ION GNSS – 2014, Tampa, Fl, Sept 8-12 2014

# Gap bridging in Precise Point Positioning

Andrew Simsky
*Septentrio NV, Belgium*

## BIOGRAPHY

Dr. Andrew Simsky holds a Ph.D. in Physics from the University of Moscow, Russia. He is working as a senior GNSS scientist at Septentrio in Leuven, Belgium. His research interests include differential and standalone navigation algorithms and performance analysis of GNSS receivers and signals.


## ABSTRACT

The vulnerability to interruption of signal tracking is a well-known problem of standard PPP technology based on float ambiguities. In conventional PPP algorithms, any outage of tracking would trigger a complete reset and re-initialization of ionosphere-free phase ambiguities. In other words, for float PPP there is no difference between a "cold start" and a "hot start". This presents a serious problem for all land applications, where a vehicle or a surveyor would occasionally pass under an obstacle. This paper presents the performance assessment of an effective gap bridging algorithm, which significantly enhances the usability of the PPP technology, especially for land applications. This algorithm is integrated into an industrial float PPP engine and has been tested in real-life applications. It has been proven to provide near-perfect bridging in benign open-sky conditions and to dramatically reduce both convergence time and positional deviations in more challenging environments.


## INTRODUCTION

The outages of tracking present a real challenge for the use of PPP with float ambiguities in land applications. Every time a vehicle goes under a bridge or in a tunnel, the tracking is interrupted, and the convergence, which typically lasts 10-20 minutes, begins afresh. This behavior is illustrated by the blue curve in Figure 1, which shows a typical height profile for a land survey. Gap bridging mends the gaps: the convergence goes on as if there were no gaps (ideally so). The purpose of gap bridging is to render float PPP more useful for land applications.

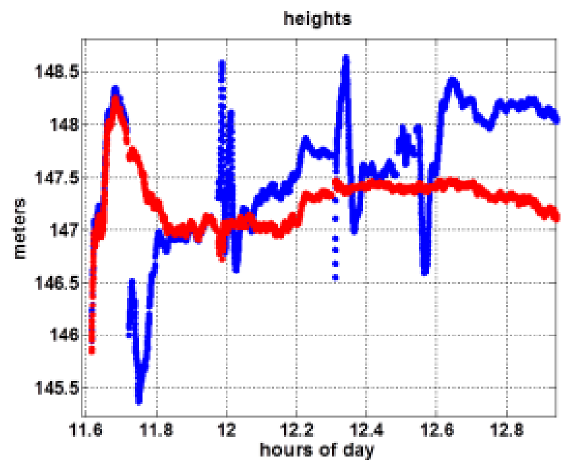


**Figure 1. A typical height profile of a land survey: blue is standard PPP, red is PPP with gap bridging.**

In this paper, we shall make a brief review of published hitherto methods of gap bridging, explain the principles of gap bridging for float PPP, and then present a number of test cases, which demonstrate the performance improvement of PPP in land applications caused by the use of gap bridging.

## GAP BRIDGING METHODS

Research on gap bridging started in 2009 with two important publications: of S. Banville and R. B. Langley [1-3] and of J. Geng [4, 5]. Publications on this subject can be clearly divided into two groups: gap bridging for PPP with float ambiguities ([1-3] and the present work), and gap bridging for PPP with integer ambiguities [4-8]. In the works of the latter group, it is typically assumed that L1/L2 ambiguities are fixed before an outage; the goal is to fix them again after the outage as quickly as possible. The works of the first group do not use any assumptions about the state of convergence: the goal is to resume convergence from where it stopped. The difference between the two groups is not only in a different treatment of ambiguities, but also, and even more importantly, in a different treatment of ionosphere delays. The methods of the second group are based on the assumption that the absolute value of ionosphere delay is known before the gap (indeed, it can be computed when L1, L2 ambiguities are known), so it can be predicted after the gap. Thus predicted absolute values of ionosphere delays are further used as constraints to facilitate fixing of L1, L2 ambiguities after the outage. Conversely, the methods of the first group do not assume any knowledge of ionosphere delays, although they do benefit from the low rate of change of ionosphere delays.

## PRINCIPLES OF GAP BRIDGING IN FLOAT PPP

Gap bridging works as follows: at some point before the signal outage (the *storage point*), the float PPP state is stored. After the gap, at another point, which we shall call the *recovery point*, it is restored (Figure 2). The general principle of gap bridging is simple: if the baseline vector between the *storage point* and the *recovery point* is known, the position of the *recovery point* can be predicted based on the known position of the *storage point*. The PPP is re-initialized from this predicted position, and the convergence of float iono-free ambiguities thus resumes from where it was interrupted.

If the baseline is known with relatively high precision, the accuracy of the predicted position of the *recovery point* will mostly depend upon the accuracy of the *storage point*. If the convergence before the gap was mature, and the position of the *storage point* is sufficiently precise, the position of the *recovery point* shall also be predicted precisely. If the convergence has just recently started, and the position of the *storage point* is, say, 50 cm off, then the position of the *recovery point* will also be 50 cm off in the same direction. Hence, re-initialized convergence will resume with the same biases as at the last epochs before the outage.

