



HAL
open science

GPS-based preliminary map estimation for autonomous vehicle mission preparation

Yohan Dupuis, Pierre Merriaux, Peggy Subirats, Rémi Boutteau, Xavier Savatier, Pascal Vasseur

► **To cite this version:**

Yohan Dupuis, Pierre Merriaux, Peggy Subirats, Rémi Boutteau, Xavier Savatier, et al.. GPS-based preliminary map estimation for autonomous vehicle mission preparation. 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014), Sep 2014, Chicago, United States. 10.1109/IROS.2014.6943160 . hal-01710415

HAL Id: hal-01710415

<https://hal.science/hal-01710415>

Submitted on 15 Feb 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

GPS-based Preliminary Map Estimation for Autonomous Vehicle Mission Preparation

Y. Dupuis¹, P. Merriaux², P. Subirats¹, R. Boutteau², X. Savatier² and P. Vasseur³

Abstract—In this paper, we tackle the problem of map estimation from small set of vehicular GPS traces collected from low cost devices. Contrary to the existing works, we rely only on GPS information. First, we propose a fast implementation of Kalman filtering of spline-based road modeling. Our approach demonstrates a significant boost of the computation speed while maintained a good estimation error. Secondly, we perform an evaluation of our algorithm on real world data. Our estimation is compared to a high grade Inertial Navigation System and vectorial data gathered from major map providers. Our results suggest that a good performance can be achieved from the fusion of multiple GPS traces collected from multiple vehicles and drivers.

I. INTRODUCTION

Autonomous vehicles have demonstrated to be able to travel long runs without any driver intervention. On the one hand, these vehicles mainly rely on the fact that the map used for navigation is known and accurate. This preparation often requires important human efforts and costly devices to achieve the required accuracy [1], [2]. On the other hand, GPS-equipped mobile devices are popular. They are cheap but less accurate than an Inertial Navigation System (INS)/GPS solutions used in autonomous vehicle mission preparation. This information can be crowd-sourced and used to obtain preliminary road location inference at a really low price.

In the literature, there exists three main approaches to infer road location from a large amount of GPS car data [3].

First of all, K-means based approaches were investigated. They consist in randomly choosing seeds in the GPS point cloud. Then K-means like techniques are used to refined the seeds position. The segment joining two seeds is considered as representative of the road [4]. The state space is composed of the latitude, longitude and bearing. As a result, this type of approach is able to deal with bidirectional roads. However, if the number of seeds is too low, the resulting road geometry is not coherent with the underlying road layout.

Secondly, curves from different GPS car traces can be merged directly [5]. They have the same advantages as the K-means based techniques. Still, as the traces are not reduced to seeds, these methods enable to obtain a resulting road geometry that is closer to reality. However, as every traces

are considered, the resulting road geometry is impacted by high error samples.

Thirdly, roads can be inferred within a probabilistic framework, based on Kernel Density Estimators (KDE) [6]. An occupancy grid is build at a given spatial resolution. Each cell is represented by the number of traces passing through it. The sensor uncertainty is applied by convolving a Gaussian kernel with the grid. The uncertainty is managed with the kernel bandwidth. A threshold is then applied to the grid to obtain a binary image on which a skeletonization techniques are used to obtained the road centerline. This technique has almost the same advantages as the trace merging approaches. On the one hand, it is not able to deal with bidirectional road as two adjacent lines often falls within the uncertainty range and the bearing is not considered. On the other hand, it is less sensitive to high error samples.

A hybrid framework, based on trace merging and a probabilistic framework, has been successfully applied to GPS data collected on trains. Train motion is constrained by the track contrary to cars where lane widths can reach 3.5 meters. This task is consequently more challenging for cars since their motion does not follow exactly the same single track. Besides, curvatures are more important on roads than on railway tracks. It mainly explains why this pipeline has never been extended to cars yet. Still this hybrid framework proposes a new and interesting insight on how to tackle the problem. In [7], [8], Hasberg *et al.* successfully modeled several segments of real-world railways with a cubic spline. The spline control points are used in the railway state vector. Two flavors of Kalman filter are investigated. The GPS position results from the fusion of raw GPS and INS data fusion. Given a new corrected GPS measurement, the railway state vector is updated based on the spline equation and measurement error. A lateral error of less than 50cm is achieved with 12 runs with respect to high precision reference map.

