VCLT: An Accurate Trajectory Tracking Attack based on Crowdsourcing in VANETs

Chi Lin[1(✉)], Kun Liu[1], Bo Xu[1], Guowei Wu[1], and Jing Deng[2]

[1] School of Software,
Dalian University of Technology
Dalian 116620, China
[2] Department of Computer Science
University of North Carolina at Greensboro
NC 27412, U.S.A.

Abstract. We investigate trajectory tracking in Vehicular Ad hoc Networks (VANETs) in this work. Previous tracking methods suffer from low accuracy and large overhead big error. In this paper, we propose a Vehicular Crowdsourcing Localization and Tracking (VCLT) scheme for mounting a trajectory tracking attack. In our scheme, crowdsourcing technique is applied to sample the location information of certain users. Then matrix completion algorithm is used to generate our predictions of the users' trajectories. To alleviate the error disturbance of the recovered location data, Kalman filter technique is implemented and the trajectories of certain users are recovered with accuracy. At last, extensive simulations are conducted to show the performance of our scheme. Simulations results reveal that the proposed approach is able to accurately track the trajectories of certain users.

Keywords: Trajectory Tracking; Crowdsourcing; Matrix Completion; Kalman Filter; VANETs

1 Introduction

In recent years, there have been growing interests and research efforts in the area of Vehicular Ad-hoc Networks (VANETs) [1, 2]. VANETs can provide users with various kinds of services, such as entertainment, commercial recommendation, location based services (LBS), and so on. Location and trajectory, the sorted list of locations through which a user travels, are fundamental to most of these services. Therefore, it is critical to provide efficient trajectory tracking. Most of prior arts used the methods of Variable-order Markov model [3], Probabilistic model [4] and Dirichlet-multinomial model [5]. These techniques suffer from the following drawbacks:

1) Expensive and complicated architecture. Many tracking methods require complex system architecture to track user's trajectory. For example, an on-board tracking device would require speed/GPS sensors and information from Road

Side Units (RSUs) to collect surrounding geospatial information and then transmit these data to location server for computation [6, 7].

2) Low accuracy & big error. In general, GPS can only provide an accuracy within 5∼30 meters in a dense urban city environment. And target to be recovered is usually represented by a matrix. However, it often has the problem of missing data or the noise pollution [8].

Motivated by the aforementioned drawbacks and to enhance the accuracy of attacking, we propose a scheme that uses a unique Matrix Completion technique in trajectory tracking or recovering area. Our schemes utilizes the concept of crowdsourcing to sample the location information of a target user. Then matrix completion technique is used to recover the location information matrix, which accurately describes the timely location information of a target user. In addition, we employ Kalman filter to further improve accuracy. Generally, the contributions of this paper can be summarized as follows.

1) We propose a trajectory tracking scheme called VCLT that takes advantages of crowdsourcing in sampling vehicles' location information.

2) To recover the trajectory of mobile users, matrix completion mechanism is used. In the matrix completion calculation process, a target's location history is sampled by crowdsourcing, then a complete trajectory will be restored.

3) To enhance the accuracy of the recovered trajectory data, we use Kalman filter to reduce the errors generated in matrix completion stage.

The rest of the paper is organized as follows. We reviewed related literatures in Section 2, and present the preliminary techniques in Section 3. The crowdsourcing based localization and tracking scheme is introduced in Section 4. We report our extensive simulation results in Section 5 and discuss and conclude the paper in Section 6.

2 Literature Review

Vehicle tracking systems [9] can be used in theft prevention, retrieval of lost vehicles, providing traffic oriented services on lanes. However, the analysis of the historical location information of a vehicle can also implicate the personal behaviors and habits of the user. For VANETs, privacy has been identified to be profoundly important.

There are two main categories for trajectory tracking attack in VANETs: Dynamic Position-aware method and Static Position-aware method. They are just corresponding to two different ways of dedicated short-range communication (DSRC) [10, 11] that is exchanging information either vehicle-to-vehicle (V2V) or vehicle-to-infrastructure (V2I). In the Static Position-aware method, it relies on the fixed-location Road Side Units (RSU) and cellular base stations. Hidden Markov Model is proposed in [12] to recover trajectory sequence by analyzing trace histories of mobile nodes and the spatial correlation between the base station and the mobile station. It has a stable trajectory recovering ability under the rigid assumption that there is a high state transmission chance between the nodes. However, V2V communication is a more viable solution for the near future [13] and Dynamic Position-aware method is well adapted to the ever-changing traffic flows. For the tracking purpose, each vehicle is assumed to be equipped with a DSRC radio and a GPS receiver. As shown in the Figure 1, each vehicle is designed to continuously report its own status by broadcasting safety messages. At the same time, each vehicle also tracks movements of neighboring vehicles based on information received from them over the shared channel [14]. Probabilistic model [4] shows its great application value in solving long-term trajectory tracking by using a small amount of mobile observation data, but the exact success probability is difficult to be measured.

