Global Positioning in Urban Areas with 3-D Maps

Vincent Drevelle, Philippe Bonnifait

Global Positioning in Urban Areas with 3-D Maps

Vincent Drevelle and Philippe Bonnifait

Abstract— **A global positioning method based on a precise 3-D drivable area map and on GPS pseudorange measurements is presented. Map and GPS measurements are represented by geometric constraints, thus turning the localization problem into a constraint satisfaction problem whose solution is the confidence domain of position. Interval analysis is employed to solve the problem by using contractions and bisections of a prior position box. If more than 3 satellites are visible, the method is robust to wrong pseudorange measurements. The system is also able to compute multiple position hypotheses in the case of ambiguities. An experimental validation using real GPS pseudorange measurements and a precise 3-D map is reported to illustrate the performance of the method with real data in an urban area, with reduced satellite visibility. Confidence domains are consistent with the truth during the whole 1 km experiment, and a 6.5 m 95% accuracy is achieved with at least two satellites in view.**

I. INTRODUCTION

Global Positioning Systems (GPS) offer a worldwide positioning solution, and can achieve high accuracy localization in open areas. Real time kinematic (RTK) enables centimeter-level accuracy positioning with phase measurements, but implies the use of a base station, and good satellite visibility. When these conditions are met, GPS can be sufficient to control autonomous vehicles in path-following tasks [1].

In urban environments, GPS signals are frequently blocked by surrounding buildings, and even reflected, causing multipath and non-line-of-sight measurement errors. This is incompatible with autonomous urban navigation tasks, requiring decimeter-accurate positioning to keep the vehicle inside the traffic lanes. Thus, local methods with exteroceptive sensors are needed to control the vehicle trajectory and to perform obstacle avoidance [2].

But, even if not precise enough to perform a robot control task, a robust global positioning method able to quantify position uncertainty, can still be needed to

- initialize a local positioning system (eg. to limit the area in where initial particles of a particle filter will be generated, or to narrow the set of visual features to be searched and tracked by a vision based algorithm).
- perform integrity checks on the local positioning system: a reliable global positioning method enables to fire an alert if the vehicle drifts from its planned trajectory.
- carry out high-level topological navigation: while reactive control is performed with local exteroceptive sensors like cameras and LIDAR, the navigation choices (eg. path planning, turn at a crossroad, respect of the

Vincent Drevelle and Philippe Bonnifait are with Heudiasyc, CNRS UMR 6599, Université de Technologie de Compiègne, Centre de Recherches de Royallieu, BP 20529, 60205 Compiegne Cedex, France, e-mail: vincent.drevelle@hds.utc.fr.

speed limit, stop at the final waypoint) need reliable global position information to be achieved.

To counterbalance the poor GPS precision and availability in urban environments, usually GPS is aided by inertial measurement units. For road vehicles, geographic information can be used as well. Much research has been done with standard maps [3][4], but precise digital surface models [5] and lane-level maps [6] are now available to provide strong constraints on position.

In this paper, we propose a robust approach to solve the positioning problem in known urban environments, with geometric constraints. For this purpose, GNSS pseudorange measurements are combined in a tightly-coupled fashion with a precise 3-D road surface model. Unlike Bayesian or optimization methods, which compute a punctual position estimate and quantify the error distribution around this point, the proposed method formulates the problem as a constraint satisfaction problem (CSP). By using interval analysis and contractors, the positioning confidence domain is computed as the set of positions compatible with all the constraints. It can even be a disconnected solution set in the case of positioning ambiguity.

The paper is organized as follows. After an overview of interval analysis and contractors, a solver for constraint satisfaction problems (CSP) is introduced. Then, the 3-D road constraint and GPS constraint are presented. A constraint for robust GPS positioning is also introduced. Finally, experimental results with real GPS data and 3-D map are presented.

II. INTERVAL ANALYSIS AND CONTRACTORS

A. Interval analysis

Interval analysis [7] involves intervals and their multidimensional extension, *interval vectors* (or *boxes*). In opposition to arbitrary sets, intervals and boxes are easy to represent and 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 is denoted $[x] = [\underline{x}, \overline{x}]$, where vectors $\underline{x}$ and $\overline{x}$ are respectively the lower and upper bounds of $[x]$.

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

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

The *box hull* $\square S$ of a set S is the smallest box that includes S. Since the union of boxes is not generally a box, the *box union* operator $\sqcup$ returns the box hull of the union of two boxes: $[x] \sqcup [y] = \square([x] \cup [y])$.

