Localization Confidence Domains via Set Inversion on Short-Term Trajectory

Vincent Drevelle, Philippe Bonnifait

Localization confidence domains via set-inversion on short-term trajectory

Vincent Drevelle and Philippe Bonnifait

***Abstract*—The knowledge of localization uncertainties is of prime importance when the navigation of intelligent vehicles has to deal with safety issues. This paper presents a robust estimation method able to quantify the localization confidence based on interval analysis and constraint propagation. Firstly, tightly-coupled position domains are computed by constraint propagation on GPS measurements and a precise 3D map of the drivable area. Since GPS is prone to satellite masking and wrong measurements in urban areas, a second stage provides localization integrity and information availability, by the use of a position and proprioceptive data history. A robust constraint propagation algorithm is employed to compute the current vehicle pose. It is able to handle erroneous positions with a chosen integrity risk. Experiments carried out in urban canyons illustrate the performance of the method in comparison with a particle filter. Despite bad satellite visibility, full positioning availability is obtained and errors are less than 5.1 m during 95% of the trial. In opposition to the particle filter, confidence domains are consistent with ground truth which confirms the high integrity of the method.**

***Index Terms*—localization, sensor fusion, GPS, 3D map**

I. Introduction

LOCALIZATION (also called ego-localization or positioning) is a prerequisite for mobile robot navigation in large areas [1]. Given a known map of the environment, the knowledge of the pose of the robot is firstly necessary for path planning. Secondly, metric localization can also been used for navigating in a constrained environment, if a map of the drivable space is known.

In this paper, we consider the navigation of intelligent vehicles in urban areas without dedicated beacons installed in the infrastructure. The goal is to reach a meter-level accuracy with high integrity and availability for way-point navigation. For this, we study the use of affordable dead-reckoning sensors, like odometers and gyros, merged with standalone Global Positioning System (GPS). Because of frequent masking of the GPS satellites that can reduce drastically the number of position fixes, pseudoranges are directly exploited in a tightly coupled manner. Recent advances related to mapping make possible to use precise 3D maps of the road network, surveyed in global coordinates with many details that allow the charting of the drivable space. If the vehicle always remains located in the drivable space of such a 3D map, this information can be exploited in the localization process as a position constraint. We describe a data fusion method that is able to exploit this prior contextual information simultaneously with the hybridization of GPS with dead-reckoning (Fig. 1).

Vincent Drevelle and Philippe Bonnifait are with Heudiasyc, University of Technology, Compiegne, France, e-mail: vincent.drevelle@tremplin-utc.net.

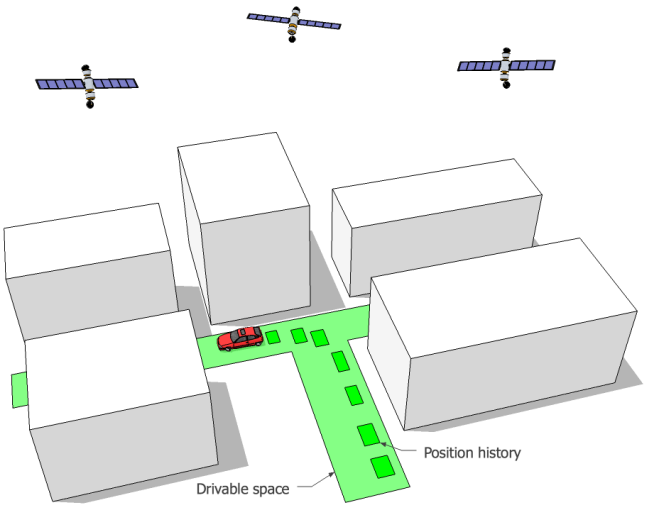


Figure 1. Localization problem constrained by the drivable space (in green) and a short-term history of positions (small rectangles).

Compared with Simultaneous Localization and Mapping (SLAM), localization is the process that refers to pose inference from known landmarks. When using radio-navigation devices based on Received Signal Strength Indicator, Time of Flight, Time of Arrival or Time Difference of Arrival measurements [2], there is no data association issue [3]. When beacons are passive, the sensor measurement association problem is easy to address with highly discernible landmarks. Examples in computer vision systems applied to outdoor navigation are the use of natural sparse features [4] or roadsigns [5].

From the integrity point of view, using exteroceptive measurements for robot localization is more risky than using proprioceptive sensors, because the robot is interacting with other systems. Exteroceptive observations can be then erroneous. For instance, radio signals are subject to multipath propagation and optical sensors are sensitive to adverse conditions. Moreover, when exploiting passive landmarks in the perception process, other sources of error arise because of landmark location errors, modification/displacement and wrong association in case of ambiguity. When the robot is used for human beings transportation, localization methods have to provide quantified position uncertainty linked to confidence figures. This has to be done in real-time since position uncertainty may vary a lot with environmental factors and geometrical configuration. Integrity quantification refers to all of these notions. The

localization method should then provide a confidence domain in real-time to the navigation module, considering the sources of error: sensor noise, spurious measurements, landmark position uncertainty, association errors, failures, etc. Monitoring confidence domains is a very efficient method to check that the localization uncertainty meets the navigation requirements specification.

In case of autonomous navigation, this can be achieved by making use of redundancy [6]. One solution is to exploit several redundant sensors, in particular proprioceptive sensors like done in [7] for fusing GPS with an inertial measurement unit (IMU). Another is to use a map of the environment as a source of information [8]. When the number of exteroceptive measurements is very low, another strategy is to do the processing on a trajectory or a position history [9].

In this paper, we focus on estimation and propagation of confidence domains. If the information from the sensor is bounded and if the bounds are well estimated, a classical set-membership predictor/estimator is well adapted. In practice, error bounds on sensor measurements and model parameters cannot be guaranteed. In this case, the confidence that can be placed on the computed domains decreases monotonically towards zero as time approaches infinity [10]. In opposite, if the system avoids the recursive processing of the date by handling a limited amount of sensor information, it is possible to keep constant the risk on the estimated domain.

In order to do a constant-risk fusion of the different information sources, we propose an interval-based method able to do the estimation process on a trajectory horizon. Indeed, interval based methods are well suited for non-linear problems [11]. Moreover, they enable to solve problems with multiple hypotheses solutions as shown by [12] for detecting ambiguous initial localization. Interval methods have also the advantage to manipulate sets, which allows to efficiently process entire domains of the state space. Finding adequate error bounds can be pessimistic or even impossible particularly when outliers may arise. A way to address this issue is to add robustness to the method by explicitly relaxing some constraints like proposed by Jaulin in [13].

The paper is organized as follows. After a general problem statement description, classical Bayesian pose estimation methods are reviewed and a constraint particle filter solver is presented for comparison purposes. Interval analysis and set-inversion with contractors are introduced in section III. Then, a two-stage real-time pose estimation strategy is presented. The first stage consists in a tightly coupled GPS - 3D map set-membership positioning method that uses time-of-flight GPS measurements, the bounds of which are dynamically chosen in order to keep constant the risk. In section V, a robust pose estimation scheme based on a position and proprioceptive data history is proposed and developed. Section VI explain how to choose the interval bounds given an integrity risk. Finally, experimental results processed in real-time are reported and compared with the particle filter.

II. Classical approach

A. Problem statement

In this paper, we aim at estimating in real-time the 2D pose of a vehicle equipped with a GPS receiver and proprioceptive sensors, with the help of a 3D-map $\mathcal{M}$ that acts as a constraint. This can be formulated using an evolution model $\mathbf{f}$ that involves proprioceptive sensors $\mathbf{u}$ and an observation model $\mathbf{g}$ in which exteroceptive measurements $\mathbf{y}$ are used:

$$\begin{aligned}\mathbf{x}_{k+1} &= \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)\\ \mathbf{y}_k &= \mathbf{g}(\mathbf{x}_k)\\ \mathbf{x}_k &\in \mathcal{M}\end{aligned}$$

The *drivable space* $\mathcal{M}$ is the surface on which the vehicle can physically evolve. For a car, the drivable space can be defined as the surface of the roadway, delimited by the sidewalks. Obstacles like poles or lane separators are also excluded from the drivable space. The raw drivable space can be eroded to take the size of the vehicle into account.

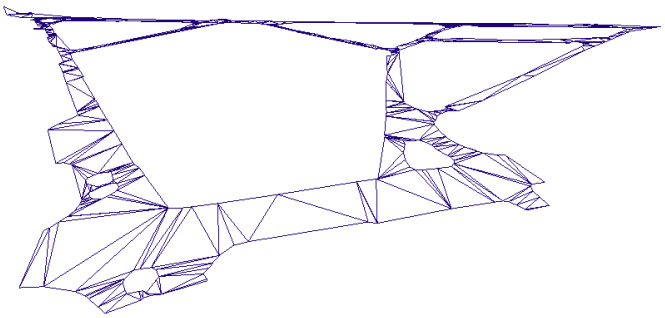


Figure 2. 3D wireframe view of the drivable space extracted from a database

The drivable space is represented in 3D by a triangular mesh, i.e. a surface made of connected triangular facets (Fig. 2). Vertices are represented by their 3-D coordinates, while facets are defined by a list of three vertices. We assume that the vehicle only evolves on the represented drivable space, which provides a very strong constraint on the position.

The vehicle is assumed to move without slipping in an horizontal planar world. Thus, pose estimation only consider the vehicle's planar coordinates (e, n) and heading ψ. The discrete non-holonomous evolution model $\mathbf{f}$ of the pose $\mathbf{x} = (e, n, \psi)$ between two samples (let denote $T_s = t_{k+1} - t_k$) is

$$\left\{\begin{array}{rl} e(t_{k+1}) = & e(t_k) + T_s \cdot v(t_k) \cdot \cos\psi(t_k) \\ n(t_{k+1}) = & n(t_k) + T_s \cdot v(t_k) \cdot \sin\psi(t_k) \\ \psi(t_{k+1}) = & \psi(t_k) + T_s \cdot \omega(t_k) \end{array}\right. \quad (1)$$

Vehicle speed and yaw-rate form the input vector $\mathbf{u} = (v, \omega)$. Vehicle speed $v(t)$ can be measured from wheel speed sensors, and, if the vehicle is equipped with a gyro, yaw-rate $\omega(t)$ can also be directly measured.

