

**POLITECNICO**  
**MILANO 1863**

Politecnico di Milano  
Department of Civil and Environmental Engineering  
Doctoral Programme in Environmental and Infrastructure Engineering

# Low cost GNSS receivers: navigation and monitoring activities

Doctoral Dissertation of:  
Marco Negretti

Supervisor:  
Prof. Ludovico Biagi

Tutor:  
Prof.ssa Giovanna Venuti

The Chair of the Doctoral Program:  
Prof. Alberto Guadagnini

XXIX cycle - 2013/2016





*To Giulio Regeni,  
waiting the true.*



# Contents

<b>Abstract</b>	<b>1</b>
<b>1 The Global Navigation Satellite System</b>	<b>5</b>
1.1 Preface . . . . .	5
1.2 The GNSS services . . . . .	5
1.2.1 Global Positioning System - GPS . . . . .	6
1.2.2 Globalnaya navigatsionnaya sputnikovaya sistema - GLONASS . . . . .	9
1.2.3 Galileo . . . . .	10
1.2.4 BeiDou Navigation Satellite System - BDS . . . . .	11
1.3 The Absolute Positioning technique . . . . .	13
1.4 The Relative Positioning technique . . . . .	19
1.4.1 Single differences . . . . .	19
1.4.2 Double differences . . . . .	20
1.5 Multipath . . . . .	22
1.6 The NMEA format . . . . .	23
<b>2 Navigation: improve accuracy using data from Reference Stations</b>	<b>27</b>
2.1 Introduction . . . . .	27
2.2 The MEP project . . . . .	27
2.3 Data available . . . . .	30
2.4 Correct rover trajectory with CORS corrections . . . . .	30
2.5 Data from static occupation . . . . .	34
2.6 Data along the test path . . . . .	43
2.6.1 Survey setup . . . . .	43
2.6.2 Path 1: survey with Smart Mobile Devices and Geodetic receivers . . . . .	51
2.6.3 Path 2: survey with Smart Mobile Devices, u-blox and Geodetic receivers . . . . .	54
<b>3 Navigation: improve accuracy using cartographic data as references</b>	<b>61</b>
3.1 Get and organize the reference cartography . . . . .	61
3.2 Correction using path segments as reference . . . . .	67
3.3 Correction out of the buildings . . . . .	72
3.4 Correction on buffers . . . . .	78
3.4.1 Basic principle . . . . .	78

3.4.2	Buffers with constant dimension . . . . .	84
3.4.3	Buffers with variable dimension . . . . .	84
3.4.4	Reclassification: segments selection in two steps . . . . .	85
3.5	Comparing results . . . . .	92
<b>4</b>	<b>Local monitoring: simulation of landslide movement</b>	<b>95</b>
4.1	Introduction . . . . .	95
4.2	Experiment 1: static survey . . . . .	96
4.3	Analysis task in monitoring dynamic displacements experiments . .	102
4.3.1	Significance analysis . . . . .	102
4.3.2	Congruence analysis . . . . .	102
4.4	Experiment 2: only horizontal displacements . . . . .	103
4.4.1	Set up and survey . . . . .	103
4.4.2	Processing and data analysis . . . . .	106
4.5	Experiment 3: horizontal and vertical displacements . . . . .	108
4.5.1	Set up and survey . . . . .	108
4.5.2	Processing and data analysis . . . . .	113
<b>5</b>	<b>Conclusions</b>	<b>123</b>
<b>A</b>	<b>Processing software</b>	<b>129</b>
A.1	Bernese GPS Software . . . . .	129
A.2	Leica Geo Office (LGO) . . . . .	129
A.3	goGPS . . . . .	130
A.4	RTKLIB . . . . .	130
<b>B</b>	<b>GNSS receivers used</b>	<b>131</b>
B.1	Leica GX 1230 . . . . .	131
B.2	u-blox . . . . .	131
B.2.1	u-blox EVK-7P . . . . .	132
B.2.2	u-blox EVK-4T . . . . .	133
<b>C</b>	<b>Correct Rover with CORS correction: graphs of Como and Milan surveys</b>	<b>135</b>
	<b>References</b>	<b>167</b>
	Bibliography . . . . .	167
	Cartography . . . . .	170
	<b>List of figures</b>	<b>171</b>
	<b>Tables</b>	<b>180</b>
	<b>Acknowledgements</b>	<b>183</b>

# Abstract

In this PhD research different conditions of using of low cost GNSS receivers were explored in **navigation** and **monitoring activities**. These receivers are cheap and widespread, but they are single-frequency receivers and then less accurate than the geodetic dual frequency receivers.

In **navigation** applications the single-frequency receivers are largely used: in car navigation systems, smart watches, tablets and smartphones they are integrated. In most of application the accuracy of tens meters usually reached in point positioning is enough, but there are some particular cases where it is needed to increase it: this is the case of the MEP (Map for Easy Path) project, studied in this thesis, where the goal is to reach an accuracy of meter order in point positioning.

The aim of the MEP project is to improve the urban mobility of the physically impaired people: mapping obstacles, identifying accessible paths and also, very important, give a feedback to identify problems and define methods to improve mobility, because for physically impaired people get around the city can be very difficult. For him it can be very important to know the obstacles before the journey starts, in order to plan the path to avoid contingencies, but the mapping activity of obstacles is expected to be carried out by volunteers mappers, with common mass market devices, with a nominal accuracy that is not enough to warrant a correct mapping activity.

Therefore the final accuracy should be improved and two different procedures were implemented and tested with the aim to correct positions estimated with these low cost receivers:

- correction using data from Continuously Operating Reference Stations (Chapter 2);
- correction using local cartography as reference (Chapter 3).

In **monitoring activities** the GNSS geodetic receivers are already used, but there are situations where these receivers are not suitable, mainly due to the great economic effort that they require. The adoption of low cost GNSS receiver can expand the field of application of GNSS techniques, but their accuracy in positioning must be increased. In Chapter 4 will be presented two experiments done to verify if it is possible to reach the accuracy of centimetres order using typical geodetic techniques in post processing with low cost receivers.

---

## Navigation: correction using data from CORS

The first method uses data from the nearest CORS to the low cost receiver (rover).

Let's consider the CORS optimally estimated position and the CORS positions estimated epoch by epoch: the first one is the actual position when the others are affected by the environmental effects presents at the epoch in which the survey was done. Therefore the difference between actual position and position at a generic epoch  $i$  it can be assumed due to specific errors present at this particular epoch. For correlated in space errors, if the rover is near the CORS, you can assume that these errors will be more or less the same that afflict estimations done with the GNSS low cost receiver. It is possible to compute the difference between actual and estimated CORS position and therefore to try to improve accuracy of rover positioning applying to the rover trajectory the corrections from CORS data, computed epoch by epoch.

In the correction procedure it must be taken into account also of the different satellites visibility between CORS and rover: rover usually is in a worse position than the CORS, so it can view only a subset of satellites viewed by CORS. Therefore a software has been developed to remove satellites not observed by rover from CORS Rinex and so to compute correction using the same set of satellites viewed by the rover.

Two different kind of test have been done:

- in ideal conditions, with a static occupation in a two fixed points, positioned on the roof of two Politecnico buildings, with a good satellites visibility;
- in normal working conditions, with data collected during several repetition of two test paths, in a typical urban scenario.

For the first test case, two test site have been used: one in Milan, near the MILA CORS station, on the roof of a Politecnico building and another one in Como near COMO CORS station, for a total of 15 days of data from static observations.

Observations were processed and corrected with data from the two near CORS stations: the final results were improved by the correction, with most of statistical errors indexes improved: therefore the tested correction method in static survey worked.

The goal of second case test was to verify the performance in an urban context. Two test paths were defined in order to analyse different conditions of satellite visibility. The test paths were repeated several times and data were collected using a geodetic receiver and some low cost receivers (single frequency). The data from the geodetic receiver were processed by a kinematic postprocessing and provided the benchmark epoch by epoch to evaluate the positioning accuracy of the low cost receivers tested.

Errors statistic were compute before and after applying corrections from the CORS like did in the first case. Unlike what it was obtained with the data from static positioning, in this case the accuracy after the correction does not improve. This is probably the effect of local noise, due to particular properties of the area where the survey was made (like multipath), that obviously can not be removed using data from reference station, where this kind of noises are not present.

### Navigation: correction using local cartography as reference

Another way to correct data collected is using local cartography as reference. As reference map it was chosen OpenStreetMap (OSM), due to the several advantages offered: it is a collaborative project, the data access is free and permits to use, download and modify data freely, without particular restrictions, keeping them always up to date. Therefore it is possible get data and organize them in a local database with the data collected during the survey activities and so to use the database tools to match data and then to correct the positioning.

To manage the cartographic database it was chosen PostgreSQL and its extension PostGIS that implements additional supports for geographic objects.

Different policies were explored to implement the corrections, using different elements of cartography as reference.

- Path segments as reference: in this case the only acceptable position for a collected point is assumed on a OSM path segment and all the points that are not on a OSM segment must be corrected.
- Building as exclusion: in this method the point position will be classified as wrong only if the point is inside a building. The point of view is reversed respect the previous approach: now it is "all the points positions are right, except those that satisfy the condition posed (to be inside a building)", when before it was "all the points positions are wrong, except that satisfy the condition posed (to be on a segment)". This policy was chosen to avoid the possible overcorrection problems present in the previous method.
- Buffer as reference: this method uses the path segments like the first correction method viewed, but in this case the point position will be considered wrong only if the distance from a segment will be greater than a certain threshold. If this condition is verified, the point in a wrong place will be moved at the threshold distance, not on the segment.

The core of this correction approach is in establishing a tolerance buffer: if the point is inside the buffer, it will not be correct, if it is outside, the position will be corrected using as new place its projection at the edge of the buffer. It's an intermediate approach between the two methods seen before.

The different correction procedures were compared checking how many points, after correction procedure, are near the actual path done, where the actual path is know and specifically defined for these tests. All the methods implemented improve the accuracy of points collected, but there is not a method that is always better to another one: the final results depend significantly on the conditions in which the survey was done, the site characteristic and from the cartography used as reference. This last aspect is very important: an incomplete or wrong reference datum can lead to consider a point position wrong because far from any element on the map, when in fact the problem can be the map itself. For this reason it is important to have a reliable cartography and easily upgradeable, like OSM, that permits to add new paths as they become available.

---

### Local monitoring

The purpose of local monitoring is to model displacements, evolution and deformations of civil engineering structures, for example bridges, dams, and local phenomena such as landslides. This monitoring activity is very important to avoid damages and possible victims. Different techniques are possible to do local monitoring, between these, in the last fifty years, the use of GNSS receivers has had a great development that permits to reach accuracy less 1 cm, in case of dual frequency receivers.

Nevertheless, the high cost of these receivers can represent a problem (a great number of receivers needed, risk to lose the instrumentation,...) then it could be interesting verify if it is possible do local monitoring with GNSS techniques, but using low cost receivers. The accuracy obtained with low cost receivers has not yet completely tested, so in this thesis it was verified:

- how reliable are the results obtained or rather, how many false or missed alarms are expected per unit of time (significance);
- the ability to identify at least the order of magnitude of a displacement (congruence).

The experiment were planned in order to assess the precision and the accuracy of low-cost GNSS receivers in monitoring dynamic displacements, with the aim of evaluating which is the precision level reachable, comparing scheduled displacements and detected displacements obtained using these kind of GNSS receivers.

Two different tests were done using two different tools to control the imposed displacements.

In the first one only horizontal displacements were applied: the receiver antenna was shifted horizontally of 5 mm for each session, three session per day, three hour per session. A total displacement of 10 cm was aimed but, due to an instrumental problem, the test stopped at 5.5 cm.

In the second test both horizontal and vertical displacements were made. From the start position the antenna was shifted, alternately, of 5 mm in horizontal and 5 mm down in vertical, every 2 hours. The total movement reached of 10 cm, both in vertical and horizontal directions.

The obtained results shown how the general displacement trend is recognizable: along the horizontal direction the results are satisfactory, with the estimated points close to the imposed positions. Along the vertical direction the results are worse: the general trend is still recognizable, but the individual movements are almost impossible to identify, probably due to the high noise of the cheap antenna used. Therefore a low cost receiver can be used, under specific constraints, in survey activities. The vertical component is not well estimated, so the the main displacement in the monitored area should be in the horizontal component to use efficiently the low-cost receiver.



# Chapter 1

# The Global Navigation Satellite System

## 1.1 Preface

In this chapter a description will be given of the main navigation satellite systems currently available at global scale or available in the next future, GPS, GLONASS, Galileo and BeiDou, with a description of the historical evolution and which developments are planned for each system in the coming years. Then two positioning techniques will be treated, absolute positioning and relative positioning, a brief introduction concerning the multipath effect and then a section about the NMEA standard to close this introduction to the basic arguments to which reference will be made widely in this thesis.

## 1.2 The GNSS services

A navigation satellite system is a constellation of artificial satellites designed to permit the determination of position, velocity and time of a point at the Earth surface using radio signals broadcast by satellites. A Global Navigation Satellite System (GNSS) is a navigation satellite system than has global coverage, therefore it can be used every where on Earth [23].

It is possible identify three main different component: space segment, control segment and user segment.

### Space segment

To provide a global navigation service a GNSS system must have a satellites constellation which insures at least four<sup>1</sup> satellites visible and their radio transmissions can be received everywhere and any time on the Earth. The number of satellites and their orbit should be calculated to optimize coverage and service availability. In every satellite the GNSS equipment includes radio transceivers to send and receive signal from Earth, atomic clock and onboard computer. The main source of power is supplied by solar panels but there are also batteries as secondary source power.

---

<sup>1</sup>In the section 1.3 will be explained why at least four satellites must be visible

### Control segment

The Control Segment is composed of a system of control stations on ground. These stations monitor satellites transmissions, perform analysis, and send commands and data to the constellation. For example they trace satellites to estimate their orbits (ephemerides), check the satellites clock errors and other parameters of which the satellite needs to generate the navigation message and they upload them to the satellites.

The Control Segment is the responsible for the proper operation of a GNSS system.

### User segment

The User Segment consists on the couple GNSS receiver - antenna which receive signals from satellites and then solve the navigation equations in order to provide:

- the receiver position coordinates;
- the time.

The accuracy with which is determined the position is of meters order, but it can be improved using particular tricks and others sources in addition to data received from the satellites.

The first GNSS systems were developed for military purposes and only later it was decided to granted access to the system also to civilian use. This openness has resulted, in a few years, to an exponential growth in the number of receivers and applications that were not originally planned. The GNSS Market Report [17] has estimated that the number of GNSS enabled devices in 2014 were about 3.6 billion units.

Main part of these devices are smartphones (3.08 bln), followed by devices used in automotive for road applications (0.26 bln). Professional devices, used for example to do surveying, mapping, GIS applications, transport networks (i.e. aviation, rail, maritime), agriculture and many others applications, are less numerous, but the impact is in any case important.

At the regional level the most of GNSS devices are in North America and Europe, but the main growth for the next years will be in Asia, opening new perspective for the GNSS market.

There are four global navigation systems, two already operational from several years, **GPS** and **GLONASS**, another, **BeiDou**, it is operational only at a local scale, but it is expanding a global scale, and then **Galileo** that it will reach the full operational in the next few years.

#### 1.2.1 Global Positioning System - GPS

GPS is the GNSS system developed by United States Government<sup>2</sup>: it is the first navigation system, the deploy started on 22 February 1978 with the launch of the first satellite and it is still the most widespread system around the world [33].

Originally developed for the U.S. military, the U.S. Government decided to make it available also for civilian purpose, when it was yet in the experimental phase, after the incident that involved the Korean Air Lines Flight 007 in 1983

---

<sup>2</sup><http://www.gps.gov>

[37]. The plane deviated of the planned flight paths, probably to cause an error in inertial navigation system, entered in a restricted airspace of the Kamchatka Peninsula and it has been shot down near Sakhalin Island by USSR air force.

In December 1993 there was the declaration of the Initial Operating Capability, with 24 operational satellites in orbit (minimum of satellites to warranties positioning 24h), followed in June 1995 by the Full Operational Capability with more than 24 satellites in orbit, to ensure operability also in case of malfunctions of some satellite. At June 2016 there were 31 operational satellites in orbit<sup>3</sup>: the operative GPS constellation is a mix of old and new satellites. The table 1.1 summarizes features of the past, current and future generations of satellites.

### GPS Navigation Message

Every GPS satellite uses at least two **carrier frequencies** to transmit navigation signals: these signals contain **binary codes** and **navigation data** [28].

In detail, you have:

- Carrier frequencies, radio frequency sinusoidal signal at a given frequency,
  - $L1 = 154f_0 = 1572.42$  MHz;
  - $L2 = 120f_0 = 1227.60$  MHz;
  - $L5 = 115f_0 = 1176.45$  MHz.

An oscillator provides to the generation of a base signal at a nominal frequency of 10.23 MHz ( $f_0$ ) from which carrier frequencies and binary messages are generated ([52] and [53]):

- Binary codes, sequences of zeroes and ones, which allow the receiver to determine the travel time of radio signal from satellite to receiver; they are called Pseudo-Random Noise sequences or PRN codes:
  - Coarse/Acquisition (C/A), civil signal;
  - Precision Code (P), moreover the Y-Code is used in place of the P-code whenever the Anti-Spoofing (A/S) mode of operation is activated;
  - L1 Civil code (L1C);
  - L2 Civil code (L2C);
  - L5 codes;
  - M-code, military, no many information are available.

The availability of the same PRN signal in both frequencies, L1 and L2, it allows to remove the errors due to ionospheric effect, as it will be discussed in the next chapters.
- Navigational data D, used to provide information on the satellite ephemeris, clock bias parameters, satellite health status and other complementary information.