B. Contraction and propagation

When the components of a vector x are linked by relations or constraints, one can define a *constraint satisfaction problem* (CSP). It consists in finding the solution set $X = \{x \in [x] | f(x) = 0\}$, where $[x]$ is the domain of the variables and $f(x) = 0$ represents the constraints, and can also represent inequalities by introducing slack variables [7].

A *contractor* $\mathcal{C}$ for a CSP is an operator that computes a smaller domain $[x_c] = \mathcal{C}([x])$ without affecting the solution set, *i.e.* $X \subset [x_c] \subset [x]$. There are many ways to implement a contractor, one of them is the forward-backward contractor based on constraint propagation [8].

C. Set inversion and subpavings

The set inversion problem consists in determining the set X such as $f(X) = [y]$, where $[y]$ is a known interval vector of m measurements. To approximate compact sets in a guaranteed way, *subpavings* can be used. A subpaving of a box $[x]$ is the union of non-empty and non-overlapping subboxes of $[x]$.

Using interval analysis, the solution $X = f^{-1}([y])$ 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 [7].

Algorithm 1 implements a SIVIA that only computes an outer approximation $\overline{X}$ of the solution set in a given domain $[x_0]$. It uses a list of boxes $\mathfrak{L}$ managed by the *push* and *pull* functions. If $\mathfrak{L}$ is a queue, the algorithm employs a breadth-first strategy. ϵ controls the sharpness of the subpaving $\overline{X}$. Boxes larger that ϵ after contraction are bisected.

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. In this paper, we present a new kind of constraint based on 3-D drivable space.

III. 3-D ROAD CONSTRAINTS

3-D roads can be used for contracting not only the East and North estimates by exploiting the road boundaries, but also altitude.

Algorithm 1 SIVIA(in: $[x_0], \epsilon$)

1: $\overline{X} := \emptyset$ // *empty subpaving*
2: push($[x_0], \mathfrak{L}$)
3: **while** $\mathfrak{L}$ is not empty **do**
4:    $[x] := \text{pull}(\mathfrak{L})$
5:    $[x] := \mathcal{C}([x])$ // *contract the box*
6:    **if** width($[x]$) $< \epsilon$ **then**
7:       $\overline{X} := \overline{X} \cup [x]$
8:    **else if** $[x] \neq \emptyset$ **then**
9:       $([x_1], [x_2]) := \text{bisect}([x])$
10:      push($[x_1], \mathfrak{L}$); push($[x_2], \mathfrak{L}$)
11:   **end if**
12: **end while**
13: **return** $\overline{X}$

A. Drivable space

In this paper, the *drivable space* denotes 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 kerbs. Obstacles like poles or lane separators are also excluded from the drivable space. Finally, since we consider the position at the center of the vehicle, erosion is performed on the drivable space to take the size of the vehicle into account.
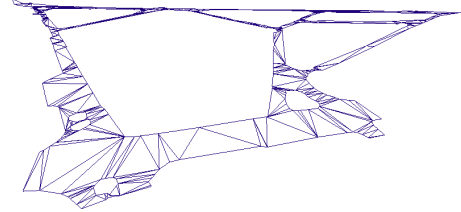

Fig. 1: Wireframe view of the drivable space extracted from a database

The drivable space is represented in 3D by a triangular mesh, i.e. a surface made of connected triangular facets (Fig. 1). 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.

B. Positioning constraint on a single facet

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

Let us consider the facet of Fig. 2. The first constraint is simply given by the plane equation obtained from the 3 vertices. The point M is on the plane defined by the points A, B, C, thus it satisfies

$$(\overrightarrow{AB} \wedge \overrightarrow{AC}) \cdot \overrightarrow{AM} = 0 \tag{1}$$

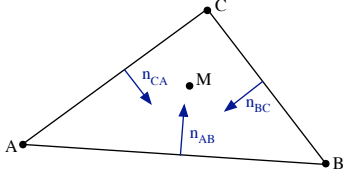Edge constraints can be expressed by enforcing the positivity of the dot products of vertex to point vectors with the

Fig. 2: Facet ABC with its edge normals

corresponding edge normal vectors:

$$\overrightarrow{AM} \cdot \overrightarrow{n_{AB}} \geq 0$$
$$\overrightarrow{BM} \cdot \overrightarrow{n_{BC}} \geq 0$$
$$\overrightarrow{CM} \cdot \overrightarrow{n_{CA}} \geq 0 \tag{2}$$

