THE JOURNAL OF NAVIGATION (2015), **68**, 827–847. © The Royal Institute of Navigation 2015
doi:10.1017/S0373463315000132

Applying Standard Digital Map Data in Map-aided, Lane-level GNSS Location

David Bétaille[1], François Peyret[1], IFSTTAR and Maxime Voyer[2]

[1](*IFSTTAR: Institut Français des Sciences et Technologies des Transports, de l'Aménagement et des Réseaux*
[2](*ESGT: Ecole Supérieure des Géomètres et Topographes*)
(E-mail: david.betaille@ifsttar.fr)

Urban positioning using the Global Positioning System (GPS) is challenging because of multipath. Urban canyons limit open sky visibility, and cause signal reflection and diffraction, resulting in significant satellite range measurement errors. The investigations reported here have been carried out in a French project called *Inturb* (an acronym derived from integrity and urban positioning). So far, the project has had two phases: first, a simple Three-Dimensional (3D) geometrical city modelling, called "Urban Trench", has been developed and engineered manually from data sets collected in different cities. Positioning improvement in terms of accuracy was quantified where the model could be applied. Second, this modelling has been automated, based on the standard national BD Topo ® map database for France, with promising results. This geometrical modelling makes it possible to distinguish between line-of-sight satellite signals and those from non-line-of-sight. The latter, apparently *bona fide*, signals are caused by strong reflections, usually from buildings with a lot of steel and glass in their construction. A correction of the pseudo-range measurements of the latter is also computed and applied in the position estimator. Positioning accuracy is improved, whilst availability is kept at its maximum. In the study both manual and automatic 3D models are used in extensive experimental campaigns. Results are: first, the possibility to cover entirely any urban area in the country; second, the reduction of the median error in 3D by more than 50% on data collected in Nantes, Paris and Toulouse for a total duration of nearly ten hours; third, the compliance with standards used in most embedded maps and geographical information systems, including an assessment of the trade-off between the model simplicity and the positioning improvement.

KEYWORDS

1. GNSS. 2. Multipath. 3. 3D City Model. 4. Standard Map Data.

Submitted: 9 January 2014. Accepted: 4 February 2015. First published online: 31 March 2015.

1. INTRODUCTION.

The Global Positioning System (GPS) has been operational since the 1990s, profoundly affecting the survey methods used in both static and kinematic applications. Other Global Navigation Satellite Systems (GNSS)

https://doi.org/10.1017/S0373463315000132 Published online by Cambridge University Press

deployed (such as GLONASS), or under development today (such as BeiDou and Galileo), will reinforce the impact – already important – of satellite positioning in the daily life of people. There is huge growth in both the requirements and applications for consumer-level positioning in cars, other vehicles and smart phones. Foreseen applications include the domain of Intelligent Transportation Systems (ITS), and they rely on inexpensive equipment with operational capabilities compatible with various urban environments where most needs concentrate. Buildings are prone to create perturbations in the satellite signal propagation, which makes the positioning solution difficult to obtain and often unacceptably erroneous. Actually, not only may signal paths to the antenna be blocked by obstacles in the satellite directions, but also multipath may occur, i.e. signal reflections with additional paths causing errors of tens of metres on the corresponding pseudo-ranges, and, in particular if only a few satellites are visible, errors of the same order of magnitude are present in the final estimated rover co-ordinates. This article addresses GNSS positioning issues in urban environments where information about building geometry is available in the form of a digital map.

1.1. *Context.* The main cause of GNSS signal distortions in city centres is due to the morphology of the local surrounding environment, particularly the building geometry. Some satellites may be hidden from the receiver point of view. Diffraction near the edges of buildings modifies the propagation path. Last, but not least, multipath also occurs, i.e. reflection from the surface of buildings, or off the road surface, that, along with a certain attenuation of the signal strength, cause a delay, and further in the tracking process, a measurement error. This is all the more severe when the reflected signal is the only one received, known as a Non-Line-Of-Sight (NLOS) signal. Braasch (1997) explains how multipath affects GNSS positioning.

1.2. *Building geometry.* Buildings are often obstacles that block Line-Of-Sight (LOS) signals, i.e. obstacles situated in between the receiver antenna and a satellite along a straight line. This makes it difficult to track the usual minimum number of four satellites to obtain a position fix.

Moreover, being blocked from both sides of a street in general, the satellites that remain visible are only located along the street direction, which results in a degraded geometry of the usable constellation. Instead of computing the usual Dilution Of Precision (DOP), it is worth considering the matrix $(\mathrm{HTH})^{-1}$ where H denotes the pseudo-range measurements' sensitivity matrix. The shape of the ellipsoid that statistically encompasses the positioning error is distorted across the street direction, which in consequence makes lateral positioning (and lane determination) very difficult. This question has been examined in detail by Wang et al. (2012) and Groves et al. (2012). This last reference mentions that in London's city centre, the pedestrian across-street DOP will typically exceed 5 for 22% of the positioning solutions, but reducing to only 12% for the along-street DOP.

1.3. *Building surface materials and multipath.* In an urban environment, building surface materials – though not as reflective as metal – cause significant multipath. If multipath interference occurs, i.e. the reception of both the direct signal from a satellite and a reflected signal, the orthogonal coding (Code-Division Multiple Access - CDMA) of the GPS signal lets the receiver correlator and discriminator design mitigate efficiently the undesirable effect of multipath. But in the eventuality that only the reflected signal is tracked, because the direct one is blocked (i.e. in NLOS), the pseudo-range measurement obtained grows unboundedly, resulting in a possibly

https://doi.org/10.1017/S0373463315000132 Published online by Cambridge University Press

large positioning deviation locally. This is because the reflected signal travels an additional significant distance compared to the corresponding direct signal.

2. EXISTING APPROACHES.

A number of different approaches exist to distinguish between whether the received satellite signals have travelled in direct path down to the antenna (satellites in LOS), or have reflected off the facades of the surrounding buildings (NLOS):

- At the hardware level, one can detect and eliminate signals arriving from below the antenna. It is generally assumed that reflected signals rising up to the antenna from the ground surface or from the roof of the vehicle itself are efficiently attenuated by its ground plane and its designed sensitivity pattern. Moreover, direct signals have right-handed circular polarization whereas many reflected signals have left-handed polarization. Also reflected signals are typically weaker. In general, signals with left-handed polarization and those with low strength are dismissed by antenna and receiver design respectively. This is done in most GNSS equipment, but with limited success, and always within the limitation imposed by an adequate balance in terms of required sensitivity.
- Other hardware approaches consist of mounting an additional sensor, when this is possible on board, for example an infrared (Meguro et al., 2009) or fish-eye (Marais et al., 2012) camera to directly measure building boundaries by image processing.
- At the navigation software level, consistency checking of the solution computed, for instance by conventional least squares, similar to Receiver Autonomous Integrity Monitoring (RAIM) is used (Jiang et al., 2011). However, should too many observations be affected, such an approach fails. Spatial constraints applied to positioning solutions can also help, these being extracted from a database such as road plane geometry for vehicles (Peyraud et al., 2013), or numerical terrain model data for land navigation in general (Jiang and Groves, 2012; Groves and Jiang, 2013), which fuses consistency checking, height aiding and also signal strength. This assumes that the accuracy of these data is at least at the same level as the GNSS measurement, which is not always true.
- Last, but not least, software approaches can also consider the use of geometric city models.

Improving GNSS positioning in urban canyons with Three-Dimensional (3D) maps has been addressed by a few research teams in the past few years: first Bradbury et al. (2007) and Suh and Shibasaki (2007) in parallel, and for the most recent references: Wang et al. (2012), Obst et al. (2012) and Peyraud et al. (2013). Whereas these articles present methods that do not model the additional path delay due to signal reflection, a few others, in contrast, will actually make this next step forward and include Bourdeau et al. (2012), Miura et al. (2013), Bétaille et al. (2013a, 2013b) and Suzuki and Kubo (2013).