The older satellites transmit only C/A and P(Y) code carried on L1 (C/A and P(Y)) and L2 (only P(Y)) frequencies. From the IIR(M) satellites series two new codes were introduced: a second civil signal on L2 (L2C) and a military code (M); with the IIF series the third civil code L5 was introduced (see table 1.1). The generation of GPS output signal is made combining the different binary signals

---

<sup>3</sup>Updated information about constellation status are available here:  
<http://www.navcen.uscg.gov/?Do=constellationStatus>

and carried frequencies, as it is shown in figure 1.1<sup>4</sup>

satellite type	launch period	Navigation Messages	Operating
1st generation			
block-I	1978-1985	C/A code on L1 P(Y) code on L1 & L2	0
2nd generation			
block-II	1989-1990	C/A code on L1 P(Y) code on L1 & L2	0
block-IIA (Advanced)	1990-1997	C/A code on L1 P(Y) code on L1 & L2	0
block-IIR (Replenishment)	1997-2004	C/A code on L1 P(Y) code on L1 & L2	12
block-IIR(M) (R. Modernized)	2005-2009	All previous signal plus 2nd civil signal on L2 (L2C) and a new military code (M-Code)	7
block-IIF (Follow-on)	2010-2016	All previous signal plus 3rd civil signal on L5 (L5)	12
3th generation			
GPS III	from 2016	All previous signal plus 4th civil signal on L1 (L1C)	in production

Table 1.1 – Past, current and future generations of GPS satellites

### GPS Performances

Two different levels of service are provided by US Global Positioning System:

- the Standard Positioning Service (SPS) [51], provided on L1 frequency, with access to coarse acquisition (C/A) code and a navigation data message (D) and available to all GPS users;
- the Precise Positioning Service (PPS) [50], provided on both L1 and L2 frequencies, with a precision (P/Y) code ranging signal with an encrypted navigation data message reserved for authorized users; this is a highly accurate military positioning.

The values of levels of performance provided by these official documents are very conservative and furthermore the performance obtained with GPS depends strongly on the kind of survey done and from the receiver used. For example using only the signals received from the satellites, the level of accuracy is nowadays [33]:

- C/A-code receivers: 5 - 10 m;
- P/Y-code receivers: 2 - 9 m.

Instead, if a GPS is used in a differential mode the results will be very different:

- C/A-code DGPS receivers: 0.7 - 3 m;
- P/Y-code DGPS receivers: 0.5 - 2.0 m.

<sup>4</sup>Figure source: "Navigation with GPS", handouts by Athanasios Dermanis, Aristotle University of Thessaloniki - Department of Geodesy and Surveying

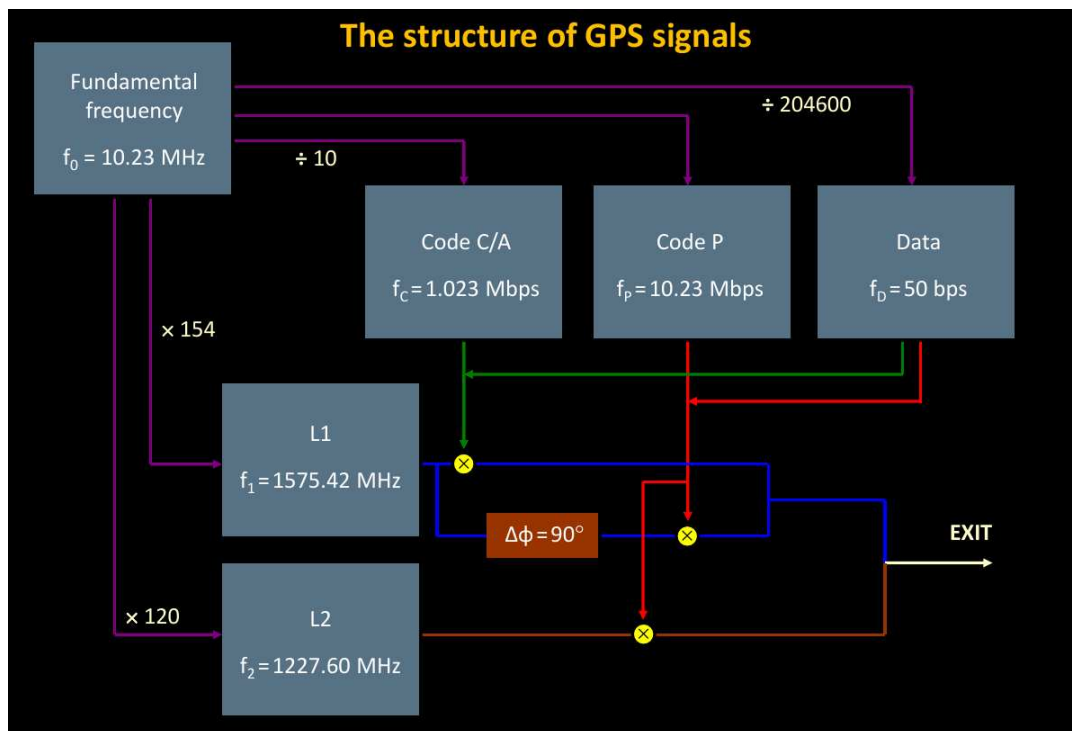


Figure 1.1 – The generation of GPS signals

### 1.2.2 Globalnaya navigatsionnaya sputnikovaya sistema - GLONASS

The satellite navigation system GLONASS<sup>5</sup> is the equivalent of GPS designed by Soviet Union [25] and now managed and developed by Russian Federation. At the end of 1960s it was started to think a system that would permit easily positioning to military force around the world, but the formal requirements were completed in 1970. In the 1976 the the Soviet Union government decided to start the development of the system: design work was carried out by specialists led by Vladimir Chermisin and the first launch took place in 1982. In 1991 Russian Federation took place of Soviet Union, it continued development of constellation and the system was declared operational in 1993, reaching its optimal status with 24 operational satellites in 1995. After reaching of the full operative, the economic crisis that hit the Russian Federation prevented launches of new satellites to replace the old ones, that would have been needed to maintain the full network. The constellation reached its lowest point, with only 6 operational satellites, in 2001, making the network basically inoperable. With the improvement of economic condition, a new program with the purpose to restoring the full constellation, was launched in 2002<sup>6</sup> and the full operative was reached again in 2012, with a small delay than expected.

GLONASS is the only satellite navigation system that employs FDMA<sup>7</sup> for the transmission of its navigation signals while GPS, Galileo and BeiDou use CDMA<sup>8</sup>

<sup>5</sup><https://www.glonass-iac.ru/en/>

<sup>6</sup>Federal Targeted Program "Global Navigation System" 2002–2011 (Government Decision n.587)

<sup>7</sup>Frequency Division Multiple Access

<sup>8</sup>Code Division Multiple Access

technique. The main difference between these two transmission techniques is that in FDMA all satellites transmit the same PRN code, but in different (dedicated) carrier frequencies when in CDMA all satellites transmit in the same carrier frequency, but with a different (dedicated) PRN code. Multiple access techniques, as CDMA and FDMA, are generally employed in communication systems where several users need to share the same medium for transmission. This is the case of GNSS satellites that transmit their signals over the same medium in the L-band.

The reason of this different choice is mainly historic, however GLONASS is moving in the direction of achieving higher interoperability with other GNSS system and from the latest class M satellites (starting from satellite 755) the signals are transmitted also in CDMA technique (L3 signal). At the present<sup>9</sup> are operational 24 satellites, plus 2 spare satellites, that all belong to M-type. An other satellite, belong to K-type, is in flight tests phase. In the K series a second frequency (L2) will be introduced also for CDMA signals starting from satellite 17L.

In table 1.2 it is possible to see the historical evolution of GLONASS satellites. For each series of satellites are indicated the navigational messages broadcast. L1, L2, L3 are the different carriers used, "O" means civil signal (standard precision), when "S" is restricted signal (high precision), the last letter indicates if is a FDMA signal (F) or a CDMA signal (C).

satellite type	launch period	Navigation Messages	
		FDMA	CDMA
GLONASS	1982-2005	L1OF L1SF, L2SF	
GLONASS-M	2003-2016	L1OF, L2OF L1SF, L2SF	L3OC for SVs (space vehicles) 755+
GLONASS-K	2011-2018	L1OF, L2OF L1SF, L2SF	L3OC, L2OC and L2SC for SVs 17L+
GLONASS-K2	2018-2024	L1OF, L2OF L1SF, L2SF	L1OC, L2OC, L3OC L1SC, L2SC

Table 1.2 – Past, current and future generations of GLONASS satellites. The access to SF and SC signals is restricted

### 1.2.3 Galileo

The GPS and GLONASS systems were built for military scope, with some encoded signals and the possibility to degrade the accuracy if needed. For example the position information from GPS could be degraded by the application of universal Selective Availability (SA) by the US military. SA was disabled on 2 May 2000 and the newer GPS satellites would not be capable of implementing Selective Availability: the satellites launched in 2009 (Block IIF) and all subsequent satellites are stated to not support SA<sup>10</sup>. When all old satellites will be replaced, SA will cease to be an option, but in any case GPS and GLONASS systems are

<sup>9</sup>GLONASS constellation status, <https://www.glonass-iac.ru/en/GLONASS/>, page visited at 2016-09-06

<sup>10</sup><http://georgewbush-whitehouse.archives.gov/news/releases/2007/09/20070918-2.html>

still partly under military control. The European Galileo system was launched by European Union through the European Space Agency<sup>11</sup> and the European GNSS Agency<sup>12</sup>, to give to GNSS users a reliable alternative that, unlike these other programs, remains under civilian control [16]. The Galileo project was started more than 10 years ago with the launch of two experimental satellites, GIOVE-A and GIOVE-B, in 2005 and 2008 respectively, serving to test the technologies that was chosen to adopt in the system. After this first phase of study four In Orbit Validation (IOV) satellites were launched (two in October 2011 and other two in October 2012) [21], very close to the final satellite design, to validate the Galileo concept and start validation tests of positioning (a minimum of 4 satellites is needed to calculate position in three dimensions, see section 1.3).

The test phases ended on 22 August 2014 with the launch of the first two Full Operational Capacity (FOC) satellites (GSAT0201 and GSAT0202). An anomaly in the flight launch injected the satellites into an incorrect orbit, so GSAT0203 and GSAT0204, launched successfully on 27 March 2015, have been the first two fully operational satellites

At the moment<sup>13</sup> there are 9 satellites operational, 2 in testing phase, 2 under commissioning and 1 not available due to technical problems.

The Galileo headquarter is in Prague, Czech Republic; there are also two main ground operations centres, one at Oberpfaffenhofen, near Munich in Germany and another one at Fucino, near L'Aquila in Italy.

#### 1.2.4 BeiDou Navigation Satellite System - BDS

The BeiDou Navigation Satellite System (BDS) is a GNSS system developed by Chinese government that will be capable of providing positioning, navigation, and timing services to users on a continuous worldwide basis [2].

The first BeiDou system, officially named BeiDou Satellite Navigation Experimental System<sup>14</sup>, included three satellites and its coverage and applications were limited to China and neighboring regions. BeiDou-1 navigation system required few satellites because the satellites were in geostationary orbit. The disadvantage of this choice is the limit of the coverage areas: BeiDou-1 service was available from longitude 70°E to 140°E and from latitude 5°N to 55°N (see figure 1.2).

The first two satellite were launched in October (BeiDou-1A) and December 2000 (BeiDou-1B). The third satellite (a backup satellite, BeiDou-1C), followed in May 2003 and this meant the start of BeiDou-1 navigation system. In February 2007 Chinese launched also a fourth satellite (BeiDou-1D, always a backup satellite).

As expected after of the experimental navigation system BeiDou-1, it is started the launch of satellites of BeiDou-2, that at the end of its implementation should become a global GNSS service, like GPS, GLONASS and Galileo.

BeiDou-2<sup>15</sup> is not an extension to the BeiDou-1, but it is a new system that it

---

<sup>11</sup>ESA - [http://www.esa.int/Our\\_Activities/Navigation/Galileo/Launching\\_Galileo](http://www.esa.int/Our_Activities/Navigation/Galileo/Launching_Galileo)

<sup>12</sup>GSA - <https://www.gsa.europa.eu/>

<sup>13</sup>Galileo constellation status, <https://www.gsc-europa.eu/system-status/Constellation-Information>, page visited at 2016-09-06

<sup>14</sup>Also known as BeiDou-1

<sup>15</sup>Previous knows as COMPASS

will have a constellation of 35 satellites:

- 5 geostationary orbit satellites, to ensure backward compatibility with BeiDou-1;
- 30 non-geostationary orbit satellites, that will offer complete positioning coverage of the globe.

In April 2007 the first satellite (Compass-M1) of BeiDou-2 was successfully put into its working orbit. Then the launches continued just to reach 8 satellites in December 2011 when the system went into operation on a trial basis: it has started providing navigation, positioning and timing data to China.

In December 2012, 16 satellites for BeiDou-2 have been launched (see table 1.3) and the system began offering services for the Asia-Pacific region: at this time, the system could provide positioning data between longitude 55°E to 180°E and from latitude 55°S to 55°N (see figure 1.3).

In the 2015 started the third phase that should bring the system at a global coverage by 2020.

Block	launch period	Satellite launched	Currently in orbit	Launch planned
1 (BeiDou-1)	2000–2007	4	0	0
2 (BeiDou-2 local)	2007–2012	16	14	0
3 (BeiDou-2 global)	From 2015	6	6	18

Table 1.3 – BeiDou satellites launch sequence

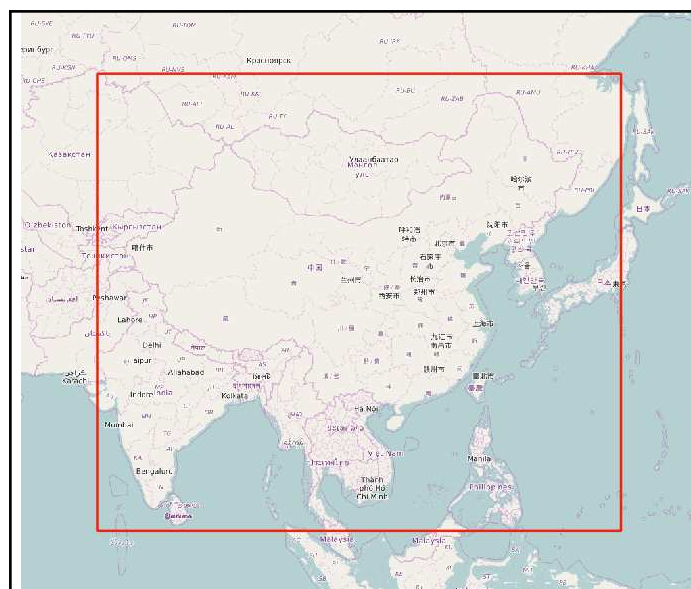


Figure 1.2 – Coverage extension of BeiDou-1 navigation system



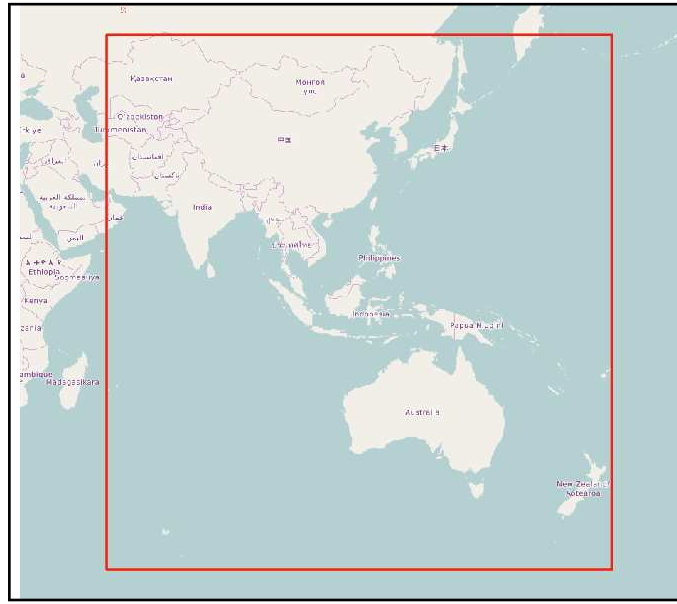


Figure 1.3 – Coverage extension of BeiDou-2 navigation system (2012)

### 1.3 The Absolute Positioning technique

The operating principle is similar to all GNSS systems mentioned in the previous section, however the most diffuse and widely studied is the GPS system (section 1.2.1): this system will be used as reference in the descriptions that follow, although the principles described are generally applicable to all GNSS system.

The absolute positioning technique [4] uses to estimate rover position only data from satellites: the pseudo random messages codes (PRN codes) and the navigational message D (section 1.2.1).

To calculate the distance rover - satellite, each receiver compares the broadcast PRN code with his internal copy<sup>16</sup>: the phase difference between the two copies of the same signal will be due to the time taken by the signal to make the satellite-receiver path.

In the example in figure 1.4, S is the PRN code sent from the satellite, R is the copy of the code generated into the receiver,  $\tau$  is the phase shift between the two signals, due to the time taken by the signal to make the satellite-receiver path.

Then, the distance  $\rho$  between the satellite and the receiver will be:

$$\rho = c\tau \quad (1.1)$$

where:

$c$  is the speed of light in vacuum;

$\tau$  is the time taken by the PRN signal to cover the satellite-receiver path.

The navigational message D contains the information about the satellite position (ephemerides predicted). The ephemerides are calculated by each satellite using the parameters received by the ground control centres (Control Segment, section 1.2). These centres continuously check the satellites position, compute and send to

<sup>16</sup>According with the kind of receiver in use, the PRN code can be the C/A code (civil receivers), the P code (military receivers) or the new civil codes recently implemented

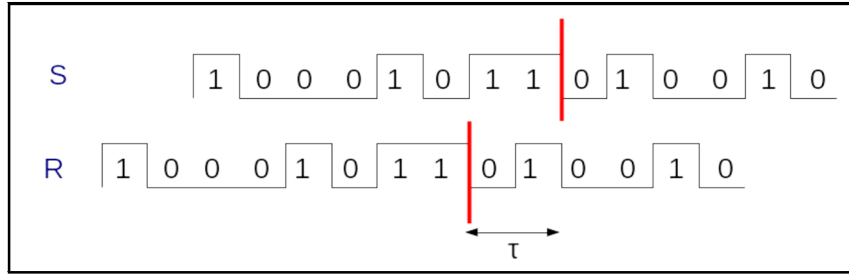


Figure 1.4 – Comparison between PRN receiver code and PRN satellite code

the satellites the ephemerides updated. The accuracy with which can be predicted the satellite position is about 1 m.

Also precise ephemeris are available<sup>17</sup>, calculated by International GNSS Service (IGS), to enhance the accuracy. With these ephemerides it is possible to reach 5 cm of accuracy, but obviously the positioning in real time is possible only with broadcast ephemerides.

With these two data, satellite position and distance receiver-satellite, it is possible place the receiver on the sphere of radius  $\rho$  and as centre the satellite position (figure 1.5).