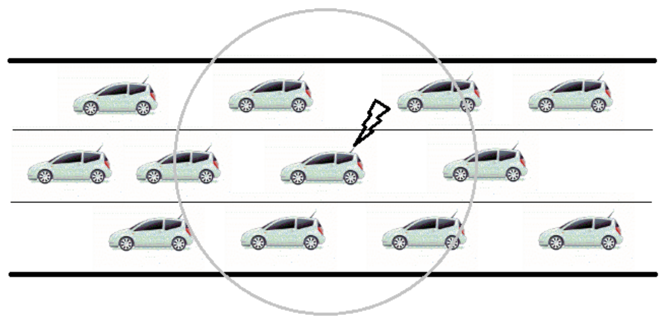

Fig. 1. Tracking the position of neighboring vehicles

Besides trajectory tracking, crowdsourcing can also play an important assistant role in the position-aware process. In general, crowdsourcing is the approach of obtaining needed content by soliciting contributions from a large group of users. Based on crowd-generated data, we can extract the information to optimize localization and tracking. In the field of indoor location [15], a hot research trend is to incorporate crowdsourcing model and built-in sensors in today's smartphone. In parallel, we are being witness of a fast growing needs for improving driving experience based on crowdsourcing like road traffic aware trip planning [16], alert broadcasting [17] or car-pooling services [18].

3 Preliminary

We present preliminary on crowdsourcing, matrix completion, and Kalman filter in this section.

3.1 Crowdsourcing Computation Model

The Crowdsourcing Computation Model is proposed to get the target vehicle's information by gathering the efforts of the vehicles during a specific period. Our

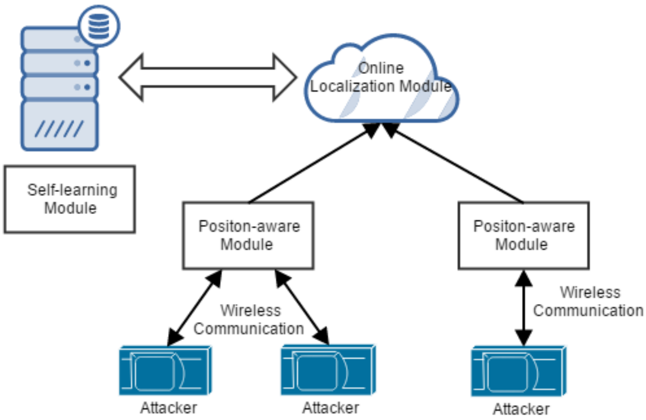

Fig. 2. Mobile Vehicle Tracking System

model is divided into three sub-module: position-aware module, online localization module and self-learning module, as shown in Figure 2.

In the position-aware module, we randomly select some nodes as the detectors. The detectors' responsibility is to collect IDs from the surrounding nodes. The ID is mapped to the Media Access Control Address of the vehicle, which is unique. We neglect the weak signal connection, which can ensure the accuracy of the location. The detectors will pack the reference point coordinate and the ID set and upload them to the server with the current time stamp. All these information is stored in dynamic location database on the server.

In the online localization module, the server analyzes the location data and extract the information matching the IDs of the target vehicles. We create a map to store the individual vehicle's location history using ID as the key. Each entry combines the source position and the sampling time like (x, y, t). The map is left to generate sparse matrix for further trajectory recovery.

In the self-learning module, we accept feedback from the comparison between the output and the observation and handle the subtle deviation. Based on the large amount of history location data, we can figure out habitual traveling route of the targeted vehicle. Dynamic location database will be continuously updated during operation, so that it can adapt to the dynamic VANETs environment.

3.2 Matrix Completion Method

Method Overview Matrix Completion (MC) [19] is an extension of Compressive Sensing [20, 21], which has become a popular research direction in the fields of Signal Processing, Machine Learning, Artificial Intelligence and so on in recent years. MC is the process of recovering a sparse matrix $M \in R^{m \times t}$ with rank r, with some unknown or missing values. It is mainly under the assumptions that the matrix is low-rank, i.e., $r \ll m$ or t. In other words, MC enables us to obtain the predicted traveling route and we compare it to the real trajectory.