When working in a valid local tangent frame, one can simplify the dot-product inequalities by working only with 2D coordinates (assuming the road is never vertical). Moreover, if facets are small enough with respect to altitude variation, constraining the position inside the bounding box of the facet is sufficient to approximate the facet plane constraint (1). Algorithm 2 presents this contractor: Initial contraction is performed with the bounding box of the facet, then dot-product constraints (2) are simply enforced by an inclusion test (line 6 of Alg. 2). If there is no point M in the box that satisfies (2), then the box is discarded. In this algorithm, interval vectors $[a], [b], [c]$ are the coordinates of the considered facet vertices.

Algorithm 2 facet_contract(in: $[x], [a], [b], [c]$)

1: $[bbox] := [a] \sqcup [b] \sqcup [c]$
2: $[x] := [x] \cap [bbox]$
3: **if** $[x] = \emptyset$ **then**
4:     return($[x]$)
5: **end if**
6: **if** $([x_{1:2}]\text{-}[a_{1:2}])\cdot \left(\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right)([b_{1:2}]\text{-}[a_{1:2}])\cap[0,+\infty]=\emptyset$
  or $([x_{1:2}]\text{-}[b_{1:2}])\cdot \left(\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right)([c_{1:2}]\text{-}[b_{1:2}])\cap[0,+\infty]=\emptyset$
  or $([x_{1:2}]\text{-}[c_{1:2}])\cdot \left(\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right)([a_{1:2}]\text{-}[c_{1:2}])\cap[0,+\infty]=\emptyset$ **then**
7:     return($\emptyset$)
8: **end if**
9: return($[x]$)

C. Positioning constraint on the road surface

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. 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.

D. Taking topology into account

When a previous position is known, road topology can be used to reduce the initial facet candidates. Let $[x_{k|k}]$ be the position at time t_k, $[x_{k+1|k}]$ the predicted location zone at time t_{k+1}, and s the distance traveled between t_k and t_{k+1}. For instance, s can be measured by an odometer.

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

1: $[xc] := \emptyset$
2: $\mathcal{F} := \text{extract\_facets}(\mathcal{M}, [x])$
3: **for** each f in $\mathcal{F}$ **do**
4:     $[xc] := [xc] \sqcup \text{facet\_contract}([x], f.[a], f.[b], f.[c])$
5: **end for**
6: **return** $[xc]$

The first step is to bound the zone $[z]$ in which the vehicle can have evolved between t_k and t_{k+1}. A rough estimate is given by

$$[z] = \frac{1}{2}\left([x_{k|k}] + [x_{k+1|k}] + [-s, s]\right) \tag{3}$$

The physical interpretation of (3) is that the traveled distance constraint is applied independently on each dimension of $[x]$.

The facets that intersect $[z]$ are extracted from the map, to get the submap $\mathcal{M}_z$. Facets in $\mathcal{M}_z$ that intersect $[x_{k|k}]$ are marked, then every facet that is connected to a marked facet in $\mathcal{M}_z$ is iteratively marked. The obtained connected components are the roads that may have been covered between t_k and t_{k+1}. The final selection is performed by only keeping the marked facets that intersect $[x_{k+1|k}]$.

This procedure replaces line 2 of Alg. 3.

IV. GPS PSEUDORANGES CONSTRAINTS

A. GPS observation model

GPS positioning with *pseudorange* measurements is a *Time of Arrival* method [9]. Pseudoranges are the distances to each visible satellite, offset by a unknown amount due to the offset between the receiver clock and the GPS time. GPS positioning using pseudoranges is thus a four-dimensional problem: along with the Cartesian coordinates (e, n, u) of the user in a local tangent frame to the Earth, the user's clock offset dt^u has to be estimated. Let $x = (e, n, u, d)$ with $d = c \cdot dt^u$.

Measured pseudoranges need to be corrected, to take into account relativity effects and the delays induced by the ionosphere and troposphere traversal. The propagation delay corrections applied to measured pseudoranges to get 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 $[\rho_i]$ whose bounds are determined given an integrity risk [10].

Satellite positions (e_i^s, n_i^s, u_i^s) are computed with broadcast ephemeris data. They are represented as boxes $[x^s] = ([e_i^s], [n_i^s], [u_i^s])$ to take ephemeris inaccuracy into account.