Therefore if a GNSS receiver can receives signals (PRN and D) from 3 satellites (figure 1.6) it will become possible define a system of three equations in three unknowns, that should give as result the coordinates of receiver:

$$\begin{cases} \rho_R^{S1} = \sqrt{(X_R - X^{S1})^2 + (Y_R - Y^{S1})^2 + (Z_R - Z^{S1})^2} \\ \rho_R^{S2} = \sqrt{(X_R - X^{S2})^2 + (Y_R - Y^{S2})^2 + (Z_R - Z^{S2})^2} \\ \rho_R^{S3} = \sqrt{(X_R - X^{S3})^2 + (Y_R - Y^{S3})^2 + (Z_R - Z^{S3})^2} \end{cases} \quad (1.2)$$

where:

$X_R, Y_R, Z_R$  are the receiver coordinates, unknowns, to be determined;

$X^{Si}, Y^{Si}, Z^{Si}$  coordinates of satellite  $i$ , known;

$\rho_R^{Si}$  distance between receiver from satellite  $i$ , also this is known.

However the clock of satellite and the clock of receiver are both characterized by an offset with respect to the standard GPS time. This means that the propagation time of the PRN code observed by the receiver is not  $\tau$ , but  $\Delta T_R^S(t)$  and it includes also the effect of these two offsets:

$$\Delta T_R^S(t) = \tau_R^S + dt_R(t) - dt^S(t) \quad (1.3)$$

where:

$\tau_R^S$  is the actual time taken to cover the satellite-receiver path;

$dt_R(t)$  is the receiver clock offset;

$dt^S(t)$  is the satellite clock offset.

Then the observation equation can be written:

$$P_R^S(t) = c\Delta T_R^S(t) = c\tau_R^S + c(dt_R(t) - dt^S(t)) \quad (1.4)$$

<sup>17</sup><http://www.igs.org/products>



Figure 1.5 – The red line represents the sphere where can be a GNSS receiver distant  $\rho$  from the satellite in figure.

But the observation equation is affected also by others effects.

Error in the PRN code ( $\nu_P$ ) generated from the receiver due to the electronic noise. The error entity depending by the type of receiver and it can be from a few decimetres to a meter: this error can not be modelled and it can not be removed from the final observation.

In the equation 1.4 it was used the speed of light in vacuum, however the signal pass through the atmosphere which affects its path and its speed. In particular there are two atmospheric layers which interact with the GPS signal: the ionosphere and the troposphere.

The ionospheric effect,  $I_R^S$ , is due to the presence, between 60 km and 1000 km<sup>18</sup> of altitude, of a layer ionized by solar radiation that influences the radio signals propagation. This effect is not constant and the main variation is between day and night: the sun radiation that causes ionisation of neutral atoms and produces free electrons and ions during the day ends during the night, therefore the recombination process prevails and free electrons and ions are recombined in neutral particles.

Different models were defined to describe the ionospheric effect, one of the most used in GPS is the Klobuchar Ionospheric Model [27]. This model describes the ionospheric status as a function of position and time using eight parameters. Like to the ephemerides, the parameters are computed by the control centres on the ground, then are sent to the satellites. The satellites include these informa-

<sup>18</sup><http://www.hq.nasa.gov/iwgsdi/Ionosphere.html>

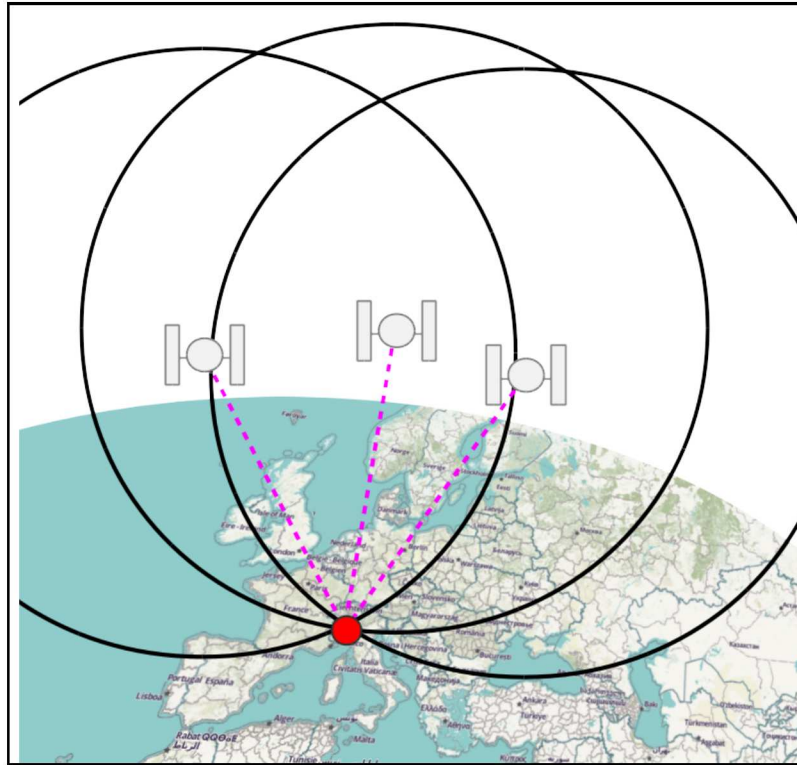


Figure 1.6 – Positioning a GNSS receiver

tion inside the navigational message (broadcast model) so that the receivers can modelled ionospheric effect and correct the error of positioning. The positioning accuracy, once the model is applied, is between 0 to 20 m.

The tropospheric effect,  $T_R^S$ , is present in the first 40 km altitude and leads to an extra delay in the measurement of the time spent by the signal to cover the satellite-receiver path. This delay depends on the temperature, pressure, humidity that change with time and with receiver location. The error is caused by the effect of refraction on the signal path and also in this case it is possible use models (for example Saastamoinen model [46]) to describe the the effect of these element on the signal. The correction process is managed by the control centres on the ground: they compute the tropospheric model and send it to the satellites, that include it in the navigational message that send to the receivers on Earth. The receiver use these information to correct the positioning error. The residual error, after model application, is about a 20 cm.

It is possible reduce the residual errors due to the atmospheric effects using the precise models that can be downloaded from the IGS website<sup>19</sup>. However in this case the data are post-processed, it is not real time positioning.

If the two effects due to atmosphere are added, the observation equation 1.4 will be:

$$P_R^S(t) = \rho_R^S(t) + c(dt_R(t) - dt^S(t)) + I_R^S(t) + T_R^S(t) \quad (1.5)$$

The final observation equation 1.5 contain 5 terms:

<sup>19</sup><http://www.igs.org/products#Atmospheric>





signal phase at the emission epoch and the satellite signal phase at the receiving epoch (figure 1.8);  $N$  cannot be observed.

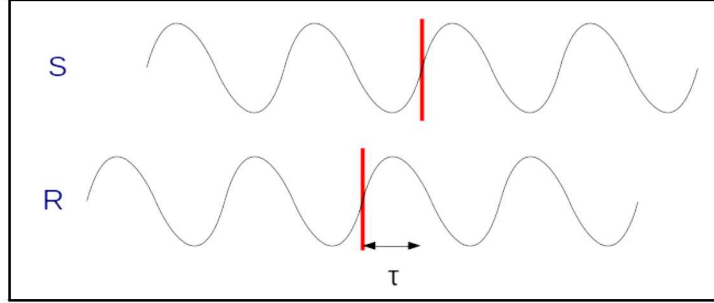


Figure 1.7 – comparison between receiver and satellite phases

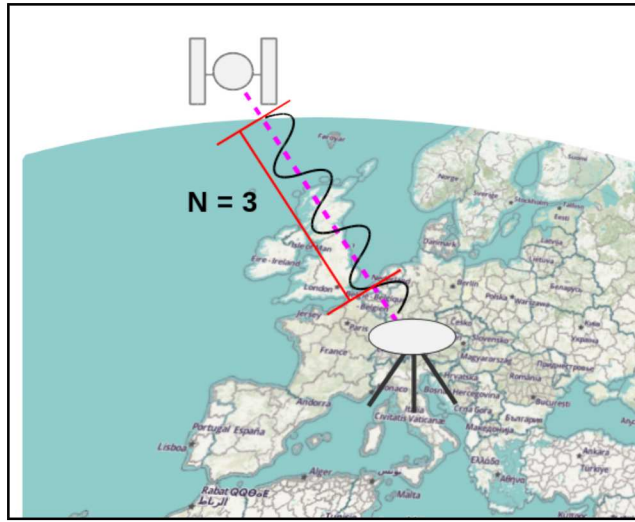


Figure 1.8 – integer ambiguity

The equation 1.7 can be multiplied by the wavelength of the carrier:

$$L_R^S(t) = \lambda \Phi_R^S(t) = c\tau_R^S + c(dt_R(t) - dt^S(t)) + \lambda(N_R^S(t) + \Phi_R - \Phi^S) \quad (1.8)$$

Ionospheric and tropospheric effects are present also into the phase observations and the equation can be written:

$$L_R^S(t) = \rho_R^S(t) + c(dt_R(t) - dt^S(t)) - I_R^S(t) + T_R^S(t) + \lambda(N_R^S(t) + \Phi_R - \Phi^S) \quad (1.9)$$

The integer ambiguity in phase observations is a multiple of 19 cm in L1 and 24 cm in L2, therefore remains into the observations, as well as the initial phases  $\Phi_R$  and  $\Phi^S$ , constants, but unknown.

No estimation of these terms is possible in single epoch: it is possible solve the phase ambiguities using observations from the same satellite during more epochs, but this require to apply complex compute algorithms.

The absolute positioning using phase is a research arguments, but for standard application the positioning using phases is made by relative positioning, illustrated in the next section, easier to be implemented than the phase absolute positioning.

The accuracy of positioning with code observations can be from tens of meters, in the case of single epoch in single frequency (low-cost receivers) to some centimetres for post-processed surveying did with double frequency receivers.

## 1.4 The Relative Positioning technique

The Relative Positioning technique [4] uses data from two or more GNSS receivers that are tracking the same satellites. This technique wants determine the vector, called baseline, between a reference site, whose position is known, and a rover in a unknown position. Therefore with the relative positioning the position of rover is determined relative to the reference and then final position computed from this. The relative positioning allows to improve the accuracy; this improvement depends on:

- kind of receivers and from observations that can be acquired;
- distance between the reference station and the rover;
- kind of survey done, time of unknown point occupation;
- kind of data processing (real-time, post-processing).

Accuracy can be from 1-2 meters with code observations in real time, to the centimetre (also sub-centimetre) with static survey with dual-frequency receivers.

### 1.4.1 Single differences

In single differences method (SD) two receivers,  $R_1$  e  $R_2$ , acquired data from one satellite  $S$  at the same epoch (figure 1.9): the single difference is the difference between the observations of the two receivers and can be considered itself a observation.

The SD code observation (equation 1.5) can be written as:

$$\begin{aligned}
 P_{R1,R2}^S(t) &= P_{R1}^S(t) - P_{R2}^S(t) \\
 &= \rho_{R1}^S(t) - \rho_{R2}^S(t) + c(dt_{R1}(t) - dt_{R2}(t)) - c(dt^S(t) - dt^S(t)) + \\
 &+ I_{R1}^S(t) - I_{R2}^S(t) + T_{R1}^S(t) - T_{R2}^S(t) = \\
 &= \rho_{R1}^S(t) - \rho_{R2}^S(t) + c(dt_{R1}(t) - dt_{R2}(t)) \\
 &+ I_{R1}^S(t) - I_{R2}^S(t) + T_{R1}^S(t) - T_{R2}^S(t)
 \end{aligned} \tag{1.10}$$

Regarding phase observations (equation 1.9) the new SD observation is:

$$\begin{aligned}
 L_{R1,R2}^S(t) &= L_{R1}^S(t) - L_{R2}^S(t) = \\
 &= \rho_{R1}^S(t) - \rho_{R2}^S(t) + c(dt_{R1}(t) - dt_{R2}(t)) - c(dt^S(t) - dt^S(t)) + \\
 &- I_{R1}^S(t) + I_{R2}^S(t) + T_{R1}^S(t) - T_{R2}^S(t) + \\
 &+ \lambda(N_{R1}^S(t) - N_{R2}^S(t) + \Phi_{R1} - \Phi_{R2} - \Phi^S + \Phi^S) = \\
 &= \rho_{R1}^S(t) - \rho_{R2}^S(t) + c(dt_{R1}(t) - dt_{R2}(t)) + \\
 &- I_{R1}^S(t) + I_{R2}^S(t) + T_{R1}^S(t) - T_{R2}^S(t) + \\
 &+ \lambda(N_{R1}^S(t) - N_{R2}^S(t) + \Phi_{R1} - \Phi_{R2})
 \end{aligned} \tag{1.11}$$

The single differences method erased from the observation equation the terms of:

- offset of satellite clock;
- satellite initial phase.

Furthermore the differentiation reduce the errors due to:

- ionospheric effect

$$I_{R1,R2} = I_{R2}^S(t) - I_{R1}^S(t)$$

- tropospheric effect

$$T_{R1,R2} = T_{R1}^S(t) - T_{R2}^S(t)$$

- satellite ephemerides

$$\rho_{R1,R2} = \rho_{R1}^S(t) - \rho_{R2}^S(t)$$

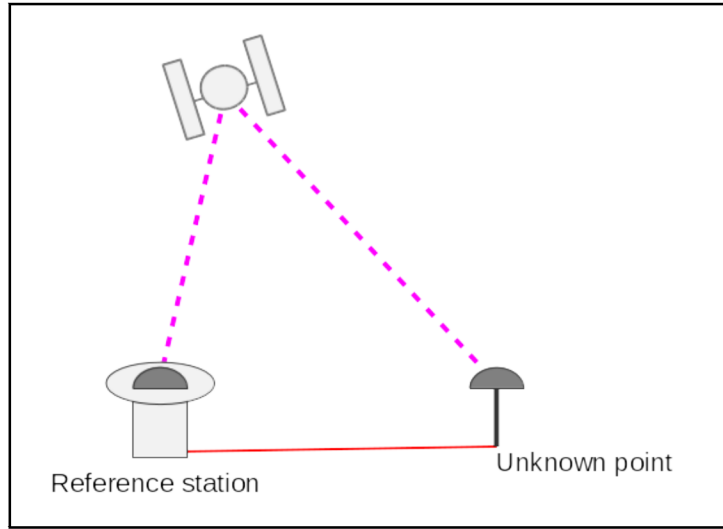


Figure 1.9 – Single Differences method

### 1.4.2 Double differences

The double difference method (DD) is the differences of two simultaneous single differences between two receivers,  $R_1$  e  $R_2$  and two satellites,  $S_1$  e  $S_2$  (Figure 1.10). Regarding code observations (equation 1.10), the double difference can be written:

$$\begin{aligned}
 P_{R1,R2}^{S1,S2}(t) &= P_{R1,R2}^{S1}(t) - P_{R1,R2}^{S2}(t) = \\
 &= \rho_{R1}^{S1}(t) - \rho_{R2}^{S1}(t) - \rho_{R1}^{S2}(t) + \rho_{R2}^{S2}(t) + \\
 &+ c(dt_{R1}(t) - dt_{R2}(t)) - c(dt_{R1}(t) - dt_{R2}(t)) + \\
 &+ I_{R1}^{S1}(t) - I_{R2}^{S1}(t) + T_{R1}^{S1}(t) - T_{R2}^{S1}(t) - (I_{R1}^{S2}(t) - I_{R2}^{S2}(t) + T_{R1}^{S2}(t) - T_{R2}^{S2}(t)) = \\
 &= \rho_{R1}^{S1}(t) - \rho_{R2}^{S1}(t) - \rho_{R1}^{S2}(t) + \rho_{R2}^{S2}(t) + \\
 &+ T_{R1}^{S1}(t) - T_{R2}^{S1}(t) - T_{R1}^{S2}(t) + T_{R2}^{S2}(t) + I_{R1}^{S2}(t) - I_{R2}^{S2}(t) - I_{R1}^{S2}(t) + I_{R2}^{S2}(t)
 \end{aligned}
 \tag{1.12}$$



If phase observations are considered (equation 1.11) the double difference equation will be:

$$\begin{aligned}
 L_{R1,R2}^{S1,S2}(t) &= L_{R1,R2}^{S1}(t) - L_{R1,R2}^{S2}(t) = \\
 &= \rho_{R1}^{S1}(t) - \rho_{R2}^{S1}(t) - \rho_{R1}^{S2}(t) + \rho_{R2}^{S2}(t) + \\
 &+ c(dt_{R1}(t) - dt_{R2}(t)) - c(dt^S(t) - dt^S(t)) + \\
 &- c(dt_{R1}(t) - dt_{R2}(t)) - c(dt^S(t) - dt^S(t)) + \\
 &+ \lambda(N_{R1}^{S1}(t) - N_{R2}^{S1}(t) - N_{R1}^{S2}(t) + N_{R2}^{S2}(t)) + \lambda(\Phi_{R1} - \Phi_{R2} - \Phi_{R1} + \Phi_{R2}) + \\
 &- I_{R1}^{S1}(t) + I_{R2}^{S1}(t) + T_{R1}^{S1}(t) - T_{R2}^{S1}(t) + I_{R1}^{S2}(t) - I_{R2}^{S2}(t) - T_{R1}^{S2}(t) + T_{R2}^{S2}(t) = \\
 &= \rho_{R1}^{S1}(t) - \rho_{R2}^{S1}(t) - \rho_{R1}^{S2}(t) + \rho_{R2}^{S2}(t) + \\
 &+ \lambda(N_{R1}^{S1}(t) - N_{R2}^{S1}(t) - N_{R1}^{S2}(t) + N_{R2}^{S2}(t)) + \\
 &+ T_{R1}^{S1}(t) - T_{R2}^{S1}(t) - T_{R1}^{S2}(t) + T_{R2}^{S2}(t) - I_{R1}^{S1}(t) + I_{R2}^{S1}(t) + I_{R1}^{S2}(t) - I_{R2}^{S2}(t)
 \end{aligned} \tag{1.13}$$

The double differences method erases from the observations equation also the offset of receiver clock and, more important, the initial phases  $\Phi_R$  and  $\Phi^S$ , leaving only the integer ambiguity term to which it is possible apply estimation and fixing algorithms. If these algorithms have successful you have a *Fixed Solution*: this means that the initial ambiguity was fixed and the integer number of initial cycles was found; in this case it will be possible reach an accuracy of centimetres order in positioning. Otherwise you have a *Float Solution*: in this case the number of initial cycles was not found, the solution is a float number; the accuracy it will remain on the meters order in case of kinematic or fast static survey. In case of static long positioning it is possible reach an accuracy at centimetre level also with float solutions.

Regarding the other terms that are still in the final DD equation, the tropospheric effect is greater the more the weather conditions between the two sites are different: it is reasonably suppose that if two sites are near (short baseline) the weather conditions will be more or less the same. The same effect concerns the ionospheric, although in this case the effect of attenuation is less evident. Overall the accuracy of positioning increases with decreasing baselines, with a relationship reported in table 1.4.