We design Vehicle Trajectory Recovery based on Matrix Completion (VTRM-C) to take advantage of space and time correlation between each source node and reduce the sampling frequency. The method acquires the result from the online localization module of the Crowdsourcing Computation Model, and recover the locations of target with tolerable deviation based on Low-rank MC.

In the online localization module, we have selected detector vehicle randomly, and denote their reference point as $P_i(x_i, y_i)$ and the time as t. We generate an original map constructed by these source vehicle ID, location vectors and the sampling time. Based on these information, we supplement the missing location vector according to timetable, that is to say the size of the original map is expanded to N $(N > \max(n_i))$. Using the map, we construct the sparse matrix including the original and missing location vectors.

In this paper, we mainly use matrix completion to solve the problem of missing information recovery. Because we have got a complete location list at every determinate time, it can help us recover the trajectory more efficiently. We adopt a high performance linear operation to get the result and the reconstruction will be addressed by solving the following optimization problem [22]:

$$\min \operatorname{rank}(X) \text{ s.t. } A(X) = B. \tag{1}$$

Here, X presents an observation matrix containing complete information, and it is also $m \times n$ order low-rank matrix. A is a linear map from X to B and B is a sparse matrix.

However, the reconstruction problem in (1) is NP-hard. The time required to solve the problem increases exponentially with the growth of the matrix size. So for the mass matrix, rank minimization method is almost unsolvable.

But for the reason that the rank r of a matrix is as same as the number of non-zero singular values of it, we use the sum of the singular values, which is nuclear norm, approximating for rank of the matrix.

Thus, problem in (1) can be converted into the problem as follows:

$$\|X\|_* \text{ s.t. } A(X) = B. \tag{2}$$

Here, we define $\|X\|_*$ as a nuclear norm of the matrix X meeting the following condition:

$$\|X\|_* = \sum_{i=1}^{n} \sigma_i(X). \tag{3}$$

Nuclear norm corresponds to l_1 norm of the vector composed of the matrix singular value, and the rank corresponds to l_0 norm of the vector.

Sparse Matrix Completion In the session, an observation matrix is builded, and the rows of the matrix represent the x(y) coordinate of single source vehicle at every determinate time, while the columns represent the time. The time interval between each columns is even.

With more vehicles involved in sampling, we have a greater opportunity of capturing the source information from the target vehicle. As a result, the accuracy of the positioning result will be higher. In our scheme, for vehicles' movement is a continuous process, our observation matrix X can be considered to have a low rank characteristics. Based on the actual situation, a vehicle's location information usually has strong association with the continuity of the route.

We donate $A_{N\times T}(X) = B$ indicate the sampling process. X is an observation matrix, and each element in the matrix represents the sampling x(y) position of vehicle i at time j. For a better expression, $A(X)$ can be defined as a matrix $Q_{N\times T}(t)$ as below:

$$Q(i,j) = \begin{cases} 1 & \text{if vehicle } i \text{ at time } j \text{ is sampled,} \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

We can compute by this formula

$$A_{N\times T}(X) = Q_{N\times T}(t) \cdot X(t) = B, \tag{5}$$

where $\cdot$ represents a dot product of two matrix.

As illustrated before, we can solve the following optimization problem to reconstruct the observation matrix:

$$\min \|X(t)\|_* \text{ s.t. } A_{N\times T}(X) = B. \tag{6}$$

3.3 Kalman Filtering

We find that there are some errors in the restored matrix after VTRMC. We need efficient ways to filter noise out of original data. So, we choose to suppress the sampling data noise by Kalman filtering.

Kalman filter [23] is the optimal filter with a minimum mean-square error. It was first used to estimate the parameters of stochastic processes, and soon gained wide application in the problems of the filtering and optimal control.

The starting point of the Kalman filter is based on the dynamic model of the system:

$$x(n) = \varPhi(n, n-1)x(n-1) + \varDelta(n)u(n) + \omega(n-1), \tag{7}$$

$$z(n) = H(n) \times n + v(n). \tag{8}$$

Here, we donate $x(n)$ as M-dimensional state vector of the system. The purpose of the Kalman filtering is in accordance with the best estimation of the state vector. The other symbols are defined in Table 1 below:

Additionally, noise vectors $\omega(n)$ and $v(n)$ are white noise vectors with the following properties:

$$E\{\omega(n)\} = 0, E\{v(n)\} = 0, \tag{9}$$

$$E\{\omega(n)\omega^T(n)\} = R, E\{v(n)v^T(n)\} = S. \tag{10}$$

Table 1. Variable definition of Kalman Filter

$u(n)$	S-dimensional system input vector
$z(n)$	L-dimensional measurement vector
$\omega(n)$	M-dimensional system noise vector
$v(n)$	L-dimensional measurement noise vector
$\Phi(n, n-1)$	M×M system transition matrix
$\Delta(n)$	M×S system input matrix
$H(n)$	L×M measurement matrix