Our main contribution is a fast and robust preliminary road location for autonomous vehicle mission preparation based on information collected from multiple conventional GPS. Contrary to the state of the art on road location estimation, we wish to perform this task from a small set of GPS traces. No digital map is used for a coarse initial guess of the road location or the final road estimate evaluation as in [8]. Our performance is evaluated against an all-in-one INS/GPS solution. Our GPS positions have not been pre-processed based on inertial fusion framework that usually enables to reduce GPS error as used in [8]. In fact, this feature does not exist on GPS-equipped mobile devices.

¹Yohan Dupuis and Peggy Subirats are with Department of Multimodal Transportation Infrastructure, CEREMA, 76120 Le Grand Quevilly, France yohan.dupuis@cerema.fr

²Pierre Merriaux, Rémi Boutteau and Xavier Savatier are with IRSEEM, ESIGELEC, 76800 Saint-Etienne-du-Rouvray, France firstname.lastname@esigelec.fr

³Pascal Vasseur is with LITIS Lab, University of Rouen, 76821 Saint-Etienne-du-Rouvray, France pascal.vasseur@univ-rouen.fr

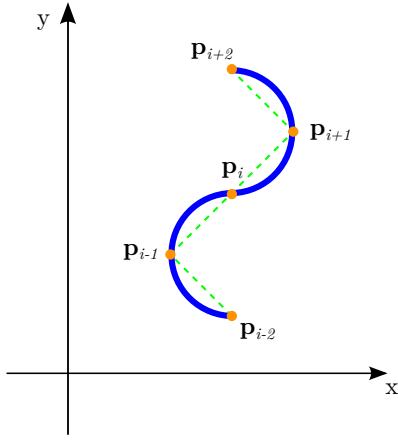


Fig. 1: Spline Parametrization
 p_i : control point; green dashed line: chord length; solid blue line: arc length

The remainder of the paper is organized as follows. Section II introduces our approach. Section III presents our results. Finally, we conclude and discuss our results in Section IV.

II. METHODOLOGY

Splines provide a nice framework to obtain a smooth piecewise geometric approximation of real world road curves. Splines can be parametrized in different manners (c.f. Figure 1). In [7], Hasberg *et al.* use chordal parametrization. In a more recent work [8], they argue that arc-length parametrization should be used as it enables stable integration of 1D kinematics states. As shown in [9], arc length parametrization is not rational for real world curves. Moreover, the integration required to achieve this parametrization really burdens the process in the case of roads as compared to rail tracks [10]. In fact, the curvature of rail tracks is negligible compared to curvatures encountered on roads at roundabouts for instance. Consequently, the bisection method will take more time to converge. Finally, the authors have not compared the performance of both parametrizations. All things considered, we use chordal spline parametrization for the rest of this paper.

A. Cubic Spline Interpolation

First, spline interpolation is performed between points $p_i = (p_i^x, p_i^y)^T$, known as supporting points, for $i \in [0, n]$. It results in the curve $s(u)$. In chordal parametrization, the parameter u_i is computed recursively as follows :

$$u_{i+1} = u_i + \|p_{i+1} - p_i\| \quad (1)$$

with $u_0 = 0$ and $i \in [0, n-1]$. Each piecewise segment $s_i(u)$ is then interpolated using the following formula :

$$s_i(u) = \begin{bmatrix} s_i^x(u) \\ s_i^y(u) \end{bmatrix} = \begin{bmatrix} a_i^x + b_i^x \Delta u_i + c_i^x \Delta u_i^2 + d_i^x \Delta u_i^3 \\ a_i^y + b_i^y \Delta u_i + c_i^y \Delta u_i^2 + d_i^y \Delta u_i^3 \end{bmatrix} \quad (2)$$

with $\Delta u_i = u - u_i$ and $u_i \leq u < u_{i+1}$

$\{a_i^j, \dots, d_i^j\}$, with $j \in \{x, y\}$, are the unknown polynomial coefficients to be estimated. They depend on the