Thus Bradbury et al. (2007) and Suh and Shibasaki (2007) could validate the various conditions of reception (i.e. LOS, diffraction and reflection) that were predicted by ray-tracing in a 3D city model, carrying out real world experiments. Following this approach, Wang et al. (2012) figure out the satellite visibility problem using similar techniques and data. A building boundary is computed, i.e. a satellite visibility

https://doi.org/10.1017/S0373463315000132 Published online by Cambridge University Press

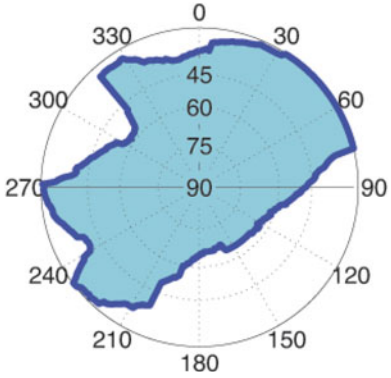


Figure 1. Sky plot of building boundaries from the perspective of GNSS users (Wang et al., 2012).

domain, represented in the usual polar diagram also called a "sky plot" (Figure 1), for any possible rover location in a grid. This building boundary makes possible LOS/NLOS separation.

Obst et al. (2012) have a similar approach, with some probabilistic aspects to cope with the uncertainty of both the model and the rover location *a priori*. This very last point (i.e. the question where to initialise and generate the geometrical computation) is actually the trickiest. First, the domain where the solution may lie can generally be constrained, in particular as concerns vehicle mobility: Jiang and Groves (2012) uses height aiding; Peyraud et al. (2013) applies road plane constraints. In Peyraud et al. (2013) no ray-tracing is done, but a virtual image is generated in the azimuth of each satellite, from which a simple geometrical process determines the corresponding critical elevation: authors have demonstrated the interest of selecting only the satellites above critical elevations. They proved the advantage of the LOS/NLOS selection versus the SNR-based (Signal-to-Noise Ratio, 3D model-free), in particular on a final position using the road constraints (Figure 2). In Groves and Jiang (2013), in addition to geometrical constraints, SNR and consistency checking (i.e. RAIM derived) have been analysed in multiple criteria combinations.

In addition, knowledge of the local building geometry makes possible the estimation of the additional path travelled due to reflection, which is an unbounded positive delay in the case of NLOS. This estimation, like LOS/NLOS selection, is also particularly dependent on the assumed position of the rover.

An exploration of the solution domain has been conducted in Miura et al. (2013). The best consistency between the assumed solution and that finally obtained from grid points ray tracing and NLOS selection and correction was sought. Bétaille et al. (2013b) proved that such consistency checking could give good results too, with the simplified street model introduced in Bétaille et al. (2013a): the idea was to test multiple lateral candidate positions across the street. Similarly, Suzuki and Kubo (2013) used particle filtering to explore the solution domain.

Bourdeau et al. (2012) show how the local building geometry can help in code tracking, which goes beyond the Position-Velocity-Time (PVT) computation stage in the receiver, and tackles the signal processing stage itself.

https://doi.org/10.1017/S0373463315000132 Published online by Cambridge University Press

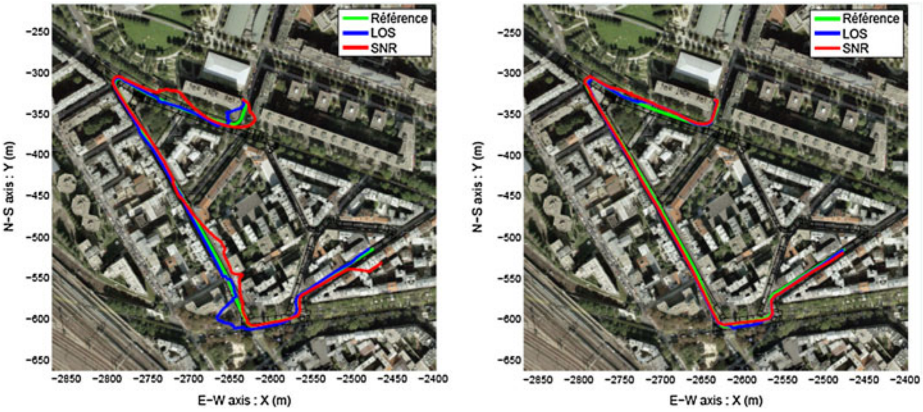


Figure 2. Overview of the solutions obtained with (LOS/NLOS or SNR) strategies without (a) and with (b) application of road constraint (Peyraud et al., 2013).

In addition to this LOS/NLOS discrimination concept, another idea (shadow matching) takes advantage of the possibility of predicting which satellites will be visible from the rover's point of view, and compares this prediction to the actual observation (Groves, 2011), and more recently in Wang et al. (2013) and Yozevitch et al. (2014). One can restrict the user's location domain by considering those satellites present in the ephemeris and for which no measurement is made. For illustration purposes, let us assume that the unique signal source considered is the Sun, this concept looks like the intuitive determination one can make of which side of a street is travelled at a given time just considering whether or not the position is in the shade or in sunlight. The same principle is used in this paper: for each GNSS satellite feasibly available in the current ephemeris file, the grid of its signal availability is determined from the city model data thus constraining what positions are realistic given the signals tracked.

These applications often seek to improve the rover position sufficiently to discriminate between different sides of a road, or between traffic lanes. Hence, they require 3D city models with geometric accuracy at the sub-metre level.

A major real-time processing problem of 3D models is due to their size and access time through embedded Geographical Information Systems (GISs). In the research work related here, urban canyons have been modelled in a very simple manner, with width and height parameters as in Bétaille et al. (2013a) but allowing for asymmetry between both sides of streets. The key point of the model produced is its simplicity, to the extent that it matches the polyline format of most digital road maps. The approximations made by this modelling have been validated in term of final positioning accuracy. The article will first develop the model (Section 3), second explain how it has been automatically computed from a public national data base available in France (Section 4), and third prove its efficacy in positioning improvement with full-scale test cases (Section 5). Last, but not least, the trade-off between the simplicity in 3D modelling, with its corresponding data storage optimality and real-time usability on the one hand, and the positioning accuracy obtained using such a model on the other hand, will be discussed, comparing our approach to the state-of-the-art (Section 6), before concluding (Section 7).

https://doi.org/10.1017/S0373463315000132 Published online by Cambridge University Press

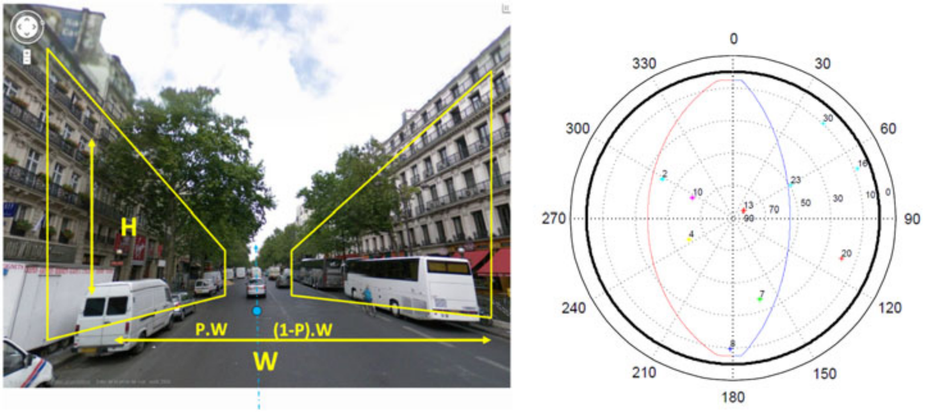


Figure 3. Urban trench and (W, H, P) parameters along a boulevard in Paris (Bétaille et al, 2013b), and an example of a "mask of visibility" with W = 30 m, H = 15 m, and P = 0·65.