Figure 3 describes a distributed estimation fusion. GPS raw data is first fused with the map altitude information, which allows to compute a point position with down to 3 time-of-flight measurements from visible satellites (*pseudoranges*). This computation is purely static. Then, the result is used to correct the drift of an odometric prediction (Eq. 1), which is

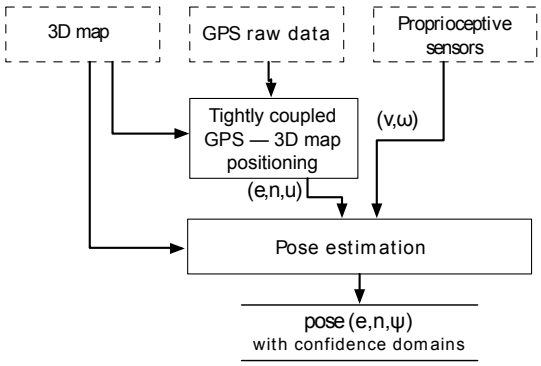


Figure 3. System overview

aided itself by the drivable space constraint. So, the system combines low sample rate position measurements with high rate proprioceptive data. Pose estimation and confidence domains have to be output at the required frequency.

B. Bayesian approach

Bayesian state filtering is a popular technique for solving localization problem [14]. Non-linear Kalman filtering is the most common estimation method. It can be robustified against measurement outliers in different ways like gating the innovation signal to reject bad observations [7] or by modeling, in the distribution of the noise, the possible occurrence of outliers using Student's t-distributions [15]. When considering prior map as a constraint, Kalman filters can still be used as the core fusion engine by adding a constraint stage after the update, like a two-step projection method proposed in [16] and used by [17] to estimate the location of a vehicle within two circular road segments.

Particle filtering has also been applied to similar problems. Bootstrap with particle pruning has been studied with a facet-based representation of the drivable space in [18]. It has also been applied to clothoidal surfaces with a representation of the connections between parallel driving lanes [19], in which a special attention has been given to the gyro bias.

We present hereafter a particle filter that addresses the problem described in Section II-A. It will be used as a reference for comparison with the proposed approach.

C. Constrained Particle Filter

As stated above, a constrained particle filter (CPF) can solve the pose estimation problem. Following the same bootstrap paradigm as in [20], at each time step, particles evolve in open loop, following the non-holonomous evolution model of Equation 1.

If the signals of at least 3 satellites are received, a least square estimate of the position is computed. The altitude is derived from the map and introduced as a measurement into the computation [21]. Particles weights are then updated with the GPS position likelihood. As GPS measurements are correlated over time, the state can be augmented with a first order autoregressive shaping filter, similar to the one used in [20] with a correlation time of 60 s and a reinitialization each time the visible satellite-set changes.

Particles weights are finally updated with a trapezoidal map likelihood as follows. Likelihood of particles that lie on the road is set to one. It tapers off linearly as particle distance to the road increases up to 50 cm. Particles that are more than 50 cm off the drivable space are so discarded.

Algorithm 1 details the main steps of the method for each time step.

Algorithm 1 Constrained Particle Filter

for every particle $\mathbf{x}_k^i$ **do**
  $\mathbf{x}_k^i \leftarrow \mathbf{f}(\mathbf{x}_{k-1}^i)$ *// Odometric evolution model*
  if $n_{sat} \geq 3$ **then**
    $\mathbf{x}_{GPS} \leftarrow \mathrm{GPS\,solver}(\mathrm{pseudoranges,\ map})$
    $w_k^i \leftarrow w_{k-1}^i \cdot \mathrm{Gaussian\,likelihood}(\mathbf{x}_k^i, \mathbf{x}_{GPS})$
  end if
  $w_k^i \leftarrow w_{k-1}^i \cdot \mathrm{Trapezoidal\,likelihood}(\mathbf{x}_k^i, \mathrm{map})$
end for
Normalize weights
Resample if necessary
Compute estimate and confidence domain

Well-known drawbacks of particle filtering are the risk of losing the correct solution during the tracking process the the over-condensation of the particles on the edges of the drivable space constraint, which mainly affects the integrity of confidence domains (as we will see later on in Fig. 20b). Another critical issue in particle filtering is the initialization, which needs often many particles to evenly fill the position space.

Bounded error approaches can provide efficient tools to avoid these kinds of behaviors, by working with outer envelopes. In the following, we present one of these approaches based on intervals.

III. Interval Analysis and Set-Inversion

Data can be represented by intervals and boxes in processing stages where uncertainty has to be taken into account. This allows propagating errors from the measurements to the pose estimation. To model inaccuracy, positions are represented as boxes which should contain the true location with a given confidence level. Measurements are represented by intervals to take noise into account. This section presents the basic concepts needed to develop the method.

A. Interval analysis

Interval analysis [22] involves intervals and their multidimensional extension, *interval vectors* (or *boxes*). In opposition to an exact representation of sets, intervals and boxes are easy to manipulate. The set of real intervals is denoted $\mathbb{IR}$, and the set of n-dimensional boxes is $\mathbb{IR}^n$. In this paper, an interval or a box $[\mathbf{x}] = [\underline{\mathbf{x}}, \overline{\mathbf{x}}]$ is written between brackets; $\underline{\mathbf{x}}$ and $\overline{\mathbf{x}}$ respectively denote the lower and upper bounds of $[\mathbf{x}]$. The width of an interval $[x]$ is $\overline{x} - \underline{x}$ and the width of a box is the largest width of its interval components.

Interval arithmetic enables computations with intervals thanks to the interval extension of classical real arithmetic

operators $+, -, \times$ and $\div$.

$$[x] + [y] = [\underline{x} + \underline{y}, \overline{x} + \overline{y}], \quad [x] - [y] = [\underline{x} - \overline{y}, \overline{x} - \underline{y}]$$

In the same way, elementary functions such as *tan, sin* and *exp* can be extended to intervals. This is done by returning the smallest interval covering the range of the input through the function.

The image of a box by a function $\mathbf{g} : \mathbb{R}^n \to \mathbb{R}^m$ is generally not itself a box, but an arbitrary set. This problem is solved using the so-called *inclusion functions*. The interval function $[\mathbf{g}]$ from $\mathbb{IR}^n$ to $\mathbb{IR}^m$ is an *inclusion function* for $\mathbf{g}$ if the image of $[\mathbf{x}]$ by $\mathbf{g}$ is included in the image of $[\mathbf{x}]$ by $[\mathbf{g}]$, *i.e.*

$$\forall [\mathbf{x}] \in \mathbb{IR}^n, \mathbf{g}([\mathbf{x}]) \subset [\mathbf{g}]([\mathbf{x}]).$$

The minimal inclusion function $[\mathbf{g}^*]$ for a function $\mathbf{g}$ returns the smallest box that contains $\mathbf{g}(\boldsymbol{x})$ — *i.e.* $[\mathbf{g}^*]([\mathbf{x}]) = \square\mathbf{g}([\mathbf{x}])$, the *interval hull* of $\mathbf{g}([\mathbf{x}])$ [22]. The natural inclusion function for g is obtained by replacing each operator in the expression of g by its interval counterpart. If each variable occurs only once in the expression of g, then the natural inclusion function is minimal.

The intersection of two boxes is an box. Since the union of two boxes is not necessarily a box (e.g. $[1, 2] \cup [3, 4]$), let us define the box union $\sqcup$ which returns the hull of the union of two boxes (e.g. $[1, 2] \sqcup [3, 4] = [1, 4]$).

To approximate compact sets in a guaranteed way, *subpavings* can be used. A subpaving of a box $[\mathbf{x}]$ is the union of non-empty and non-overlapping sub-boxes of $[\mathbf{x}]$. A guaranteed approximation of a compact set X can be made by bracketing it between an inner subpaving $\underline{X}$ and an outer subpaving $\overline{X}$ such as $\underline{X} \subseteq X \subseteq \overline{X}$.

B. Contractors and Constraint Propagation

When the components of a vector $\mathbf{x}$ are linked by relations or constraints, one can define a *constraint satisfaction problem* (CSP). It consists in finding the solution set $X = \{\mathbf{x} \in [\mathbf{x}] | \mathbf{g}(\mathbf{x}) = \mathbf{0}\}$, where $[\mathbf{x}]$ is the domain of the variables and $\mathbf{g}(\mathbf{x}) = \mathbf{0}$ represents the constraints, and can also represent inequalities by introducing slack variables [22].

A *contractor* $\mathcal{C}$ for a CSP is an operator that computes a smaller domain $[\mathbf{x}_c] = \mathcal{C}([\mathbf{x}])$ without affecting the solution set, *i.e.* $X \subset [\mathbf{x}_c] \subset [\mathbf{x}]$. There are many ways to implement a contractor. One of them is the forward-backward contractor based on constraint propagation [23].

C. Set-inversion via interval analysis

The set inversion problem consists in determining the set $X = \mathbf{g}^{-1}([\mathbf{y}]) = \{\mathbf{x} \mid \mathbf{g}(\mathbf{x}) \in [\mathbf{y}]\}$, where $[\mathbf{y}]$ is a known interval vector of m measurements. Using interval analysis, the solution X can be approximated between two subpavings $\underline{X}$ and $\overline{X}$ such that $\underline{X} \subseteq X \subseteq \overline{X}$. The *SIVIA* algorithm allows performing such a set inversion, by recursively bisecting an initial box [22].

Since we are seeking to characterize the positioning confidence domain, we only need to compute the outer subpaving $\overline{X}$ of the set that fulfills positioning constraints. Algorithm 2 implements a SIVIA that only computes an outer approximation $\overline{X}$ of the solution set in a given domain $[\mathbf{x}_0]$, given an inclusion function $[\mathbf{g}]$ for $\mathbf{g}$ and a contractor $\mathcal{C}_{\mathbf{g},[\mathbf{y}]}$. It uses a list $\mathfrak{L}$ in which boxes are stored an retrieved. When $\mathfrak{L}$ is implemented as a stack, the algorithm employs a depth-first strategy. If $\mathfrak{L}$ is a queue, then the algorithm uses a breadth-first strategy. ϵ controls the sharpness of the subpaving $\overline{X}$. Boxes larger that ϵ after contraction are bisected along the dimension of largest-width component.