Suppose the estimate of the location vector at time n is $\hat{x}(n)$. Thus, we define error estimation vector as

$$e(n) = x(n) - \hat{x}(n). \tag{11}$$

Estimation error covariance matrix is

$$P(n) = E\{e(n)e^T(n)\} = E\{[x(n) - \hat{x}(n)][x(n) - \hat{x}(n)]^T\}. \tag{12}$$

We can get the estimation error variance:

$$\xi(n) = \sum_{i=1}^{M} E\{e_i^2\}, \tag{13}$$

or

$$\xi(n) = T_r p(n). \tag{14}$$

Kalman Filter is the system described by the above formulas (Φ, Δ, H, R, S are known). According to the input vector $u(n)$ and measurement vector $z(n)$, we can find out the best estimate $\hat{x}(n)$ of the state vector making the estimation error variance $\xi(n)$ minimum.

We assume that it is known the best estimate at the time $(n-1)$ and the corresponding covariance matrix

$$P(n-1) = E\{[x(x-1) - \hat{x}(x-1)][x(n-1) - \hat{x}(n-1)]^T\}. \tag{15}$$

Kalman Filtering Algorithm get the best estimate $\hat{x}(n)$ at time n in two steps:

Step 1: Prediction. Based on $\hat{x}(n-1)$ at time (n-1), we can get the best estimate $\hat{x}(n, n-1)$ at time n according to formula (7):

$$\hat{x}(n, n-1) = \Phi(n, n-1)\hat{x}(n-1) + \Delta(n)u(n-1). \tag{16}$$

The corresponding prediction variance matrix is

$$P(n, n-1) = \Phi(n, n-1)P(n-1)\Phi^T(n, n-1) + R. \tag{17}$$

On the basis of $\hat{x}(n, n-1)$ and the regulation of the vehicle movement, we can get the predicted value:

$$z(n) = H(n)\hat{x}(n, n-1). \tag{18}$$

Step 2: Filtering. According to the actual measured value, we can get the predicted value:

$$\hat{x}(n) = \hat{x}(n, n-1) + K(n)[z(n) - H(n)\hat{x}(n, n-1)]. \tag{19}$$

Here, K represents the gain matrix.

It can be proved that the best treatment is showed as follows:

$$K(n) = P(n.n-1)H^T(n)[H(n)P(n-1)H^T(n) + R]^{-1}, \tag{20}$$

$$P(n) = [I - K(n)H(n)]P(n, n-1). \tag{21}$$

Generally speaking, the dynamic system represented by the formula (7), and its Kalman filter equations include formula (16-21).

According to the initial value of the state vector $\hat{x}(0)$ and error variance $P(0)$, we can receive the best estimate of the locations of target vehicles at each time with these equations.

4 Crowdsourcing Localization and Tracking

4.1 Problem Statement

Crowdsourcing-based Localization and Tracking scheme encourages the vehicle users to participate to improve the accuracy of the estimated position of the targeted vehicle. Figure 3 shows our thoughts in designing the scheme.

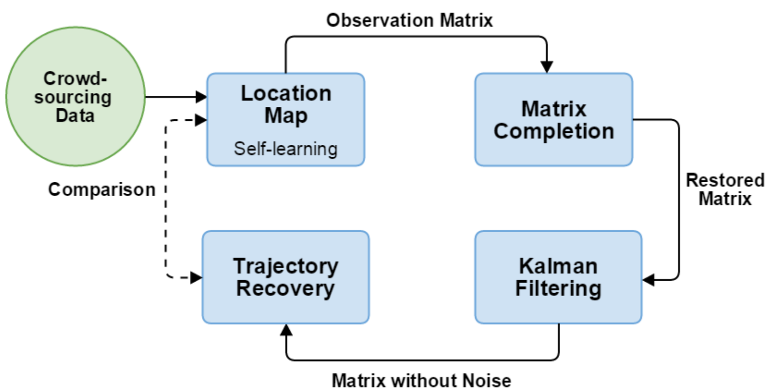

Fig. 3. Vehicle Tracking Setup

As illustrated before, we consider a vehicle network composed of a set of N vehicles including detectors and target vehicles. A detector is capable of tracking the neighboring vehicles based on information (e.g., ID, position, time) received from them, recorded as (id, x, y, t). The duty of the detectors is tracking and reporting to the control center timely the location information of any vehicle including the target one in VANETs. After the original information is processed with Matrix Completion and Kalman Filtering, we extract the information of the target vehicles like $G_t(ID) = \{P_0, P_1, P_2, \cdots, P_{t-1}\}$, which can generate the estimated trajectory.