3. THE URBAN TRENCH MODEL. For the initial proof of concept already published (Bétaille et al., 2013a), streets were considered as parallelepipeds, so called "urban trenches", from which LOS/NLOS satellite signal discrimination was made possible. The method consists of four phases:

- Characterisation of the 3D environment;
- Detection of the NLOS satellite signals;
- Correction of the additional path in their measurements;
- Computation of the positioning solution.

We made the choice intentionally to call our model of urban streets "urban trench" and not "urban canyon" to clearly differentiate the two concepts. "Urban canyon" is indeed often used in the literature to depict a situation when high buildings are surrounding the receiver, creating propagation problems. "Urban trench" depicts a simplified model of any street, considering that the rows of buildings can be approximated by an average geometry. Any kind of street can be modelled by the "urban trench" model, even those with low buildings that are far from being "urban canyons".

3.1. *The characterisation of the 3D environment.* Only street sections with "homogeneous geometry" are concerned (see Figures 3 and 4), while elsewhere a conventional positioning algorithm will operate. Criteria defining these sections correspond to considering streets as parallelepipeds, i.e.:

- A constant street direction;
- The same width (W) of the street all along the section;
- The same height (H) of building all along the section, and on both sides;
- A unique lateral position (P) occupied by the vehicle in this section, i.e. no lane change: this is measured from left to right, in percentage of the total width from side to side, and with the direction of the street duly defined.

https://doi.org/10.1017/S0373463315000132 Published online by Cambridge University Press

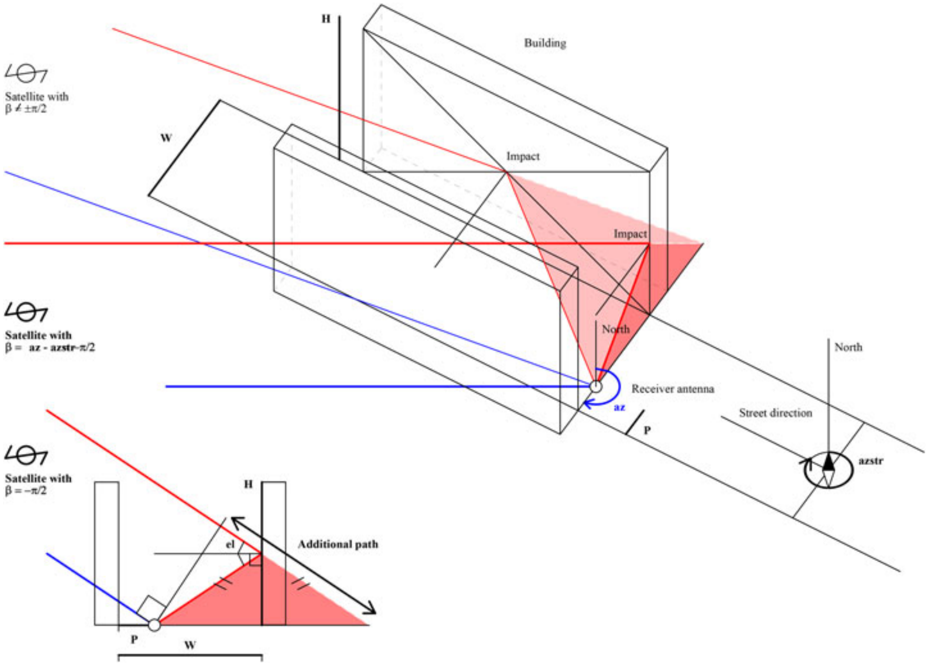


Figure 4. Urban trench geometry with W, H and P parameters and 2 NLOS satellites (one with an azimuth orthogonal to the street direction, i.e. β = –π/2); β denotes the angle difference between the satellite azimuth and the street direction, this being arbitrarily set in the data base, irrespective of the driving direction.

Several test campaigns were performed, with data collected in Nantes, Paris and Toulouse. The VERT (Véhicule d'Essais et de Reference Trajectographique, the Ifsttar Geoloc lab test vehicle) provided a decimetre-accurate GPS time-stamped reference trajectory (Ortiz et al., 2013), even in the deepest urban canyons of these city centres, by means of its embedded instruments: a LandINS from IXSEA coupling an inertial measurement unit with post-processed kinematic solutions from a Novatel dual-frequency DLV3 GNSS receiver and permanent network data. The receiver used in the test was the automotive and high-sensitivity GPS Ublox LEA6-T, 5 Hz, mounted on the roof of the vehicle.

Throughout the test areas in Nantes, Paris and Toulouse, width and height have been estimated from Google Earth views. Horizontal distances were measured manually with the rule available in Google tools, whilst total floor numbers, visible from Street View gave an approximation of vertical distances, considering floor heights of 2·5 or 3 m depending on the construction architecture.

The position of the vehicle in the across-street direction was approximated (Table 1), taking into account the number of lanes, including cycling lanes and pedestrian avenues at road section centres (in addition to sidewalks), which are common in large boulevards.

The principle here is to model the characteristics of the urban trench, where feasible, using a parameter triplet (W, H, P), which is used in the subsequent modelling and estimation scheme. This is illustrated in Figure 3.

https://doi.org/10.1017/S0373463315000132 Published online by Cambridge University Press

Table 1. Street configurations and lateral position values.

<table>
<tr><th>Configurations</th><th>Schematic representations</th><th>P values</th></tr>
<tr><td>One direction</td><td>|-_-|</td><td>0·5</td></tr>
<tr><td>Two directions or two lanes</td><td>|-__-|</td><td>0·35 or 0·65 depending on the direction or the lane travelled</td></tr>
<tr><td>Two times two directions and central tree-lined walk</td><td>|-__—__-|</td><td>0·1, 0·2, 0·8 or 0·9 depending on the direction and the lane travelled</td></tr>
</table>

Table 2. Critical elevations and additional distances for satellites situated on the left side of the street, i.e. $(-\pi < \beta < 0)$ or $(\pi < \beta < 2\pi)$.

	Critical elevations	Additional distances
1 reflection	$el_1 = el_b = arctg\left\|\frac{H}{PW}sin(\beta)\right\|$	$\Delta_{d1} = 2W(1-P)cos(el)sin(\beta)$
2 reflections	$el_2 = arctg\left\|\frac{H}{(2-P)W}sin(\beta)\right\|$	$\Delta_{d2} = 2Wcos(el)sin(\beta)$
3 reflections	$el_3 = arctg\left\|\frac{H}{(2+P)W}sin(\beta)\right\|$	$\Delta_{d3} = 2W(2-P)cos(el)sin(\beta)$

Note: for satellites situated on the right side, i.e. $(-2\pi < \beta < -\pi)$ or $(0 < \beta < \pi)$, just replace P by 1-P.

3.2. *The detection of the NLOS satellite signals.* Using the local triplet of parameters, a "mask of visibility" can be computed (Figure 3), with the assumption that the street is infinite in length. It is not necessarily symmetric, even if H is the same on both sides, because P is not always equal to 0·5. This mask will separate the sky plot into two domains: the one with the highest elevation (containing LOS satellites), the other with the lowest elevation (respectively NLOS). A set of mathematical formulae makes possible the LOS/NLOS separation, as well as, in the NLOS case, the number of specular reflections affecting each satellite signal. A maximum number of three reflections is considered. The detection of the NLOS satellites is made in real-time, as long as the appropriate streets have been modelled.

3.3. *The correction of additional path length.* In case of one or several reflections and no interference with the direct signal, part of the measured pseudo-range is directly due to the additional path travelled by the signal. Geometrical considerations are again used to compute this error, as shown in Figure 4 in the case of a unique reflection. The critical elevations and additional distances corresponding to 1, 2 and 3 reflections are given in Table 2.