Length baseline	Survey conditions		
	Good	Standard	Poor
1 km	0.1 cm	0.2 cm	1.0 cm
10 km	0.5 cm	2.0 cm	10.0 cm
100 km	5.0 cm	20.0 cm	100.0 cm

Table 1.4 – Double Difference method: error in function of baseline length

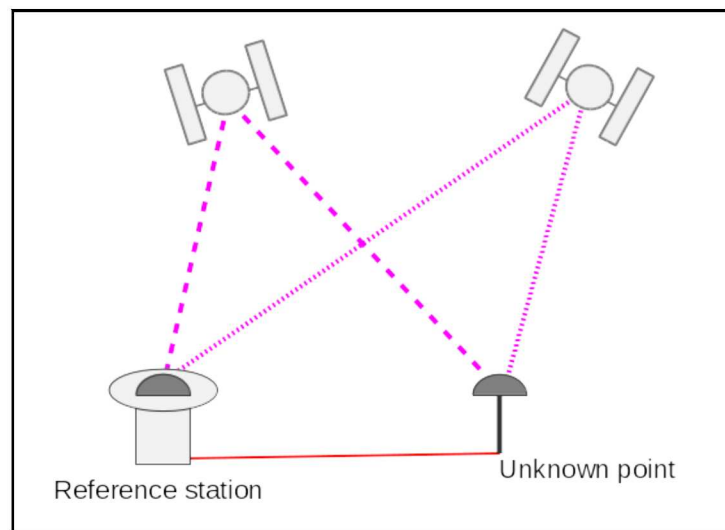


Figure 1.10 – Double Differences method

## 1.5 Multipath

The multipath is an interference that concerns both phase measurements and code measurements and is generated when a signal arrives, by different ways, at the antenna (see figure 1.11). The main cause is due to reflecting structures near the antenna and it is more important when the signal comes from the satellites with low elevation, because the difference between actual path and reflected path can be in this case more relevant. In case of the code measurements the error can reach 15-20 m, but typically is less than 2 or 3 meters. In case of the phase measurements, its theoretical maximum value can reach 5 centimetres, but it is typically less than 1 centimetre.

To minimise the effect of multipath error different actions are possible: choose a good position for the antenna, away from reflecting objects, but this can not be always possible; improve the antenna quality to mitigate the signal coming from certain low elevation directions.

It should be clear how in particular conditions, like kinematic survey made in urban environment the multipath interference can be relevant and highly variable. Also, if a mass market device with an integrated antenna, like a smartphone or a tablet, is used, a higher contribution of reflected signal can be present in final estimation positioning.

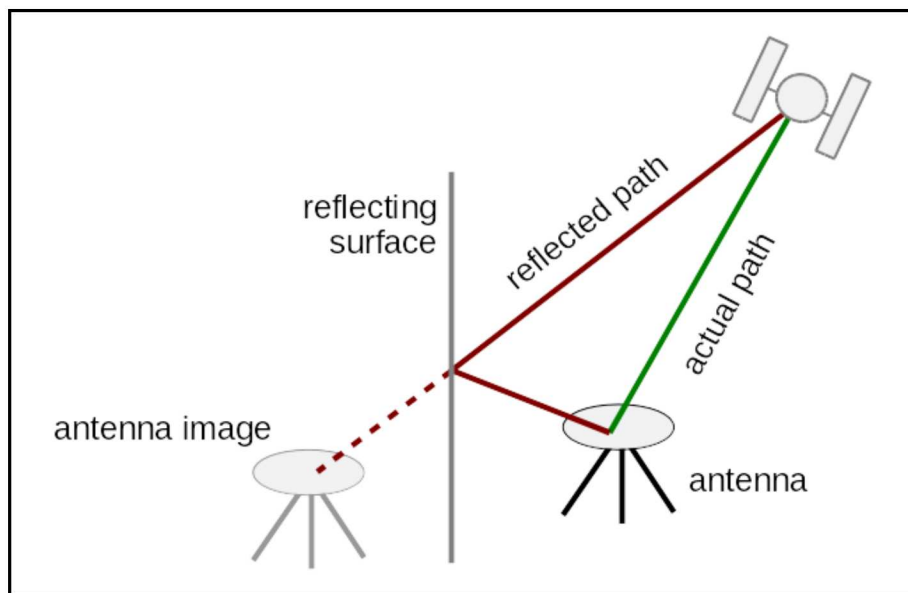


Figure 1.11 – Multipath: example of overestimated path

## 1.6 The NMEA format

NMEA is for National Marine Electronics Association<sup>20</sup> the organization that has developed the NMEA Standard [35], that defines the communication interface between its electronic equipment.

These specification are largely used also by GNSS devices to storage and exchange data and typically smartphones and tablets save GNSS data following this standard, they were used in this thesis and for that a brief explanation of this standard is given in the following.

The idea of NMEA is very simple: it is a text file where data are organised by lines called sentences. Every sentence is totally self contained and independent from other sentences and can be no longer than 80 characters (plus the line terminators). Comma character is used to separate the different data fields inside the same sentence.

A sentence start always with '\$' simbol, followed by two letter that defines to which device is referd the sentence, for example GP for GPS receivers and GN for GLONASS. Then there is a three letter sequence that defines the sentence contents. The last field in each sentece is the checksum, that consists of a '\*' and two hex digits representing an 8 bit exclusive OR of all characters between the '\$' and '\*'.

In figure 1.12 an example of output in NMEA format produced by the Samsung Galaxy Tab2 during the survey done the 16th October 2014. Each row start with the letter GP, so the data are related to GPS satellites, then in this file you have three different sentences GSA, GGA and RMC.

- GGA essential fix data which provide 3D location and accuracy data; let's take as example the first row in figure 1.12:

**112514.00** ⇒ Fix taken at 11:25:14 UTC

<sup>20</sup><http://www.nmea.org/>

**4548.134500,N** ⇒ Latitude 45 deg 48.1345' N

**00905.655351,E** ⇒ Longitude 9 deg 05.655351' E

**1** ⇒ Fix quality:

- 0 = invalid
- 1 = GPS fix (SPS)
- 2 = DGPS fix
- 3 = PPS fix
- 4 = Real Time Kinematic
- 5 = Float RTK
- 6 = estimated (dead reckoning)
- 7 = Manual input mode
- 8 = Simulation mode

**15** ⇒ Number of satellites being tracked

**1.2** ⇒ Horizontal dilution of position

**204.6,M** ⇒ Altitude, Meters, above mean sea level

**48.1,M** ⇒ Height of geoid (mean sea level) above WGS84 ellipsoid

**(empty)** ⇒ time in seconds since last DGPS update

**\*67** ⇒ the checksum data

- GSA data about satellites status, DOP and active satellites; let's take as example the second row in figure 1.12:

**A** ⇒ Type of selection of 2D or 3D fix (M = manual, A = Auto)

**3** ⇒ 3D fix - values include:

- 1 = no fix
- 2 = 2D fix
- 3 = 3D fix

**08,16,18,...** ⇒ PRNs of satellites used for fix (space for 12)

**1.9** ⇒ Position dilution of precision (PDOP)

**1.1** ⇒ Horizontal dilution of precision (HDOP)

**1.5** ⇒ Vertical dilution of precision (VDOP)

**\*37** ⇒ the checksum data

- RMC essential GNSS data (position, velocity, time); let's take as example the third row in figure 1.12:

**112514.00** ⇒ Fix taken at 11:25:14 UTC

**A** ⇒ Status: A=active, V=Void

**4548.134500,N** ⇒ Latitude 45 deg 48.1345' N

**00905.655351,E** ⇒ Longitude 9 deg 05.655351' E

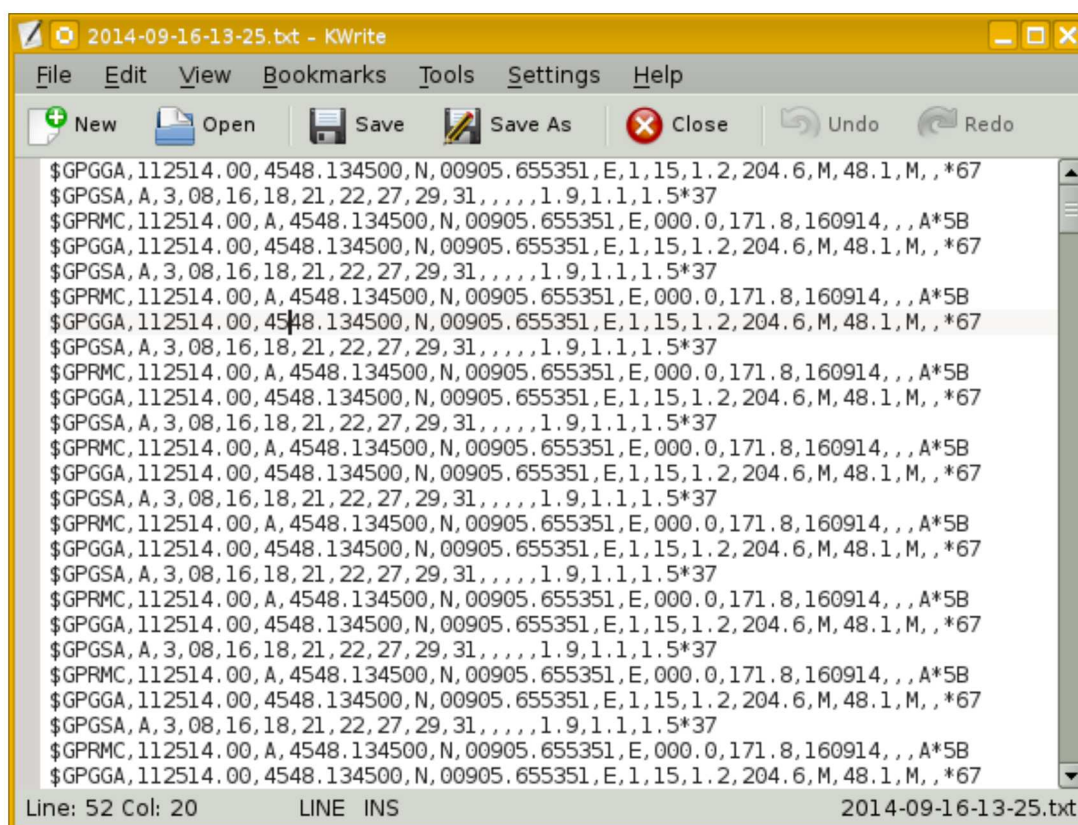
**000.0** ⇒ Speed over the ground in knots

**171.8** ⇒ Track angle in degrees True

**160914** ⇒ Date - 16th of September 2014

**(empty)** ⇒ Magnetic Variation

**\*5B** ⇒ The checksum data



```
$GPGGA,112514.00,4548.134500,N,00905.655351,E,1,15,1.2,204.6,M,48.1,M,,*67
$GPGSA,A,3,08,16,18,21,22,27,29,31,,,,,1.9,1.1,1.5*37
$GPRMC,112514.00,A,4548.134500,N,00905.655351,E,000.0,171.8,160914,,A*5B
$GPGGA,112514.00,4548.134500,N,00905.655351,E,1,15,1.2,204.6,M,48.1,M,,*67
$GPGSA,A,3,08,16,18,21,22,27,29,31,,,,,1.9,1.1,1.5*37
$GPRMC,112514.00,A,4548.134500,N,00905.655351,E,000.0,171.8,160914,,A*5B
$GPGGA,112514.00,4548.134500,N,00905.655351,E,1,15,1.2,204.6,M,48.1,M,,*67
$GPGSA,A,3,08,16,18,21,22,27,29,31,,,,,1.9,1.1,1.5*37
$GPRMC,112514.00,A,4548.134500,N,00905.655351,E,000.0,171.8,160914,,A*5B
$GPGGA,112514.00,4548.134500,N,00905.655351,E,1,15,1.2,204.6,M,48.1,M,,*67
$GPGSA,A,3,08,16,18,21,22,27,29,31,,,,,1.9,1.1,1.5*37
$GPRMC,112514.00,A,4548.134500,N,00905.655351,E,000.0,171.8,160914,,A*5B
$GPGGA,112514.00,4548.134500,N,00905.655351,E,1,15,1.2,204.6,M,48.1,M,,*67
$GPGSA,A,3,08,16,18,21,22,27,29,31,,,,,1.9,1.1,1.5*37
$GPRMC,112514.00,A,4548.134500,N,00905.655351,E,000.0,171.8,160914,,A*5B
$GPGGA,112514.00,4548.134500,N,00905.655351,E,1,15,1.2,204.6,M,48.1,M,,*67
$GPGSA,A,3,08,16,18,21,22,27,29,31,,,,,1.9,1.1,1.5*37
$GPRMC,112514.00,A,4548.134500,N,00905.655351,E,000.0,171.8,160914,,A*5B
$GPGGA,112514.00,4548.134500,N,00905.655351,E,1,15,1.2,204.6,M,48.1,M,,*67
$GPGSA,A,3,08,16,18,21,22,27,29,31,,,,,1.9,1.1,1.5*37
$GPRMC,112514.00,A,4548.134500,N,00905.655351,E,000.0,171.8,160914,,A*5B
$GPGGA,112514.00,4548.134500,N,00905.655351,E,1,15,1.2,204.6,M,48.1,M,,*67
```

Line: 52 Col: 20                      LINE INS                      2014-09-16-13-25.txt

Figure 1.12 – Example of NMEA produced during the survey done



## Chapter 2

# Navigation: improve accuracy using data from Reference Stations

### 2.1 Introduction

In navigation applications the GNSS single-frequency receivers are largely used in smart mobile devices: in car navigation system, smart watch, tablet and smartphone are integrated this kind of receivers. The technique used to get up the position is the absolute point positioning (1.3) that reach an accuracy of tens meters. In most of applications this accuracy is enough, but there are some particular cases where it could be better increase it, like in the MEP project, the case study used in this thesis, where the goal is reach an accuracy of meter order in point positioning.

Two different procedures were implemented and tested to correct data collected with low cost receivers and then improve final accuracy:

- correction using data from **reference stations** (described in this chapter);
- correction using local **cartography as reference** (described in chapter 3).

### 2.2 The MEP project

The MEP (Map for Easy Path) project<sup>1</sup> [8] won the Polisocial Award of Politecnico di Milano in 2014; it is financed by this award and it started in October 2014. The Polisocial Award<sup>2</sup> is the academic social responsibility programme supported by 5x1000 fund raising campaign of the Politecnico di Milano. It rewards Politecnico best research projects with social purposes, with the aim of support the development of projects with an high social impact and promote responsible, sustainable and equitable practices of development. The aim of MEP project is improve the urban mobility of the physically impaired people: mapping obstacles, identifying accessible paths and also, very important, give a feedback to identify problems and define methods to improve mobility, because for physically impaired

---

<sup>1</sup><http://mep5x1000.wixsite.com/mepapp>

<sup>2</sup><http://www.polisocial.polimi.it>

people get around the city can be very difficult: it can be very important to know the obstacles position before the journey start, in order to know the presence of obstacles and dangers and to plan the path to avoid contingencies.

MEP solution is based on smart mobile devices with an app to collect and deliver data: these data can be collected in a implicit way, like position data from GNSS satellites and in a explicit way, when the user maps a point of interest, gives a description, takes a photo and so on.

Three departments of Politecnico di Milano are involved:

- DEIB, the department of electronic, computer science and bioengineering, to manage the harvesting of explicit data;
- DICA, environmental and civil engineering department to manage GNSS data;
- Design department, to define how the end users can interact with this project.

In figure 2.1 it is shown the organization schema of the project activities.

The mapping obstacles activities is expected to be carried out by volunteered mapper (Volunteered Geographic Information - VGI [20]), with common mass market devices, widespread, but also with some problems of lack of accuracy (single frequency receiver). As example in figure 2.2 it is shown a case where there is an obstacle at the left side of the street (green diamond), but with mass market devices we are not able to identify it correctly (the red diamond is the wrong position detected). For that one of the activity of the project was to find a way to improve the accuracy of positioning of mass market devices: the aim is to warrant the identification the right side of the road, if it possible, otherwise at least the road. In the MEP project were used many techniques to improve positioning accuracy, using all the all the information available during a mapping activity with smart devices, not only GNSS data, like direction from compass, but the focus of this thesis is on the use of GNSS data and only methods to improve position using these data will be treated.

In this chapter data from a CORS station will be used to try to improve accuracy, instead in next chapter it will be used cartography as reference to correct data collected with the low cost receiver.





Figure 2.1 – Organization schema of MEP project



Figure 2.2 – An example of erroneous positioning: the green diamond is the actual position of the obstacle, but the position detected is not at the right side of the street (red diamond)

## 2.3 Data available

To increase the accuracy of positioning of smart mobile devices we are generally limited by the data that are available from the device:

- only coordinates and the list of satellites used to estimate them (from tablets and smartphones based on Android operative system<sup>3</sup>);
- only coordinates (from devices based on IOS operative system<sup>4</sup>).

In this chapter it will be discussed how to correct data from low cost receivers using data from Continuously Operating Reference Stations (CORS), in particular data from COMO reference station<sup>5</sup> will be experimented.

Two different approaches are possible to use data from CORS to improve accuracy in position estimate:

- geodetic approach to compute rover trajectory, in this case corrections are applied in the space of observations;
- correct rover trajectory with CORS corrections, in this case corrections are applied in the space of positions.

The geodetic approach used the double differences method (section 1.4) between a CORS and a rover to estimate rover position. This method needs the raw data of rover so at the moment, if smartphones and tablets are used as rover, it will be not possible to use it.

The second method requires the list of satellites that the receiver used to estimate the position with the absolute positioning technique. The list of satellites, as said before, is generally available as output data in the most of mass market devices so it was decided to study this method, considering its most wide applicability.

## 2.4 Correct rover trajectory with CORS corrections

Let's consider the CORS official position and the CORS positions estimated epoch by epoch. The first one is the actual position when the estimated positions are affected by environmental effects presents at the epoch in which the estimation was done. So the difference between actual position and position at epoch  $i$  can be supposed due to particular conditions present at this epoch that affects the estimation process. In the examples in figure 2.3 the green triangle is the actual position of CORS, the yellow circles are its positions estimated epoch by epoch. If the rover is near to the CORS these temporary effects will be more or less the same that afflict estimations done with the GNSS low cost receiver. In figure 2.3 the orange circles are the estimated positions epoch by epoch of the rover, when the actual rover position (unknown) is the orange triangle.