Algorithm 2 SIVIA(in: $[\mathbf{x}_0], [\mathbf{y}], \epsilon$)

```
1:  X̄ ← ∅    // Empty subpaving
2:  Store [x_0] in 𝔏
3:  while 𝔏 is not empty do
4:     Retrieve a box [x] from 𝔏
5:     if [g]([x]) ⊂ [y] then
6:        X̄ ← X̄ ∪ [x]    // [x] is included in X
7:     else
8:        [x] ← C_{g,[y]}([x])    // Contract the box
9:        if Width([x]) < ε then
10:          X̄ ← X̄ ∪ [x]
11:       else if [x] ≠ ∅ then
12:          ([x_1], [x_2]) ← bisect([x])
13:          Store [x_1] in 𝔏; Store [x_2] in 𝔏
14:       end if
15:    end if
16: end while
17: return X̄
```

IV. Tightly coupled GPS – 3D map positioning

The proposed bounded error approach follows the same distributed estimation fusion strategy as described in Figure 3. Let us start by studying the fusion of GPS pseudorange measurements with the 3D drivable space constraint.

A. Drivable space constraint

Being located on a 3D triangular facet can be expressed by four simple constraints: one constraint representing the facet plane, and three constraints for the edges. Optimal

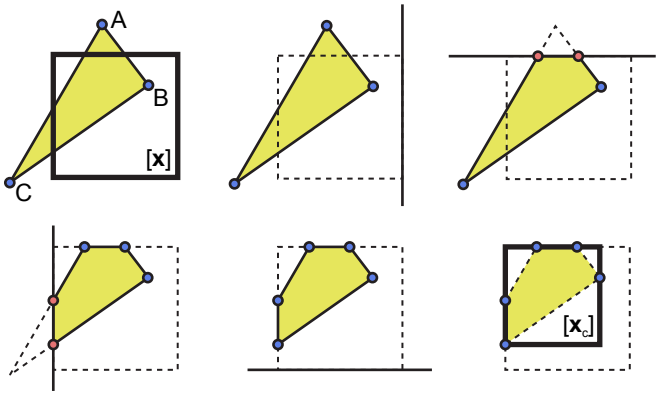


Figure 4. Contraction of box $[\mathbf{x}]$ with facet ABC. Facet is clipped with the Sutherland-Hodgman algorithm. $[\mathbf{x}_c]$ is the contracted box.

contraction of a box with respect to a single 3D facet can be performed with the Sutherland-Hodgman polygon clipping algorithm [24]. It consists in successively clipping the facet with each plane of the box, by finding intersections and discarding exterior vertices (Fig. 4). The considered facet is clipped with the box; then the bounding box of the resulting polygon is returned as the contraction of the original box.

The whole road constraint is simply the union of the constraints from each facet of the map $\mathcal{M}$. Algorithm 3 contracts a box with the entire road. To speed up the road contractor, facets are stored with their bounding boxes, to enable quick pruning of incompatible facets. The *extract_facets* function returns the set of facets whose bounding box intersects the prior box; It enables to focus only on the interesting part of the map in order to save computation time. Map inaccuracy is handled by using interval coordinates for the vertices (typically ± 5 cm horizontally).

Algorithm 3 road_contract(in: $[\mathbf{x}], \mathcal{M}$)

$[\mathbf{xc}] \leftarrow \emptyset$
$\mathcal{F} \leftarrow \text{extract\_facets}(\mathcal{M}, [\mathbf{x}])$
for each f in $\mathcal{F}$ **do**
  $[\mathbf{xc}] \leftarrow [\mathbf{xc}] \sqcup \text{facet\_contract}([\mathbf{x}], f)$
end for
return $[\mathbf{xc}]$

B. GPS pseudorange constraints

GPS positioning is a *Time of Arrival* method, which involves *pseudorange* measurements from each visible satellite [21]. Pseudoranges are ranges offset by a unknown amount due to the time base difference between the receiver and the GPS system. GPS positioning using pseudoranges is thus a four-dimensional problem: along with the "East, North, Up" Cartesian coordinates (e, n, u) of the user in a local tangent frame, the receiver clock offset dt_u has to be estimated. Satellite positions at their emission times can be retrieved from broadcast ephemeris data using well known procedures [25]. A frame transformation is then applied to have the satellites positions $(e_i^{\text{s}}, n_i^{\text{s}}, u_i^{\text{s}})$ in the working frame.

Corrections are applied to measured pseudoranges to compensate for relativity and atmosphere propagation delays. Corrected pseudoranges ρ_i are imprecise because of model and parameter errors. Moreover, the receiver measurement errors should also be taken into account. Corrected pseudorange measurements are thus represented as intervals $[\rho_i]$. Each pseudorange introduces a constraint on the solution. The constraint induced by the i^{th} pseudorange measurement is represented by the natural inclusion function of the GPS pseudorange observation function (c is the speed of light):

$$[g_i]([e],[n],[u],[dt_u]) = \sqrt{([e]-[e_i^{\text{s}}])^2 + ([n]-[n_i^{\text{s}}])^2 + ([u]-[u_i^{\text{s}}])^2} + c \cdot [dt_u] \quad (2)$$

A forward-backward contractor [23] allows constraints to be propagated in an optimal order for each measurement, using (2). Constraints are split in an elementary constraints tree, and constraint propagation is performed from the leafs to the root, then back to the leafs (Fig. 5). Since there is no multi-occurrence of a variable in the expression (2), this is an optimal contraction.

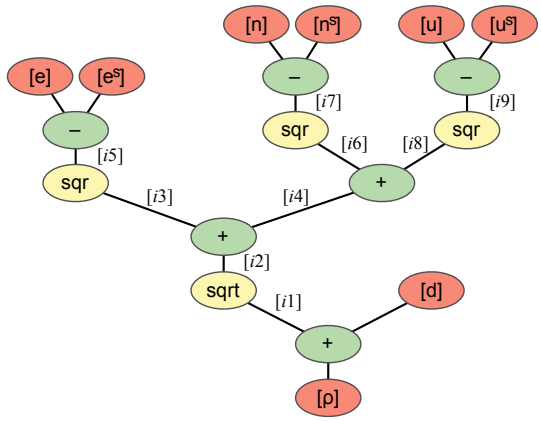


Figure 5. Elementary constraint decomposition of pseudorange constraint. Intervals $[i_j]$ represent represent auxiliary variables in the computation.

A GPS contractor can be built by successively applying the pseudorange contractor with each available measurement, until a fixed contracted box is obtained. Because of the loops in the GPS positioning constraint graph, pseudorange contractors have to be applied several times to reach a fixed solution.

C. Position computation

Position computation is then performed using the *SIVIA* algorithm presented previously (Alg. 2) with the proposed road and GPS pseudorange contractors (Alg. 4). This algorithm computes an outer approximation of the set that satisfies both the GPS and road constraints.

Algorithm 4 $\mathcal{C}_{road\_gps}$(in: $[\mathbf{x}], \mathcal{M}, [\rho]_{1\ldots m}, [\mathbf{x}^s]_{1\ldots m}$)

repeat
  $[\mathbf{x}] \leftarrow \text{road\_contract}([\mathbf{x}], \mathcal{M})$
  $[\mathbf{x}] \leftarrow \text{gps\_contract}([\mathbf{x}], [\rho]_{1\ldots m}, [\mathbf{x}^s]_{1\ldots m})$
until no more significant contraction can be done on $[\mathbf{x}]$
return $[\mathbf{x}]$

To reduce positioning ambiguities and limit the number of road facets used for contraction, a road-topology aware facet selection step can be implemented before calling the SIVIA algorithm [26]. Similarly to a a map-matching algorithm, the topological facet selection step makes use of facet connections, covered distance and prior position to determine the candidate triangles of the map.

V. Robust pose estimation on data history

The pose estimation process is now done by constraint propagation on a data history, as shown in Fig. 6.

Map, positions and proprioceptive data are merged using a robust algorithm applied to a buffer of recent positions and proprioceptive data (Fig. 7). A data buffer management algorithm supervises buffers filling, keeping a reasonable buffer size and ensuring buffers hold enough information to estimate the pose.

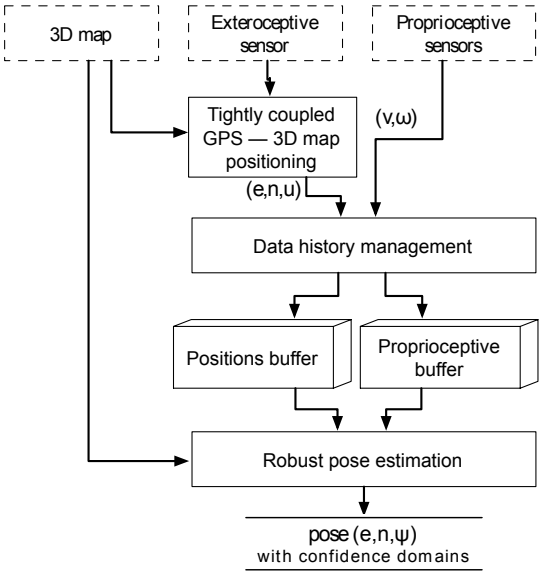


Figure 6. Data fusion system overview

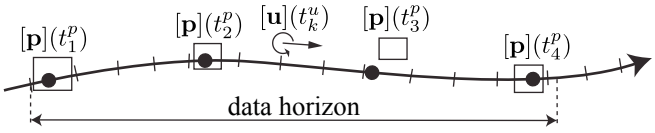


Figure 7. Data horizon. *Ticks on the trajectory represent proprioceptive measurements. Circles represent true position at GPS measurements epochs. The boxes show the computed GPS solutions. Notice an erroneous GPS position box at time t_3^p.*

A. Data history

To allow pose estimation based on a finite number of past observations, two data history buffers are used:

- the position history buffer, containing o position boxes: $L_p(t) = \{[\mathbf{p}](t_1^p), \ldots, [\mathbf{p}](t_o^p)\}$
- the list of proprioceptive inputs boxes, containing n boxes: $L_u(t) = \{[\mathbf{u}](t_1^u), \ldots, [\mathbf{u}](t_n^u)\}$, with $[\mathbf{u}](t) = ([v](t), [\omega](t))^T$, $v(t)$ and $\omega(t)$ denoting respectively the linear speed and the angular speed of the vehicle at the midpoint of the rear axle.

Each record in the data history buffer is timestamped with its time of acquisition, to allow variable acquisition rate processing [27]. It also enables to cope with out of sequence measurements [28].

Data history buffers are managed to keep a tractable size. The growth of the list of position observations is limited. When the size limit is reached, the oldest data is removed to make room for incoming data. The list of proprioceptive inputs is then cleaned from obsolete data related to the position previously discarded.