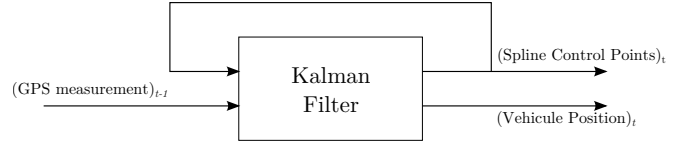


Fig. 2: Recursive Spline-based road model estimation

supporting points, smoothness and continuity conditions on the spline itself as well as the spline first and second order derivatives at these points [8].

The supporting points p_i can be grouped into a single vector q^j along each dimension:

$$q^j = [p_0^j, \dots, p_n^j]^T \quad (3)$$

As a result, the unknown polynomial coefficient formulation can be rewritten into a linear matrix form:

$$\begin{aligned} a^j &= Aq^j & b^j &= Bq^j \\ c^j &= Cq^j & d^j &= Dq^j \end{aligned} \quad (4)$$

More details on the computation of A , B , C and D are given in [8]. This formulation allows a significant speedup in the polynomial coefficient computation.

For a given u such that $u_i \leq u < u_{i+1}$, we wish to select the corresponding segment i and compute $s_i(u)$. As a result, k_i , a masking vector, is introduced. It enables to select the segment i as follows:

$$s_i^j(u) = k_i^T \cdot s^j(u) = \begin{bmatrix} \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \end{bmatrix}^T \cdot \begin{bmatrix} \vdots \\ s_{i-1}^j(u) \\ s_i^j(u) \\ s_{i+1}^j(u) \\ \vdots \end{bmatrix} \quad (5)$$

As shown in Equation (2), $s_i^j(u)$ is expressed as a function of the polynomial coefficients. Consequently, Equation (5) can be rewritten as follows:

$$\begin{aligned} s_i^j(u) &= k_i^T \cdot s^j(u) \\ &= k_i^T \cdot [Aq^j + Bq^j \Delta u_i + Cq^j \Delta u_i^2 + Dq^j \Delta u_i^3] \\ &= k_i^T \cdot [A + B\Delta u_i + C\Delta u_i^2 + D\Delta u_i^3] q^j \\ &= G(u)q^j \end{aligned} \quad (6)$$

The curve $s(u)$ can now be rewritten as:

$$s(u) = \begin{bmatrix} s^x(u) \\ s^y(u) \end{bmatrix} = \begin{bmatrix} G(u) & 0 \\ 0 & G(u) \end{bmatrix} \cdot \begin{bmatrix} q^x \\ q^y \end{bmatrix} \quad (7)$$

B. Kalman Filter

As we wish to estimate the road location, we need to handle GPS measurement noise. We also have to deal with error in our spline model. Kalman filters offer an elegant framework to manage measurement and model noises as well as estimation of state vectors through indirect measurements contaminated by noise [11]. Let us consider x_t a state vector

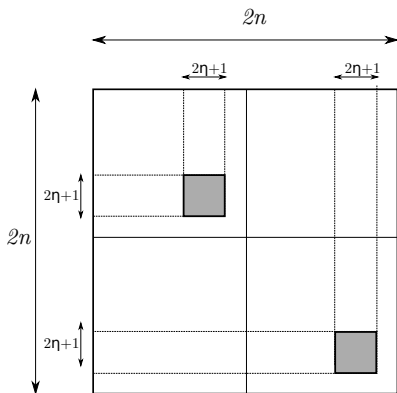


Fig. 3: $\hat{\mathbf{P}}$ update

at time t and \mathbf{z}_t a measurement vector at time t . We define the following dynamic model :

$$\mathbf{x}_{t+1} = \mathbf{F}_t \mathbf{x}_t + \epsilon_t \quad (8)$$

where ϵ_t is the system noise assumed to be Gaussian with a covariance matrix \mathbf{R}_t and \mathbf{F}_t is the state transition model.