The location zone computation consists in characterizing the set X of all locations compatible with the measurements and the satellite position intervals. Each pseudorange introduces a constraint on the solution. The constraint induced by the ith corrected pseudorange measurement is represented
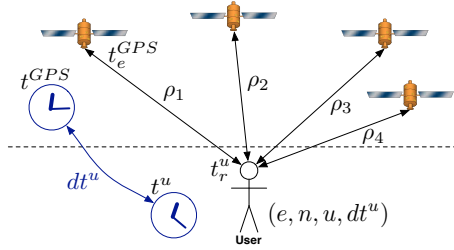
Fig. 3: GPS positioning with pseudoranges

by the natural inclusion function of the GPS pseudorange observation function:

$$[\rho_i] = \sqrt{([e] - [e^s_i])^2 + ([n] - [n^s_i])^2 + ([u] - [u^s_i])^2} + [d] \tag{4}$$

B. GPS contractor

A contractor for a single pseudorange constraint is given in Alg. 4. It is a forward-backward contractor [8], which performs constraint propagation in an optimal order, following the elementary constraints decomposition of the pseudorange measurement function (Fig. 4).

Algorithm 4

pseudorange_contract(in: $[x], [\rho], [x^s]$)

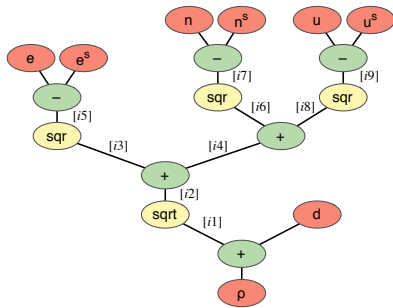| // Forward	// Backward
$[i9] := [u] - [x^s_3]$	$[d] := [d] \cap ([\rho] - [i1])$
$[i8] := \text{square}\,[i9]$	$[i1] := [i1] \cap ([\rho] - [d])$
$[i7] := [n] - [x^s_2]$	$[i2] := [i2] \cap \text{square}\,[i1]$
$[i6] := \text{square}\,[i7]$	$[i3] := [i3] \cap ([i2] - [i4])$
$[i5] := [e] - [x^s_1]$	$[i4] := [i4] \cap ([i2] - [i3])$
$[i4] := [i6] + [i8]$	$[i5] := [i5] \cap \text{square}^{-1}\,[i3]$
$[i3] := \text{square}\,[i5]$	$[i6] := [i6] \cap ([i4] - [i8])$
$[i2] := [i3] + [i4]$	$[i7] := [i7] \cap \text{square}^{-1}\,[i6]$
$[i1] := \sqrt{[i2]}$	$[i8] := [i8] \cap ([i4] - [i6])$
$[\rho] := [\rho] \cap ([i1] + [d])$	$[i9] := [i9] \cap \text{square}^{-1}\,[i8]$
	$[e] := [e] \cap ([i5] + [x^s_1])$
	$[n] := [n] \cap ([i7] + [x^s_2])$
	$[u] := [u] \cap ([i9] + [x^s_3])$

return $[x]$

Fig. 4: Elementary constraints decomposition of pseudorange measurement inclusion function

The constraint for GPS positioning using pseudoranges consists in fulfilling the pseudorange constraints for all

visible satellites. With m satellites in use, a GPS contractor is given in Alg. 5.

Algorithm 5

gps_contract(in: $[x], [\rho]_1, \dots, [\rho]_m, [x^s]_1, \dots, [x^s]_m$)

1: **for** $i := 1$ **to** m **do**
2:    $[x] := [x] \cap \text{pseudorange\_contract}([x], [\rho_i], [x^s]_i)$
3: **end for**
4: **return** $[x]$

C. Robust GPS contractor

GPS measurements are prone to errors: while satellite failures may rarely occur, erroneous measurements are frequent in urban environments, due to multipath propagation, or non-line-of-sight measurements (when the direct signal is blocked and only reflected signals are measured). In those cases, the error bounds set on pseudoranges are no more consistent with the real pseudorange error.

Inconsistent error bounds may lead either to an empty solution (in this case, the presence of an inconsistency is detected), or to a misleading solution.