Since the absolute positions buffer is of limited size, adding new observation data implies forgetting older position data. Adding every position to the observation buffer may lead to heading estimation issues when the length of the buffered trajectory is in the same order of magnitude as the position boxes width. This problem arises when the vehicle slows down or stops: the system starts to accumulate redundant position observations, while discarding older parts of the trajectory and thus loosing the constraint on heading.

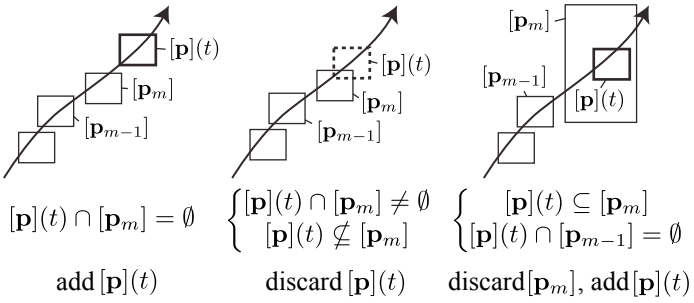


Figure 8. Data horizon management policy

To address this issue, observation buffer filling is based on a spatial criterion (Fig. 8). If the new position box to be added intersects the last position in the buffer, it is not added to the buffer. This rule can however lead to the loss of informative position information when the last position box is too large, preventing any new smaller (thus more informative) box to be added. To counteract this side-effect, the buffer filling policy is complemented by a second rule: if the box to be added is included in the last box of the buffer, and if it does not intersect the penultimate box of the buffer, then the buffer's last box is replaced by the new box. The spatial density of stored positions is controlled by enlarging or shrinking boxes with a *density* ratio before testing intersections.

B. q-relaxed intersection

In the presence of spurious measurements, a robust method has to be used. A robust set inversion method is the *q-relaxed set inversion*. The q-relaxed set inversion consists in tolerating a given number q of wrong measurements. The solution set, with m measurements, is thus the set of solutions *at least compatible with $m - q$ measurements*.

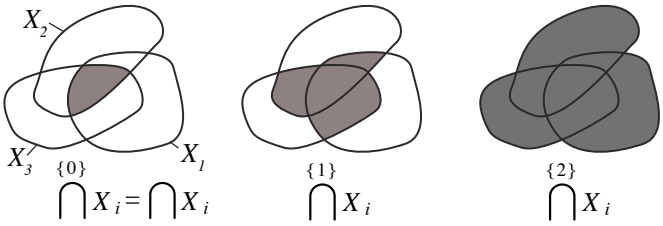


Figure 9. q-relaxed intersection of three sets for $q \in \{0, 1, 2\}$

Considering m sets $X_1, \ldots, X_m$ of $\mathbb{R}^n$, the *q-relaxed intersection* $\bigcap^{\{q\}} X_i$ is the set of $\mathbf{x} \in \mathbb{R}^n$ which belongs to at least $m - q$ of the X_i's (Fig. 9).

By considering the solution sets $X_i = g_i^{-1}([y_i])$ for each measurements, the *Robust Set Inverter via Interval Analysis* (RSIVIA) solver [22] guarantees the computation of a q-relaxed solution set $X = \mathbf{g}^{-1}([\mathbf{y}])$ with all the measurements. This algorithm returns an outer subpaving of the q-relaxed solution. Provided that a contractor $\mathcal{C}_i$ is available for each measurement, a contractor $\mathcal{C}_{relax}$ for a q-relaxed constraint

is presented in algorithm 5. A detailed explanation of the q-relaxed contractor is given in Appendix A.

Algorithm 5 $\mathcal{C}_{relax}$(in: $[\mathbf{x}_0]$, q, $\mathcal{C}_1$, ..., $\mathcal{C}_m$)
q-relaxed contractor for m constraints.

1: **for** $i = 1$ to m **do**
2: $[\mathbf{x}]^i \leftarrow \mathcal{C}_i([\mathbf{x}_0])$ // contract $[\mathbf{x}_0]$ w.r.t the i^{th} constraint
3: **end for**
4: $[\mathbf{x}] \leftarrow \square \left(\bigcap_{i\in\{1,\ldots,m\}}^{\{q\}} [\mathbf{x}]^i \right)$ *hull of the q-relaxed intersection of m boxes*
5: **return** $[\mathbf{x}]$

The q-relaxed set inversion is a robust method in the sense that the solution-set remains consistent with the truth as long as there are no more that q outliers in the measurements vector. If a measurement is wrong and inconsistent with the other measurements, it is automatically excluded from the solution, and it can be identified as an outlier. This way, when there is enough redundancy to allow fault detection and identification, the solution set is defined by the good measurements. When outlier rejection is not possible, the multiple solution hypotheses are combined in the solution set.

C. Robust pose estimation from previous positions

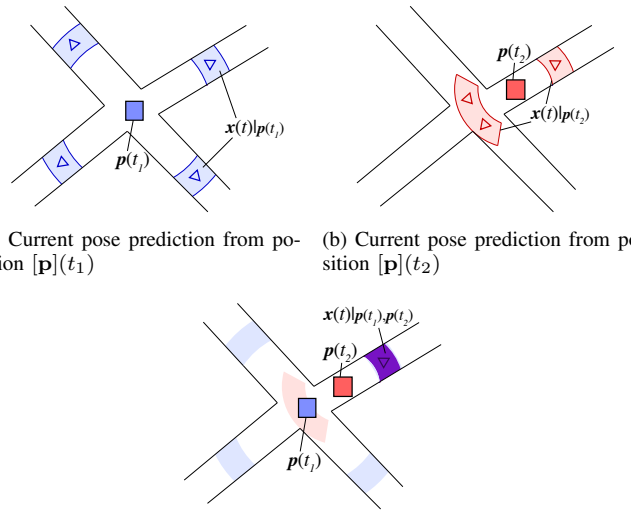
Each past position constrains the current pose (at time t) in a subset of the pose space. Fig. 10 shows the estimation of the current pose at time t, given two positions at past times t_1 and t_2, an estimate of the proprioceptive displacement and a map. Since the heading is not measured, the knowledge of a past position constrains the current pose inside a ring whose radius depends on the vehicle displacement. At each point of the ring corresponds a particular vehicle heading interval. The map information is an additional constraint for pose prediction (Fig. 10a). Moreover, the parts of the ring that do not correspond to a trajectory inside the drivable space are discarded (10b). As shown in figure 10c, when a history of past positions is available, the current pose $[\mathbf{x}](t)$ is located at the intersection of the constraints imposed by each position information:

$$[\mathbf{x}](t) = \bigcap_{k=1\ldots o} [\mathbf{x}^+](t) \mid [\mathbf{p}](t_k^p), [\mathbf{u}](t_1^u), \ldots, [\mathbf{u}](t_n^u) \quad (3)$$

where $[\mathbf{x}^+]$ denotes a prediction given a past position and the knowledge of the proprioceptive sensors.

To deal with erroneous positions in the data buffer, a given number q of erroneous measurements in the buffer is tolerated, using q-relaxed intersection. The robust pose estimation consists in computing the vehicle's position and heading at time t, given a finite number of prior position measurements and inputs, under the hypothesis that at most q position measurements are wrong. The robust pose estimate $[\mathbf{x}]^{\{q\}}(t)$ is given by Eq. (4).

$$[\mathbf{x}]^{\{q\}}(t) = \bigcap_{k=1\ldots o}^{\{q\}} [\mathbf{x}^+](t) \mid [\mathbf{p}](t_k^p), [\mathbf{u}](t_1^u), \ldots, [\mathbf{u}](t_n^u) \quad (4)$$



(a) Current pose prediction from position $[\mathbf{p}](t_1)$

(b) Current pose prediction from position $[\mathbf{p}](t_2)$

(c) Current pose is at the intersection of the two predicted poses

Figure 10. Estimation of current pose, given two absolute positions, an estimate of the proprioceptive displacement and a map. In this illustration, the displacement is supposed to be straight. Arrows indicate heading.

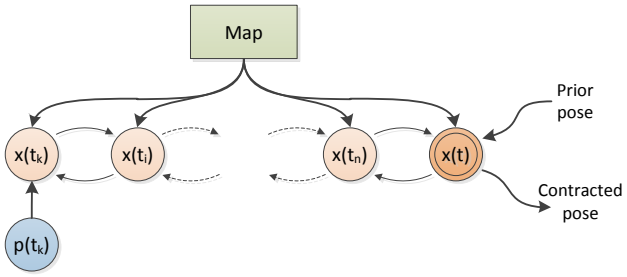


Figure 11. Backward-forward constraint propagation of pose $[\mathbf{x}](t)$ with one position $[\mathbf{p}](t_k)$ and proprioceptive measurements. Map constraint is applied at each time step.

Computation of $[\mathbf{x}^+](t) \mid [\mathbf{p}](t_k^p), [\mathbf{u}](t_1^u), \ldots, [\mathbf{u}](t_n^u)$ is done using a backward-forward constraint propagation with the natural inclusion function $[\mathbf{f}]$ of the evolution function given in Eq. (1), the map $\mathcal{M}$ and the proprioceptive inputs (see Alg. 6 and Fig. 11). Backward propagation is performed with the natural inclusion function of the inverted evolution model $\mathbf{f}_{inv}$ defined as above:

$$\begin{cases} e(t_k) = & e(t_{k+1}) - T_s \cdot v(t_k) \cdot \cos\psi(t_k) \\ n(t_k) = & n(t_{k+1}) - T_s \cdot v(t_k) \cdot \sin\psi(t_k) \\ \psi(t_k) = & \psi(t_{k+1}) - T_s \cdot \omega(t_k) \end{cases}$$

$[\mathbf{x}]^{\{q\}}(t)$ is computed by using the SIVIA algorithm with the q-relaxed contractor $\mathcal{C}_{relax}([\mathbf{x}_0], q, \mathcal{C}_{p_1}, \ldots, \mathcal{C}_{p_o})$, relaxing q constraints out of the o constraints from stored positions. Since each $[\mathbf{p}](t_k^p)$ only contains position information, $[\psi](t)$ cannot be estimated directly by independent contraction with respect to each $[\mathbf{p}](t_k^p)$. Thus, in the SIVIA algorithm, bisections are prioritarily performed on $[\psi](t)$ to enable heading estimation, exactly like done in [29], [27]. This way, the forward-backward contractor is called with prior heading values, which enables effective contraction as illustrated by Fig. 12. Since the contractors $\mathcal{C}_{p_k}$ and the q-relaxed contractor $\mathcal{C}_{relax}$ involve