The measurement model of the true vector state is given by:

$$\mathbf{z}_t = \mathbf{H}_t \mathbf{x}_t + \delta_t \quad (9)$$

where δ_t is the measurement noise assumed to be Gaussian with a covariance matrix \mathbf{Q}_t and \mathbf{H}_t is the observation model.

The Kalman filter performs the state vector estimation in two main steps. First, the state vector \mathbf{x}_t is estimated as follows :

$$\hat{\mathbf{x}}_{t|t-1} = \mathbf{F}_{t-1|t-1} \mathbf{x}_t \quad (10)$$

$$\hat{\mathbf{P}}_{t|t-1} = \mathbf{F}_t \mathbf{P}_{t-1|t-1} \mathbf{F}_t^T + \mathbf{R}_t \quad (11)$$

where $\hat{\mathbf{P}}_{t|t-1}$ and $\hat{\mathbf{x}}_{t|t-1}$ are the estimated a priori covariance matrix and state vector respectively. Secondly, when a measurement \mathbf{z}_t is available, the residual $\hat{\mathbf{y}}_t$ is used to update the state vector:

$$\hat{\mathbf{y}}_t = \mathbf{z}_t - \mathbf{H}_t \hat{\mathbf{x}}_{t|t-1} \quad (12)$$

$$\mathbf{K}_t = \hat{\mathbf{P}}_{t|t-1} \mathbf{H}_t^T (\mathbf{H}_t \hat{\mathbf{P}}_{t|t-1} \mathbf{H}_t^T + \mathbf{Q}_t)^{-1} \quad (13)$$

$$\hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t \hat{\mathbf{y}}_t \quad (14)$$

$$\hat{\mathbf{P}}_{t|t} = (\mathbf{I} - \mathbf{K}_t \mathbf{H}_t) \hat{\mathbf{P}}_{t|t-1} \quad (15)$$

where $\hat{\mathbf{P}}_{t|t}$ and $\hat{\mathbf{x}}_{t|t}$ are the estimated a posteriori covariance matrix and state vector respectively.

C. Fast Road Spline Model estimation by Kalman Filtering

In the following framework, we consider that the cars follow the same itinerary. Contrary to [7], [8], we do not use any digital map to provide the initial guess. We rely solely on GPS data. In fact, one of the GPS traces is randomly chosen as our initial guess. A spline interpolation is performed on this trace. In order to avoid singular situations, duplicate position measurements are removed within each GPS trace.

The spline is also re-parametrized to have a constant chordal length l_u . These preliminary operations give us $\hat{\mathbf{S}}_0(u)$.

Given a GPS-based position measurement \mathbf{z}_t and this preliminary itinerary estimate, we first find \hat{u} that minimizes the innovation $\hat{\mathbf{y}}_t$. It corresponds to the closest point on the spline curve. It gives us the corresponding segment and enables to define \mathbf{k}_i and the value of Δu_i . From Equation (7) and Equation (9), we define :

$$\mathbf{H}_t = \begin{bmatrix} \mathbf{G}(\hat{u}) & 0 \\ 0 & \mathbf{G}(\hat{u}) \end{bmatrix} \quad (16)$$

$$\mathbf{x}_t = \begin{bmatrix} \mathbf{q}^x \\ \mathbf{q}^y \end{bmatrix} \quad (17)$$

In our implementation, we consider time invariant states (*i.e.* $\mathbf{F}_t = \mathbf{I}$) and a null system noise (*i.e.* $\mathbf{R}_t = 0$). The variance of estimated error given by GPS from all the measurements is used to define the measurement noise covariance matrix \mathbf{Q}_t . We consider it as diagonal. This same variance is used to fill the initial estimate covariance matrix $\hat{\mathbf{P}}_{0|0}$, which is also defined as a diagonal matrix.

We want to be able to perform a road estimation in a fast manner. We also want our computation time to be invariant to the length of the itinerary. In Sections II-A and II-B, the matrices are rectangular matrices of sizes proportional to n , where n is the number of control points. As the itinerary grows and l_u decreases, the size of the matrices grows quadratically and so is time required to perform the computations. The matrix size impact is really significant in Equations (13) and (15). Even if in theory, it may be interesting to use all the supporting points for small road segments, it is totally useless for road networks. As it can be seen in Figure 4a, GPS measurements in the north part of the map will not influence the shape of road in its southern part.