3.4. *The computation of the positioning solution.* Contrary to Peyraud et al. (2013) where NLOS detected satellites were eliminated, leading to a severe decrease in availability, the solution suggested here uses LOS satellites and as many NLOS ones as necessary to obtain at least the four satellites needed for a least squares computation of the solution. So, if necessary, NLOS with one reflection were included all together first, and also NLOS with two and even three reflections successively. Of course, the additional distances in raw measurements are corrected before their use in the solution. A simple least squares estimation scheme was implemented to emphasise the impact of the LOS/NLOS detection and correction, rather than Kalman filtering for instance. Moreover, no elevation or SNR weighting strategy has been applied.

To conclude, in our test areas in Nantes, Paris and Toulouse, urban trenches were defined by a manual process, which is quite time consuming. That way, about half of the total distance travelled for test cases was covered. The main problems were:

- Poor spatial resolution of the model, due to the generalisation of a unique height for a long street section and for both street sides;
- Possible errors inevitably affecting a repeated process performed manually;
- Structure of the output data, produced for the test cases, not easily reusable for new trajectories, and totally unlinked with the usual GIS data, either street polylines or building layer.

This last point mainly motivated the initiative of additional investigations that would yield an automatic determination of geometric parameters applicable to urban streets, with an organisation similar to that of a standard city database.

Actually, to enhance the potential use of an algorithm of NLOS detection and correction in a navigation process, it is of key importance that common data used in car navigation systems be considered. In particular, a specification requirement applicable to an automatic characterisation of a city by urban trenches consists of reusing existing polylines, by just adding new attributes that depict the surrounding constraints imposed by buildings. Therefore, existing map-matching algorithms should deliver not only the road segment identity as they already do, but also the parameters required by the NLOS algorithm. These will be processed further along with the satellite observations and broadcast ephemeris in order to refine the final solution.

4. AUTOMATIC CHARACTERISATION BY URBAN TRENCHES AND ADAPTATION OF THE NLOS DETECTION AND CORRECTION.

The aim of this automatic characterisation is to determine width and height attributes of polylines, i.e. those arc segments representing the streets in position (and direction). That way, the complete geometry of urban trenches will be obtained. First we will make the hypothesis that our vehicle runs on the nearest map-matched arc segment.

To approximate the urban trench model, we use the database called BD Topo ® produced by the French Institute of Geographic and Forest Information (IGN):

- The distances on the left and the right to the closest building;
- The heights on the left and the right of the buildings along the street.

The final data item to determine is the direction of the arc segment. All these values were computed with an algorithm working with the open-source software: PostgreSQL and its extension: PostGIS (http://postgis.net/docs). Simple SQL requests have been programmed. Sections 4·1 to 4·4 will completely explain the process. It includes a map-matching step, for which we could not use the *a priori* least squares solutions because of their inaccuracy. Instead, the reference trajectory has been used so far.

https://doi.org/10.1017/S0373463315000132 Published online by Cambridge University Press

The output of the characterisation process is a simple table giving width on the left, width on the right, height on the left, height on the right and the street direction at every epoch during the tests performed.

4.1. *Overview of the method.* We now describe the process that will automatically compute W, H and P parameters from a geographic database, (the previously mentioned BD Topo ® from the IGN). It is available throughout France and provides building heights with metre accuracy, and hence is at least as accurate as the previous manual approach described in Section 3.1.

Nevertheless, we must make a few preliminary adjustments. First, despite each arc of the BD Topo ® road layer containing a width attribute, this does not include the width of the sidewalk. It is therefore necessary to directly calculate the distance from building to building facing each other in order to get W.

The position P is more difficult to determine. Indeed, the road layer provides only a set of arcs drawn in the middle of the roadway. We will assume that the position of the rover is matched onto the closest arc. Then we determine the left width (W1) and right width (W2), corresponding to the distance between the arc and the closest building on the left or on the right. Finally, the approximate location P equals W2 / (W1 + W2), and the corresponding total width W is (W1 + W2).

Concerning the parameter H, two different heights of left (H1) and right (H2) buildings will be calculated. The height H of the first model would correspond to the average of the two heights, i.e. (H1 + H2) / 2.

We can say that the new urban trench model consists of the quadruplet:

- Distance from the arc segment to the left of the street, or left width;
- Distance from the arc segment to the right of the street, or right width;
- Height of buildings located to the left;
- Height of buildings located to the right;

instead of the previous W, H and P triplet.

This paradigm shift will lead to:

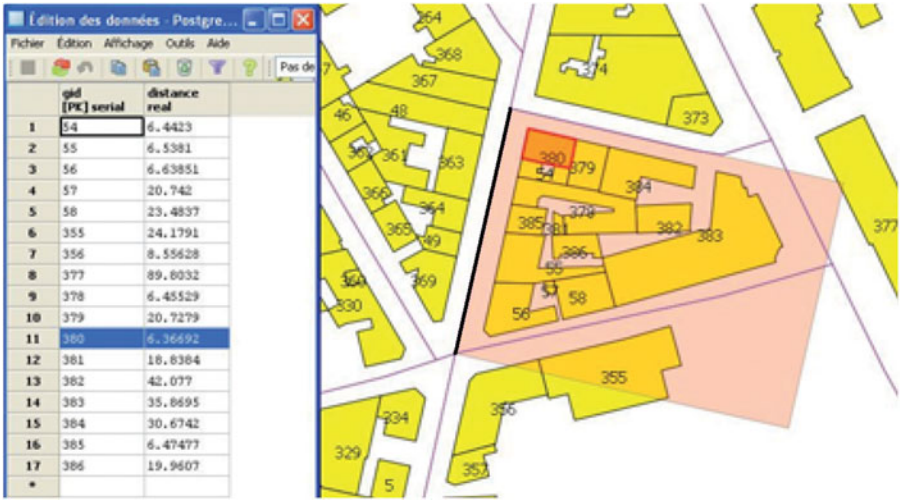
- Representing asymmetric trenches lined with buildings on one side only, or different heights of buildings on each side of the street;
- Assuming that the rover is *a priori* located on one of the arcs of the road layer, instead of using the lateral position value P, previously set using the reference trajectory.

4.2. *Algorithm for urban trench modelling.* The aim is to extract the values described in the previous paragraph, namely the distances from each arc to the left and right buildings, and their respective heights. We must therefore know the arc direction to determine unambiguously the left and right sides of a trench. We produce, for each arc of the BD Topo ®, a table (id, W1, W2, H1, H2, direction). For this we will use the PostGIS spatial database.

The shapefiles of the BD Topo ®, specifically road and building layers, are used. A selection box in Quantum GIS (a freeware software package) enables us to specify a zone windowing the experiment area. The corresponding data from the layers will be processed using PostgreSQL through the *shp2pgsql* PostGIS tool.

4.2.1. *Determination of widths.* Measuring the distance from an arc to a building uses the *ST_Distance* PostGIS function that returns the minimum distance between

https://doi.org/10.1017/S0373463315000132 Published online by Cambridge University Press



Édition des données - Postgre...

	gid [PK] serial	distance real
1	54	6.4423
2	55	6.5381
3	56	6.63851
4	57	20.742
5	58	23.4837
6	355	24.1791
7	356	8.55628
8	377	89.8032
9	378	6.45529
10	379	20.7279
11	380	6.36692
12	381	18.8384
13	382	42.077
14	383	35.8695
15	384	30.6742
16	385	6.47477
17	386	19.9607
•		

Figure 5. Half-buffer (100 m wide, see Section 4.3) used for the determination of side width; the arc under consideration is underlined.

two geometric objects, whether they are of the same type (point, line, surface) or not. As a first step, it is therefore necessary to identify buildings lining the street (left or right). For this we will use a buffer area.

The buffer takes a geometric object (e.g. a line or an arc) and a distance given by the user. In our case, we will use a half-buffer: this is a buffer area from only one side of the source geometric object.

Taking a half-buffer extended from the arc of the road layer, and a large enough input distance parameter, we will be able to identify buildings contained or intersected in this area through *ST_Intersects* function (Figure 5). The identifiers of each detected building will then be stored in a temporary table as well as their distance from the arc that is to be characterised as an urban trench.