Algorithm 6 $\mathcal{C}_{p_k}$(in: $[\mathbf{x}](t_n^u), [\mathbf{p}](t_k^p), L_u, \mathcal{M}$)
Backward-forward constraint propagation with one position, proprioceptive measurements and map constraint

find j that verifies $t_{j-1}^u < t_k^p \leq t_j^u$
for $i = n-1$ down to j **do**
  $[\mathbf{x}](t_i^u) \leftarrow [\mathbf{f}_{inv}]([\mathbf{x}](t_{i+1}^u), [u](t_i^u))$
  $[\mathbf{x}](t_i^u) \leftarrow \text{road\_contract}([\mathbf{x}](t_i^u), \mathcal{M})$
end for
$[\mathbf{x}](t_k^p) \leftarrow [\mathbf{f}_{inv}]([\mathbf{x}](t_j^u), [u](t_{j-1}^u))$
$[\mathbf{x}]_{1..2}(t_k^p) \leftarrow [\mathbf{x}]_{1..2}(t_k^p) \cap [\mathbf{p}](t_k^p)$
$[\mathbf{x}](t_j^u) \leftarrow [\mathbf{x}](t_j^u) \cap [\mathbf{f}]([\mathbf{x}](t_k^p), [u](t_{j-1}^u))$
for $i = j$ to $n-1$ **do**
  $[\mathbf{x}](t_{i+1}^u) \leftarrow [\mathbf{x}](t_{i+1}^u) \cap [\mathbf{f}]([\mathbf{x}](t_i^u), [u](t_i^u))$
  $[\mathbf{x}](t_{i+1}^u) \leftarrow \text{road\_contract}([\mathbf{x}](t_{i+1}^u), \mathcal{M})$
end for
return $[\mathbf{x}](t_n^u)$

computing the intersection of two poses, the heading angle interval is first transformed into a sine and cosine representation to simplify computations. The interval extension of the four-quadrant inverse tangent is used after the contraction to retrieve the angle.

VI. Integrity risk

A. Position estimates

Position estimates (e, n, u), see Fig. 6, are computed each time the receiver is tracking at least 3 satellites in line-of-sight.

Usually, confidence intervals are set on measurements once for all. However, as the number of measurements increases, the risk of having an outlier increases too. While a simple solution consists in setting the measurements error bounds to cover the worst case (the maximum number of visible satellites), this adds pessimism to the position solution.

Our approach consists in adapting the measurements intervals to keep a constant confidence in the result, regardless to the number of available pseudoranges. Thus, the bounds set on the pseudorange measurements depend on the number m of visible satellites and are adjusted to keep a given integrity risk r in the position solution. This risk corresponds to the probability that the box $[\mathbf{p}]$ does not include the unknown true position $\mathbf{p}$.

$$r = \Pr\left(\mathbf{p} \notin [\mathbf{p}]\right) \tag{5}$$

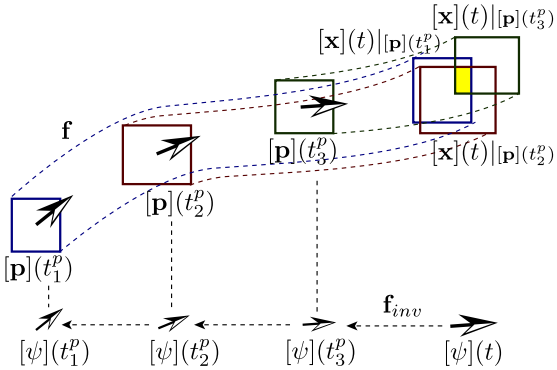
Assuming the pseudorange errors are independent, the risk $r_{[\rho]}$ on each measurement and the computation of the bounds $\pm\alpha\sigma$ can be directly derived using the following formula:

$$\begin{aligned} r &= 1 - (1 - r_{[\rho]})^m \\ r_{[\rho]} &= 1 - \sqrt[m]{1-r} \end{aligned} \tag{6}$$

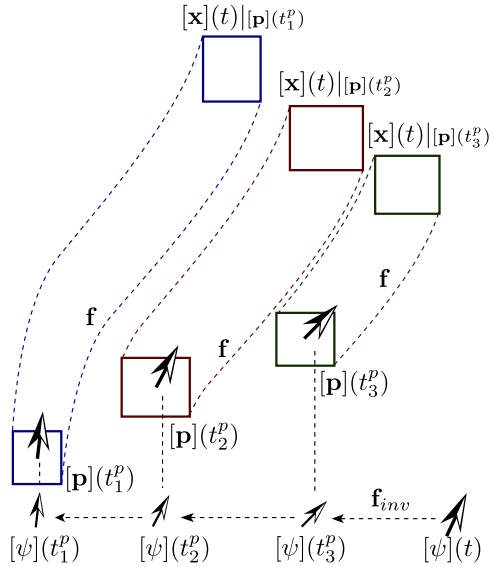
With Φ denoting the cumulative distribution function of the standard normal distribution, on can compute $\alpha = -\Phi^{-1}\left(\frac{r_{[\rho]}}{2}\right)$ and using the standard deviation σ provided by the GPS data, the measurement intervals are given by

$$[\rho] = [\rho - \alpha\sigma, \rho + \alpha\sigma].$$

Then, the position box is computed by set inversion with the contractor of section IV-C.



(a) Prior $[\psi](t)$ interval is compatible with measurements history, since the boxes after forward propagation intersect.



(b) Prior $[\psi](t)$ interval is inconsistent with measurements history. Intersection of boxes after forward propagation is empty.

Figure 12. Backward-forward constraint propagation with three position measurements, for two different prior values of $[\psi](t)$. Big arrows represent the midpoints of the heading intervals. Starting with $[\psi](t)$, headings $[\psi](t_i)$ at past times are computed using the inverse model $\mathbf{f}_{inv}$, and then forward propagation is done.

B. Constant-risk pose estimation from data history

As previously shown, data history is useful for robustness. It is also interesting for strengthening the hypotheses that are made regarding independence. Indeed, the spatial sampling strategy allows reducing the correlation of position errors, since multipath errors are location dependent. It also decreases time correlation coming from the receiver's low-level signal tracking stages.

It is established in [10] that the lower bound of the confidence in the pose estimation tends to zero when the number of used positions o tends to infinity, since each position in the history has an associated risk. In order to guarantee a minimal risk on the pose estimation result, it is crucial to limit the size of the position buffer. The ability to provide a solution with constant risk is a main feature of our method.

Let us consider from now on that the risk r associated with each position box in the buffer is constant:

$$r = \Pr\left(\mathbf{p}(t_k^p) \notin [\mathbf{p}](t_k^p)\right),\ k \in \{1 \dots o\} \tag{7}$$

Let N_{ok} be the number of position boxes that are consistent with the truth. The probability of having exactly i good positions out of o is given by the binomial distribution:

$$\Pr(N_{ok} = i) = \binom{o}{i}(1-r)^i r^{o-i} \tag{8}$$

where $\binom{o}{i} = \frac{o!}{i!(o-i)!}$ is the binomial coefficient.
Thus, by summing (8) over successive values of i, the probability of having at least $o-q$ good position boxes is:

$$\Pr(N_{ok} \geq o-q) = \sum_{i=o-q}^{o} \binom{o}{i}(1-r)^i r^{o-i} \tag{9}$$

The proposed pose estimation algorithm computes a guaranteed approximation $[\mathbf{x}](t)$ of the solution at time t. Moreover, if the number of spurious positions in the buffer is less than or equal to the number q of relaxed positions, then the solution is guaranteed to be consistent with the true pose $\mathbf{x}(t)$. This way,

$$N_{ok} \geq o-q \Rightarrow \mathbf{x}(t) \in [\mathbf{x}](t)$$

which leads to

$$\Pr(x(t) \in [x](t)) \geq \Pr(N_{ok} \geq o-q) \tag{10}$$

$$\Pr(x(t) \notin [x](t)) \leq 1 - \Pr(N_{ok} \geq o-q) \tag{11}$$

$$R \leq 1 - \sum_{i=o-q}^{o} \binom{o}{i}(1-r)^i r^{o-i} \tag{12}$$

where R denotes the maximum risk associated with a pose computed from the data history.

In practice, Eq. (12) can be used in several ways: one can choose the size o of the position buffer, the number of tolerated outliers q, and the confidence in each position box and then deduce the corresponding risk. Another way is to specify the risk R and then compute the corresponding r given o and q. In the reported results, r was fixed, o was limited by real-time constraints and q was set to one.

VII. Results

A. Experiment

Data acquisition was performed on the *Stereopolis* vehicle from the French *Institut Géographique National* (IGN), for the *CityVIP* research project. It consists in three laps of a 1 km loop in the 12th arrondissement of Paris, which roughly corresponds to a 15-minute drive around the local town hall (Fig. 13).

The drivable space is provided by the *IGN*. It is hand made from precise aerial photographs [30]. It has been converted from its original Lambert93 projection to the local working frame in which the GPS positioning is performed. We use GPS pseudorange measurements acquired at 2 Hz with a *Septentrio PolaRx2* receiver, all pseudoranges with L1 carrier to noise ratio below 35 dBHz were filtered out. Satellite positions and pseudorange corrections were computed with the open source *GPSTk* library [25]. Ground truth is provided by a post-processed *Applanix* inertial navigation system.

The acquired data is very challenging for autonomous GPS, since 85% of the measurement epochs have less than 4 satellites in use. There are no more than 2 satellites in use during 56% of the test run (Fig. 14).

