves. Here $\alpha = 0$.

The effectiveness of VLOCI when $\alpha = 0.1$ and 0.20 and with multiple cars skewed to one side is shown in figures 9a and 9b respectively.

6 Discussion and Analysis

The results show a definite improvement on the location error. Figures 5a, 5b and 6 show that with accurate distance measurements (i.e. $\alpha = 0$) VLOCI does indeed improve the average location error. When $\alpha = 0$ and the half-neighbourhood size is set to $M = 7$, after 10 iterations the average location error reaches a value of 2.38 meters—an improvement of 52%. With the accuracy set



(a) Iterations 1–10.

(b) Iterations 8–10.

Fig. 7. Effects of the accuracy of distance measurements on location error. Half-neighbourhood size 7.

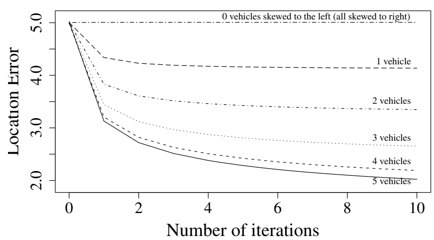
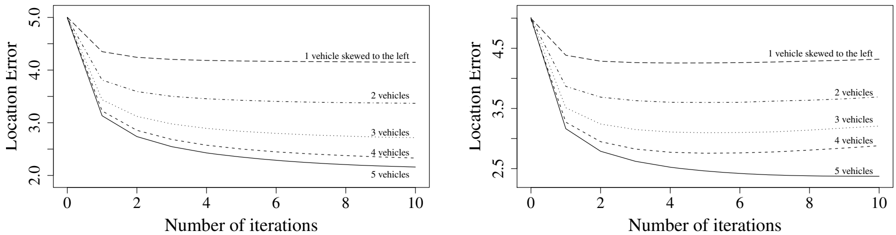


Fig. 8. Effects of skewness in initial positions. Half-neighbourhood size 7. $\alpha = 0$.

to $\alpha = 0.20$, the average location error improves by 41%, before increasing from the fifth iteration. This is still acceptable as there is an overall improvement on the location error after 10 iterations and due to the mobile nature of VANETs, the vehicles sometimes cannot perform time-consuming operations restricting the number of iterations possible.

Figure 7 shows that VLOCI still improves the location error when inaccurate distance measurements are introduced. For values of $\alpha < 0.20$ the location error, after 10 iterations, has improved to values less than 2.7 metres. If VLOCI is set to iterate 10 times, this value of α is an upper bound on the average location error.

Figures 8, 9a and 9b shows how VLOCI is affected when the initial position estimates of the vehicles, relative to their actual positions, is set to have a portion of vehicles skewed to one side (Section 4.7). When all the vehicles are skewed to one side, the location error does not noticeably increase or decrease within the first 10 iterations. This is because while the vehicles have adjusted their estimated positions such that the distance between them is consistent with the true distance, they are still skewed to the same side. There needs to be at least one vehicle skewed to the other side to 'pull' the other vehicles from one side of their true position to the other. As expected, the best results occur when approximately half the vehicles are skewed to one side (and the other half skewed to the other side). Even when only one vehicle is skewed to one side, the average location error still improves.



(a) $\alpha = 0.1$ (b) $\alpha = 0.2$

Fig. 9. Effects of skewness in initial positions. Half-neighbourhood size 7. $\alpha = 0.1$.

7 Conclusions

The VLOCI algorithm presented in this paper is shown to improve every vehicle's initial position estimate. Assuming the vehicles are connected via a VANET and are equipped with a distance measuring device, along with a GPS device to provide the initial position estimate, VLOCI is still able to improve locations estimation when erroneous distance measurements are included in the computations. The effect of skewness is also shown in this paper. Only in the situation where all vehicles are skewed to one side does VLOCI not improve the average location error. For the remaining situations, VLOCI still reduces the average location error.

References

1. LaMarca, A., Chawathe, Y., Consolvo, S., Hightower, J., Smith, I., Scott, J., Sohn, T., Howard, J., Hughes, J., Potter, F., Tabert, J., Powledge, P.S., Borriello, G., Schilit, B.N.: Place lab: Device positioning using radio beacons in the wild. In: Gellersen, H.-W., Want, R., Schmidt, A. (eds.) PERVASIVE 2005. LNCS, vol. 3468, pp. 116–133. Springer, Heidelberg (2005)
2. Fleming, W.J.: Overview of automotive sensors. IEEE Sensors Journal 1, 296–308 (2001)
3. Fölster, F., Rohling, H.: Data association and tracking for automotive radar networks. IEEE Transactions on Intelligent Transportation Systems 6, 370–377 (2005)
4. Priyantha, N.B., Balakrishnan, H., Demaine, E., Teller, S.: Anchor-free distributed localization in sensor networks. In: 1st International Conference on Embedded Networked Sensor Systems, SenSys 2003 (2003)
5. Barani, H., Fathy, M.: An algorithm for localization in vehicular ad-hoc networks. Journal of Computer Science 6(2), 168–172 (2010)
6. Thomas, F., Ros, L.: Revisiting trilateration for robot localization. IEEE Transactions on Robotics 21, 93–101 (2005)
7. Liu, B.-C., Lin, K.-H.: Enhanced location accuracy for hyperbolic positioning technique based on sssd measurements in wireless cellular networks via least-square algorithm. In: Global Telecommunications Conference, GLOBECOM 2006, December 27, pp. 1–5. IEEE, Los Alamitos (2006)

8. Xu, L., Deng, Z., Ren, W., Wang, H.: A location algorithm integrating gps and wsn in pervasive computing. In: Third International Conference on Pervasive Computing and Applications, ICPCA 2008, vol. 1, pp. 461–466 (2008)
9. Zou, D., Deng, Z., Xu, L., Ren, W.: Seamless lbs based on the integration of wsn and gps. In: International Symposium on Computer Science and Computational Technology, vol. 2, pp. 91–96 (2008)
10. Savarese, C., Rabaey, J.M., Beutel, J.: Locationing in distributed ad-hoc wireless sensor networks. In: Proceedings of 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2001), vol. 4, pp. 2037–2040 (2001)
11. Cheng, X., Thaeler, A., Xue, G., Chen, D.: TPS: a time-based positioning scheme for outdoor wireless sensor networks. In: Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 2004, 7–11, vol. 4, pp. 2685–2696 (2004)
12. Lazos, L., Poovendran, R., Čapkun, S.: ROPE: robust position estimation in wireless sensor networks. In: Fourth International Symposium on Information Processing in Sensor Networks, IPSN 2005, pp. 324–331 (April 2005)
13. Benslimane, A.: Localization in vehicular ad hoc networks. In: Proceedings of Systems Communications, pp. 19–25 (August 2005)
14. Boukerche, A.: Algorithms and Protocols for Wireless, Mobile Ad Hoc Networks. Wiley-IEEE Press (2008)
15. Bellusci, G., Janssen, G.J.M., Yan, J., Tiberius, C.C.J.M.: Modeling distance and bandwidth dependency of TOA-based UWB ranging error for positioning. Research Letters in Communications, 1–4 (January 2009)
16. Powell, S., Shim, J. (eds.): Wireless Technology: Applications, Management, and Security. Springer, Heidelberg (2009)