It is possible to compute the difference between actual and estimated CORS position and assume that this difference is caused by the same temporary effects that afflict the estimation of rover position. Therefore it is possible to try to

---

<sup>3</sup>The raw data (pseudoranges, dopplers and carrier phase) should be available in the next version of Android, Android N: <http://gpsworld.com/google-opens-up-gnss-pseudoranges/>

<sup>4</sup>verified with iPhone 5

<sup>5</sup>COMO CORS is part of EUREF Permanent Network since 2004, under the name COMO00ITA - <http://epncb.oma.be/>

improve the accuracy of rover positioning applying to the rover trajectory the corrections from CORS data, computed epoch by epoch.

In this correction procedure you have to take account also of the different satellite visibility between CORS and rover. The reference station typically observes all visible satellites at a certain epoch and the position is estimated using all of them. Rover usually is in a worse position than the CORS, so it can view a subset of satellites used by CORS to determine its position (as it is illustrated in figure 2.4). When the correction are computed the CORS position should be determined using the same set of satellites viewed by the rover.

For this reason a software (*ExtractRinex*) has been developed to remove satellites not observed by rover from CORS Rinex. The software has been written in C language, in Linux<sup>6</sup> environment, using GCC<sup>7</sup> as compiler. Anyway, only standard C instructions were used, so it should be compile and therefore use the software in whatever operating system that has a C standard compiler.

In figure 2.5 is shown the flow chart followed: first of all the list of satellites used to estimate the rover positions are read from the rover NMEA file<sup>8</sup>. Then *ExtractRinex* reads the list of satellites visible from CORS at the same epoch, compares the two lists and it removes from CORS Rinex the data of satellites that were no visible from rover.

The position, epoch by epoch, of the reference station is estimated using this modified Rinex, the difference between estimated position and official position is computed and then applied as correction to the rover position. To estimate epoch by epoch CORS position we used two different software: goGPS<sup>9</sup> and RTKLIB<sup>10</sup>. It has been decided to do two different kind of test:

- in ideal conditions with a static occupation in a two fixed points, to verify the principles of correction method: see section 2.5;
- in normal working conditions, with data collect during several repetition of two test paths in a urban environment (Path 1 - Figure 2.14 and Path 2 - Figure 2.15): see section 2.6.

---

<sup>6</sup>Gentoo Linux - <https://www.gentoo.org/>

<sup>7</sup>GCC, the GNU Compiler Collection, version 4.9.3 - <https://gcc.gnu.org/>

<sup>8</sup>Typically the data from smarthphones and tablets are delivered using this format for other information about this standard see section 1.6

<sup>9</sup>Appendix A.3

<sup>10</sup>Appendix A.4

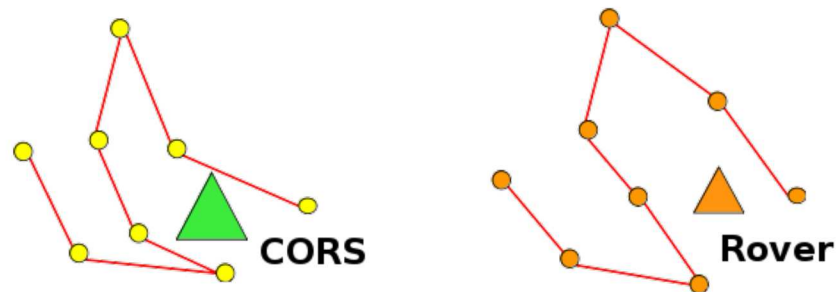


Figure 2.3 – The green triangle is the actual position of CORS, the yellow circles are its estimated positions epoch by epoch. It can be assumed that a similar trend affect the rover: the orange triangle is the unknown actual position, than the orange circles are its estimated positions epoch by epoch

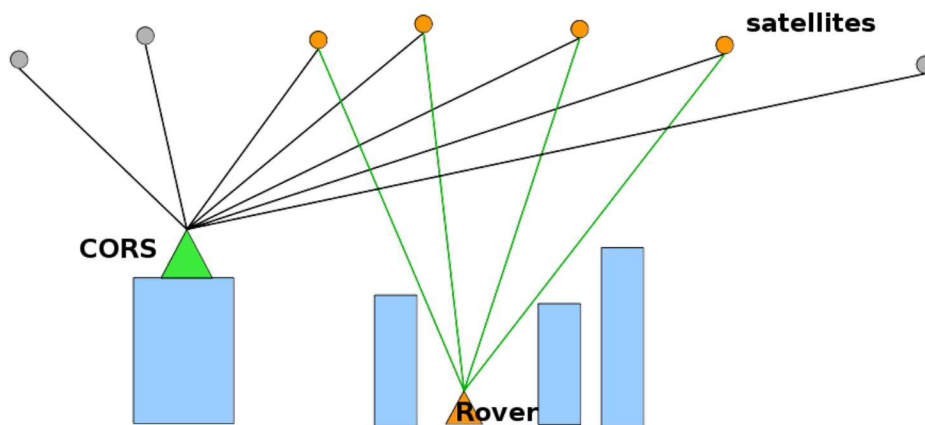


Figure 2.4 – Example of different visibility of the satellites between CORS and rover: the circles are the satellites viewed by CORS, the orange circles are the subset of satellites viewed by the rover

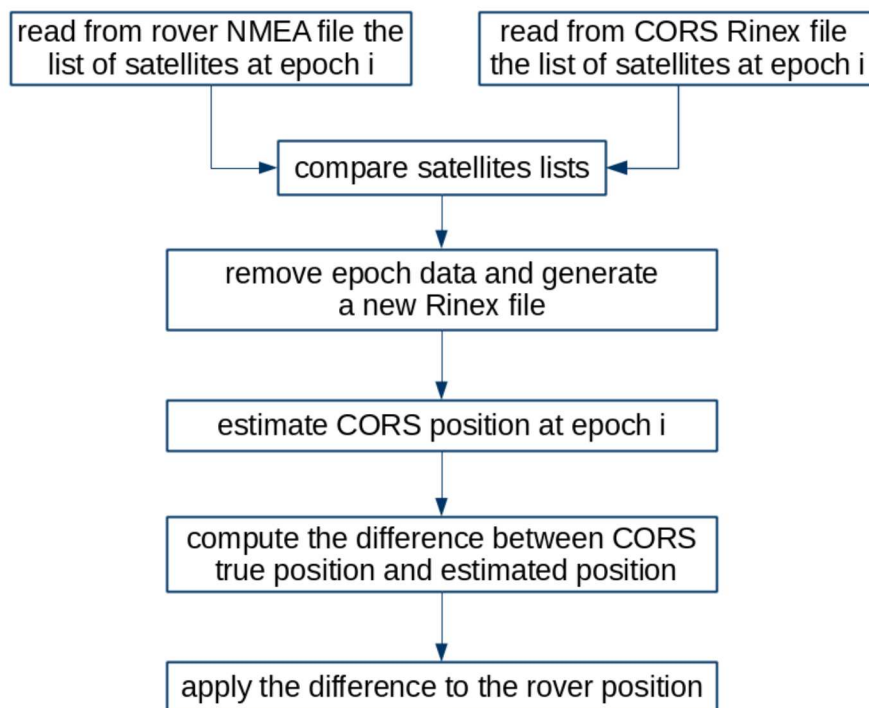


Figure 2.5 – Flow chart followed to implement *ExtractRinex* software

## 2.5 Data from static occupation

The test can be split in two different steps, data acquisition and data processing.

- Data acquisition:
  - test site selection: this should be near a CORS station and the actual position should be note, or in any case should be determined with a good accuracy, for example using a geodetic receiver;
  - test site survey: survey with a low cost receiver, during a a significant range of time.
- Data processing:
  - creation of the CORS Rinex with only the contribution of the satellites observed by the rover (application of the algorithm showed in figure 2.5);
  - estimation of corrections than must be applied to correct the low cost receiver data;
  - comparison between the data corrected and the actual point position.

Only planimetrics data (East and North coordinate) will be considered during the experiments: elevations data (UP) are not significant for our scope of obstacles identification.

Two test site have been used: one in Milan (figure 2.6), near the CORS station named MILA, on the roof of a Politecnico building<sup>11</sup> and another one in Como (figure 2.7), near COMO CORS station already cited before (section 2.3). Data and actual position of Milan site were already available<sup>12</sup>: one week from 14-073 to 14-079<sup>13</sup>.

About Como test site a geodetic receiver (Leica GX 1230<sup>14</sup>) was used to estimate its actual coordinates (figure 2.8) and then data were collected during a week, from 15-128 to 15-135<sup>15</sup> using a single frequency receiver. A u-blox NEO-7P<sup>16</sup> was used as rover [49]: it was chosen this device instead than a common smart mobile devices because it was necessary collect data for a long period, outdoor, in condition that could damage smartphones or tablets. In any case the u-blox chip-set is largely used as GNSS chip in many mass market devices so it is proper to use it for this kind of test. No improvement was made to warrant a better isolation of antenna to be nearest as possible to usual work condition of smart devices.

---

<sup>11</sup>MILA CORS is part of "Servizio di Posizionamento Interregionale GNSS di Regione Piemonte e Regione Lombardia", a local positioning service, available in Lombardy and Piedmont areas: <http://www.spingnss.it/spiderweb/frmIndex.aspx>

<sup>12</sup>Thanks to Dr. Mirko Reguzzoni that provided these data

<sup>13</sup>from March 14 to March 20, 2014

<sup>14</sup>See appendix B.1

<sup>15</sup>from May 8 to May 15, 2015

<sup>16</sup>See appendix B.2





Figure 2.6 – The positions of MILA CORS (Red triangle) and the rover (yellow circle); the distance of the two points is less than 80 meters.



Figure 2.7 – The positions of COMO CORS (Green triangle) and the rover (yellow circle); the distance of the two points is less than 150 meters.

To acquire data continuously for at least a week it was prepared a particular hardware configuration to warranty that the devices used were continuously powered (figures 2.9 and 2.10) and safer against problems due to bad weather conditions (figure 2.11).

Positioning data from Como and Milan test sites were compared before and after the application of CORS corrections using the statistic software R<sup>17</sup>: a script it was built to compare data and give as output the errors statistic.

In figure 2.12 is reported, as example, the effect of position corrections relative to the North coordinate in the day 130-15. It is possible to note how the red line, that is the CORS position estimation, goes up and down the actual north value, following a random pattern as it was aspect. The blue line is the oscillation of the position estimation of the rover: the oscillations are greater than in the CORS data, so in this case, as expected, the effect of noise is more relevant. In any case it is possible see how the rover data pattern (blue line) follow that described from the CORS data pattern (red line). Applying the correction to the rover estimated positions the common pattern disappears and you obtain as result the green line. In the appendix C are shown the graphs regarding all the other days of survey and it is possible to see the presence of a common pattern in all the cases.

The CORS positions estimated with goGPS and rtkLIB were similar: in the following tables are reported the statistics of corrections estimated with goGPS, but it possible made the same considerations using results estimated with rtkLIB.

In table 2.1 and 2.2 are reported all the results obtained applying the position correction for all the two sites considered: statistics of the errors before any kind of correction, after the correction using the CORS differences and after the correction using the CORS difference processed using only the subset of satellites viewed by rover. Therefore we computed two times the corrections, one time using the original Rinex file from CORS (column "rover corrected") and another time using the Rinex processed with ExtractRinex program. In both the cases, mean and standard deviation, are generally better using correction from CORS; instead the difference between correction done considering all the satellites visible from reference station and considering only the subset of satellites viewed by rover is not so important. This is mainly due to the good position of the rover during the test, so the number of satellites viewed by the rover was not so different of than viewed by the CORS.

---

<sup>17</sup>R is a free software environment for statistical computing and graphics, it is licensed under GPL-2 — GPL-3. Web site: <https://www.r-project.org/>





Figure 2.8 – The setup of test point in Como



Figure 2.9 – The rack used to power the devices using during the test

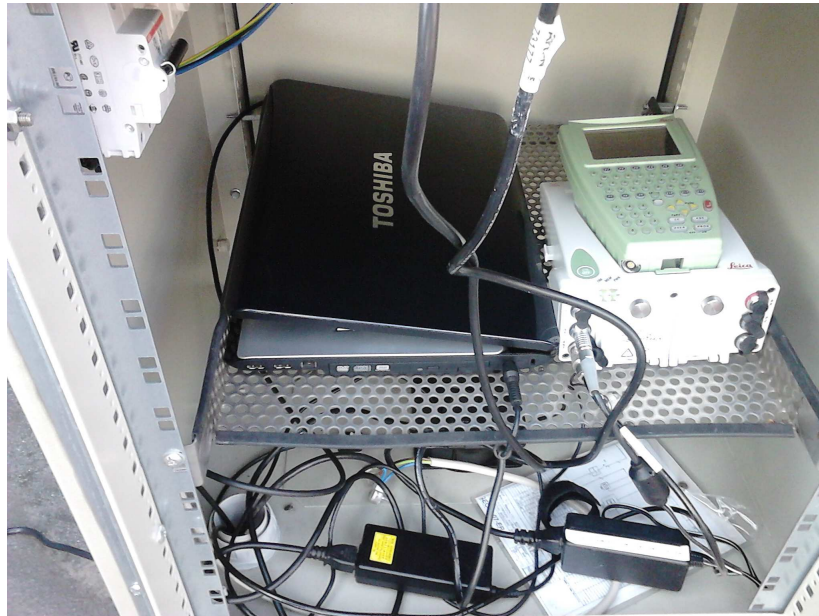


Figure 2.10 – A detail, inside the rack: a PC was used to power u-blox and collect data from it

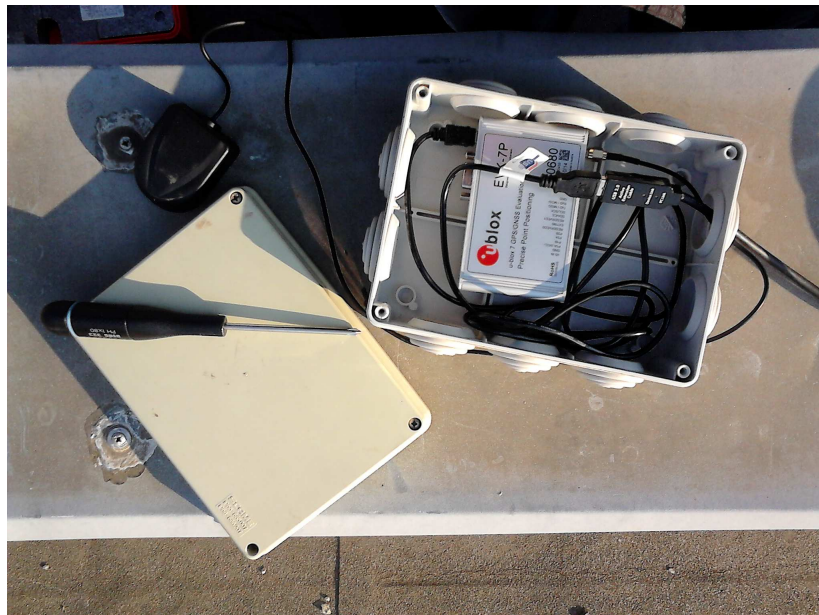


Figure 2.11 – The waterproof box used to protect the u-blox during the survey



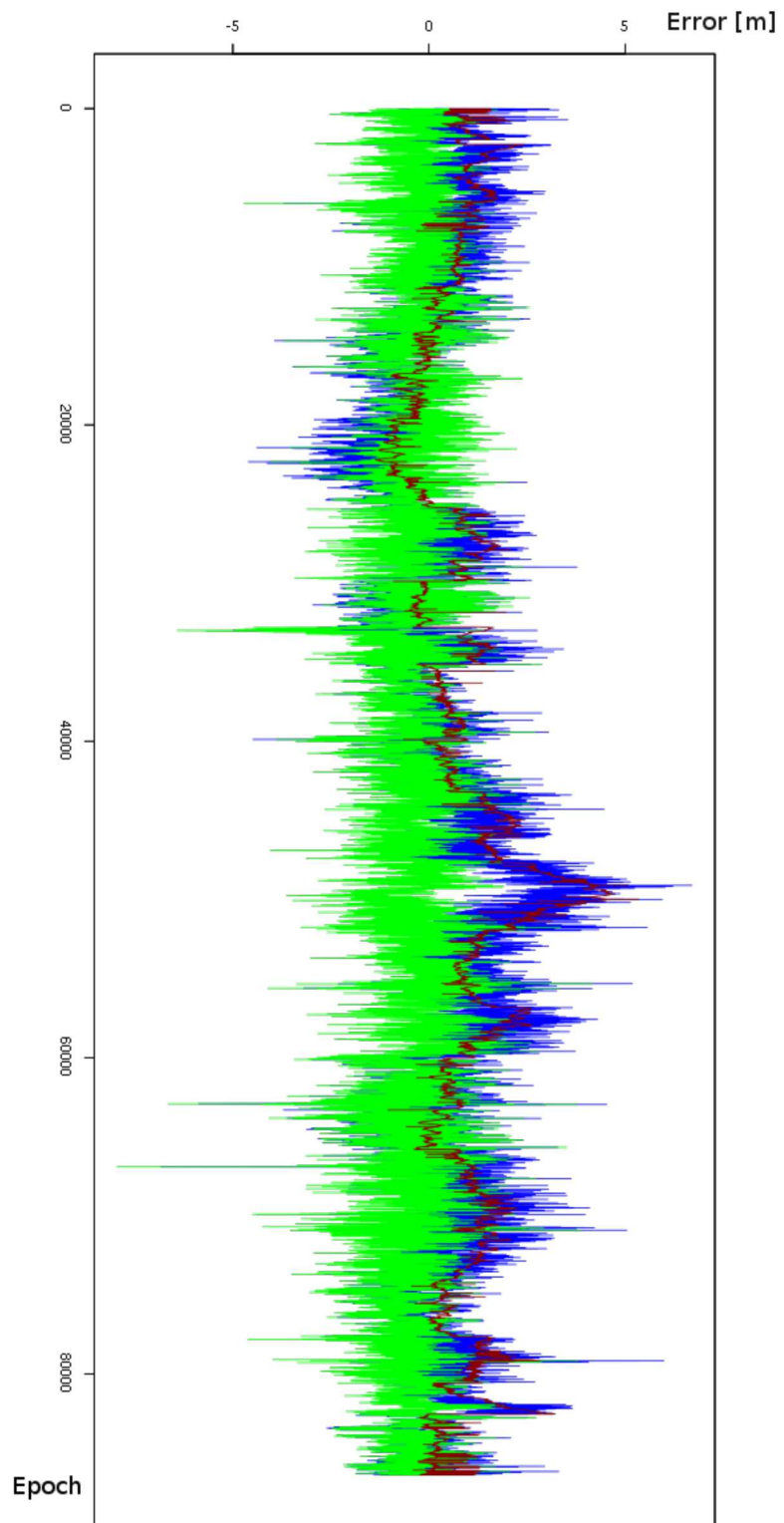


Figure 2.12 – Statistical results for the day 15-130 for the north coordinate: the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections