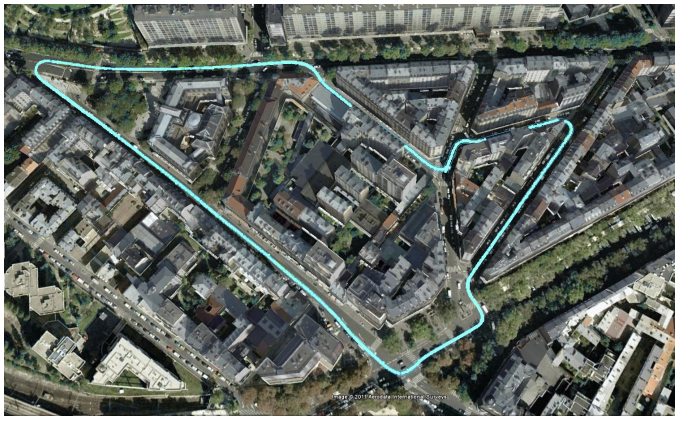
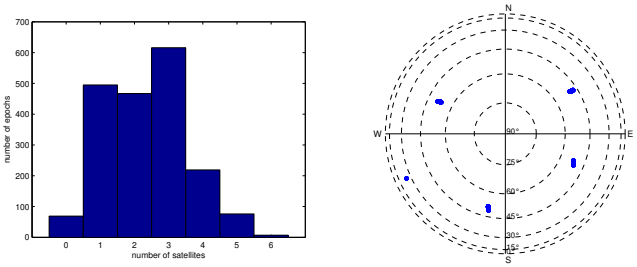


Figure 13. Trajectory around the 12th arrondissement town hall in Paris



(a) Histogram of the number of satellites (b) Skyplot during the first lap

Figure 14. Satellite visibility with $C/N_0 \geq 35$dBHz

B. Real-time implementation

SIVIA algorithm and contractors are implemented in C++. The bisection strategy is "largest box first", which enables to evenly explore the solution space, even if the computation has to be stopped after a timeout to satisfy real-time requirements. As more time is allotted to set inversion, the solution gets more precise, thanks to smaller boxes in the subpaving, which tend to reduce pessimism and wrapping effect.

Since computation on each box is completely independent from the other boxes of the subpaving, SIVIA benefits a lot from parallel processing. In our implementation, the workload is distributed in several threads to take advantage of multicore architectures.

The tightly coupled GPS-map fusion is a rather fast computation. On the contrary, the process of robust pose estimation on the data horizon is time consuming. The system has been designed to enable real-time positioning despite the high latency of robust pose estimation. For this purpose, two pose computation tasks are simultaneously running (Fig. 15):

- The robust pose estimation based on the map and the horizon of previous positions and proprioceptive data. Since it requires heavy computations, this task yields results too late for real-time.
- An quick pose predictor, which computes the new pose each time a proprioceptive measurement is received. It only uses the previous pose, the map and the vehicle evolution model. This predictor, which only relies on map and odometry, enables to get low-latency results while keeping the robustness to erroneous positions.

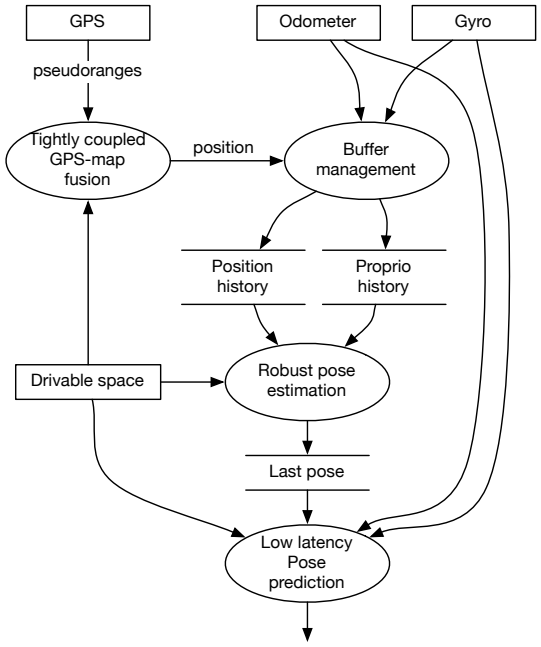


Figure 15. Real-time implementation of the robust pose estimation system

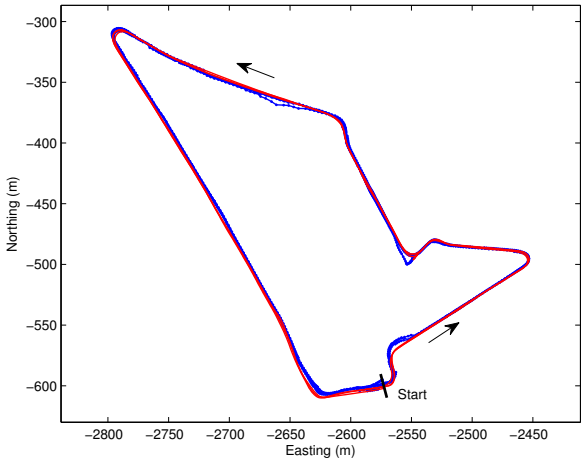


Figure 16. 2D computed trajectory (center of gravity of the solution set) for the three laps (thick blue). *Thin red line is ground truth.*

C. Pose estimation results

In this experiment, the risk has been specified to be less than $R = 10^{-3}$ in each pose estimate, using equation (12). This is the common risk requirement in mobile robotics.

After several preliminary tests, the position buffer length was set to $o = 10$ boxes, with a spatial density of two. Two values of q have been retained. A first experiment has been done with $q = 0$, *i.e.* no robustness to wrong position in the horizon. This enables a comparison with the bootstrap constrained particle filter (CPF) presented in section II-C. A second run has been done with $q = 1$, *i.e.* one wrong position box was tolerated in the horizon, in order to evaluate the pessimism introduced by fault tolerance.

With $q = 0$, the tightly-coupled GPS 3D-map positioning is therefore tuned with a risk $r = 10^{-4}$. Assuming Gaussian pseudorange error with 1 m variance, this leads for example to ± 4.15 m pseudorange error bounds with 3 satellites in view and ± 3.89 m with one satellite. With $q = 1$, the inversion of Eq. (12) gives $r = 4.8 \cdot 10^{-3}$.

The pose estimation process provides a list of boxes up to 20 Hz. One can remark that even if the last pose (Fig. 15) is a subpaving, the list of boxes can overlap because of the prediction step. While this representation gives a good knowledge of confidence domain, a punctual estimate is useful for comparison with the CPF. We thus also compute a pose estimate as the center of gravity of the solution set (Fig. 16).

At the beginning of the test run, the punctual pose estimate only roughly follows the ground truth trajectory (see the overshoot on Fig. 16). Indeed, the pose estimation algorithm starts with no prior knowledge of the vehicle's heading: the prior value for $[\psi]$ is set to interval $[-\pi, \pi]$. This leads to multiple pose hypotheses that offset the pose estimate (Fig. 17a). When enough data is gathered, wrong hypotheses are discarded (boxes "die" because of inconsistency), and a more precise estimate is computed (Fig. 17b).

(a) Ambiguous pose at the beginning of the test

(b) Non-ambiguous pose

Figure 17. Solution subpaving at two epochs (thick blue boxes). *Thin boxes are the position history, green lines point to the used satellites.*

The narrow streets of the trial which induces strong urban canyons (Fig. 13). Despite this bad GPS satellite visibility, the fusion system enables positioning information along the whole trajectory.

During the whole test, the pose subpaving remains consistent with ground truth. Figure 18 shows the errors on the position estimate (center of gravity of the subpaving), along with the bounds of the subpaving. In the first 30 seconds of the experiment, the heading is not well initialized. When enough positions are gathered in the buffer, a precise heading estimated is obtained. Reduction of the heading uncertainty enables a better rejection of wrong hypotheses at crossroads. During the try, longitudinal error is generally larger than lateral error. This is mainly due to the strong map constraint: the vehicle's itinerary mainly follows narrow streets. Once road ambiguity is resolved, GPS pseudoranges mainly provide a longitudinal constraint which can be biased due to measurement and corrections errors, especially when only one or two

satellites are used.

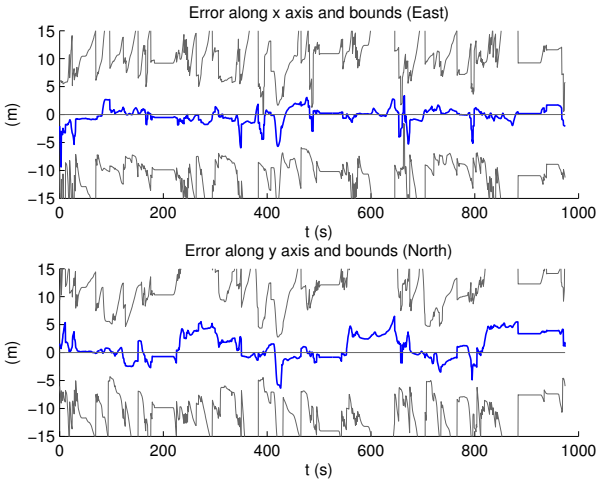


Figure 18. Set-Membership position estimation error (blue) and confidence upper and lower bounds (gray).

Figure 19 shows the outputs of the CPF. Estimation errors are of the same level as the set-membership horizon filter (SMHF). In order to have the same level of confidence ($R = 10^{-4}$), bounds are computed with $\pm 3.29\sigma$. Compared to the proposed approach, confidence domains are smaller, but remain in the same order of magnitude. Nevertheless, one can notice that numerous samples show too optimistic confidence intervals.

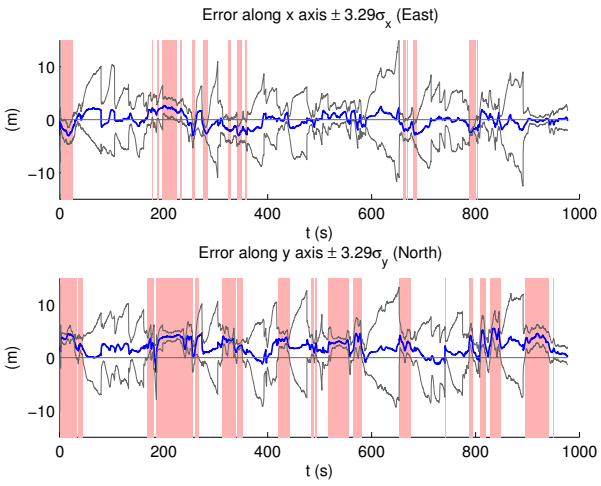
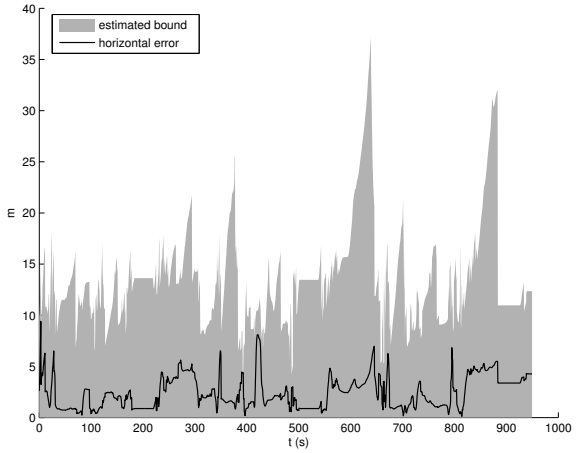
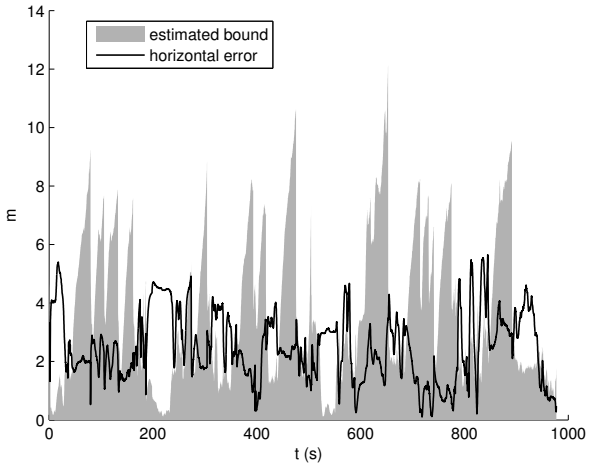


Figure 19. CPF position estimation error (blue) and confidence upper and lower bounds (gray). Pink areas highlight instants where confidence bounds are inconsistent.