In [1-3], the *storage point* is treated as an RTK base station, while the state after the gap is treated as a rover. The baseline between the *storage point* and the *recovery point* is computed with cm-level accuracy as long as the standard resolution of double-differenced ambiguities becomes possible. Following the logic of [1-3], the new values of iono-free ambiguities can be computed directly, based on resolved double-differenced ambiguities, without the need to initialize them from the predicted position of the *recovery point*. Operating in the ambiguity domain is ultimately equivalent to operating in the position domain. The implementation of this method in GNSS receivers is feasible in principle, but is cumbersome, taking into account actual limitations of the receiver's FW environment. Therefore, our design for the gap-bridging algorithm, although initially inspired by [1-3], was substantially influenced by practical concerns. Our algorithm, currently implemented in the firmware of Septentrio's receivers, is simple, and, at the same time, sufficiently robust to handle real-life 'messy' outages when satellites are randomly coming and going. This algorithm uses two GNSS constellations: GPS and GLONASS, but can easily be expanded to include any other GNSS used in PPP.

.

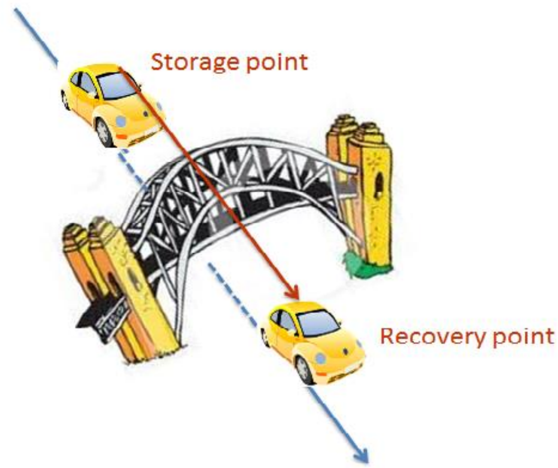


**Figure 2. The principle of gap bridging: the baseline vector connecting the *storage point* and the *recovery point* is shown in red.**

## TESTING CAMPAIGN

The goal of the testing campaign was to test the behavior of our gap bridging algorithm in real-life environments representative of land applications where PPP can be used. In all the tests, real-time PPP corrections provided by the TerraStar global augmentation service were employed. This was a natural choice because TerraStar, as its name suggests, was designed for land applications. We present here some typical results of these tests to demonstrate the measure of performance improvement, which the gap bridging can bring in actual applications.

## STATIC OPEN-SKY TEST

We begin with a basic roof test. The antenna was installed on the rooftop of Septentrio's office in Leuven, Belgium. The GNSS signals from the antenna were interrupted (by an electrical switch) every hour for 30 seconds. This is a typical configuration used to test the convergence of float PPP. In Figure 3, red is gap bridging, blue is standard PPP; the plot speaks for itself. Three cycles are zoomed (Figure 4 - Figure 6) to see in a more detailed manner how gap bridging is handled. In fact, it works almost ideally, with some very short re-convergence 'tails', typically no longer than 10 seconds.