Como		rover		rover corrected		rover corrected with modified Rinex	
day	[m]	East	North	East	North	East	North
15-128	Mean	-0.1	0.8	0.0	-0.3	0.0	-0.3
	Snd Dev	0.8	1.2	0.6	0.8	0.6	0.8
	Max	4.4	5.7	4.7	4.4	4.7	4.4
	Min	-4.2	-5.3	-4.3	-6.8	-4.3	-6.8
	Median	-0.1	0.7	0.0	-0.3	0.0	-0.3
15-129	Mean	0.3	0.6	0.0	-0.3	0.0	-0.3
	Snd Dev	0.8	1.4	0.6	0.8	0.6	0.8
	Max	4.4	6.2	4.1	4.6	4.1	4.0
	Min	-4.4	-5.0	-4.0	-9.1	-3.7	-9.1
	Median	0.3	1.0	0.0	-0.3	0.0	-0.3
15-130	Mean	0.2	0.4	0.1	-0.4	0.1	-0.4
	Snd Dev	0.8	1.2	0.7	0.8	0.6	0.8
	Max	5.1	6.7	4.9	4.2	4.9	4.2
	Min	-4.8	-6.8	-4.3	-7.9	-4.3	-7.9
	Median	0.2	0.3	0.1	-0.4	0.1	-0.4
15-131	Mean	0.2	0.7	0.1	-0.5	0.1	-0.5
	Snd Dev	0.9	1.1	0.7	0.8	0.7	0.8
	Max	5.3	6.2	4.8	5.5	4.8	5.5
	Min	-5.8	-5.8	-6.5	-5.6	-5.0	-5.6
	Median	0.2	0.7	0.1	-0.5	0.1	-0.5
15-132	Mean	-0.3	0.5	0.1	-0.5	0.1	-0.5
	Snd Dev	0.9	1.1	0.7	0.8	0.6	0.8
	Max	5.0	7.5	4.8	6.4	4.8	6.4
	Min	-11.1	-7.8	-10.2	-8.6	-10.2	-8.6
	Median	-0.3	0.5	0.1	-0.4	0.1	-0.4
15-133	Mean	-0.2	1.1	0.0	-0.4	0.0	-0.4
	Snd Dev	0.9	1.2	0.7	0.9	0.7	0.9
	Max	4.3	7.2	4.6	4.7	4.6	4.7
	Min	-6.1	-17.9	-5.8	-18.9	-5.8	-18.9
	Median	-0.2	1.1	0.1	-0.4	0.0	-0.4
15-134	Mean	0.1	0.2	0.1	-0.5	0.1	-0.5
	Snd Dev	0.9	1.2	0.6	0.8	0.6	0.8
	Max	6.5	8.4	5.4	7.4	5.4	7.4
	Min	-4.1	-5.8	-3.7	-6.6	-3.7	-6.6
	Median	0.0	0.1	0.1	-0.5	0.1	-0.5
15-135	Mean	0.4	0.5	0.1	-0.3	0.1	-0.3
	Snd Dev	0.8	1.3	0.6	0.8	0.6	0.8
	Max	4.3	5.8	4.4	4.4	4.4	4.4
	Min	-4.6	-5.8	-4.7	-5.4	-4.7	-5.3
	Median	0.4	0.4	0.1	-0.3	0.1	-0.3

Table 2.1 – Como test point: statistical errors

Milan		rover		rover corrected		rover corrected with modified Rinex	
day	[m]	East	North	East	North	East	North
14-073	Mean	-0.6	1.2	-0.7	0.2	-0.7	0.2
	Snd Dev	1.4	2.1	1.4	2.0	1.4	1.9
	Max	8.0	13.0	8.3	12.1	8.2	12.1
	Min	-9.8	-13.6	-9.7	-15.6	-9.7	-15.6
	Median	-0.7	1.2	-0.7	0.2	-0.7	0.2
14-074	Mean	-0.4	1.3	-0.7	0.2	-0.7	0.2
	Snd Dev	1.5	2.1	1.4	2.0	1.4	2.0
	Max	9.4	16.1	9.1	15.8	9.3	13.3
	Min	-7.3	-11.0	-7.8	-13.5	-7.8	-13.5
	Median	-0.3	1.4	-0.7	0.2	-0.7	0.2
14-075	Mean	-0.6	1.1	-0.7	0.2	-0.7	0.2
	Snd Dev	1.5	2.1	1.4	2.0	1.4	2.0
	Max	9.0	48.1	9.2	47.5	9.1	47.5
	Min	-18.6	-12.1	-18.1	-12.3	-18.1	-12.3
	Median	-0.6	1.1	-0.7	0.2	-0.7	0.3
14-076	Mean	-0.7	1.3	-0.7	0.2	-0.7	0.2
	Snd Dev	1.5	2.3	1.4	2.0	1.4	2.0
	Max	8.9	15.0	8.7	15.0	8.4	14.9
	Min	-8.7	-9.2	-8.4	-10.7	-8.4	-10.7
	Median	-0.7	1.3	-0.7	0.3	-0.7	0.3
14-077	Mean	-0.9	1.7	-0.7	0.2	-0.7	0.2
	Snd Dev	1.4	2.4	1.4	2.0	1.4	2.0
	Max	7.6	30.2	7.7	26.7	7.7	26.7
	Min	-8.4	-9.4	-8.0	-11.2	-8.0	-11.1
	Median	-0.9	1.6	-0.7	0.3	-0.7	0.3
14-078	Mean	-0.7	1.4	-0.7	0.2	-0.7	0.2
	Snd Dev	1.5	2.6	1.4	2.1	1.4	2.0
	Max	6.8	16.4	6.6	15.2	6.6	14.9
	Min	-7.5	-9.8	-7.4	-12.9	-7.4	-12.9
	Median	-0.7	1.4	-0.7	0.3	-0.7	0.3
14-079	Mean	-0.6	1.8	-0.7	0.2	-0.7	0.3
	Snd Dev	1.6	2.7	1.5	2.1	1.5	2.1
	Max	8.9	29.4	8.0	27.9	8.0	27.9
	Min	-6.7	-9.1	-9.4	-10.8	-9.4	-10.7
	Median	-0.7	1.7	-0.7	0.3	-0.7	0.3

Table 2.2 – Milan test point: statistical errors

## 2.6 Data along the test path

### 2.6.1 Survey setup

The mapping activity in MEP project is mainly relevant to a urban context and it will be done mainly by volunteers using tablet and smartphone. But the mapping activities done with these tools, in such environment, can reach a very low accuracy, even worse then ten meters, above all in bad visibility condition. Therefore it is important to define a process that permits to improve the final accuracy of survey taking into account these specific conditions of use.

The first step was define the experiment conditions and for this two test paths have been defined in Como town (figures 2.14 and 2.15), in order to simulate the possible different conditions of satellite visibility that can occur in a urban context:

- urban canyons (figure 2.16), in the old town, red area in the figure 2.14;
- areas of fair visibility (figure 2.17), outside the old town, yellow area in the figure 2.14 and slightly part of path in figure 2.15) covered by trees;
- open areas (figure 2.18), mainly near the lake, green area in the figure 2.14 and for most of the path in figure 2.15.

The test paths have been repeated several times and the data were collected using a geodetic receiver<sup>18</sup>, some smart mobile device, tablets and smartphones<sup>19</sup>, and one u-blox<sup>20</sup>; in figures 2.19 and 2.20 it is showed the survey setup. Data from geodetic receiver were needed to verify epoch by epoch the positioning accuracy of the low cost receivers: the positioning obtained with geodetic receiver has been used as reference, the actual position, respect to which evaluate the the low cost receiver goodness. The difference between geodetic and low cost receiver coordinates is the positioning error (figure 2.21). The surveys were done in kinematic conditions, in an urban context, with obstacles and interference which can deteriorated the quality of satellite signal, therefore the estimated points accuracy for each epoch can be different, depending to the quality of collected signal from the geodetic receiver.

To be reliable as reference, the data acquired with the geodetic receiver have been post-processed<sup>21</sup> with the relative positioning technique (section 1.4), using as reference station COMO CORS, the nearest reference station, with a baseline less than 5 km. The process parameters used are reported in figure 2.13, extracted from the report produced by LGO. With the double difference method (section 1.4.2), according to the table 1.4, in the case studied, it will be possible to reach an error of few centimetres, but only if it is possible to obtain a fixed phase solution. For these points a good accuracy is reached and it reasonable to use its as actual position to evaluate the data from low cost receiver, while the other points should be rejected because the estimation errors are too high to be considered a reference.

---

<sup>18</sup>Leica GX 1230, see appendix B.1

<sup>19</sup>Samsung Galaxy Tab2, Samsung Galaxy Tab4 and Samsung Galaxy S5

<sup>20</sup>u-blox EVK-7P evaluation kit, see appendix B.2

<sup>21</sup>data were precessed using LGO, see appendix A.2)

<b>Processing Parameters</b>	
<b>Parameters</b>	<b>Selected</b>
Cut-off angle:	10°
Ephemeris type:	Broadcast
Solution type:	Automatic
GNSS type:	Automatic
Frequency:	Automatic
Fix ambiguities up to:	80 km
Min. duration for float solution (static):	5' 00"
Sampling rate:	Use all
Tropospheric model:	Saastamoinen
Ionospheric model:	Klobuchar
Use stochastic modelling:	Yes
Min. distance:	8 km
Ionospheric activity:	Automatic

Figure 2.13 – Parameters used in LGO software to process data from geodetic receiver (extract of LGO report)



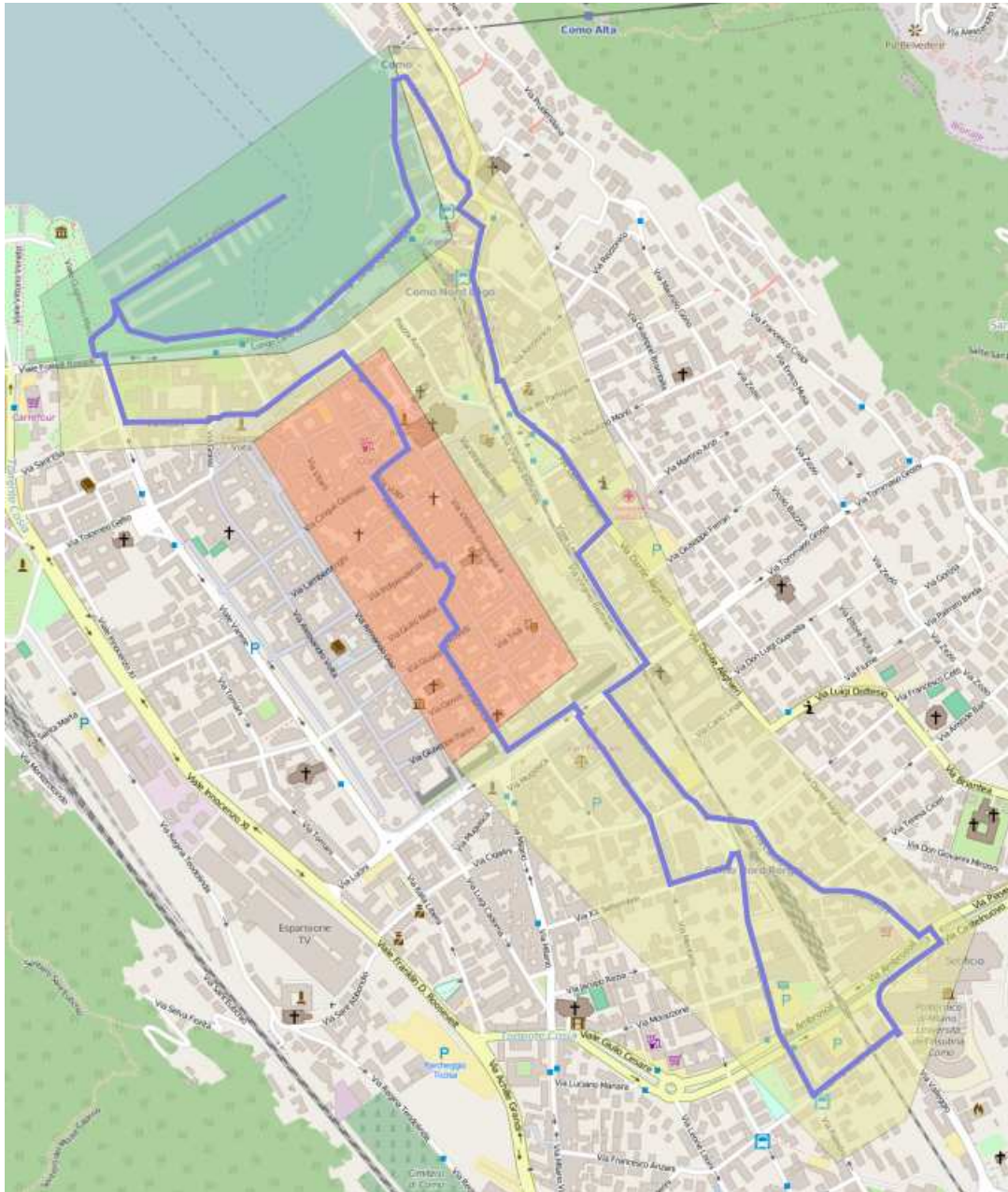


Figure 2.14 – Path 1: urban canyon (red), area of fair visibility (yellow), open areas (green)



Figure 2.15 – Path 2



Figure 2.16 – Example of urban canyon: Luini Street, in the old town





Figure 2.17 – Example of area with fair visibility: XX settembre street



Figure 2.18 – Example of open area: breakwater Caldirola



Figure 2.19 – The pack setup with the instruments used for the tests



Figure 2.20 – The position of geodetic antenna (red circle) and u-blox antenna (red circle) during the test; smartphones was taken by hand



Figure 2.21 – Yellow triangles are the points estimated with the geodetic receiver, red circles are the positions collected with the low cost receiver during the same survey. The different of positioning between points at the same epoch is the error (the orange dotted line)

## 2.6.2 Path 1: survey with Smart Mobile Devices and Geodetic receivers

Three different surveys were made along the Path 1 (figure 2.14) on the days 14-259, 14-267 and 14-268<sup>22</sup>, using a tablet, Samsung Galaxy Tab2, and a geodetic receiver, Leica GX 1230<sup>23</sup>, as reference as explained before.

In table 2.3 are showed the results, related of these three survey, of the double differences processing applied to the raw data of the geodetic receiver: there is a fixed solution for less than half of the points: kinematic float solution are not relate and it was possible evaluate the positioning accuracy of the tablet only for these points. Therefore the results from geodetic receiver and tablet were compared epoch by epoch where reference data were available: in table 2.4 it is possible to see the statistic errors: the standard deviation is between 2.5 m and 5.5 m as it was expected. Along the path, mostly in the area with bad satellites visibility, it is possible that the error has been greater, but in these areas reference solution were not available to make comparison, because the solution obtained from geodetic receiver data was not fixed in phase.

The next step it was compute the corrections to be applied: the same method, already saw to correct data from static occupation in the section 2.5, was applied. For each epoch the CORS Como coordinates were computed using the modified Rinex with only the contribution of satellites viewed by the rover. The difference between estimated coordinates epoch by epoch and actual CORS coordinates were computed and used to correct rover position. In the two last columns of table 2.4 it is possible to see the statistics relative the errors after the application of the corrections. Unlike what it was obtained with the data from static positioning, where correction generally improved the accuracy (tables 2.1 and 2.2), in this case the results after the correction sometimes are better and sometimes are worse than the results without corrections. This is probably the effect of local effects, manly multipath (see section 1.5), due to particular properties of the area where the survey was made, that obviously can not be removed using data from reference station, where they are not present.

For example, in figure 2.22 it is shown a particular area where a systematic error occurred for all three times: in this case the problem is probably due to the surface of a nearby building that it is very reflective (figure 2.23) and causes multipath effects.

day	N° of points fixed/total	% fixed solutions
14-259	1805/4038	45%
14-267	1313/3634	36%
14-268	1198/3547	33%

Table 2.3 – Survey with geodetic receiver along the Path 1: number of points with fixed phase solution compared to total points collected

---

<sup>22</sup>respectively October 16th, October 24th and October25th, 2014

<sup>23</sup>see appendix B.1









Figure 2.23 – The building in Gerbetto square that can be the possible source of the systematic error

### 2.6.3 Path 2: survey with Smart Mobile Devices, u-blox and Geodetic receivers

In the previous section the application of CORS corrections to the data did not improve the accuracy of the position: one of the reasons could be due to the difficulties encountered along the way, bad visibility and lack of signal. Therefore a new path has been defined (the Path 2, figure 2.15) with less obstacles, to verify if at least in discrete conditions the correction method can improve positions like it was done in static occupation (section 2.5).

The starting point of this new path was placed in Cavour Square, then the path continues in north west direction, along the lake, until reaches the War Memorial and then comes back to the start point. The path is generally well exposed, with few buildings and only occasionally with trees that could interfering with satellite signals.

The survey along Path 2 has been made with a geodetic receiver, Leica GX 1230<sup>24</sup>, the u-blox EVK-7P evaluation kit<sup>25</sup> and a smartphone, Samsung S5; three repetitions of the entire path were made on the days 16-139, 16-207 and 16-208<sup>26</sup>.

Reading the table 2.5 it is possible realize how the satellites visibility on this path is really better than on Path 1: after the processing of data collected with the geodetic receiver the number of fixed solutions are generally more than 80%, when in Path 1 was largely less then 50% (table 2.3) in all the three repetitions.

The difference between u-blox and smartphone can be easily viewed comparing the results on the map: in figure 2.24 the results of survey made with the smartphone and in figure 2.25 those made with the u-blox. The first one is very noisy and it is possible guess the inaccuracy also at this small scale. On the contrary the results obtained with the u-blox seem smoother and follow better the path.

This great difference between collected data with devices that share the same technical characteristic is probably due to the different antenna used. The GNSS antenna in smartphones is integrated, instead u-blox uses an external antenna, that probably permits a better signal reception.

In tables 2.6 and 2.7 are reported the statistical errors obtained at the end of the three repetitions of survey.