The graphs of Fig. 20 express in a compact isotropic representation the consistency of the filters. They show the positioning error in the horizontal plane. The confidence radius is computed along the error vector. For the Bayesian filter, the confidence domain is an ellipse and the confidence radius is determined by using the covariance matrix eigenvalues. For the SMHF, the error radius is the distance between the estimate and the intersection of the bounding box with the



(a) Set-membership horizon filter (10-position horizon, $q = 0$)



(b) Constrained Particle Filter

Figure 20. 2D position error (black) and confidence radius (gray)

error vector. In opposition to Fig. 18, the y-axis is not truncated in order to appreciate the full range of the parameters. Fig. 20b highlights frequent inconsistencies for the CPF when the black curve (estimate) goes out of the gray zone (confidence radius). Fig. 20a clearly show the consistency of the SMHF.

With the center of gravity of the subpaving as a pose estimator, the horizontal position error is less than 5.1 m for 95% of the measurement epochs and is less than 2.1 m half of the time. (Fig. 21). The same figures for the CPF error are 4.5 m 95% and less than 2.3 m half of the time. Globally speaking the CPF provides less extreme values errors, and more biased estimates than the SMHF.

The same figure shows that robustifying the SMHF by relaxing a constraint in he buffer ($q = 1$) has very little impact on the performance (5.47 m 95%, 1.8 m 50%). The two cumulative error distributions match very well (see Fig. 21). This shows that relaxing 10% of the exteroceptive information provides the same level of accuracy. Moreover, as shown in Fig. 22, the confidence radius are roughly the same than the non-robust SMHF, with some higher peaks in very strong

urban canyons (eg at $t \simeq 700$ s). Detailed error statistics are reported in table I.

Table I
HORIZONTAL POSITION ERROR (HPE) STATISTICS

	CPF	SMHF (q=0)	SMHF (q=1)
Mean HPE	2.50 m	2.43 m	2.45 m
Median HPE	2.36 m	2.09 m	1.81 m
95th percentile HPE	4.52 m	5.1 m	5.47 m
Max HPE	5.65 m	9.40 m	13.02 m

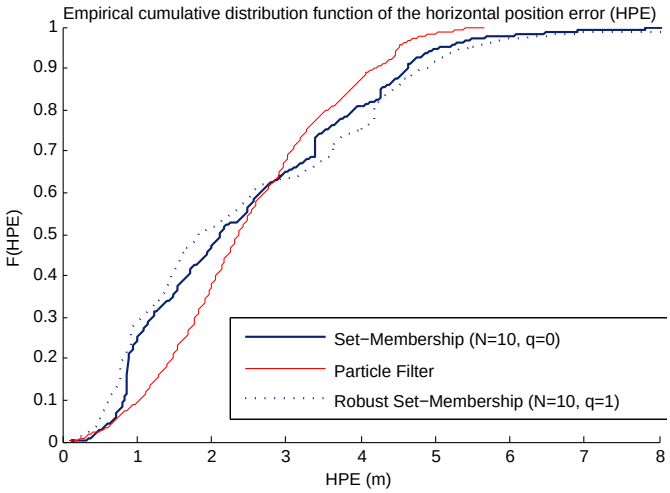


Figure 21. Cumulative distribution function of the horizontal position error

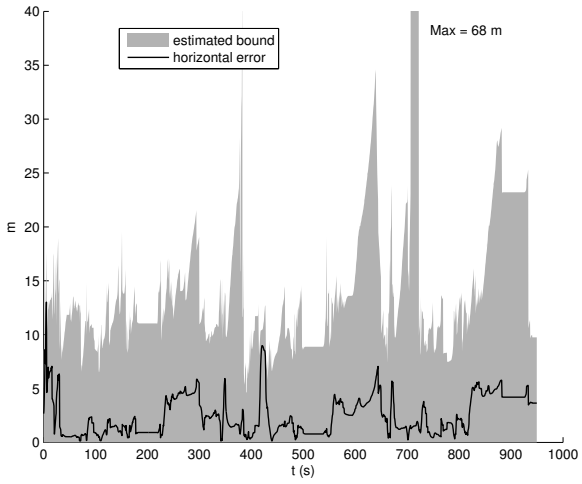


Figure 22. 2D position error (black) and confidence radius (gray) for the robust SMHF with $q = 1$ (10 positions in horizon)

Figures 23 and 24 are integrity plots. They represent the estimated error bounds versus the actual position error. The bisector cuts the graph in two areas: the upper area corresponds to a safe operation, while the lower one corresponds to misleading information (i.e. over-confident). Usually, an alarm limit is plotted to check localization availability. Here, since there are no specified performance with respect to this requirements, this threshold is omitted.

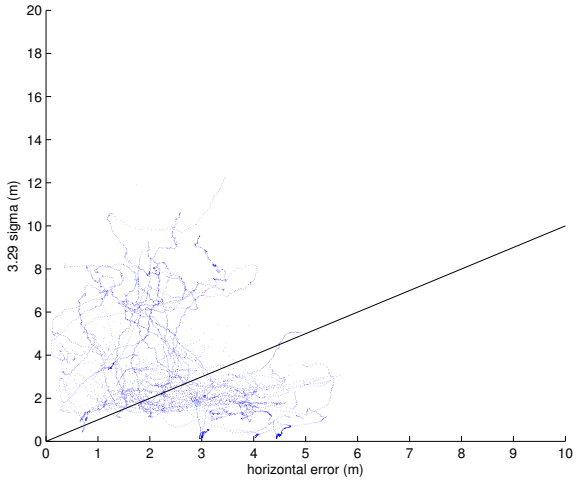


Figure 23. Integrity plot of the constrained particle filter

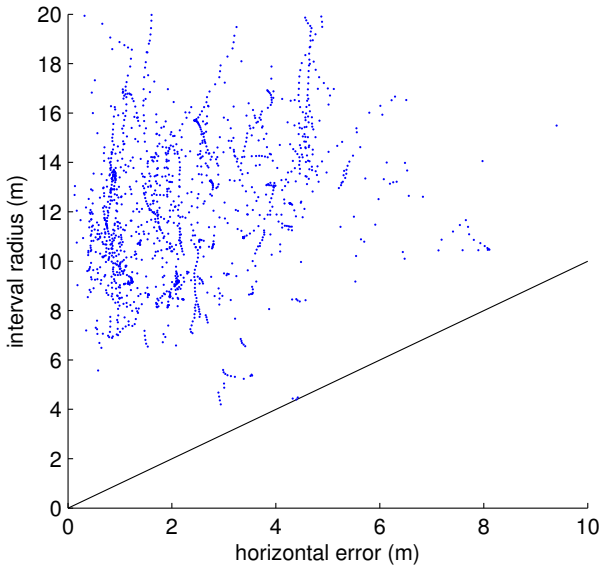


Figure 24. Integrity plot of the set-membership horizon filter

One can notice, as stated before, that the CPF is often over-confident and that the SMHF provides consistent bounds with ground truth. Indeed, with the SMHF, the true position is always inside the confidence domain. True position touches the edge of the confidence domain in 0.15% of the epochs which is approximately the tuning of the filter. Conversely, the $\pm 3.29\sigma$ confidence bounds of the CPF are violated 41% of the time.

Figure 24 shows a tendency in the SMHF error bounds to grow linearly with the actual error, which makes computed error bounds a good indicator of positioning errors in real-time. The CPF does not exhibits such a nice characteristic (Fig. 23).

VIII. Conclusion

A fusion localization method that continuously provides a vehicle pose estimate with high integrity has been presented in this article. Since urban environments are difficult for GPS positioning, extensive use of a precise 3D map of the drivable space has been done to constrain the localization problem. Positions are first computed by a tightly coupled fusion of GPS pseudoranges and the map. The problem is solved by a set inversion algorithm based on interval analysis where both pseudoranges and the map are considered as geometrical constraints on position. Thanks to the 3D map, position computation with as low as two satellites in view is possible if a rough prior position is known (i.e if the road segment is known). Since wrong GPS measurement may lead to empty or erroneous solutions, and since satellite visibility can be greatly reduced in urban canyons, this first positioning step cannot ensure 100% positioning availability. Computed positions are thus gathered in a data history along with proprioceptive measurements, to enable robust pose estimation in a second step. Position data history filling strategy uses spatial criteria to ensure that heading is still observable, even after a long stop. Road data is introduced in the robust constraint propagation process to tighten the solutions, to eliminate the dead-reckoning drift and to reduce positioning ambiguities after GPS outages. To enable high-rate positioning in real-time conditions, a quick prediction based on a box particle filter constrained by the map is performed between pose estimation results.

Experimental validation was performed in very challenging GPS conditions, i.e. an urban canyon with at most two satellites in view half of the time. A constrained particle filter (CPF) has been implemented for comparison purpose. Real-time processing tests showed the system's ability to provide full positioning availability, pose confidence domains consistent with ground truth, and positioning errors below 5.1 m 95% of the time. Under the same conditions, the CPF yield comparable point-positioning results, but only half of the confidence domains were consistent with ground-truth.