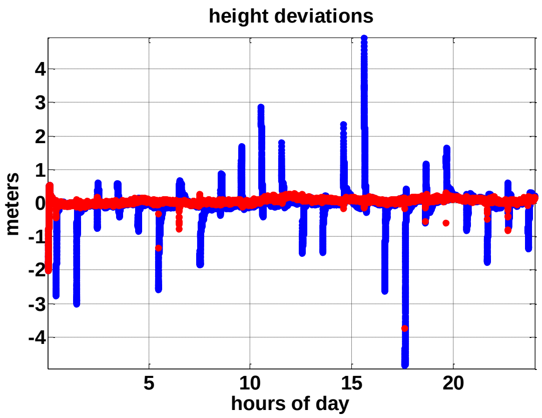


**Figure 3. The roof test: the antenna signal was interrupted every hour for 30 seconds.**

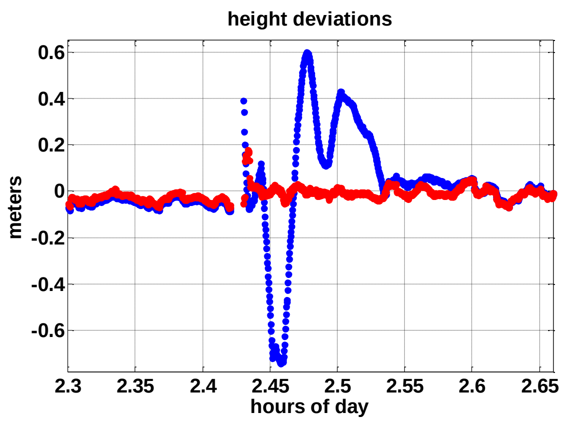


**Figure 4. The same roof test, one of the cycles is zoomed (case #1).**

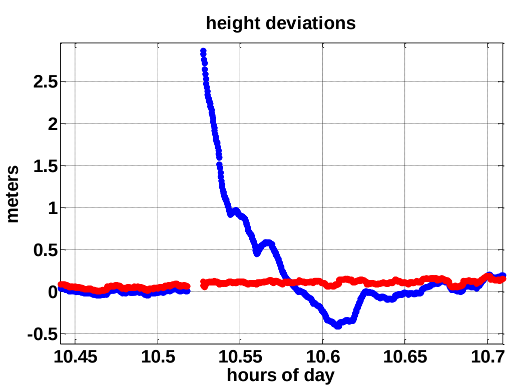


**Figure 5. The same roof test, one of the cycles is zoomed (case #2).**

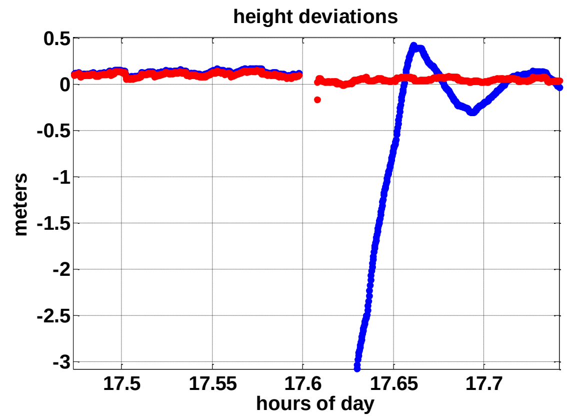


**Figure 6. The same roof test, one of the cycles is zoomed (case #3).**

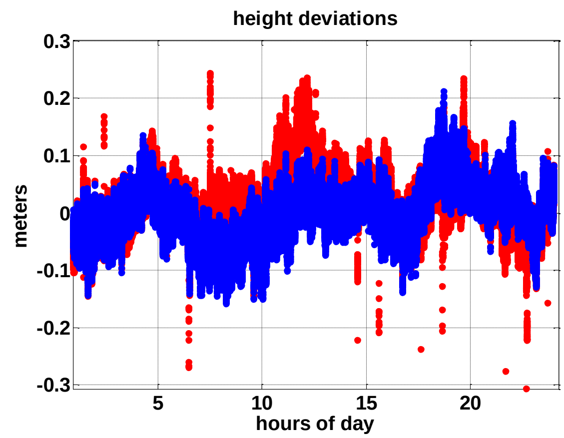


**Figure 7. Height deviations for gap bridging in the same roof test compared with continuous PPP processing (blue curve, no interruptions).**

**Figure 8. Environment of the roof test.**

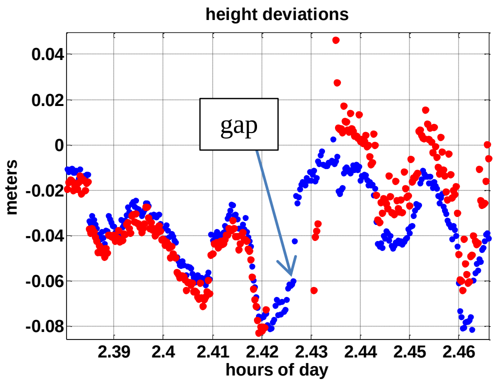


**Figure 9. One of the gaps is zoomed to appreciate how well the gap-bridged solution after the gap agrees with the continuous solution.**

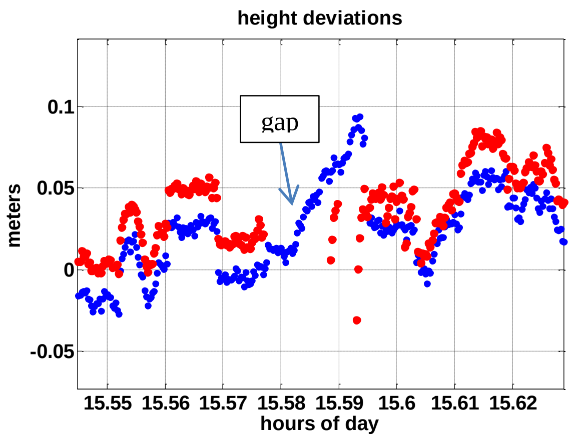


**Figure 10. Same as the previous figure, another gap.**

| RMS | height | east | north | 3D |
|---|---|---|---|---|
| Gap bridging | 5.5 | 3.1 | 3.0 | 7.1 |
| continuous | 4.5 | 2.2 | 2.4 | 5.6 |

**Table 1. Accuracy of gap bridging with 24 30-sec interruptions during the day compared to the continuous processing of a concurrently collected daily data set w/o interruptions. These statistics correspond to the data presented in Figure 7**

In this test we observe near-perfect performance of gap bridging owing to a near-perfect open-sky environment on the rooftop of Septentrio's building. Figure 8 shows that surrounding buildings are on the same level with ours, hence there are no reflectors higher than our antennas. The high quality of gap-bridging in this open-sky environment is further illustrated by the comparison of gap-bridged and continuous solutions shown in Figure 7, Figure 9, Figure 10, and in Table 1. The statistics in Table 1 demonstrate that with gap-bridging active, the outages cause almost no deterioration in comparison with the continuous solution (with no interruptions) shown in blue in Figure 7. The details zoomed in Figure 9 and Figure 10 show that after the gap (an interruption of the red curve), the red points of the gap-bridged solution join, after a short transient, with the blue points of the continuous solution.