The minimum distance in the temporary table finally fills in the width parameter for left and right sides successively, and for every arc contained in the zone.

4.2.2. *Determination of heights.* The building layer of the BD Topo ® database contains the height of each building. The distance between the arc studied and the nearest building on each side is also known, so we will again use a half-buffer to identify the buildings lining the street. However, the search width used will correspond to the width of side already calculated in the previous step to which we add a few metres to consider only the closest buildings to the street.

Once again we use the *ST_Intersects* function and store the identifier of the building and its height in a temporary table. Finally, we calculate the average height using the *avg* PostgreSQL function for each side of the arc, which fills in the height parameter for left and right sides. This is repeated for every arc in the zone.

4.2.3. *Determination of the arc direction.* The last data to extract from the map database, the direction of each arc, is essential in the following calculation of the rover position since it indicates the direction of the arc and sets the left and right sides. The *ST_Azimuth* function is used, which returns the heading between two

https://doi.org/10.1017/S0373463315000132 Published online by Cambridge University Press

given points, these being returned by the *ST_StartPoint* and *ST_EndPoint* functions applied to the arc concerned (in PostGIS 2).

4.3. *Settings of the process.*

4.3.1. *Buffers.* First, our buffers are not circular, but rectangular at their extremities: this is obtained by adding the parameter "flat" in *ST_Buffer* PostGIS function.

For the width of the street, we have to use a search distance large enough to reach the sides of the street. Indeed, our cases apply to cities (Nantes, Paris and Toulouse) where some boulevards are of a grand scale (up to 60 metres in width). In addition, the BD Topo ® road layer may contain several polylines (i.e. series of arcs) in large streets with lateral lanes or sidewalks. When the process considers an arc very close to the right-hand side, for example, the half-buffer must cover the entire left-hand side of the street before arriving at the foot of the opposite buildings. For this reason, the width of the half-buffer is fixed at 100 metres.

To retrieve the buildings along an arc (for the calculation of heights), we will apply a 5 metres half-buffer along a line offset by the W1 (or W2) value already recovered. Buildings located at a distance equivalent to the width but also slightly recessed are then included within the buffer and the correct data is recorded. Note that this 5 metres parameter may have to be adapted in case tall buildings exist behind smaller ones: however this adaptive process, if envisaged, has still not been implemented.

4.3.2. *Sampling arcs and segment resolution.* The arcs of the road layer are divided by shape points identifying a change of direction. This was done by the data editor, here the IGN, and reused with no change. Thus, the extraction of (W1, W2, H1, H2, direction) is repeated for each segment belonging to the arcs of the road layer.

4.4. *Map-matching and full output table.* Once the urban trenches have been modelled, a table with the following data is obtained: Id, W1, W2, H1, H2, direction. The next step is to determine which section is the more relevant at any one epoch in the problem of solving for the vehicle position: this is a problem of map-matching of the trajectory on the arc segments of our zone.

The measurement campaigns made in urban areas resulted in GNSS positions *a priori* widely dispersed because of the very poor environment, causing errors of up to several tens of metres. This poses a problem in the search for map-matching the points, which seeks to identify the correct urban trench (and not another). This concern is not trivial to solve, and it will be addressed in Section 6 as a perspective of this research. Failing that, we will use the reference trajectories obtained through the VERT vehicle, precise enough to allow reliable matching.

The software part of the map-matching, a basic point-to-segment algorithm (Quddus et al., 2007), starts from the GNSS points file, and produces, for each point, the identifier of the arc segment associated with it. We then convert the text file format into *dbf* to insert it into PostgreSQL through *shp2pgsql* tool.

Then, we just do a data join on the identifier of the arc segment to get a full output table. It will display for each point the GPS time, urban trench characterised and data corresponding to the attributes of the trench. The output table is exported into a text file for easier use in Matlab and final calculation of the rover position by least squares.

4.5. *Summary.* Our method of advanced positioning within urban trenches consists of four phases (see Section 3): determination of the 3D urban environment, detection of LOS or NLOS satellites, calculation of the additional distance and final vehicle position. So far, only the first phase, i.e. the automatic characterisation by urban trenches, has been described.

https://doi.org/10.1017/S0373463315000132 Published online by Cambridge University Press

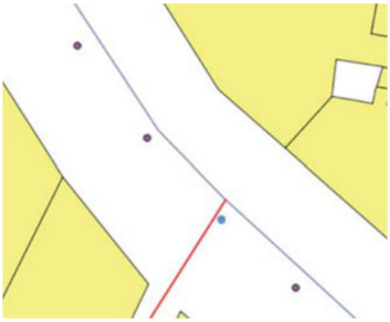


Figure 6. Perturbation of the simple map-matching algorithm used.

For the second and third phases, we have modified the scripts already written in Matlab for Table 2, replacing P by W1/(W1 + W2) in the computation of the critical elevation angles and additional distances. We did not forget, by the way, to lower the height by two metres, due to the height of the rover antenna above the road. The final calculation of the vehicle position – fourth phase – was unchanged.

4.6. *Artefacts.* The simple map-matching algorithm used does not take into account certain parameters (such as direction). Some points may mismatch. Typically, in Figure 6, the point near the crossroad will be associated with the arc perpendicular to the route, so it will not be assigned the correct parameters of urban trench. Quddus et al. (2007) reviews more sophisticated algorithms where this problem is fixed.

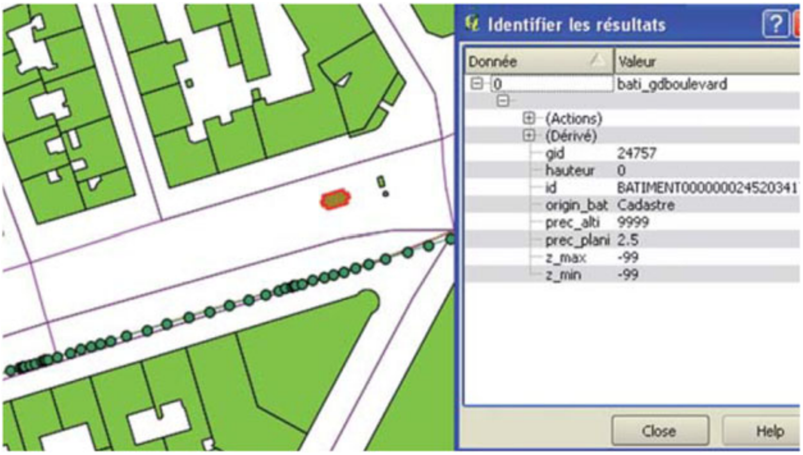
There are also artefacts related to the BD Topo ® building layer. In some areas there are "buildings" that would be more regarded as street furniture. Their height is zero (since they were borrowed directly from a map source that does not specify geometry but instead cadastre information). In Figure 7, the selected item (a kiosk) is in the middle of the street, and the reference trajectory appears in a series of points. The arc map-matched will not have correct urban trench attributes because on one side buildings are properly identified, but on the other, the width will be too low and of zero height because of the presence of this booth. It was therefore decided to remove from the building layer all the elements whose height is zero before retrieving urban trench data.

Last, the BD Topo ® road layer also includes bridges, which are shown as "crossing" arcs that actually do not intersect. These linear features are poorly characterised. The search for the nearest building returns null, so we consider that the visibility extends to the whole sky. Locally, we will not expect any improvement in the calculation of the vehicle position in these configurations.