Let's start considering the results obtained with the smartphone (table 2.6): they are quite variable, with the standard deviation in the day 16-139 that it is always under 2.5 m, when for the other surveys it goes by 4 m to almost 5 m.

One possible explanation of this great difference could be due to the different seasons when the surveys were made: the first one, with smaller standard deviations was made in May, the others two in July and the different cover effect due to the trees may have affected the accuracy of measures.

In figure 2.26 are compared position collected during the three days of survey near the lake in a open area. Note that there are two repetitions of the path, the outward and the return. Seeing this figure there is not a clear evidence of a better accuracy of the 16-139 survey.

But, if it is considered an area partially covered by trees (figure 2.27), it will see how the path tracked in May is clearly more close to the actual path then the

---

<sup>24</sup>see appendix B.1

<sup>25</sup>see appendix B.2

<sup>26</sup>respectively May 18th, July 25th and July 26th 2016

others two, mainly respect the survey made in day 16-208.

Consider now the results obtained with the u-blox receiver (table 2.7): the standard deviation is generally under 1.5 m, with only one case very bad, but in any case it is less than 2.5 m; the mean is less 0.5 m in almost all cases, or slightly higher. These results are significantly better than these obtained with smartphone and then in the u-blox data the effect due to the seasonal is not so evident like in the smartphone data, confirming a better signal receiving.

Let's check the path in some particular circumstances to try to explain the difference shown in tables 2.6 and 2.7. In figure 2.28 it is shown a piece of path very well exposed: it is possible to see how estimated points from geodetic receiver (green triangles) and from u-blox receiver (blue circles) are very close, when points from smartphone (yellow circles) are more fluctuating around the actual path. This shows how, also in good conditions, the accuracy of smartphone survey is affected by an error that sometimes is significant, as in the area just before the Memorial War (figure 2.29): this explain why smartphone standard deviation is general the double of u-blox standard deviation.

Where geodetic data processing didn't give fixed solutions it is not possible to have a reference to use as actual path (as explained in section 2.6.1), but could be interesting do at least a qualitative analysis if a quantitative is not possible. In figure 2.30 it is shown a piece of the path where the signal was obstructed by trees: the red triangles are the point estimated after processing data collected with geodetic receiver whereby ambiguity solution was not fixed. It is possible see how in this area the points collected with u-blox continue to follow approximately the actual path, when the points collected with smartphone worsen. So, also in the areas not included in the statistics reported in tables 2.6 and 2.7 the general data trend is confirmed.

Let's consider now the application of CORS corrections to the estimated position. In the last two columns of the tables 2.6 and 2.7 are reported the errors statistics: in both of cases, u-blox and smartphone, these corrections do not improve the survey accuracy, confirming what already seen in the case of Path 1 (table 2.4).

Also in this case, despite better conditions, the local errors are greater than the errors shared with the CORS station and so the corrections are not applicable with success. The figure 2.31 well explains the effect of local errors: the effect of CORS corrections on the points position (brown circles) is minimum if compared with the general positioning error and it can not significantly affect the accuracy of the survey. Moreover there is an area (in the middle of path in figure 2.31) where the CORS corrections even worsens the accuracy instead improve it.

day	N° of points fixed/total	% fixed solutions
16-139	1886/2144	88%
16-207	1841/2017	91%
16-208	1596/1811	88%

Table 2.5 – Survey with geodetic receiver along the Path 2: number of points with fixed phase solution compared to total points collected



Figure 2.24 – Path 2 detected with smartphone - Day 16-208



Figure 2.25 – Path 2 detected with u-blox - Day 16-208

		Smartphone		Smartphone corrected	
Day	[m]	East	North	East	North
16-139	Mean	-0.8	1.8	-0.9	2.1
	Snd Dev	2.0	2.4	2.0	2.4
	Max	6.0	14.6	5.5	15.2
	Min	-9.3	-4.8	-9.5	-4.7
	Median	-0.5	1.7	-0.6	1.9
16-207	Mean	-1.4	-2.3	-1.3	-2.3
	Snd Dev	4.0	4.4	4.0	4.5
	Max	13.6	13.5	14.0	13.9
	Min	-28.1	-21.0	-28.3	-21.5
	Median	-0.8	-2.4	-0.7	-2.4
16-208	Mean	0.9	-0.9	0.9	-0.8
	Snd Dev	4.9	4.0	4.9	4.0
	Max	13.3	13.5	13.4	13.1
	Min	-13.2	-11.7	-12.9	-11.6
	Median	0.1	-1.4	0.0	-1.2

Table 2.6 – Smartphone statistical errors before and after the correction with CORS data along the Path 2

		u-blox		u-blox corrected	
Day	[m]	East	North	East	North
16-139	Mean	-0.6	0.5	-0.7	0.6
	Snd Dev	0.9	1.3	1.1	1.4
	Max	3.2	5.0	3.0	5.3
	Min	-2.5	-2.9	-3.4	-2.3
	Median	-0.8	0.2	-0.8	0.3
16-207	Mean	-0.1	0.6	-0.1	0.3
	Snd Dev	1.4	2.4	1.4	2.8
	Max	4.3	19.7	4.5	22.1
	Min	-4.1	-5.2	-3.9	-5.8
	Median	0.0	0.3	0.1	0.1
16-208	Mean	-0.2	-0.2	-0.3	-0.1
	Snd Dev	1.2	1.1	1.2	1.2
	Max	2.1	2.5	2.2	2.6
	Min	-3.9	-5.1	-3.9	-4.6
	Median	-0.1	-0.2	-0.2	-0.1

Table 2.7 – u-blox statistical errors before and after the correction with CORS data along the Path 2



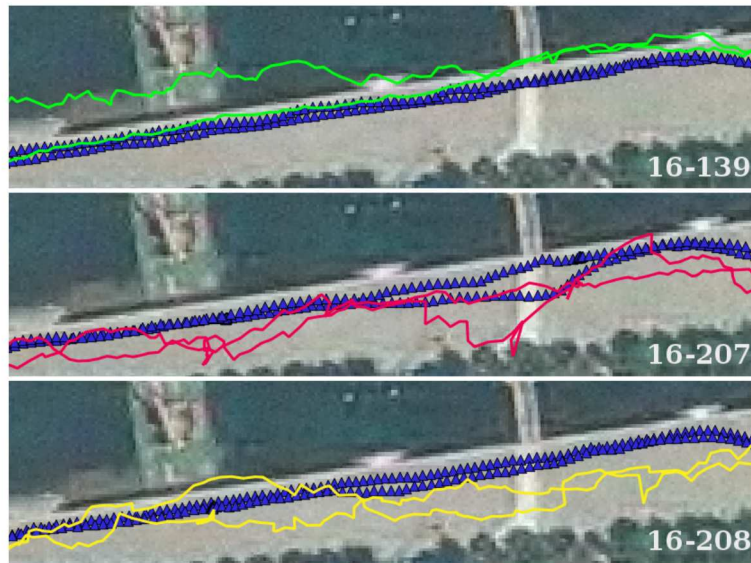


Figure 2.26 – Comparison between data collected with smartphone in different days in an open area: 16-139 (green circles), 16-207 (red circles), 16-208 (yellow circles). The actual path are the blue triangles

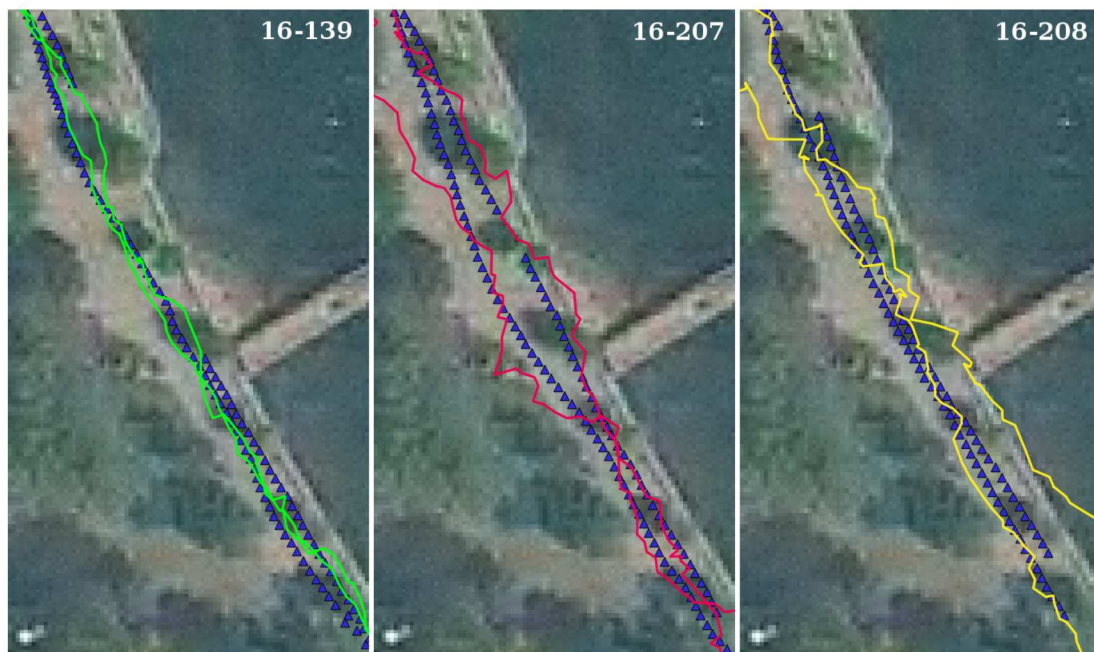


Figure 2.27 – Comparison between data collected with smartphone in different days in a area partially covered by trees: 16-139 (green circles), 16-207 (red circles), 16-208 (yellow circles). The actual path are the blue triangles



Figure 2.28 – Detail of the area in front of the lake with good exposition: points collected with geodetic receiver (green triangles), u-blox (blue circles) and smartphone (yellow circles) - Day 16-208



Figure 2.29 – Detail of the area near War Memorial: points collected with geodetic receiver (green triangles), u-blox (blue circles) and smartphone (yellow circles) - Day 16-208





Figure 2.30 – Detail of an area with bad exposition: points collected with geodetic receiver (green triangles when the ambiguity solution was fixed, red triangles when it was not fixed), u-blox (blue circles) and smartphone (yellow circles) - Day 16-208



Figure 2.31 – Data collected with u-blox: before (light blue circles) and after the CORS corrections (brown square) and actual point (green triangles) - Day 16-208



## Chapter 3

# Navigation: improve accuracy using cartographic data as references

### 3.1 Get and organize the reference cartography

The positioning in a urban environment with the accuracy request by the MEP project (section 2.2) using GNSS mass market devices<sup>1</sup> can be very difficult like it has been already seen in the chapter 2. The main problems are due to obstacles and surface that prevent or interfere with the correct data reception, that deteriorates positioning accuracy. In urban context, however, you can use other resource to try to correct errors in positioning: a local map, with a good representation of buildings, streets and other elements, can be used to find possible mistakes and correct trajectories.

For example, if you are surveying a path along a road and one of the points is detected on a roof, the position of this point will be clearly wrong and you have to correct it. The point should be moved in the right position, but the correction is not easy, because there could be more candidates as true position. In figure 3.1 a trivial example is shown where it is easy know that the point is in a wrong position, but it is very hard find where to move it.

The mass market devices were used only for the points detected activities, when the corrections were made at server side: data were sent from the smart mobile devices to a server, where the correction procedures were implemented and applied.

The purpose of procedures studied and described in this chapter is therefore to find a good way to correct estimated data that are in bad position, using cartography elements as main reference [40].

As reference map was chosen OpenStreetMap<sup>2</sup> [41]: OSM is a collaborative project with the aim of creating a collection of geographic data on a global scale. The data access is free, with a particular license [36] that permits to use, download and modify data freely, without particular restrictions. The advantages of this choice are obvious: with OSM data it is possible to get data and organize them

---

<sup>1</sup>In the following examples smartphone and tablets

<sup>2</sup>OSM - <http://www.openstreetmap.org>

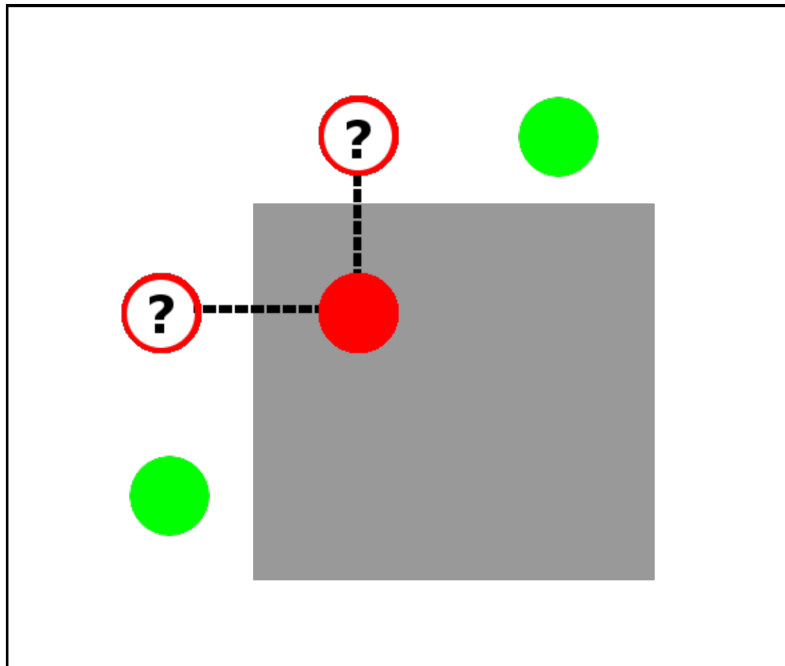


Figure 3.1 – Example of wrong point positioning: the green circles are the points correctly positioned, the red one is inside a building (grey rectangle) and so in wrong position. The circles with question mark are two possible right positions

in a local database with the estimated data during the survey activities, and use them together with the database tools to correct the estimated positioning. Data are constantly updated and, in any case, if something wrong is found, it will be possible correct it directly in MEP database or, better, update the OSM database and contribute to the project sustainability and development.

To manage the cartographic database it was chosen PostgreSQL<sup>3</sup> [44], one of the most important and powerful open source object-relational database systems. It includes most SQL data types, including *integer*, *numeric*, *boolean*, *char*, *varchar*, *date*, *interval*, *timestamp* and it has full support for joins, views, triggers, and support multiple procedural languages. It runs on all major operating systems, including Linux, UNIX and Windows; in this particular case all the software to perform the tests were installed on a Gentoo Linux system<sup>4</sup>.

Furthermore, very important, PostgreSQL has a database extension named PostGIS<sup>5</sup> [12] that implements additional supports for geographic objects, allowing to use PostgreSQL as a spatial database for geographic information systems, much like ESRI's SDE or Oracle's Spatial extension.

This extension add dedicated features to deal efficiently 3D data into PostgreSQL, like:

- new data types, as *point*, *linestring*, *polygon*, *multipoint*, *multilinestring*, *multipolygon*, *geometrycollection*, so you can define data as geographical data;
- new functions to build geographical multidimensional queries (for example

---

<sup>3</sup><http://www.postgresql.org>

<sup>4</sup><https://www.gentoo.org/>

<sup>5</sup><http://postgis.net/>

to check if a point is inside a polygon) and to create new map objects (such as buffers);

- the information about the data reference system<sup>6</sup> [26] and tools to convert data from a reference system to another one;
- tools to allows direct access to the data from many GIS software.

Furthermore many tools are available to easily import data in the database from the most common formats for geospatial data (like shapefiles) but also from OSM. These additional features are essential to permit the building of analysis procedures and correction of the estimated positions with the smart mobile devices.

Therefore all the data to perform the test were collected in a PostgreSQL/PostGIS database, in detail:

- points detected with mass market devices, estimated during the various surveys made<sup>7</sup>. This data were organized as shown in table 3.1;
- line (item key = *highway*) that describe a generic path (pedestrian, street, footway, cycling path,...) and buildings (item key = *building*), described by own perimeter, from the OSM<sup>8</sup> maps in the neighborhood of the points surveyed, organized as shown in table 3.2;
- script to implement the data correction procedures.

The field defined as *geometry* contains the coordinates of spatial item: it is defined according to the geometry (*point, linestring,...* ) and the reference system of the objects stored. For these data the reference system is WGS 84, to which corresponds the EPSG code 4326<sup>9</sup>.

All the data downloaded from OSM are without any kind of topology information. In figures 3.2, 3.3 and 3.4 are reported as example some data rows extracts from these tables.

In figure 3.5 is reported the implemented architecture scheme.

<b>gpsdata</b>		
Column	Type	Description
id	integer	point identifier
coord	geometry(Point,4326)	point coordinates
time	time without time zone	epoch of point collection

Table 3.1 – GPS data table

---

<sup>6</sup>PostGIS uses, as many other open source software, the EPSG notation (defined by the International Association of Oil & Gas Producers - <http://www.epsg.org/EPSPGhome.aspx>) to associate the reference system to the data.