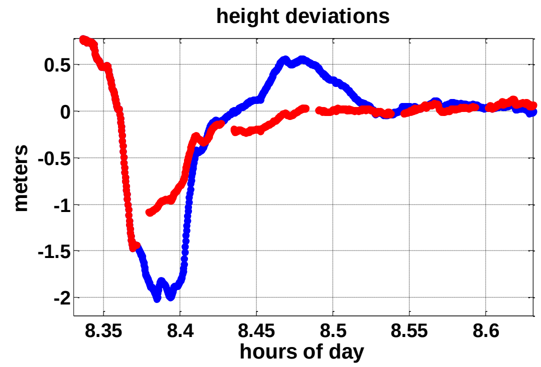


**Figure 11. Gap bridging with interruptions during the initial stage of convergence, example 1.**

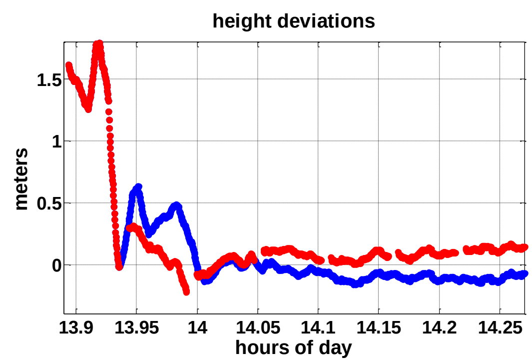


**Figure 12. Gap bridging with interruptions during the initial stage of convergence, example 2.**

Figure 11 and Figure 12 show that gap bridging is effective at any stage of convergence. It does not require the convergence to be complete or mature in any sense. Here five 30-sec interruptions were introduced every 3 minutes starting from the beginning of convergence. It can be seen that although the convergence is somewhat disturbed by the gaps, it still goes on in about the same manner as without interruptions (blue curve) and takes about the same time to complete.

## STATIC SURVEY TEST

The next static test was performed in a more high-multipath environment (Figure 14), in a spot between the buildings in the block where Septentrio's office is situated (Figure 13). This spot is surrounded with 3-storeyed buildings, which generate multipath and occlude a significant portion of the sky. The antenna was static (on a tripod) and was covered with a screen each 15 min, making sure that the tracking has stopped.

**Figure 13. The environment of the static survey test.**

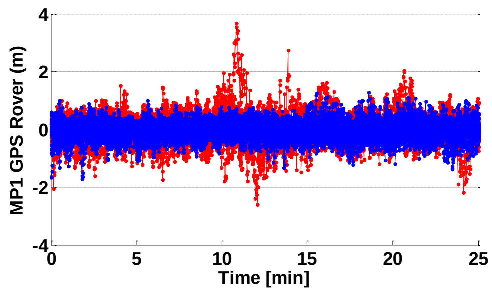


**Figure 14. Multipath of L1-CA code in the static survey test (red) compared to the roof (blue).**

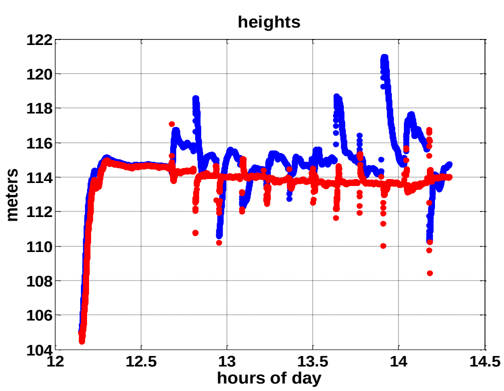


**Figure 15. Results of the static survey test #1. In this and all the following plots, the gap bridging solution is shown with red color, while standard PPP solution is blue.**

The results, presented in Figure 15 and Figure 16 show that the gap-bridging algorithm experiences greater difficulties than in open-sky tests. The re-convergence lasts longer (up to 1-2 min) and initial deviations sometimes reach values of 1-2 meters. Nevertheless the gap-bridged solution is still a lot more successful in handling gaps than standard PPP. The usual convergence by the standard PPP algorithm also takes longer and suffers from higher positional deviations than in open-sky tests.