5. THE TEST CASES AND POSITIONING IMPROVEMENT.

First tests were performed in Nantes for feasibility purposes, in the area known as *Ile de Nantes*, for 20 minutes, from which 5 minutes were included in urban trenches. Then in Paris, *XIIth district*, 2 h½ (40 minutes in urban trenches) and *Haussmann Grands Boulevards*, 5 h½ (3 h¼ in urban trenches), and in the business centre of *La Défense* (1 h and 20 minutes). In the latter, it was not possible to define properly any urban trench,

https://doi.org/10.1017/S0373463315000132 Published online by Cambridge University Press



Identifier les résultats

Donnée	Valeur
0	bati_gdboulevard
(Actions)	
(Dérivé)	
gid	24757
hauteur	0
id	BATIMENT000000024520341
origin_bat	Cadastre
prec_alti	9999
prec_plani	2.5
z_max	-99
z_min	-99

Figure 7. A kiosk (of null height) that perturbs the urban trench determination.

Table 3a. 3D median error and gain achieved in Nantes for the different test cases.

MANUAL	ALL SVS SOLUTIONS			LOS ONLY SOLUTIONS			
nb of ep. 1455	av. % 100%	nb. svs. 7·2	error med. 8·3 m	av. % 88·1%	nb. svs. 5·76	error med. 5·0 m	gain 39·5%
				NLOS CORRECTED SOLUTIONS			
				av. % 100%	nb. svs. 5·81	error med. 5·4 m	gain 35·1%
AUTOMATIC	ALL SVS SOLUTIONS			NLOS CORRECTED SOLUTIONS			
same ep. 1455	av. % 100%	nb svs. 7·2	error med. 8·3 m	av. % 100%	nb. svs. 6·0	error med. 5·2 m	gain 36·9%
AUTOMATIC	ALL SVS SOLUTIONS			NLOS CORRECTED SOLUTIONS			
other ep. 3962	av. % 100%	nb. svs. 8·1	error med. 6·7	av. % 100%	nb. svs. 7·0	error med. 5·9 m	gain 12·4%

except (10 minutes) along the *Avenue Charles de Gaulle* in Neuilly, which is quite similar to a *grand boulevard*. Lastly, a test case of 50 minutes was carried out in the city centre of Toulouse (20 minutes in urban trenches), with a building architecture very different from Paris and Nantes. All these tests were done in 2012. Tables 3a to 3e gives the results obtained in the different test cases:

- "nb of ep" is the number of epochs under consideration (5 Hz receiver data);
- "av. %" is the availability, i.e. the fraction of the number of epochs, in %, when a solution could be computed by conventional least squares;
- "nb svs" is the average number of satellites used in the computed solutions;
- "error med" is the 3D median error in position;
- "gain" is the relative decrease of the 3D median error obtained using the urban trench approach with respect to that obtained using all satellites.

https://doi.org/10.1017/S0373463315000132 Published online by Cambridge University Press

Table 3b. 3D median error and gain achieved in *XIIth district* for the different test cases.

MANUAL	ALL SVS SOLUTIONS			LOS ONLY SOLUTIONS			
nb of ep.	av. %	nb. svs.	error med.	av. %	nb. svs.	error med.	gain
11635	100%	7·3	22·4 m	68·6%	5·1	6·1 m	72·6%
				NLOS CORRECTED SOLUTIONS			
				av. %	nb. svs.	error med.	gain
				100%	5·3	8·5 m	61·9%
AUTOMATIC	ALL SVS SOLUTIONS			NLOS CORRECTED SOLUTIONS			
same ep.	av. %	nb. svs.	error med.	av. %	nb. svs.	error med.	gain
11635	100%	7·3	22·4 m	100%	5·3	9·4 m	58·2%
AUTOMATIC	ALL SVS SOLUTIONS			NLOS CORRECTED SOLUTIONS			
other ep.	av. %	nb. svs.	error med.	av. %	nb. svs.	error med.	gain
33894	99·9%	8·1	14·7	99·9%	6·4	6·2 m	57·6%

Table 3c. 3D median error and gain achieved in *Grands Boulevards* for the different test cases.

MANUAL	ALL SVS SOLUTIONS			LOS ONLY SOLUTIONS			
nb of ep.	av. %	nb. svs.	error med.	av. %	nb. svs.	error med.	gain
57938	99·7%	7·3	32·9 m	82%	4·84	8·8 m	73·3%
				NLOS CORRECTED SOLUTIONS			
				av. %	nb. svs.	error med.	gain
				99·7%	4·88	9·9 m	70·0%
AUTOMATIC	ALL SVS SOLUTIONS			NLOS CORRECTED SOLUTIONS			
same ep.	av. %	nb. svs.	error med.	av. %	nb. svs.	error med.	gain
57938	99·7%	7·3	32·9 m	99·7%	5·2	11·1 m	66·2%
AUTOMATIC	ALL SVS SOLUTIONS			NLOS CORRECTED SOLUTIONS			
other ep.	av. %	nb. svs.	error med.	av. %	nb. svs.	error med.	gain
37739	99·9%	7·7	13·5 m	99·9%	6·1	7·8 m	42·5%

The purpose was not only to validate the automatic model with respect to the manual one (which is indeed feasible where urban trenches have been manually determined), but also to measure the impact of the new modelling anywhere else, because the last model is actually available all over France.

For each zone, the statistics considered are the average number of satellites used in the solution, and the 3D median error (this, contrary to the mean error, is not affected by possible outliers). The error is defined as the difference between the computed solution and the reference trajectory at the same epoch.

The different test cases are respectively:

- Epochs where urban trenches were constructed by the manual model
  - solution (always by least squares) using all satellites
  - solution using LOS satellites only, which decreases the availability
  - mixed solution using LOS and NLOS satellites, with corrected ranges

Table 3d. 3D median error and gain achieved in *La Défense* for the different test cases.

MANUAL	ALL SVS SOLUTIONS			LOS ONLY SOLUTIONS			
nb of ep.	av. %	nb. svs.	error med.	av. %	nb. svs.	error med.	gain
3414	100%	7·4	37·8 m	99·9%	6·6	5·0 m	86·7%
				NLOS CORRECTED SOLUTIONS			
				av. %	nb. svs.	error med.	gain
				100%	6·6	5·0 m	86·7%
AUTOMATIC	ALL SVS SOLUTIONS			NLOS CORRECTED SOLUTIONS			
same ep.	av. %	nb. svs.	error med.	av. %	nb. svs.	error med.	gain
3414	100%	7·4	37·8 m	100%	6·5	5·0 m	86·9%
AUTOMATIC	ALL SVS SOLUTIONS			NLOS CORRECTED SOLUTIONS			
other ep.	av. %	nb. svs.	error med.	av. %	nb. svs.	error med.	gain
20827	93·5%	7·1	25·2	93·5%	6·1	15·5 m	38·4%

Table 3e. 3D median error and gain achieved in *Toulouse* for the different test cases.

MANUAL	ALL SVS SOLUTIONS			LOS ONLY SOLUTIONS			
nb of ep.	av. %	nb. svs.	error med.	av. %	nb. svs.	error med.	gain
5845	100%	7·6	23·7 m	69·5%	5·5	14·5 m	38·7%
				NLOS CORRECTED SOLUTIONS			
				av. %	nb. svs.	error med.	gain
				100%	5·7	16·0 m	32·6%
AUTOMATIC	ALL SVS SOLUTIONS			NLOS CORRECTED SOLUTIONS			
same ep.	av. %	nb. svs.	error med.	av. %	nb. svs.	error med.	gain
5845	100%	7·6	23·7 m	100%	5·4	14·2 m	40·0%
AUTOMATIC	ALL SVS SOLUTIONS			NLOS CORRECTED SOLUTIONS			
other ep.	av. %	nb. svs.	error med.	av. %	nb. svs.	error med.	gain
9396	99·9%	7·8	18·3	99·9%	5·6	10·1 m	45·0%

- The same epochs as with the manual model, but using the automatically generated trench models
  - solution using all satellites
  - mixed solution using LOS and NLOS satellites, with corrected ranges
- On all other epochs where the automatic model works but not the manual one
  - solution using all satellites
  - mixed solution using LOS and NLOS satellites, with corrected ranges.

The improvement is always given with respect to the basic navigation solution using all satellites.