As a consequence, we propose to use only points in the vicinity of the spline segment $i_{\hat{u}}$ where \hat{u} was found. We define this parameter η . For a given η , the state of a supporting point \mathbf{p}_i is updated if and only if $i_{\hat{u}} - \eta \leq i \leq i_{\hat{u}} + \eta$. Moreover, only the polynomial coefficients affected by these supporting points can be used. Only $2\eta + 1$ columns of \mathbf{A} , \mathbf{B} , \mathbf{C} and \mathbf{D} should be considered. \mathbf{H}_t has now a dimension of $2 \times 2\eta + 1$. Only a subset of the estimate covariance matrix $\hat{\mathbf{P}}_{t|t-1}$ can be updated (c.f. Figure 3).

III. RESULTS

A. Datasets

A data set summary is presented in Table I. As we can see, three datasets were collected from an Android Tablet embedding a Broadcom GPS chip supporting GPS, GLONASS and EGNOS. The data were collected at 1 Hz on three different days from three different vehicles driven by three different drivers. Set one was recorded in summer 2013, set two and three in January 2014 but six days apart from each other. They include ten, ten and four traces respectively. They represent 9451 GPS position measures. The device was placed somehow at the center of the vehicle windshield. The

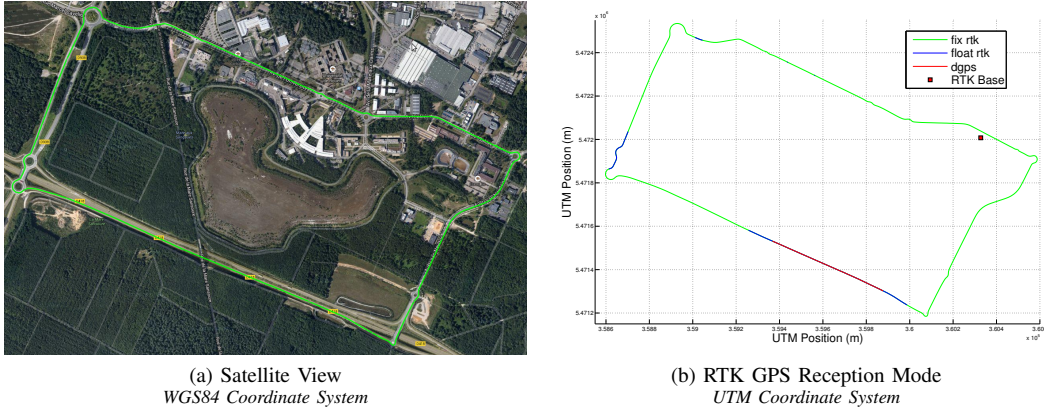


Fig. 4: Experiment Site

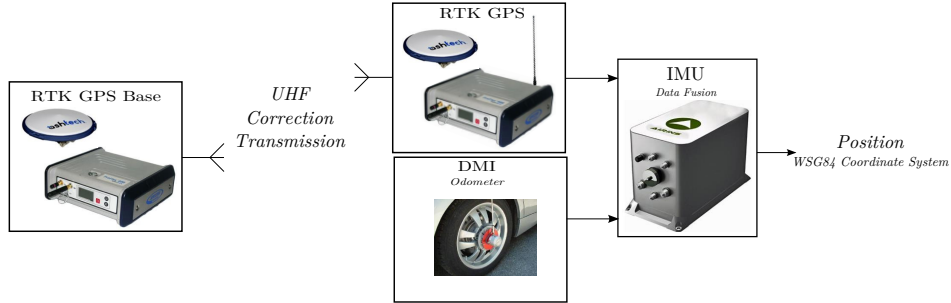


Fig. 5: Ground Truth Collection Setup
DMI: Distance Measurement Indicator; RTK: Real Time Kinematic

Dataset	1	2	3	Reference
Device	Android Tablet	Android Tablet	Android Tablet	RTK GPS/IMU/DMI
Position Update Frequency (Hz)	1	1	1	200
Number of Traces	10	10	4	1
Vehicle ID	1	2	3	3
Driver ID	1	2	3	3
Date of Acquisition	August 2013	January 2014	January 2014	January 2014

TABLE I: Dataset Summary

drivers were asked to use the right-most lane on multilane roads.