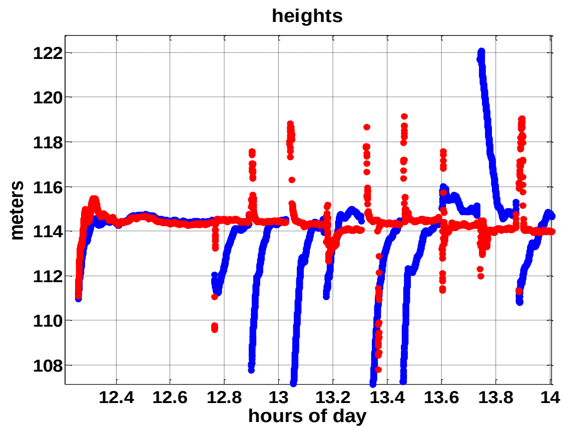


**Figure 16. Results of the static survey test #2.**

## PEDESTRIAN SURVEY

This test occurred in an open-sky environment (Figure 17) which is typically used to test standard float PPP. In such a survey there are no physical reasons for interruptions of tracking, but accidents may happen. This time the antenna got disconnected, then re-connected again. This event is catastrophic for a standard PPP solution, shown in blue in Figure 18, but causes little disturbance for gap bridging.

**Figure 17. The environment of the pedestrian survey.**

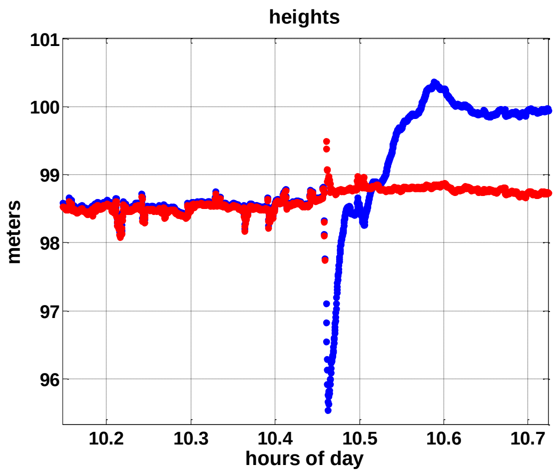


**Figure 18. Height profile for the last part of the pedestrian survey. The antenna was disconnected and then quickly re-connected about 15 min before the end of the test.**

## A DRIVE THROUGH THE WOODS

When driving through Belgian woods in summer (Figure 19), one often finds himself under a fairly dense canopy. The reception of GPS signals is intermittent, and PPP restart attempts are alternating with periods of DGPS solution.

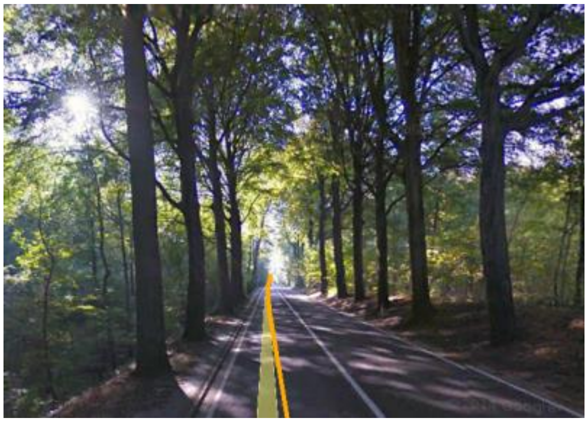

**Figure 19. The environment of the drive test through the wood; the area is near Leuven, Belgium**

In this case, the whole drive through a densely forested area took about 3 minutes and happened to be a valid example of a 'messy' outage: the satellites are coming and going randomly. The logic of the gap bridging algorithm skips 'the mess' and latches onto the right solution after a final exit from the wood. The failed attempts to restart PPP inside the wooded area are ignored. The whole 3-min drive is thus treated as one long gap.

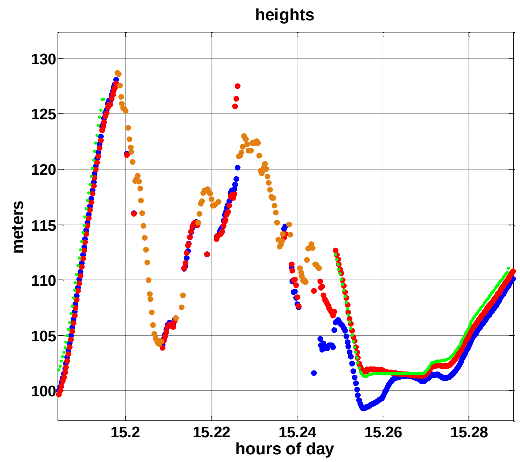


**Figure 20. Height profile for the drive through the woods. Here and in the following figures the green curve indicates RTK reference, while orange color indicates DGPS.**

The height profile (Figure 20) shows that, after the car leaves the forest, the gap-bridging solution gets onto the right track; its height profile is very close to the reference RTK solution (shown in green), while standard PPP makes a jump greater than 5m. Figure 21 is the Google Earth image of the whole forested area which was crossed.