4.2 Crowdsourcing Modeling

In obedience to the wishes of the users, the vehicles participating are treated as nodes uploading the identification key of the nearby nodes in the range of communication at every determinate time. They act as detectors in the VANETs. All the data will be examined seriously. When it comes to repetition, we calculate the closest value based on the position estimation algorithm. Figure 4 shows the situation of vehicle communication at a given time in which the circles represent the communication distance. From the data transmitted from the nodes, we can fetch the information belonging to the target attacked vehicles.

Fig. 4. Vehicle Communication

4.3 Trajectory Recovering

The original data is shown in the map as discrete points. We use VTRMC method to recover the undetected location information and adopt Kalman filtering to eliminate these errors in the the recovered matrix .

Algorithm shows the operational details of the Crowdsourcing Tracking and Recovery algorithm.

Algorithm 1 Crowdsourcing Tracking and Recovery

1: **Input:** VANET Map, Scenarios, Tracking parameters
2: **Output:** Reconstruction matrix $\hat{B}_{N\times T}$
3: VANET Initialization
4: Crowdsourcing Computation Model Initialization
5: Record the target vehicles location list over time within a map
6: Generate observation matrix $X_{N\times T}$, corresponding two-valued matrix Q
7: Compute the estimate location matrix $B_{N\times T}$ in accordance with Equation 2
8: Generate the prediction location matrix based on the state of vehicles
9: Kalman filtering noise out of $B_{N\times T}$ according to the prediction matrix $\mathcal{C}_l$

Algorithm above proceeds as follows: Firstly, we import the VANET map along with scenarios in which we set the initial state of the vehicles and the street. We also input some attacking parameters, such as vehicle communication interval, the amount of vehicles and sampling frequency. After the initialization of VANETs, the vehicles are driven towards the destination. Meanwhile, the tracking attack process begins.

The detectors report the received information to the control center. The crowdsourcing computation model and will first record the location list over time for each vehicles in a map, and then generate the observation matrix $X_{N\times T}$ according to the map. Matrix completion method will be performed to generate $B_{N\times T}$ as recovery matrix of original matrix. At last, it handles noise out base on Kalman Filter and work out the reconstruction matrix $\hat{B}_{N\times T}$. The algorithm will finally return $\hat{B}_{N\times T}$ as the output, which can be treated as the recovered coordination matrix of all the vehicles running in the sampling region.

5 Simulations

We conduct the simulations to evaluate the performance of VCLT in realistic settings. Our data is generated from VANETsim [24], a simulator for security and privacy concepts in VANETs.

5.1 Simulation Setup

For vehicles' movement in simulation, we use trajectories produced by VANET-sim. This scenario corresponds to a medium speed of 30km/h. Upon receiving information from other cars, each vehicle uploads its communication records with neighboring vehicles. We choose Crowdsourcing-based Localization and Tracking scheme for tracking attacked vehicles. The street maps of three different cities in China, Dalian, Changsha, and Wuhan, have been chosen from the OpenStreetMap (OSM) project [25]. We use only a small region in each city to simulate our scheme. Related simulation parameters are listed in Table 2.

Table 2. Simulation Parameters

Parameters	Values
Sampling Interval	60s
Average Speed	30km/h
Max Communication Distance	100m
Simulation Time	6 hours
Number of Vehicles	200
Map Size	10km by 7km (Dalian) 10km by 10km (Changsha) 10km by 8km (Wuhan)

5.2 Vehicle Network Establishment

The construction of road network data for Vehicle Network is based on OS-M, which provides us with original geographic data which improve our attack mechanism's credibility. We reconstruct the road network data by removing unnecessary geographic elements, as shown in Figure 5.

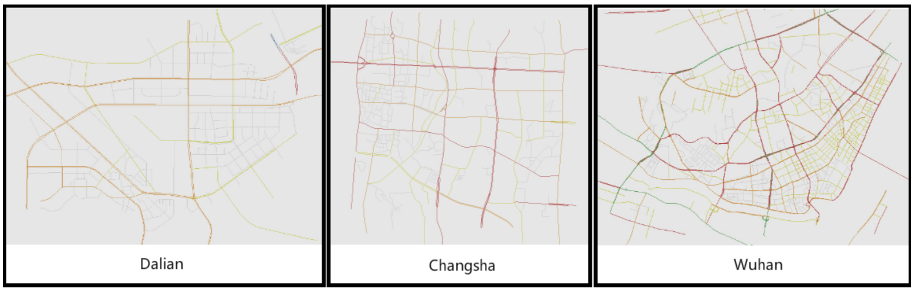

Fig. 5. Road Maps of Dalian, Changsha & Wuhan

We create the scenarios by randomly sampling locations and destination of the vehicles. In order to simulate the real situation of the vehicle, we set the speed, communication distance, braking rate and acceleration rate and so on. We assure that Wi-Fi connection is available to each vehicle.