Appendix A
Fast q-relaxed intersection contractor

Computing the q-relaxed intersection of m n-dimensional boxes is the core step of the robust set inversion algorithm and the q-relaxed contractor. Unfortunately, this problem has an $O(m^n)$ complexity [13]. When the dimension n of boxes is fixed, the problem is polynomial with respect to the number of boxes. In our case, $n = 4$ because the intersection is done on $([e], [n], [\cos\psi], [\sin\psi])$.

A fast contractor for the q-relaxed intersection can be implemented using axis-projection of constraints (Fig. 25). Each dimension is considered separately. Constraint propagation is applied to the input box, thus obtaining a smaller box constrained by only one measurement. The obtained upper and lower bounds for the considered axis are added to a list of bounds and associated values: each opening bracket (lower bound) is associated with the $+1$ value while each upper bound is associated with the -1 value. A similar list of bounds is constructed for each dimension and populated by applying constraint propagation of each available measurements to the input box.

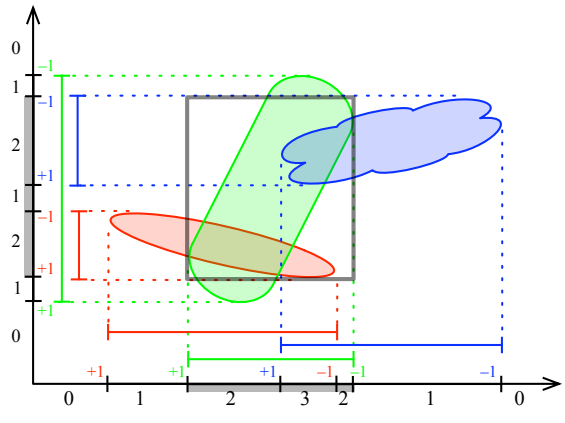


Figure 25. One iteration of the axis-projection q-relaxed contractor, with three constraints *(shown in red, green and blue)* and $q = 1$. The result is in gray.

For each axis, bounds are sorted in ascending order and a counter is set to 0. Then, the bounds corresponding to the axis are examined from the lowest to the highest. Each time a bound is encountered its corresponding value is added to the counter. The first bound that makes the counter hit $m - q$ is set as the axis's lower bound. The last bound that causes the counter to fall below $m - q$ is set as the upper bound for the considered axis. The contracted box is the Cartesian product of the contracted intervals obtained on each dimension. The contraction process is iterated until the box size can be reduced no further.

This contractor is not optimal – *i.e* it does not reach the interval hull of the q-relaxed intersection of boxes in the general case – but its complexity is low: assuming a sorting algorithm of linearithmic complexity, the fast q-relaxed intersection contractor performs in $O(n \cdot m \log(m))$. After contraction, the box is bisected and enqueued so as to narrow the result further in subsequent processing.

Acknowledgment

This work has been supported by ANR (French National Agency) CityVIP project under grant ANR-07_TSFA-013-01. The system has been tested on sequences recorded by the test platform "Stereopolis". The authors would like to thank particularly the IGN staff for the ground truth and the 3D map.

References

[1] M. Buehler, K. Iagnemma, and S. Singh, *The DARPA Urban Challenge: Autonomous Vehicles in City Traffic*. Springer Verlag, 2010, vol. 56.

[2] C. Röhrig and M. Müller, "Indoor location tracking in non-line-of-sight environments using a IEEE 802.15.4a wireless network," in *Conference Proceedings of IROS09. St. Louis*, 10 2009, pp. 552–557.

[3] H. Kloeden, D. Schwarz, E. M. Biebl, and R. H. Rasshofer, "Vehicle localization using cooperative RF-based landmarks," in *2011 IEEE Intelligent Vehicles Symposium (IV)*, Baden-Baden, Germany, June 5-9 2011, pp. 387–392.

[4] E. Royer, M. Lhuillier, M. Dhome, and J.-M. Lavest, "Monocular vision for mobile robot localization and autonomous navigation," *International Journal of Computer Vision*, vol. 74, pp. 237–260, 2007.

[5] H. Li, F. Nashashibi, and G. Toulminet, "Localization for intelligent vehicle by fusing mono-camera, low-cost GPS and map data," in *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, sept. 2010, pp. 1657 –1662.

[6] E. M. Nebot and H. Durrant-Whyte, "A high integrity navigation architecture for outdoor autonomous vehicles," in *Robotics and Autonomous Systems*, no. 26, 1999, pp. 81–97.

[7] S. Sukkarieh, E. Nebot, and H. Durrant-Whyte, "A high integrity IMU/GPS navigation loop for autonomous land vehicle applications," *Robotics and Automation, IEEE Transactions on*, vol. 15, no. 3, pp. 572 –578, jun 1999.

[8] R. Toledo-Moreo, D. Betaille, and F. Peyret, "Lane-level integrity provision for navigation and map matching with gnss, dead reckoning, and enhanced maps," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 11, no. 1, pp. 100 –112, march 2010.

[9] O. Le Marchand, P. Bonnifait, J. Ibanez-Guzmán, and D. Betaille, "Vehicle localization integrity based on trajectory monitoring," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, oct. 2009, pp. 3453 –3458.

[10] L. Jaulin, "Set-membership localization with probabilistic errors," *Robotics and Autonomous Systems*, vol. 59, no. 6, pp. 489 – 495, 2011.

[11] L. Jaulin, M. Kieffer, E. Walter, and D. Meizel, "Guaranteed robust nonlinear estimation with application to robot localization," *IEEE T. Syst. Man. Cy. C.*, vol. 32, no. 4, pp. 374–382, 2002.

[12] D. Meizel, O. Leveque, L. Jaulin, and E. Walter, "Initial localization by set inversion," *IEEE T. Robotic. Autom.*, vol. 18, no. 6, pp. 966–971, 2002.

[13] L. Jaulin, "Robust set-membership state estimation; application to underwater robotics," *Automatica*, vol. 45, no. 1, pp. 202–206, 2009.

[14] Y. Bar-Shalom, X. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation*. Wiley Online Library, 2001.

[15] G. Agamennoni, J. Nieto, and E. Nebot, "An outlier-robust kalman filter," in *IEEE International Conference on Robotics and Automation*, 2011.

[16] S. Julier and J. LaViola, "On kalman filtering with nonlinear equality constraints," *Signal Processing, IEEE Transactions on*, vol. 55, no. 6, pp. 2774 –2784, june 2007.

[17] J. Dunik, M. Simandl, and O. Straka, "Multiple-model filtering with multiple constraints," in *American Control Conference (ACC), 2010*, 30 2010-july 2 2010, pp. 6858 –6863.

[18] F. Chausse, J. Laneurit, and R. Chapuis, "Vehicle localization on a digital map using particles filtering," in *IEEE Intelligent Vehicles Symposium, 2005*. Las Vegas, Nevada, U.S.A.: IEEE, 2005, pp. 243–248.

[19] A. Selloum, D. Bétaille, E. Le Carpentier, and F. Peyret, "Robustification of a map aided location process using road direction," in *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*. IEEE, 2010, pp. 1504–1510.

[20] I. Miller, M. Campbell, and D. Huttenlocher, "Map-aided localization in sparse global positioning system environments using vision and particle filtering," *Journal of Field Robotics*, vol. 28, no. 5, pp. 619–643, 2011.

[21] E. Kaplan and C. Hegarty, *Understanding GPS: Principles and Applications Second Edition*. Artech House Publishers, 2006.

[22] L. Jaulin, M. Kieffer, O. Didrit, and É. Walter, *Applied Interval Analysis*. Springer-Verlag, 2001.

[23] F. Benhamou, F. Goualard, L. Granvilliers, and J.-F. Puget, "Revising hull and box consistency," in *Int. Conf. on Logic Programming*. MIT press, 1999, pp. 230–244.

[24] I. Sutherland and G. Hodgman, "Reentrant polygon clipping," *Communications of the ACM*, vol. 17, no. 1, pp. 32–42, 1974.

[25] B. Tolman, R. B. Harris, T. Gaussiran, D. Munton, J. Little, R. Mach, S. Nelsen, and B. Renfro, "The GPS Toolkit: Open source GPS software," in *Proceedings of the 16th International Technical Meeting of the Satellite Division of the Institute of Navigation*, Long Beach, California, September 2004.

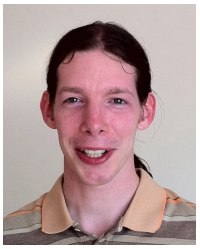
[26] V. Drevelle and P. Bonnifait, "Global positioning in urban areas with 3-D maps," in *IEEE Intelligent Vehicle Symposium, 2011*, Baden-Baden, Germany, 2011, pp. 764–769.

[27] ——, "Robust positioning using relaxed constraint-propagation," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. Taipei, Taiwan: IEEE, 2010, pp. 4843–4848.

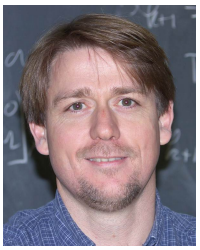
[28] A. Ranganathan, M. Kaess, and F. Dellaert, "Fast 3D pose estimation with out-of-sequence measurements," in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, 29 2007-nov. 2 2007, pp. 2486 –2493.

[29] F. Abdallah, A. Gning, and P. Bonnifait, "Box particle filtering for non linear state estimation using interval analysis," *Automatica*, vol. 44, pp. 807–815, December 2008.

[30] N. Paparoditis, C. Thom, and H. Jibrini, "Surface reconstruction in urban areas from multiple views of aerial digital frame cameras," *International Archives of Photogrammetry and Remote Sensing*, vol. 33, no. Suppl. B3, pp. 43–50, 2000.



Vincent Drevelle obtained his Ph.D. degree in Computer Science and Engineering at the the Université de Technologie de Compiègne (UTC), France in 2011. He graduated as a computer science engineer in 2007; then he obtained a master degree in system and information technologies from the UTC. Since 2008, he has been with Heudiasyc UMR 6599, a joint research unit backed by CNRS and UTC. His current research interests are in set-theoretic methods and interval analysis, for multi-sensor localization of vehicles.



Philippe Bonnifait is a professor in the Computer Science and Engineering department of the Université de Technologie de Compiègne (UTC), France. He obtained his Ph.D. degree in automatic control and computer science from the École Centrale de Nantes in 1997. Since 1998, he has been with Heudiasyc UMR 6599, a common research laboratory between UTC and CNRS. His research interests are in Intelligent Vehicles. He is currently working on high integrity positioning and map-matching to retrieve contextual information stored in a navigable database for Advanced Driving Assistance Systems.