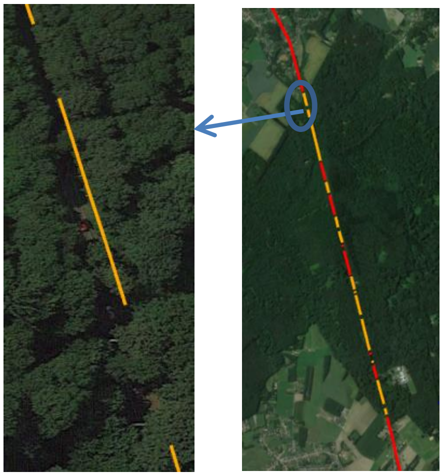

**Figure 21. The satellite image on the right shows the whole forested area, which was crossed in the direction from bottom of the image to its top. Red line shows the gap-bridged PPP solution, while orange line shows route segments with DGPS. The image on the left is a zoom; it shows how the canopy is blocking the road.**

**Figure 22. Septentrio's test vehicle used in driving tests.**

## A FREEWAY DRIVE

Driving on freeways, we are getting quite often under overhead crossings and bridges. In this example we show the situation when two bridges were separated only with a short time interval of about 15 s (points in Figure 24 corresponds to 1-Hz samples).

**Figure 23. Two bridges on a freeway drive; positioning results are shown in Figure 24.**

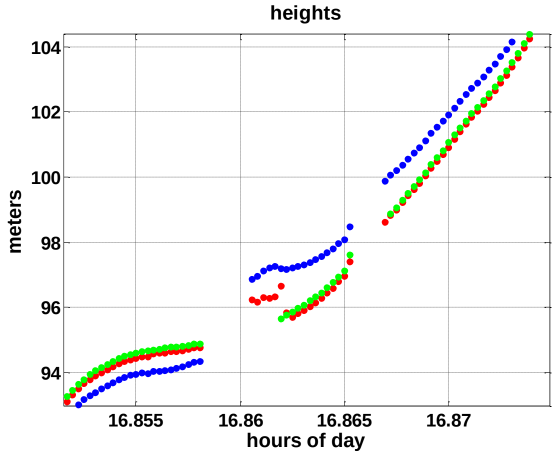


**Figure 24. This height profile corresponds to the track shown in Figure 23.**

The height profile is shown in Figure 24. It can be seen that the standard PPP solution (blue curve) makes a jump at the first bridge, while the gap-bridged solution is always consistent with the reference except for a few seconds after the first bridge.

## DRIVING THROUGH A TUNNEL

In our street testing, we often choose the ring road of Leuven, which includes a tunnel under a railway station.

**Figure 25. Satellite image of the area around the tunnel under the Leuven railway station.**

We present here two cases, when this tunnel occurred in our testing. In one case (Figure 27) the standard PPP handles the signal outage fairly well, and both PPP solutions are close to reference. In the second case (Figure 28) the standard PPP solution makes a jump, while the gap bridging is still close to reference.

**Figure 26. Street view of the exit from the tunnel.**

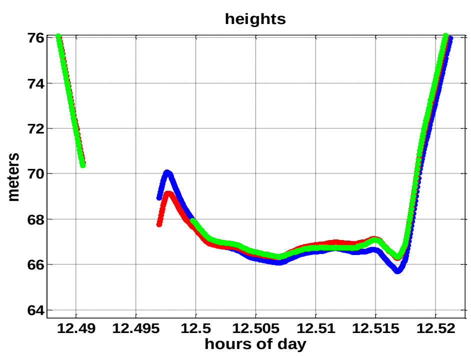


**Figure 27. Height profile after the tunnel, case #1.**

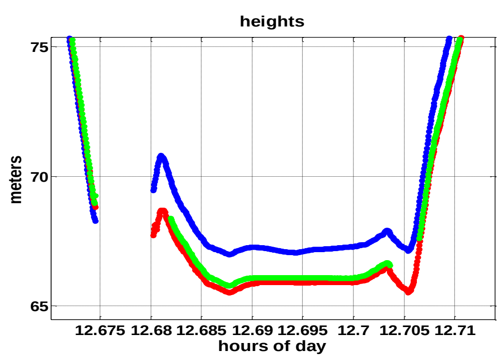


**Figure 28. Height profile after the tunnel, case #2. Here the standard PPP (blue curve) makes a jump.**

## URBAN 'CANYON'

In cities, urban canyon conditions may occur in narrow streets even if the buildings are relatively low. The case shown in Figure 29 - Figure 31 was apparently quite challenging for both versions of PPP. After driving through the streets along the waterway, when a half of the sky was masked out, the vehicle went into an 'urban canyon', where the PPP solution became intermittent.