The ground truth was collected from a calibrated IXSEA LANDINS, an all-in-one INS/GPS solution, at the same time as set 3. The GNSS device is a Spectra Precision Proflex 800 Real Time Kinematic (RTK) GPS. A RTK base station was placed at the location indicated in Figure 4. A Distance Measurement Indicator (DMI) was placed on the rear left-hand wheel. Its expected accuracy is 2.4mm/pulse. Inertial Measurement Unit (IMU) information from the LANDINS IMU are fused with the RTK GPS and DMI information within the IXSEA device. The fusion is based on a Kalman filter and vehicle model. The trace used as reference was directly outputted by the LANDINS device. As it can be seen in Figure 4b, three reception modes are available: Differential GPS (DGPS), Float RTK and RTK. The expected precisions are respectively 1cm, 5cm and 25cm. The vehicle estimated position is updated at 200 Hz. The recording was performed at 20 Hz. The devices were placed as indicated in Figure 5.

B. Performance Measure

All our experiments are conducted in UTM coordinates. Contrary to the works mentioned in our introduction, our resulting road estimation is compared to the trace resulting from the inner fusion of the RTK GPS, IMU and DMI of the IXSEA device. We use the median of all orthogonal distances of points on the spline to the reference trace as the position error. At the maximum speed on our test itinerary (*i.e.* 25m/s), two consecutive points from the reference trace can be 1.25m apart. As a result, a piecewise linear interpolation is performed in order to compute the orthogonal distance from the resulting line to the spline points. The spline curve is sampled every 1 meter along the chordal parametrization.

C. Experiments

As shown in Figure 9a, we consider our problem as an open loop problem. Despite our test configuration, itineraries are often not closed. As we want to propose a generic framework, no loop closure techniques are used.

In Figure 6a, we investigate the impact of l_u , the supporting point inter distance, on both computation time and

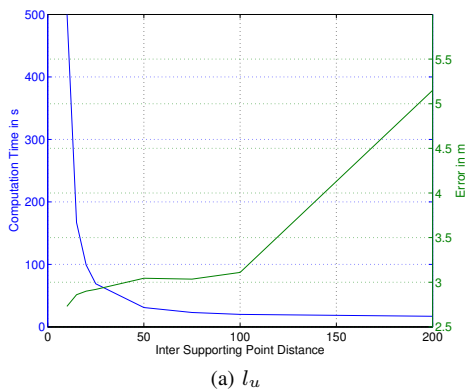
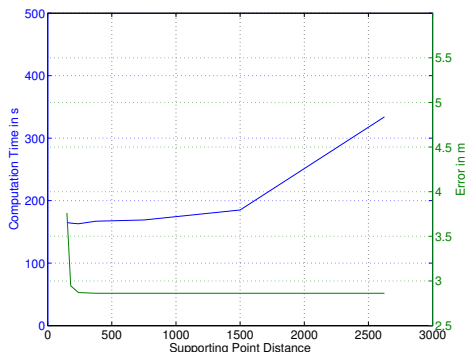
(a) l_u (b) η

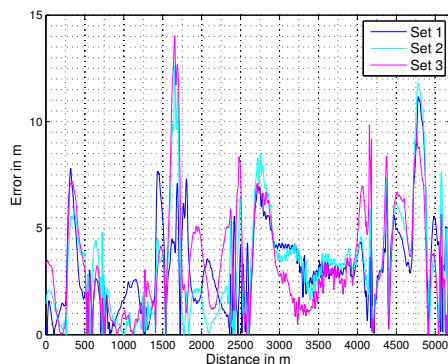
Fig. 6: Algorithm Parameter Impact Analysis