Instead of enforcing the constraints of all pseudoranges, the robust GPS constraint consists in relaxing a given number q of pseudorange constraints. This way, q wrong measurements are tolerated, and the constraint corresponds to the set of solutions *at least compatible with $m - q$ measurements*. This is the so-called *q-relaxed intersection* of the pseudorange constraints [11].
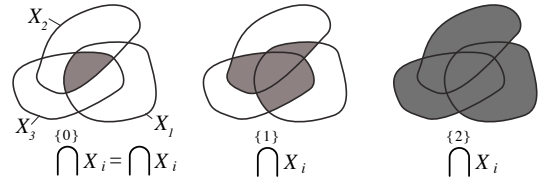

Fig. 5: q-relaxed intersection of three sets for $q \in \{0, 1, 2\}$

Considering m sets $X_1, \dots, X_m$ of $\mathbb{R}^n$, the q-relaxed intersection $\overset{\{q\}}{\bigcap} X_i$ is the set of $x \in \mathbb{R}^n$ which belongs to at least $m - q$ of the X_i's (Fig. 5).

The robust GPS contractor [10] is given in Alg. 6. One should notice that, in opposition to the standard intersection of boxes, the q-relaxed intersection of several boxes is not generally a box. As a consequence, the box-hull of the q-relaxed intersection is used at line 5 of Alg. 6.

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 [10].

D. Receiver clock offset estimation/prediction

As previously stated, GPS positioning does not only consists in finding the receiver's position, but also its clock offset dt^u. Over a short period of time, receiver time can be considered as a clock running at constant a constant speed. The receiver clock drift $\dot{dt}^u$ can thus be estimated to help

Algorithm 6
gps_contract_qrel($[x], [\rho]_1, \ldots, [\rho]_m, [x]_1^s, \ldots, [x]_m^s, q$)

1: **repeat**
2:    **for** $i := 1$ **to** m **do**
3:       $[xc](i) := \text{pseudorange\_contract}([x], [\rho]_i, [x]_i^s)$
4:    **end for**
5:    $[x] := \square \left(\bigcap_{i \in \{1,\ldots,m\}}^{\{q\}} [xc](i) \right)$ *hull of the q-relaxed intersection of m boxes*
6: **until** no more contraction can be done on $[x]$
7: **return** $[x]$

---

the prediction of dt^u. This enables positioning with a single satellite in view, as long as there is no ambiguity in the current road selection [12].

The clock offset of many receivers is not continuous: it is often maintained into a specific range, which causes jumps in dt^u value. In those cases, dt^u contraction fails because of inconsistency, and the receiver clock offset estimator is reset. In such a case, the inconsistency is always detected, because typically a clock jump is about 1 ms and the confidence in dt^u is about 50 ns.

V. EXPERIMENTAL RESULTS

A. Experiment

Data acquisition was performed on the *Stereopolis* vehicle from the French *Institut Geographique National* (IGN), for the *CityVIP* research program. It is a 1 km loop in the 12th arrondissement of Paris, a 5 minute drive around the local town hall (Fig. 6).
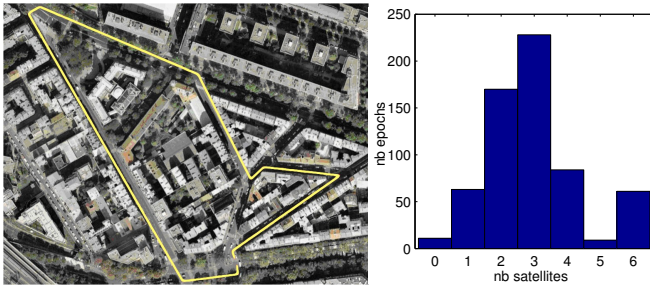

Fig. 6: Trajectory around the 12th arrondissement town hall in Paris and histogram of the number of visible satellites.

The drivable space is provided by the *IGN* [5]. It is hand made from precise aerial photographs (5 cm horizontal precision, 25 cm in altitude). It has been converted from its original Lambert93 projection to a local tangent frame. GPS positioning is also performed in the same local tangent frame. We use GPS pseudorange measurements acquired at 2 Hz with a *Septentrio PolaRx2* receiver. Satellite positions and pseudorange corrections are computed with the open source *GPSTk* library [13]. Ground truth is provided by a post-processed *Applanix* inertial navigation system.

The acquired data are very challenging for GPS positioning, since there are less than 4 satellites in view during 77%

of the trial, and even less than 3 satellites in view in 40% of the measurement epochs (Fig. 6).