The tracked vehicle is selected randomly during simulation, and its real trajectory will be recorded.

5.3 Matrix Completion and Filtering

To illustrate the accuracy and energy-saving features of our scheme, we extract 100 vehicles experiments data to construct the original sampling matrix. Matrix completion make trajectory tracking more efficient by abandoning extra communication cost in VANETs.

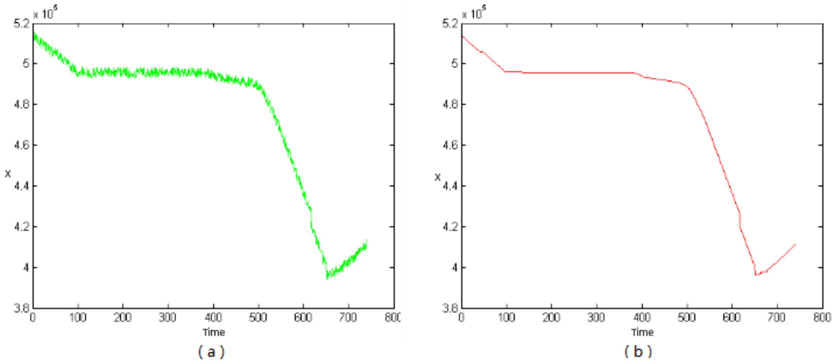

Fig. 6. A contrast between trajectory recovering (a) The position curve with noise, and (b) The position curve after filtering

We capture the location information of the target from the restored matrix. The diagrams show the performance of VCLT test. The left Figure 6(a) shows the curve of the position of vehicle on the x-axis coordinates over time. After removing the noise included in the restored information, we get an ideal curve in Figure 6(b) that displays the position change process of the vehicle.

Citing Dalian as an example, Figure 7(a) shows the real trajectory of a vehicle user in a period of time, as the blue line shows. The circle marks its starting point, and the cross represents the ending point. The medium Figure 7(b) reflects the recovered trajectory and the red points stand for the sampling points during the crowdsourcing period. From right Figure 7(c), the effect of trajectory recovery is in the continuing optimization, as the recovery route (green) and the filtered route (blue) show.

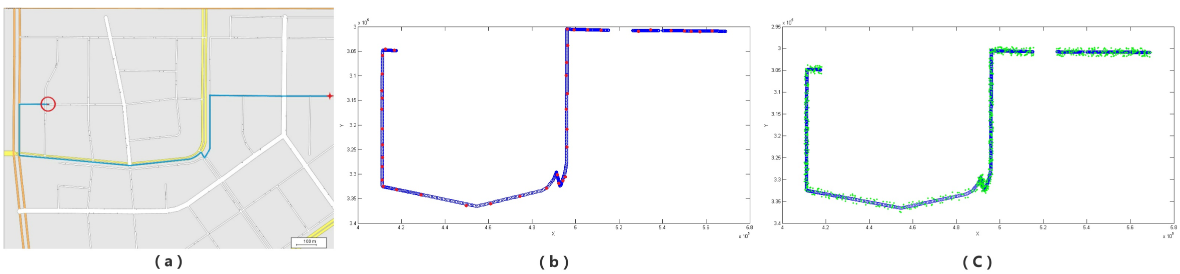

Fig. 7. A trajectory tracking example: (a) The position curve with noise, (b) The position curve after filtering and (c) Contrast of before and after filtering

5.4 Recovered Trajectory

To illustrate that VCLT is capable of multi-vehicle tracking, we randomly select 3 vehicles in tracking region and recover their locations and trajectory. Based on the restored information without noise, we can determine a rough trajectory of the target vehicle. Based on the road network map, we use the estimated position to restore a credible traveling route. The Figure 8 shows the intermediate achievement of three target attacked vehicle after the Trajectory Recovering.