Set \ Set	1	2	3
1	1	0.89	0.84
2	0.89	1	0.93
3	0.84	0.93	1

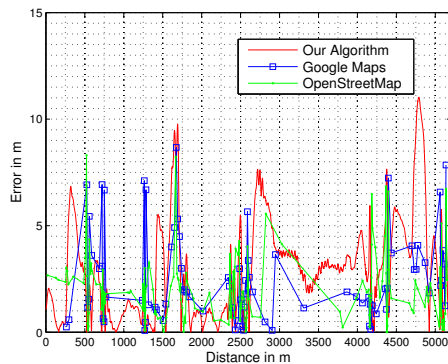
TABLE II: Set Error Normalized Maximum Cross-Correlation

estimation error. The computations were performed on Matlab. As it can be seen, a reduced l_u tends to decrease the estimation error but increases significantly the computation time. Altogether, we chose $l_u = 15$ as it offers a good trade-off between both performance metrics. Figure 6b proposes a similar analysis but is focused on the influence of η , which controls the neighbor supporting points updated after each measurement. A reduced neighborhood decreases the computation time. The influence on the error is less significant as it remains stable from 325m to 2500m. As a consequence we picked $\eta = 325$. For algorithm parameters $\{l_u, \eta\} = \{15, 325\}$, the overall runtime is 167s, *i.e.* an average measurement update time of 14ms. The original implementation of Kalman filtering of spline-based road modeling [8] involves to use all the supporting points in the road network. In our case, it would involve to choose $\eta = 2600$. As it can be noticed in Figure 6b, it would take 334sto process the entire dataset. Our approach enables to divide by two the processing time while maintaining the same error.

Figure 7 analyzes the estimation error with respect to the



(a) Individual Orthogonal Error Analysis



(b) Error Comparison of our approach versus Google Maps and OpenStreetMap vectorial data

Fig. 7: Spatial Spline Model Estimate Error

reference trace. In Figure 7a, we compare the error estimate for the three datasets. On the one hand, large errors happen in the same regions among the different sets. It indicates that local phenomena such as road environment influence the GPS position measurement. On the other hand, seasonal effects can be noticed. In fact, Set 2 and 3 curves have a behavior similar to each other (c.f. Table II). It results from noise on the GPS and GLONASS signals themselves. Figure 8 reveals that the error distributions are different for the three sets. Set 3, which was recorded simultaneously to the reference trace, achieves the smallest median position error. However, its error dispersion is also the largest of all. It can be explained by the fact that less traces were used to perform the estimation. The fusion of the three sets is really important as it allows to achieve the same median position error while reducing the error dispersion as well as the outlier spread.

In Figure 7b, we compare the performance of our estimation based on 24 GPS traces to Google Maps and OpenStreetMap vectorial data. We measured the position error of the nodes with respect to the reference trace. Vectorial data are obtained from the fusion of governmental data and GPS traces collected by the map provider. Governmental data are obtained by land surveying. As a result, it may takes a considerable amount of time to collect the data over a given itinerary. As mentioned in the previous paragraph, less large errors can be seen on the red curve compared to the graphs

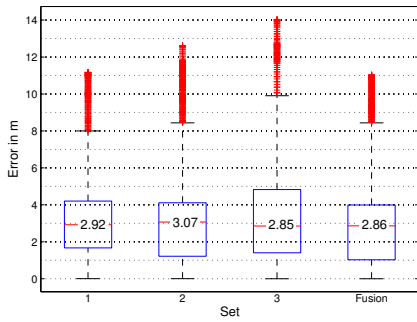
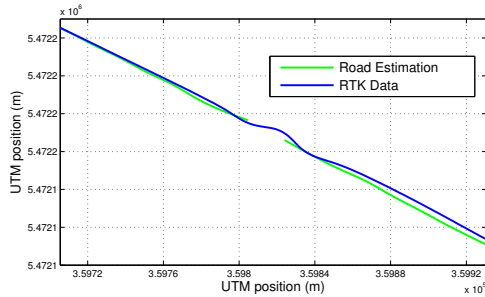
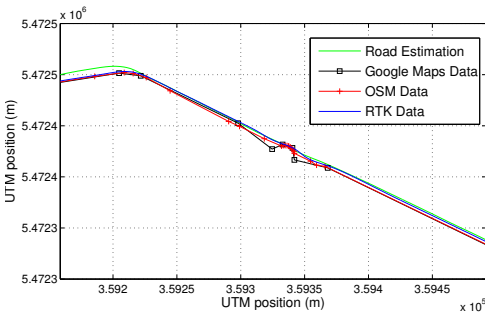