B. Results of the positioning algorithm

Localization is performed by using the road contractor of section III together with the robust GPS contractor of section IV-C. We use the SIVIA algorithm (Alg. 1), with the contractor $\mathcal{C}_{road\_gps}$ implemented in Alg. 7. It successively applies the road and GPS contractors until no more contraction happens. The improved facet selection algorithm of section III-D is employed. An odometer provides the traveled distance $s(t_k, t_{k+1})$ between two epochs. The traveled distance is also employed to compute a prior box $[x_{k+1|k}]$ from the bounding box of the previous solution $[x_{k|k}]$.

Algorithm 7 $\mathcal{C}_{road\_gps}$(in: $[x], \mathcal{M}, [\rho]_{1\ldots m}, [x^s]_{1\ldots m}, q$)

1: **repeat**
2:    $[x] := \text{road\_contract}([x], \mathcal{M})$
3:    $[x] := \text{gps\_contract\_qrel}([x], [\rho]_{1\ldots m}, [x^s]_{1\ldots m}, q)$
4: **until** no more contraction can be done on $[x]$
5: **return** $[x]$

The number q of tolerated wrong measurements in the robust GPS contractor is dynamically set as a function of the number m of visible satellites. With less than 4 satellites in view, we use a non-robust contractor ($q_{m<4} = 0$). With at least 4 satellites in view, the robust contractor is employed: $q_{m=4} = 1$, $q_{m>4} = 2$. Pseudorange intervals are set to ensure a risk of less than 10^{-5} assuming normally distributed pseudorange errors with $\sigma = 1$ m [10].

Fig. 7 shows the bounds of the confidence domains with respect to ground truth. Confidence domains are consistent with ground truth along the whole trial. The center of gravity of the solution subpaving is used as a punctual position estimate for error computations.

The test starts with good satellite visibility, before entering urban canyons with 2 or 3 satellites in view. At epoch #69 (Fig. 8a), accurate positioning is achieved with only two satellites in view, thanks to the road constraint. At epoch #220 (Fig. 8b), road selection is ambiguous at a crossroad. Ambiguous positioning occurs at epoch #553 (Fig. 8c) after a crossroad, since two satellites are not enough to select only the good road segment.

Around epoch #280, the system in unable to efficiently contract position with zero then only one satellite in view. Nevertheless, between epochs #560 and #580, the system is able to contract position with only one satellite in view, thanks to the receiver clock offset prediction.

The center of gravity of the subpaving is a good position estimate. With at least 2 satellites in view, positioning error is less than 6.5 m 95% of the time (Fig. 9). In the same conditions, the radius of the computed confidence domain stays within 16 m 95% of the time.

VI. CONCLUSIONS

A global localization method has been presented in this paper. It uses bounded-error GPS measurements and a pre-
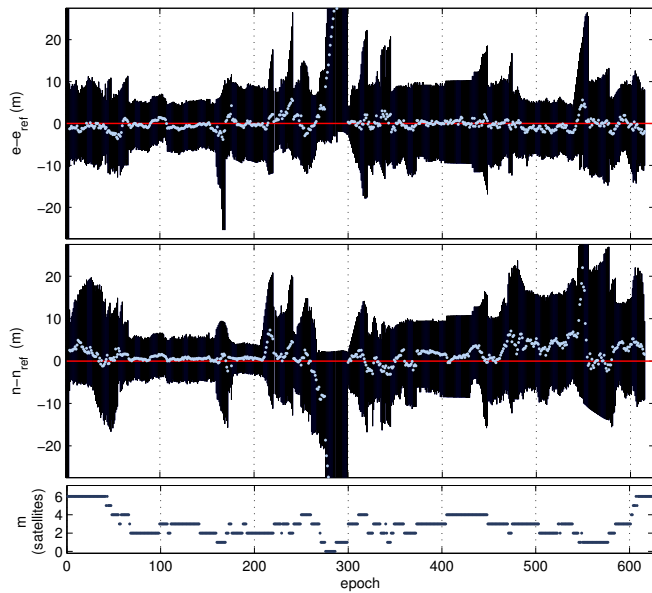
Fig. 7: Bounds of the computed position subpaving. *Zero ordinate is ground truth. Points are the center of gravity of the solution subpaving.*
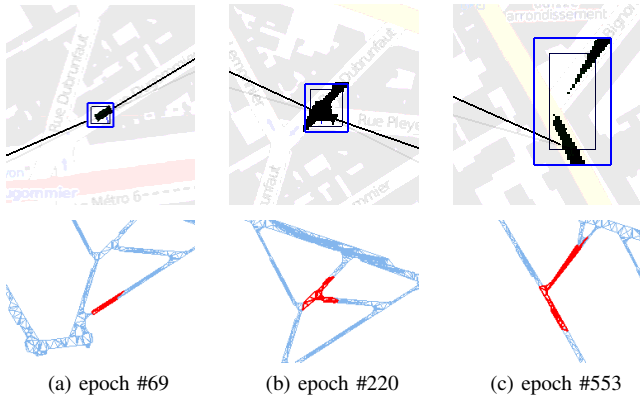