**Figure 29. The street view of this 'urban canyon'.**

**Figure 30. The satellite image of the area around the 'urban canyon' and the route of the test vehicle (de Vaart, Leuven, Belgium).**

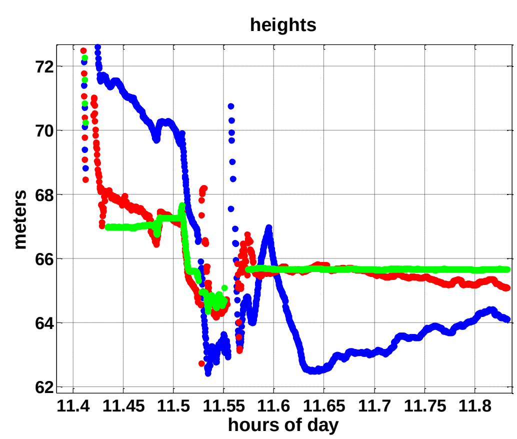

**Figure 31. The height profile for the ride through the 'urban canyon'.**

The height profile for this case can be found in Figure 31. This case is difficult for both PPP solutions, but gap bridging still provides a solution much closer to the RTK reference than standard PPP, which is significantly more vulnerable to code multipath at initial stages of convergence.

## A RIVER TEST

River navigation can be considered as a kind of a 'land application' even though it occurs on water. When moving via inland waterways, a boat goes regularly under bridges and its situation is not very different from the situation of a land vehicle. Therefore the boat test was included in this set of tests. It took place in the city of Bordeaux, France, on the Garonne river. The barge used in this test (Figure 32) was moving through the city in the direction of the sea.

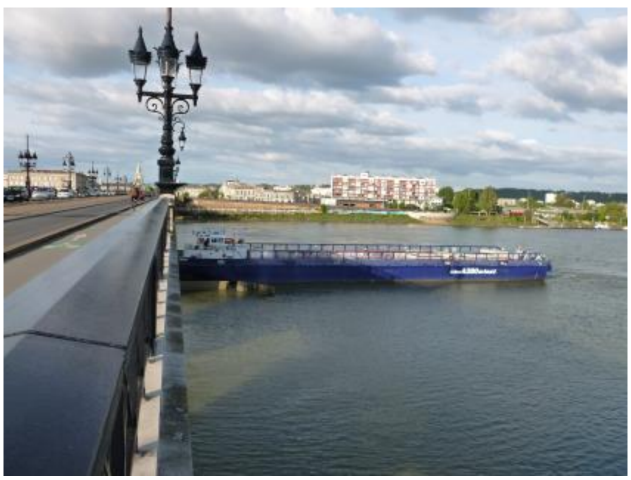

**Figure 32. The barge used in this test goes under a bridge.**

**Figure 33. Part of the route through the city of Bordeaux with two bridges.**

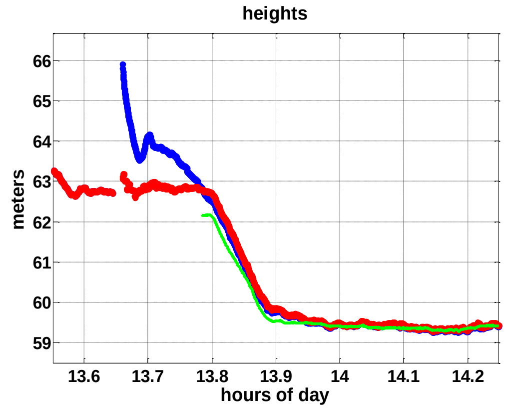


**Figure 34. The first interruption of tracking in this test. The convergence was still immature, that is why one can see a 50 cm height bias with respect to the RTK reference (green). Later on the bias phases out.**

The first interruption in the course of this test is shown in Figure 34. At the moment of the interruption, the convergence was yet incomplete, and a 50-cm height bias with respect to the reference can be observed. This bias phases out in about 20 minutes and is not present in further plots (Figure 36, Figure 38), which show two cases of the outages caused by bridges.