Fig. 8: Error Analysis



(a) Open loop formulation



(b) Vectorial maps - Local comparison
OSM: OpenStreetMap

Fig. 9: Local Results

in Figure 7. We can notice similar local errors. On the one hand, our algorithm outperforms the vectorial data in regions where the road is straight (c.f. Figure 9b). On the other hand, the error is lower for the vectorial data in region where the direction changes. This error originates from the sampling rate of our GPS chip. Moreover, filters used in the GPS chips tend to smooth GPS traces in order to remove the noise induced by the error on the GPS signals. While we achieve a median error of 2.86m, Google Maps and OpenStreetMap data reach 1.90m and 1.62m error respectively. Our performance is fair as we solely rely on GPS data. As a result, our framework provides an interesting alternative to obtain preliminary map information where vectorial information are outdated or not available.

IV. CONCLUSION

In this paper, we investigated an approach to use crowd-sourced vehicular GPS data to build preliminary maps for autonomous vehicle mission. In the first part, we propose a fast implementation of Kalman filtering for road spline model estimation. Our results show a significant speedup of the computation and an improvement of error by choosing the appropriate supporting point density. The second part highlights that the fusion of GPS traces captured from low cost devices improves the estimation error. The fusion of sets taken at different time of year also contributes to improve the estimation quality. Despite different drivers and vehicles were used, we achieve a 2.8m median error on 5.2km long test itinerary. Our approach achieves a fair performance as compared to vectorial information obtained from the aggregation of a large amount of data. Future works will be focused on extended our results to more tracks and multiple scenarios. Geo-referenced images will also be added to improve the spline-based road model estimation. Moreover, vehicular dynamics will be used in the Kalman filter to constraint the resulting estimated road curve.

REFERENCES

- [1] A. Geiger, M. Lauer, F. Moosmann, B. Ranft, H. Rapp, C. Stiller, and J. Ziegler, "Team AnnieWAY's Entry to the 2011 Grand Cooperative Driving Challenge," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 13, no. 3, pp. 1008–1017, Sept 2012.
- [2] J. Funke, P. Theodosis, R. Hindiyeh, G. Stanek, K. Kritatakirana, C. Gerdes, D. Langer, M. Hernandez, B. Muller-Bessler, and B. Huhnke, "Up to the limits: Autonomous Audi TTS," in *Intelligent Vehicles Symposium (IV), 2012 IEEE*, 2012, pp. 541–547.
- [3] J. Biagioni and J. Eriksson, "Inferring Road Maps from Global Positioning System Traces," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2291, no. 1, pp. 61–71, 2012.
- [4] G. Agamennoni, J. Nieto, and E. Nebot, "Robust and accurate road map inference," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, 2010, pp. 3946–3953.
- [5] L. Cao and J. Krumm, "From GPS traces to a routable road map," in *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2009, pp. 3–12.
- [6] J. Biagioni and J. Eriksson, "Map inference in the face of noise and disparity," in *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*. ACM, 2012, pp. 79–88.
- [7] C. Hasberg and S. Hensel, "Online-estimation of road map elements using spline curves," in *Information Fusion, 2008 11th International Conference on*, June 2008, pp. 1–7.
- [8] C. Hasberg, S. Hensel, and C. Stiller, "Simultaneous Localization and Mapping for Path-Constrained Motion," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 13, no. 2, pp. 541–552, June 2012.
- [9] R. T. Farouki and T. Sakkalis, "Real rational curves are not unit speed," *Computer Aided Geometric Design*, vol. 8, no. 2, pp. 151 – 157, 1991. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/016783969190040I>
- [10] H. Wang, J. Kearney, and K. Atkinson, "Arc-Length Parameterized Spline Curves for Real-Time Simulation," in *Curve and Surface Design*, 2002.
- [11] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2006.