Note that only LOS statistics have not been added in Tables 3 for the automatic model, but they are almost similar to those for the manual model. They are nevertheless aggregated and displayed in Figure 8.

https://doi.org/10.1017/S0373463315000132 Published online by Cambridge University Press

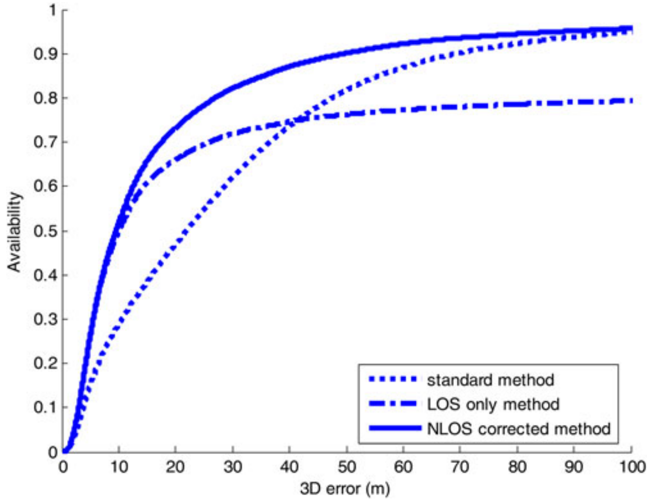


Figure 8. Cumulative distribution function of the absolute error in 3D, for standard and urban trench solutions, using the automatic 3D model, for all epochs and all test sites in Nantes, Paris and Toulouse (10 hours of data); note that the asymptote is lower when using LOS only, because solutions are often not computable, which can also be interpreted as an infinite error

Several conclusions arise: first of all, a gain (between 30% and 90%) is always achieved in urban trenches, despite various urban environments. It is slightly less (a few points less) when NLOS satellites are introduced in the least squares estimates (after range correction), but the improvement this strategy brings in terms of availability makes it valuable anyway. Actually, where the availability was reduced by 10% to 30%, almost full availability is recovered. The second conclusion concerns the automatic versus manual modelling: the gains are not significantly different. Last, but not least, a gain (between 10% and 60%) is also achieved where the automatic model is applied, which would not be achieved when the manual model is supplied. These conclusions justify the effort of studying the problem of LOS satellites selection along with NLOS ranges correction (when necessary to get epoch per epoch least squares solutions).

Figure 8 aggregates all epochs and all test sites (i.e. 10 hours of data), irrespective of whether or not they belong to the former manually modelled urban trenches. The cumulative 3D error distribution is displayed, for the standard estimator, for the estimator using LOS satellites only, and for the estimator mixing LOS and NLOS satellites, applying the new automatically modelled urban trenches.

Whereas only 4½ hours were modelled manually, here the BD Topo ® database was used automatically everywhere, which led to a near 10 hour final data set. The LOS only method availability is around 80%, while the standard and NLOS corrected methods reach almost 100% (*La Défense* showing the trickiest environment: this is Paris business centre, where tall buildings behind smaller ones have probably not been included in the 5 metres parameter used for the automatic determination of heights, see Section 4.3). The median error in 3D has been reduced from 21·7 m down to 9·4 m, i.e. 56% improvement globally.

https://doi.org/10.1017/S0373463315000132 Published online by Cambridge University Press

6. TRADE-OFF BETWEEN MODEL SIMPLICITY AND POSITIONING IMPROVEMENT.

Both the manual and automatic building models are obviously very crude compared to the approaches used by other research groups working on 3D-model-aided GNSS. Let us just mention that actually:

- the accuracy of the geometric objects in BD Topo ® is one order of magnitude below that of other databases, e.g. in France: BATI 3D ®, used by (Peyraud et al., 2013), i.e. the metre versus the decimetre; note however that BATI 3D ® covers only very few cities, like Paris *XIIth district*;
- moreover, in our approach the geometry is averaged per road arc, which penalises this approach where the buildings vary in height and/or if there are gaps between them so they do not make continuous facades;
- another drawback is due to the assumption that the urban trenches are infinite in length: at a road junction, consequently, the geometry of the street one comes from will be used until the junction centre – unless specific road arcs exist in the junction itself – despite such an area offering a relatively open sky in general. The most advanced state-of-the-art approaches already mentioned in Section 2 consider that the building geometry is specific in each satellite azimuth, whereas our approach regards urban trenches as infinite;
- no grid of possible rover positions or exploration of the solution domain is used here, but one assumes that the closest road arc is occupied to feed the building boundaries determination process.

Even if in practice the urban trench model can be computed everywhere, it will actually apply optimally only where the building geometry is regular and simple enough. For example, a grid of building boundaries (Figure 1) could provide a much better positioning performance in more singular and complex environments.

The objective in Section 6 is to measure the performance penalty in terms of positioning accuracy compared to a sophisticated model, and determine the benefit in counterpart in terms of computing performances.

In a preliminary proof of concept made in our group of research (Peyret et al., 2011), the same BD Topo ® source database was used directly and the LOS/NLOS separation, i.e. the computation of the critical elevations, was based on the use of virtual images generated in the azimuth of the tracked satellites, an original process equivalent of ray tracing for that purpose. A comparison of the selection of satellites and final accuracy using those LOS only is possible, between this method and that using urban trenches.

A series of tests were done in Nantes, in 2011, in the same area as the 2012 tests already mentioned in Section 5. Ublox LEA4-T receiver data were logged at 4 Hz, for approximately 45 minutes in the morning and in the afternoon. Tables 4 display the confusion matrix between the two approaches for the selection of satellites.

The urban trench approach obviously over-determines NLOS configurations in this data set, which confirms that its geometrical assumptions do not completely apply. In particular at intersections, the hypothesis that urban trenches are infinite in length causes false building boundaries in crossroad directions, where in fact open sky conditions apply.

In terms of positioning accuracy, the 3D median error using LOS satellites only grows from 9·0 m (using ray tracing) to 10·3 m (using urban trenches) in the morning, and from 10·8 m to 12·7 m in the afternoon, which is still below the accuracy of the solution using all satellites: 14·3 m and 15·2 m respectively. To conclude, the

https://doi.org/10.1017/S0373463315000132 Published online by Cambridge University Press

Table 4a. Confusion matrix (morning test).

98578 observations for 11249 epochs	LOS observations by urban trenches approach	NLOS observations by urban trenches approach
LOS observations by ray tracing approach	59·90%	15·48%
NLOS observations by ray tracing approach	2·81%	21·82%

Table 4b. Confusion matrix (afternoon test).

100404 observations for 11791 epochs	LOS observations by urban trenches approach	NLOS observations by urban trenches approach
LOS observations by ray tracing approach	58·44%	15·82%
NLOS observations by ray tracing approach	3·77%	21·97%

performance penalty of the urban trench modelling globally equals 16·7% for this data set (relative error growth).

On the one hand, there is an error growth, but on the other hand, much better computing performances have been achieved.

Using the previous example for computational efficiency is not straightforward. Let us simply quote that Peyraud et al., (2013) – based on the same ray tracing approach – mentions "an off-line process in replay mode": in other words, a very challenging real-time implementation. In terms of data storage, one can refer to Groves et al. (2012), which states that "a 4 GB flash drive could store building boundary data for 2,500–25,000 km of road network, depending on compression". This represents an interval between 0·16–1·6 MB/km. For Nantes centre, our model covers 188 km with less than 1 MB, i.e. 0·005 MB/km, near two orders of magnitude below, because we store a few urban trench parameters only per arc segment: Id, co-ordinates and direction, W1, W2, H1, H2 (actually we have just added two widths and two heights). So this makes the urban trench approach highly competitive for real-time implementation in navigation systems and embedded ITS applications.