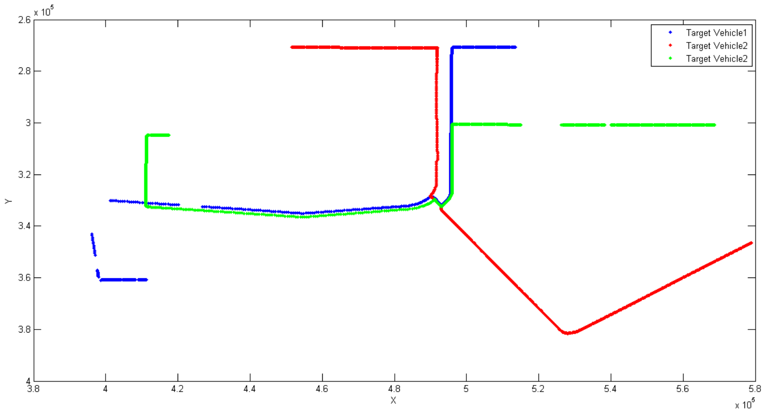

Fig. 8. The recovered tracking results

5.5 Influence of Crowdsourcing Vehicle Amount

The amount of vehicles participating in crowdsourcing has a lot to do with positioning accuracy. With the increase of vehicles, accuracy is improved. We conduct the simulation for verification. As is shown below in Figure 9(a), we can see that with the rise in vehicle number, errors decrease. When the amount of detector vehicles running is 50 in the determined region, the chance to meet and communicate with the target vehicles is greatly reduced. In addition, we notice that the relative error of Wuhan is the highest. It is for the reason that the number of intersections and complexity of road network will also influence the accuracy. In short, vehicle amount needs to match the sampling area, and adapt to road network capacity in the reality. According to our road network in simulation, we set the number of vehicle is 200, which lets VCLT scheme run efficiently with the appropriate data that it requires.

5.6 Influence of Vehicle Communication Interval

In this session, we mainly discuss the influence of the vehicle communication interval which is related to the sampling frequency. As is shown in Figure 9(b), the

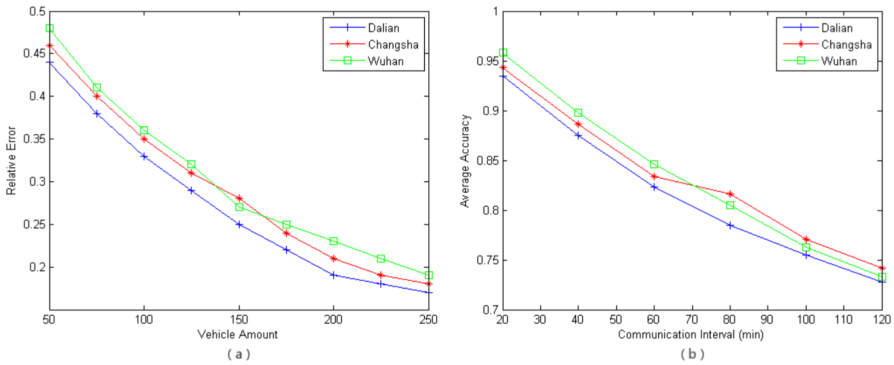

Fig. 9. Comparisons with Different (a) Vehicle Amount and (b) Comm. Interval

shorter time interval will lead to the lower average tracking accuracy. It means our crowdsourcing computation model needs appropriate sampling frequency to reduce the error and improve the tracking success rate. From Figure 9(b), we find that the denser the distribution of the streets is, it's of the greater chance to communicate with neighboring vehicles, that is why the accuracy of Wuhan is comparatively higher than others.

6 Conclusion

In this paper, we proposed a VANET localization and tracking crowdsourcing model VCLT. Our model does not require using the additional sensors, except for the availability of Wi-Fi wireless connection. We believe that crowdsourcing is an ideal way for tracking that benefits from the mass participation and effective network connection. An innovative and practical method VTRMC was proposed to accurately estimate the position and restore missing information based on the continuity of the trajectory. We filter noise out of restored data. We built the vehicle network and simulate the process. Our extensive evaluation results indicate that our crowdsourcing model successfully handles complex city road network structure and simultaneously provides good performance in localization cost and trajectory recovering.

As part of our future work, we will mainly study how to avoid colluding attacks based on VCLT.

Acknowledgment

This research is sponsored in part by the National Natural Science Foundation of China (contract / grant number: No.61173179, No.61202441, No.61402078) and Program for New Century Excellent Talents in University (NCET-13-0083). This research is also sponsored in part supported by the Fundamental Research Funds for the Central Universities (No.DUT14YQ212, No.DUT14RC(3)090).