**Figure 35. One of the bridges, corresponds to the height profile in Figure 36.**

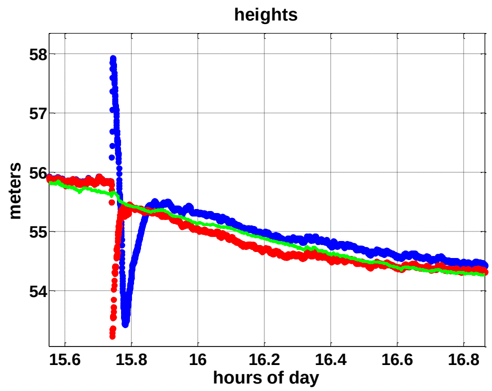


**Figure 36. Height profile for passing under a bridge shown in Figure 35.**

**Figure 37. Bridge case #2, corresponds to the height profile in Figure 38.**

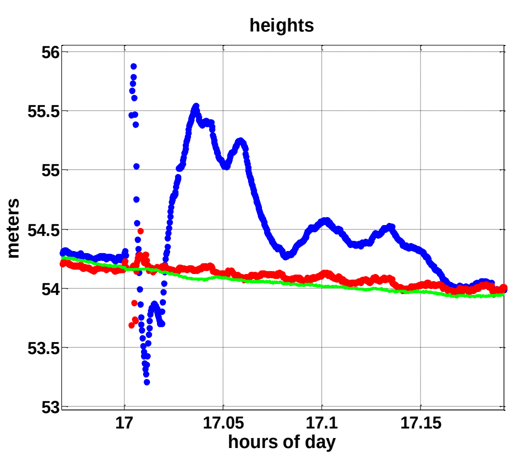


**Figure 38. Height profile for passing under a bridge shown in Figure 37.**

## HANDLING OUTAGES OF PPP CORRECTIONS

Our gap bridging algorithm can also handle gaps in the stream of PPP corrections. In practical applications such gaps may occur due to the occlusion of geostationary satellites, which transmit these corrections in L-band. This problem is more typical for high-latitude areas where elevation angles of geostationary satellites are low. The gaps of PPP corrections cause interruptions in PPP (after corrections are timed out), although the tracking might still be continuous. In the absence of tracking interruptions, mending gaps in corrections is a relatively easy task.

Situations with tracking gaps occurring during corrections outages are somewhat more challenging, but can still be handled by our algorithm. Figure 39 presents an example when a tracking gap of 30 s was caused to occur in the middle of a simulated corrections outage, which lasted 40 minutes. The tracking gap in this case is not visible because PPP solution is absent more than 30 minutes.

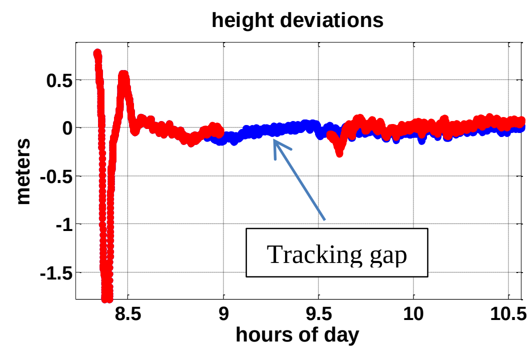


**Figure 39. A 40-min outage of PPP corrections is combined with a tracking gap of 30 sec in the middle of the outage.**

## CONCLUSIONS

An efficient gap bridging algorithm was developed and successfully integrated into Septentrio's GPS/GLONASS float PPP engine. The purpose of gap bridging is to mend outages of tracking, which normally would cause the restart of convergence in a standard float PPP solution. With gap bridging, convergence resumes approximately in the same state in which it had stopped before the outage, and the solution looks continuous (in an ideal case, as if there were no gaps). Typical duration of re-convergence is 10-20 s in benign open-sky environments; this duration can grow to 1-2 minutes in more challenging cases with more multipath and masked satellites.

To evaluate the performance of gap bridging, a test campaign was organized which included static and kinematic tests, pedestrian as well as automotive and on a river. It was confirmed that gap bridging significantly improves the performance of float PPP for land applications.

**ACKNOWLEDGEMENTS**

I am thankful to my colleagues at Septentrio for support and discussions, especially to Frank Boon, Bruno Bougard, Frank Kleijer, Richard Deurlo, Joao Viana, and Pierre Nemry.

**REFERENCES**

1. Banville, S., Langley, R. B.., *Improving real-time kinematic PPP with instantaneous cycle-slip correction*, Proceedings of ION GNSS 2009.
2. Banville, S., Langley, R. B. *Instantaneous cycle-slip correction for real-time PPP applications*. NAVIGATION, Journal of the Institute of Navigation, **57**, 4, pp. 325-334 (2010).
3. Banville, S., Langley, R. B. *Cycle-slip correction for single-frequency PPP*. Proceedings of ION GNSS 2012.
4. Geng, J., *Rapid re-convergence in real-time Precise Point Positioning with ambiguity resolution*. ION GNSS 2009, pp. 2437-2448.
5. Geng, J., Meng, X., Dodson, A. H., Ge, M., Teferle F. N. *Rapid re-convergences to ambiguity-fixed solutions in precise positioning.* Journal of Geodesy, **84**, 2010, pp. 705-714.
6. Leandro, R., Talbot, N. Ch., Vollath, Ul. *GNSS signal processing with ionospheric bridging for re-convergence*, US patent, 2012.
7. Leandro, R., Landau, H., Nitschke, M., Glocker, M., Seeger, S., Chen, X., Deking, A., Ben Tahar, M., Zhang, F., Ferguson, K., Stolz, R., Talbot, N., Lu, G., Allison, T., Brandl, M., Gomez, V., Cao, W., Kipka, A., *RTX positioning: the next generation of cm-accurate real-time GNSS positioning.* Proceedings of ION GNSS 2011, Portland, OR, US, 19-23 September 2011, pp. 1460-1475.
8. Mervart, L., Rocken, C., Iwabuchi, T., Lukes, Z. Kanzaki, M., *Precise Point Positioning with Fast Ambiguity Resolution – Prerequisites, Algorithms and Performance*. ION GNSS+, 2013.