7. CONCLUSIONS AND PERSPECTIVE.

We managed to automate the creation of urban trenches from the BD Topo ® database using an approach independent of the reference trajectory of the rover. This is a major evolution compared to the initial manual modelling. To start the geometric calculation of multipath, the vehicle position is map-matched onto the nearest road arc segment (we actually do not use the reference trajectory except for determining surely by map-matching in which urban trench it is located). We can also say that this assumption is relevant since the results are consistent with those obtained initially with the manual model. It is feasible to apply the technique throughout France (entirely covered by the BD Topo ® database), with a successful applicability to streets that fit the simplified building boundaries.

We have several future problems to address:

- The positioning *a priori* (before map-matching) could use a generic street geometry, and an SNR-based threshold; it is also very promising to run a filter (e.g. Kalman) instead of a least squares estimator, in order to predict this initial positioning;

https://doi.org/10.1017/S0373463315000132 Published online by Cambridge University Press

- The map-matching could be regarded as a multiple hypotheses problem, first, because multiple road segments should be considered where the position uncertainty is large, and second, because next generation maps will be certainly based on a multiple lanes road representation;
- The positioning *a posteriori* (after satellite selection and range correction) should feed the map-matching process and consequently ensure a total independence of the complete process with respect to reference trajectory;
- Instead of regarding urban trenches as infinite, the elevation mask could be characterised more finely, sampling azimuth from the rover point of view: a trade-off between computation through-put and modelling should be found. In the current research, 10 hours of data could be automatically processed, whereas only 4½ hours had been made manually, i.e. less than half obviously considered in urban trenches. A future outcome of our research would be the estimated proportion of streets where the urban trench method works best, despite its simplified building boundary representation. Its systematic application until the extremities of road arcs (i.e. even at junctions) should be reconsidered.

ACKOWLEDGEMENTS

The authors would like to thank Stephan Miquel and Leïla Fontenay, Société de Calcul Mathématique SA, for engineering the trench method initially, and Miguel Ortiz, IFSTTAR, who proceeded to the experimental campaigns. This work has been funded by the Ministry responsible for transport in France, DGITM (General Directorate for Infrastructure, Transport and the Sea), in the frame of the project named *Inturb*.

REFERENCES

Braasch, M. S. (1997). Multipath effects. *Chapter 14 in GPS: theory and applications*, Volume 1, edited by B. W. Parkinson and J. J. Spilker, Progress in Astronautics and Aeronautics Series, AIAA.

Bétaille, D., Peyret, F., Ortiz, M., Miquel, S. and Fontenay, L. (2013a). A New Modelling based on Urban Trenches to Improve GNSS Positioning Quality of Service in Cities. *IEEE Intelligent Transportation Systems Magazine*, **5**(3), 59–70.

Bétaille, D., Peyret, F., Ortiz, M., Miquel, S. and Fontenay, L. (2013b). GNSS Accurate Positioning Including Satellite Visibility Check in a Multiple Hypotheses 3D Mapping Framework. *ENC GNSS Congress*, Vienna.

Bourdeau, A., Sahmoudi, M. and Tourneret, J.-Y. (2012). Tight Integration of GNSS and a 3D City Model for Robust Positioning in Urban Canyons. *Proceedings of the 20th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2012)*, Nashville, TN.

Bradbury, J., Ziebart, M., Cross, P. A., Boulton, P. and Read, A. (2007). Code Multipath Modelling in the Urban Environment Using Large Virtual Reality City Models: Determining the Local Environment. *The Journal of Navigation*, **60**, 95–105.

Groves, P. D. (2011). Shadow Matching: A New GNSS Positioning Technique for Urban Canyons. *The Journal of Navigation*, **64**, 417–430.

Groves, P. D., Jiang, Z., Wang, L. and Ziebart, M. (2012). Intelligent Urban Positioning using Multi-Constellation GNSS with 3D Mapping and NLOS Signal Detection. *Proceedings of the 20th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2012)*, Nashville, TN.

Groves, P. D. and Jiang, Z. (2013). Height Aiding, C/N0 Weighting and Consistency Checking for GNSS NLOS and Multipath Mitigation in Urban Areas. *The Journal of Navigation*, **66**, 653–659.

Jiang, Z., Groves, P., Ochieng, W. Y., Feng, S., Milner, C. D. and Mattos, P. G. (2011). Multi-Constellation GNSS Multipath Mitigation Using Consistency Checking. *Proceedings of the 19th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2011)*, Portland, OR.

https://doi.org/10.1017/S0373463315000132 Published online by Cambridge University Press

Jiang, Z. and Groves, P. D. (2012). GNSS NLOS and Multipath Error Mitigation using Advanced Multi-Constellation Consistency Checking with Height Aiding. *Proceedings of the 20th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2012)*, Nashville, TN.

Meguro, J.-I., Murata, T., Takiguchi, J.-I., Amano, Y. and Hashizume, T. (2009). GPS Multipath Mitigation for Urban Area using Omnidirectional Infrared Camera. *IEEE Transactions on Intelligent Transportation Systems*, **10**(1), 22–30.

Marais, J., Ambellouis, S., Flancquart, A., Lefebvre, S., Meurie, C. and Ruichek, Y. (2012) Accurate Localisation Based on GNSS and Propagation Knowledge for Safe Applications in Guided Transport. *Transport Research Arena*, Athens.

Miura, S., Hisaka, S. and Kamijo, S. (2013). GPS Multipath Detection and Rectification using 3D Maps. *Proceedings of the IEEE Intelligent Transportation Systems Conference*, The Hague.

Obst, M., Bauer, S., Reisdorf, P. and Wanielik, G. (2012). Multipath Detection with 3D Digital Maps for Robust Multi-Constellation GNSS/INS Vehicle Localization in Urban Areas. *Proceedings of the IEEE Intelligent Vehicles Symposium*, Alcalá de Henares.

Ortiz, M., Peyret, F., Renaudin, V. and Bétaille, D. (2013). From Lab to Road Test: Using a Reference Vehicle for Solving GNSS Localization Challenges. *Inside GNSS*, September-October 2013, 20–34.

Peyraud, S., Bétaille, D., Renault, S., Ortiz, M., Mougel, F., Meizel, D. and Peyret, F. (2013) About Non-Line-Of-Sight Satellite Detection and Exclusion in a 3D Map-Aided Localization Algorithm. *Sensors*, 13, 829–847.

Peyret, F., Bétaille, D. and Mougel, F. (2011). Non-Line-Of-Sight GNSS signal detection using an on-board 3D model of buildings. *Proceedings of IEEE International Conference on ITS Telecommunications*, Saint-Petersburg.

Quddus, M. A., Ochieng, W. Y. and Noland, R. B. (2007). Current map-matching algorithms for transport applications: State-of-the-art and future research directions. *Elsevier Transportation Research Part C*, **15**, 312–328.

Suh, Y. and Shibasaki, R. (2007). Evaluation of Satellite-Based Navigation Services in Complex Urban Environments Using a Three-Dimensional GIS. *IEEE Transactions on Communications*, E90-B, No. 7, 1816–1825.

Suzuki, T. and Kubo, N. (2013). Correcting GNSS Multipath Errors Using a 3D Surface Model and Particle Filter. *Proceedings of the 26th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2013)*, Nashville, TN.

Wang, L., Groves, P. D. and Ziebart, M. (2012). Multi-Constellation GNSS Performance Evaluation for Urban Canyons Using Large Virtual Reality City Models. *The Journal of Navigation*, **65**, 459–476.

Wang, L., Groves, P. D. and Ziebart, M. (2013). GNSS Shadow Matching: Improving Urban Positioning Accuracy using a 3D City Model with Optimized Visibility Scoring Scheme. *NAVIGATION*, **60**, 195–207.

Yozevitch, R., Ben-Moshe, B. and Dvir, A. (2014). GNSS Accuracy Improvement Using Rapid Shadow Transitions. *IEEE Transactions on Intelligent Transportation Systems*, **15**(3), 1113–1122.

https://doi.org/10.1017/S0373463315000132 Published online by Cambridge University Press