(a) epoch #69   (b) epoch #220   (c) epoch #553

Fig. 8: Upper row: position solution subpaving (in black). *Blue box is the prior predicted position, lines are GPS satellites lines of sight.* Lower row: drivable space. *Candidate facets are in red.*

cise 3-D map of the drivable space as geometric constraints on position. These data are fused in a branch and prune algorithm, which combines a robust GPS pseudorange contractor and a road contractor that takes topology into account, to characterize the user location zone.

Experimental validation has been performed with real GPS data in Paris. With at least 2 satellites in view, the method provides consistent error bounds with 16-meter average radius location zones, and a 6.5 m horizontal accuracy (95%).

Since the method does not estimate nor predict the full vehicle pose, confidence domains computed with less that 2 satellites in view are very large. Integrating a vehicle evolution model will enable to predict the vehicle pose, which will enable to get tighter confidence domains when GPS visibility is reduced. It could also permit the use of a
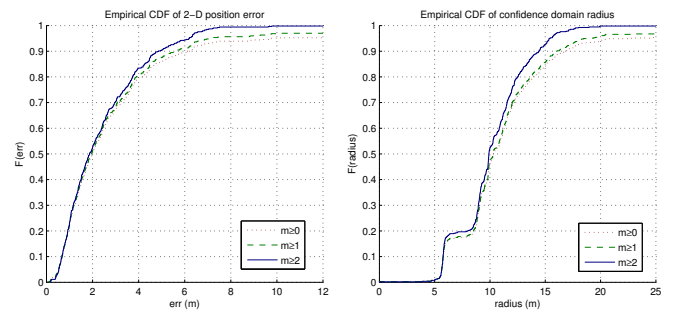

Fig. 9: Empirical cumulative distribution functions of the 2-D position error of the center of gravity of the solution, and of the confidence domain radius. *CDFs are plotted for all epochs ($m \geq 0$), and for epochs with at least one ($m \geq 1$) or two ($m \geq 2$) satellites in view.*

robust GPS contractor with less than four satellites in view.

REFERENCES

[1] B. Thuilot, R. Lenain, P. Martinet, and C. Cariou, "Accurate GPS-based guidance of agricultural vehicles operating on slippery grounds," in *Focus on Robotics Research*, J. Liu, Ed. Nova Science Publishers, 2006, pp. 185–239, Chapter 6 ISBN: 1-59454-594-4.

[2] I. Miller, S. Lupashin, N. Zych, P. Moran, B. Schimpf, A. Nathan, and E. Garcia, "Cornell University's 2005 DARPA Grand Challenge entry," *Journal of Field Robotics*, vol. 23, no. 8, pp. 625–652, 2006.

[3] A. Lahrech, C. Boucher, and J.-C. Noyer, "Accurate vehicle positioning in urban areas," *IECON Proceedings (Industrial Electronics Conference)*, vol. 5, pp. 486–490, 2005.

[4] S. Syed and M. Cannon, "Map-aided GPS navigation," *GPS World*, vol. 16, no. 11, pp. 39–44, 2005.

[5] 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.

[6] D. Bétaille, R. Toledo-Moreo, and J. Laneurit, "Making an enhanced map for lane location based services," in *Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on*. IEEE, 2008, pp. 711–716.

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

[8] 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.

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

[10] V. Drevelle and P. Bonnifait, "High integrity GNSS location zone characterization using interval analysis," in *Proceedings of ION GNSS 2009*, 2009, pp. 2178–2187.

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

[12] C. Fouque and P. Bonnifait, "Multi-hypothesis map-matching on 3D navigable maps using raw gps measurements," in *13th IEEE International Conference on Intelligent Transportation Systems*, Madeira Island Portugal, 09 2010, pp. 1498–1503.

[13] 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.