References

1. Zeadally S, Hunt R, Chen Y, et al. Vehicular ad hoc networks (VANETS): status, results, and challenges[J]. Telecommunication Systems, 2012, 50(4):217-241.
2. Samara G, Al-Salihy W A H, Sures R. Security issues and challenges of Vehicular Ad Hoc Networks (VANET)[C]. //International Conference on New Trends in Information Science & Service Science. IEEE, 2010:393 - 398.
3. Reza T A, Barbeau M, Lamothe G, et al. Non-cooperating vehicle tracking in VANETs using the conditional logit model[C]. //International IEEE Conference on Intelligent Transportation Systems. IEEE, 2013:626 - 633.
4. Hao P, Boriboonsomsin K, Wu G, et al. Probabilistic model for estimating vehicle trajectories using sparse mobile sensor data[C]. //IEEE International Conference on Intelligent Transportation Systems. IEEE, 2014:1363 - 1368.
5. Xue G, Luo Y, Yu J, et al. A novel vehicular location prediction based on mobility patterns for routing in urban VANET[J]. EURASIP Journal on Wireless Communications and Networking, 2012, 2012(1): 1-14.
6. Sampigethaya K, Huang L, Li M, et al. CARAVAN: Providing Location Privacy for VANET[J]. in Embedded Security in Cars (ESCAR, 2005.
7. Shafiee K, Leung V C M. Connectivity-aware minimum-delay geographic routing with vehicle tracking in VANETs[J]. Ad Hoc Networks, 2011, 9(2): 131-141.
8. Tsai M, Wang P, Shieh C, et al. Improving positioning accuracy for VANET in real city environments[J]. Journal of Supercomputing, 2014:1-21.
9. Thangavelul A, Bhuvaneswari K, Kumar K, et al. Location Identification and Vehicle Tracking using VANET (VETRAC)[J]. 2007.
10. Cheng L, E. Henty B, D. Stancil D, et al. Mobile Vehicle-to-Vehicle Narrow-Band Channel Measurement and Characterization of the 5.9 GHz Dedicated Short Range Communication (DSRC) Frequency Band[J]. IEEE Journal on Selected Areas in Communications, 2007, 25(8):1501-1516.
11. Jiang D, Delgrossi L. IEEE 802.11p: Towards an international standard for wireless access in vehicular environments[C]. //IEEE Vehicular Technol Conf. 2008:2036 - 2040.
12. Vasquez D, Fraichard T, Laugier C. Growing Hidden Markov Models: An Incremental Tool for Learning and Predicting Human and Vehicle Motion[J]. International Journal of Robotics Research, 2009, 28:1486-1506.
13. Fallah C H, Sengupta Y P, Krishnan R, et al. Adaptive intervehicle communication control for cooperative safety systems[J]. Network IEEE, 2010, 24(1):6 - 13.
14. Rezaei S, Sengupta R, Krishnan H, et al. Tracking the position of neighboring vehicles using wireless communications[J]. Transportation Research Part C Emerging Technologies, 2010, 18(3):335-350.
15. Jiang Z, Zhao J, Li X, et al. Communicating Is Crowdsourcing: Wi-Fi Indoor Localization with CSI-based Speed Estimation[J]. Journal of Computer Sciences&stechnology, 2014, 29(4):589-604.
16. Artikis A, Weidlich M. Heterogeneous Stream Processing and Crowdsourcing for Urban Traffic Management[J]. Edbt, 2014.
17. Suriyapaibonwattana K, Pomavalai C. An Effective Safety Alert Broadcast Algorithm for VANET[C]. //International Symposium on Communications & Information Technologies. IEEE, 2008:247 - 250.
18. Corporation H P. Mobility Crowdsourcing: Toward Zero-Effort Carpooling on Individual Smartphone[J]. International Journal of Distributed Sensor Networks, 2013, 12(3):188-192.

19. Cands E J, Recht B. Exact Matrix Completion via Convex Optimization[J]. Foundations of Computational Mathematics, 2009, 9(6):717-772.
20. Wang J, Tang S, Yin B, et al. Data gathering in wireless sensor networks through intelligent compressive sensing[C]. //Infocom, IEEE. IEEE, 2012:603 - 611.
21. Wang H, Zhu Y, Zhang Q. Compressive sensing based monitoring with vehicular networks[C]. //Infocom, IEEE. IEEE, 2013:2823 - 2831.
22. Lin Z, Chen M, Ma Y. The Augmented Lagrange Multiplier Method for Exact Recovery of Corrupted Low-Rank Matrices[J]. Eprint Arxiv, 2010.
23. Welch G, Bishop G. An Introduction to the Kalman Filter[J]. University of North Carolina at Chapel Hill, 1995.
24. VANET Simulator, `http://svs.informatik.uni-hamburg.de/vanet/`
25. OpenStreetMap, `http://www.openstreetmap.org`