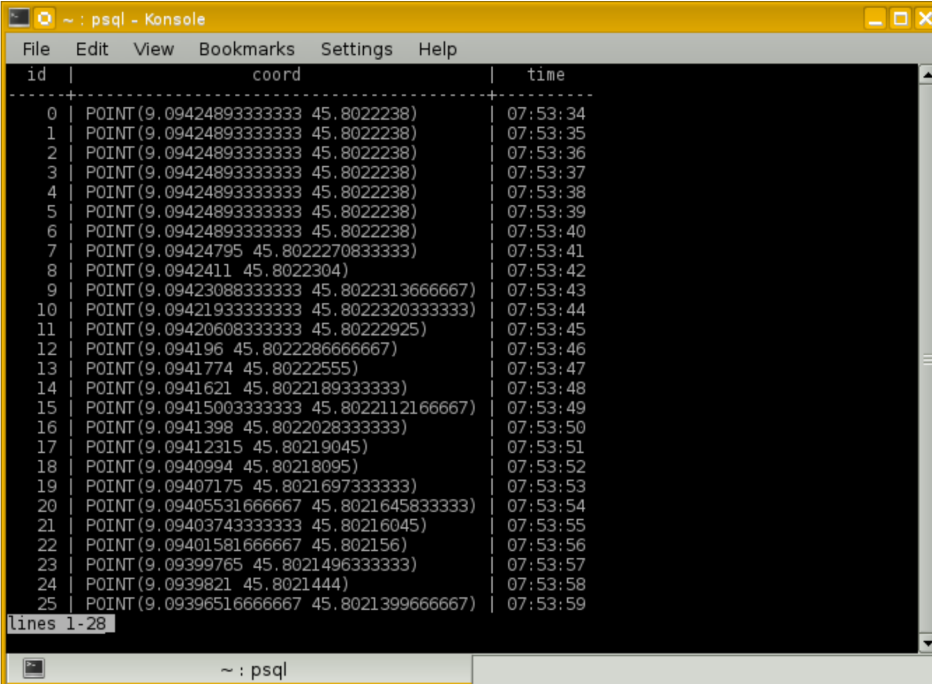
<sup>7</sup>The data used in these tests are those estimated on days 14-259, 14-267 and 14-268 on Path 1 (figure 2.14) and already used in section 2.6.2

<sup>8</sup>©OpenStreetMap contributors

<sup>9</sup>The online registry of EPSG Geodetic Parameter Dataset is available at <https://www.epsg-registry.org/>

highway.routes and building.routes		
Column	Type	Description
ogc_fid	line integer	line identifier
wkb_geometry	geometry(LineString,4326)	line coordinates: an ordered list of points that define the line path
name	character varying	name associated to street or building

Table 3.2 – OSM tables: two tables with the same structure, where data downloaded from OSM are storage: *highway.routes* is the table with the path data and *building.routes* is the table with the building data



```

psql - Konsole
File Edit View Bookmarks Settings Help
id | coord | time
-----+-----+-----
0 | POINT (9.09424893333333 45.8022238) | 07:53:34
1 | POINT (9.09424893333333 45.8022238) | 07:53:35
2 | POINT (9.09424893333333 45.8022238) | 07:53:36
3 | POINT (9.09424893333333 45.8022238) | 07:53:37
4 | POINT (9.09424893333333 45.8022238) | 07:53:38
5 | POINT (9.09424893333333 45.8022238) | 07:53:39
6 | POINT (9.09424893333333 45.8022238) | 07:53:40
7 | POINT (9.09424795 45.80222708333333) | 07:53:41
8 | POINT (9.0942411 45.8022304) | 07:53:42
9 | POINT (9.09423088333333 45.8022313666667) | 07:53:43
10 | POINT (9.09421933333333 45.8022320333333) | 07:53:44
11 | POINT (9.09420608333333 45.80222925) | 07:53:45
12 | POINT (9.094196 45.8022286666667) | 07:53:46
13 | POINT (9.0941774 45.80222555) | 07:53:47
14 | POINT (9.0941621 45.8022189333333) | 07:53:48
15 | POINT (9.09415003333333 45.8022112166667) | 07:53:49
16 | POINT (9.0941398 45.8022028333333) | 07:53:50
17 | POINT (9.09412315 45.80219045) | 07:53:51
18 | POINT (9.0940994 45.80218095) | 07:53:52
19 | POINT (9.09407175 45.8021697333333) | 07:53:53
20 | POINT (9.09405531666667 45.8021645833333) | 07:53:54
21 | POINT (9.09403743333333 45.80216045) | 07:53:55
22 | POINT (9.09401581666667 45.802156) | 07:53:56
23 | POINT (9.09399765 45.8021496333333) | 07:53:57
24 | POINT (9.0939821 45.8021444) | 07:53:58
25 | POINT (9.09396516666667 45.8021399666667) | 07:53:59
lines 1-28
~ : psql

```

Figure 3.2 – List of some rows of gpsdata table

ogc_fid	wkb_geometry	name
1	LINESTRING(9.0941185 45.8022038,9.0933611, 45.8017537,9.0929965 45.801537,9.0922956, 45.801163)	Via Francesco Anzani
2	LINESTRING(9.0935461 45.8027475,9.0937052, 45.8026155)	Via Valleggio
3	LINESTRING(9.0937052 45.8026155,9.0937893, 45.8027824,9.0942801 45.8031192)	Via Valleggio
4	LINESTRING(9.0937052 45.8026155,9.0941185, 45.8022038,9.0941662 45.8021542,9.094591, 45.8017128,9.0951324 45.8011728,9.095723, 8 45.8005455,9.0960535 45.8000654,9.09613, 9 45.7998962)	Via Valleggio
5	LINESTRING(9.0932415 45.8028617,9.0927325, 45.802631)	Via Ambrosoli
6	LINESTRING(9.0927749 45.8023795,9.0929293, 45.8024539,9.0933727 45.8026674)	Via Castelnuovo
7	LINESTRING(9.0933727 45.8026674,9.0935461, 45.8027475,9.0942801 45.8031192,9.094701, 3 45.803358,9.0949436 45.8034983,9.095177, 3 45.80362,9.0954577 45.8037476,9.0959315, 45.8039266,9.0965229 45.8041726,9.096879, 3 45.8043229,9.0972674 45.8044649,9.09773, 14 45.8045836,9.0981164 45.8046473,9.0985, 187 45.8047264,9.0986983 45.8047763,9.098, 7923 45.8048025)	Via Castelnuovo
8	LINESTRING(9.0938551 45.803241,9.0941599	

Figure 3.3 – List of some rows of highway\_routes table

ogc_fid	wkb_geometry	name
13	LINESTRING(9.0947183 45.8021458,9.0948899 45.8019649,9.094612 4, 5.8018368,9.0944404 45.8020176,9.0947183 45.8021458)	
14	LINESTRING(9.0935319 45.8026245,9.0933817 45.8025422,9.0935265, 45.8024076,9.093666 45.8024861,9.0935319 45.8026245)	
15	LINESTRING(9.0953243 45.8017221,9.0952412 45.8017371,9.0951741, 45.8017595,9.0950749 45.8018156,9.0949917 45.8018923,9.0948523, 45.8018306,9.0947476 45.8017838,9.0947986 45.8017277,9.0948312, 45.8016924,9.0948663 45.8016581,9.0949193 45.8016174,9.0949971, 45.8015725,9.095054 45.801545,9.0951258 45.801522,9.0952304 45, 8015015,9.0953458 45.8014884,9.0954289 45.8014865,9.095504 45.8, 014959,9.0955791 45.8015071,9.0956676 45.8015277,9.0957454 45.8, 015501,9.0958125 45.8015763,9.095893 45.8016137,9.0959707 45.80, 16642,9.0957535 45.8018381,9.0956837 45.8017988,9.095622 45.801, 7707,9.0955255 45.801739,9.0954209 45.8017203,9.0953243 45.8017, 221)	Valleggio - VA
16	LINESTRING(9.0950475 45.8021087,9.0950368 45.8020732,9.095028 4, 5.8020307,9.0950307 45.8019877,9.0950449 45.8019458,9.09507 45, 8019065,9.0951052 45.8018712,9.0951493 45.8018411,9.0952006 45, 8018172,9.0952574 45.8018004,9.0953177 45.8017913,9.0953794 45, 8017902,9.0954403 45.8017972,9.0954983 45.8018119,9.0955513 45, 8018339,9.0955975 45.8018624,9.0956402 45.8019022,9.0956698 45, 8019474,9.0956846 45.801996,9.0956842 45.8020457,9.0956684 45.8, 020941,9.0956394 45.8021394,9.0957286 45.8022054,9.0949401 45.8, 021772,9.0950475 45.8021087)	Valleggio - V
17	LINESTRING(9.0949625 45.802532,9.0949611 45.8025754,9.0951217 4, 5.8025782,9.095135 45.8022338,9.0949741 45.8022308,9.0949625 45, 802532)	Valleggio - VP
18	LINESTRING(9.0947062 45.8013365,9.094815 45.8012274,9.0949638 4,	

Figure 3.4 – List of some rows of building\_routes table

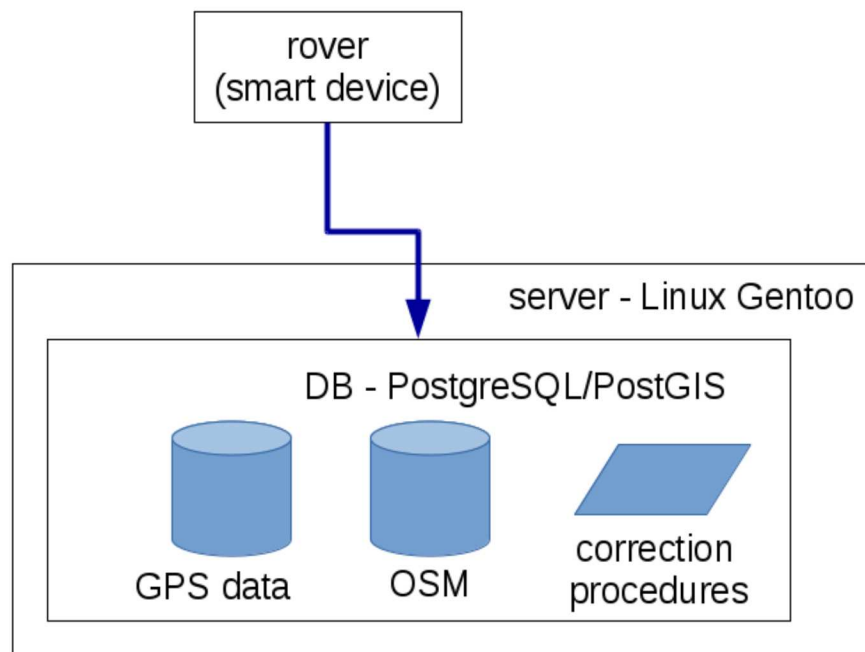


Figure 3.5 – The implemented architecture scheme

### 3.2 Correction using path segments as reference

The first procedure implemented uses the path segments (table 3.2, *highway.routes*) as reference [10] to make correction. This procedure assumes that the only acceptable position for a estimated point is on a segments and than corrects the points positions to move every point on the "pertinent segment".

The most critical phase of this procedure is to identify correctly the pertinent segment for each point, that it can not be simply the nearest segment. In figure 3.6 it is shown how, in correspondence of intersections or of more parallel segments, the point correction to the nearest segment jumps from a segment to another one, following every time the nearest, with a final result that is not acceptable.



Figure 3.6 – Correction on the nearest segment: the reds circles are positions of the points collected, the green circles are the positions after the correction to the nearest path

To avoid these kind of problems several indicators were considered to chose the "pertinent segment":

- the distance from the estimated point to the candidate segment;
- the bearing from the previous to the current point;
- the bearing of the candidate segment;
- the distance from the current corrected point to the previous corrected point.

In order to apply this method (but also the followings) it is needed consider each segment as a single feature vector, but in the data downloaded from OSM a feature vector generally is composed by more than one segment, like in figure 3.7.

The splitting of *highway.routes* elements into segment (figure 3.8) is obtained by the query 3.1 that select the points that define each path and uses these points to define the segments (as start and end point). The PostGIS Functions [39] (*ST\_\**) are used in this query and in all the following. To storage these data a new table *highway.segment* (table 3.3) is created. In figure 3.9 is possible to see as example some data row of this new table.

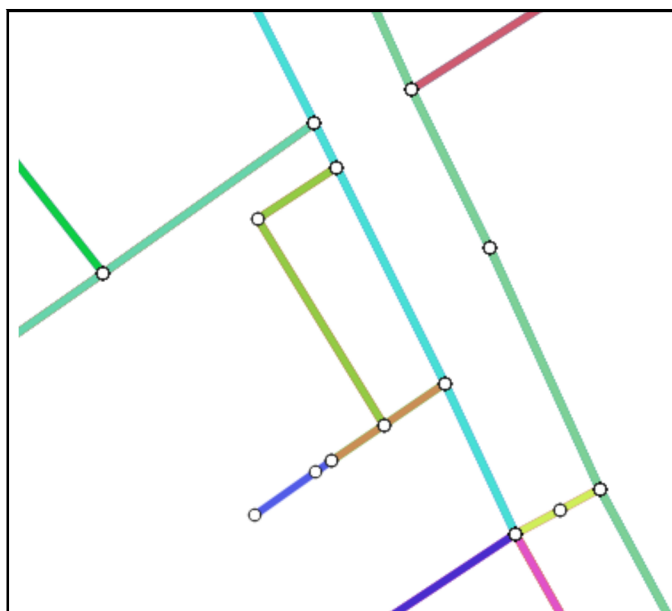


Figure 3.7 – Data organization in table *highway.routes*: each vector feature (lines with the same color) is composed by one or more segment

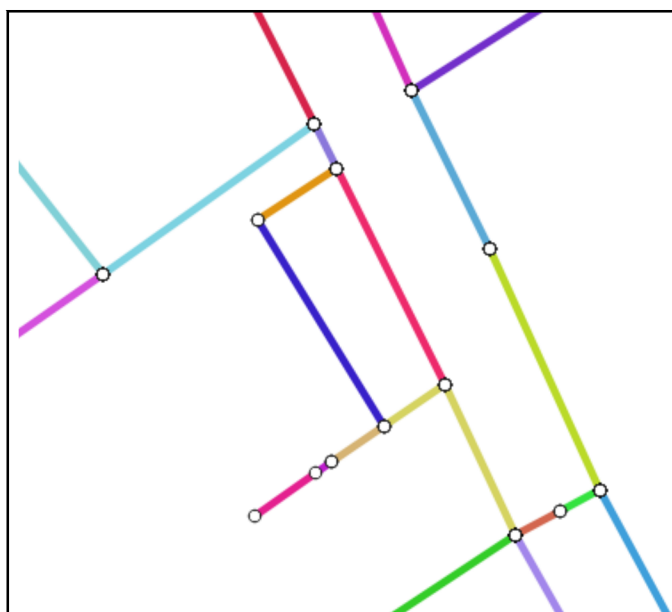


Figure 3.8 – Data organization in table *highway.segment*: each vector feature is composed by only one segment



**Query 3.1**

```

CREATE TABLE highway_segment AS
SELECT row_number() over(order by segment) as id, name,
ST_SetSRID(ST_MakeLine(sp,ep),4326) AS seg
FROM (
  SELECT DISTINCT
    ST_AsText(
      ST_PointN(wkb_geometry, generate_series(1, ST_NPoints(wkb_geometry)-1))
    ) AS sp,
    ST_AsText(
      ST_PointN(wkb_geometry, generate_series(2, ST_NPoints(wkb_geometry)))
    ) AS ep,
    name
  FROM highway.routes
) AS segment;

/* The query selects an ordered list of points from the specified
linestring field (wkb_geometry), then uses the couples
of consecutive points to define the segments */

```

<b>highway.segment</b>		
Column	Type	Description
id	integer	line identifier
seg	geometry(LineString,4326)	segment coordinates: start point and end point coordinates
name	character varying	name associated to street of which the segment is part

Table 3.3 – The *highway.segment* table

id	seg	name
1	LINESTRING(9.0739569 45.8133218,9.0740572 45.8133215)	Viale Filippo Corridoni
2	LINESTRING(9.0740572 45.8133215,9.0741368 45.8133174)	Viale Filippo Corridoni
3	LINESTRING(9.0741089 45.8109533,9.074206 45.810978)	Via Sant'Elia
4	LINESTRING(9.0741368 45.8133174,9.074197 45.813305)	Viale Filippo Corridoni
5	LINESTRING(9.074197 45.813305,9.0742695 45.8132869)	Viale Filippo Corridoni
6	LINESTRING(9.074206 45.810978,9.0742608 45.8109919)	Via Sant'Elia
7	LINESTRING(9.0742418 45.8126465,9.074119 45.8126365)	Viale Fratelli Rosselli
8	LINESTRING(9.0742608 45.8109919,9.0743109 45.8110092)	Via Sant'Elia
9	LINESTRING(9.0742695 45.8132869,9.0743811 45.8132547)	Viale Filippo Corridoni
10	LINESTRING(9.0743109 45.8110092,9.0746176 45.8111186)	Via Sant'Elia
11	LINESTRING(9.0743811 45.8132547,9.0744779 45.8132268)	Viale Filippo Corridoni
12	LINESTRING(9.0744779 45.8132268,9.0754565 45.8129598)	Viale Filippo Corridoni
13	LINESTRING(9.0745494 45.8126714,9.0742418 45.8126465)	Viale Fratelli Rosselli
14	LINESTRING(9.0746176 45.8111186,9.0752558 45.8113292)	Via Sant'Elia
15	LINESTRING(9.0746844 45.8135191,9.0744779 45.8132268)	Viale Filippo Corridoni
16	LINESTRING(9.0749075 45.8135609,9.0746844 45.8135191)	Viale Filippo Corridoni
17	LINESTRING(9.0749161 45.8138421,9.0749391 45.8139655)	Viale Filippo Corridoni
18	LINESTRING(9.0749391 45.8139655,9.0749208 45.8141804)	Viale Filippo Corridoni
19	LINESTRING(9.0749391 45.8139655,9.07514 45.8140204)	Viale Filippo Corridoni
20	LINESTRING(9.074959 45.8137524,9.0749161 45.8138421)	Viale Filippo Corridoni
21	LINESTRING(9.0750469 45.8115945,9.0751717 45.811436)	Viale Filippo Corridoni
22	LINESTRING(9.07514 45.8140204,9.0752892 45.8140715)	Viale Filippo Corridoni
23	LINESTRING(9.0751717 45.811436,9.0747447 45.811295)	Viale Filippo Corridoni
24	LINESTRING(9.0751717 45.811436,9.0752558 45.8113292)	Viale Filippo Corridoni
25	LINESTRING(9.0751774 45.8116956,9.0750469 45.8115945)	Viale Filippo Corridoni
26	LINESTRING(9.0751774 45.8116956,9.0752085 45.811675)	Viale Filippo Corridoni

Figure 3.9 – List of some rows of highway.segment table

The correction method selects, from the set of the closest segments to the estimated point, the segment that has similar direction to the previous estimated direction (to avoid jump from one segment to another in correspondence of intersections) and that minimizes the distance from the previous corrected point (to prevent jumps between parallel paths).

The selection of the segment is done in real time when a new estimated point is inserted into the database: a spatial query is recalled by a trigger on data table. The query 3.2 extracts the list of the nearest segments, calculates the distance from previous corrected point and projects the point on the pertinent segment; the simple query 3.3 evaluates bearing between previous and current estimated point.

The result of query 3.2 is an ordered list (1,...,n) by the distance (dgps) between the estimated point and the path segments. The list is evaluated starting to the nearest segment to the estimated point (row 1) up to the farthest (row n), to search the best "pertinent segment" to use to correct the point position. Therefore the nearest segment to the estimated point is the first segment selected as "pertinent segment".

Let's consider now the generic row  $i$  of the list, we have:

- $id\_seg_i$  segment  $i$ ;
- $dprevpoint_i$  distance between candidate corrected point  $i$  and previous corrected point;
- $angle\_line_i$  bearing of segment  $i$ .

If we indicate with  $c$  the row with the current segment selected as "pertinent segment" we have:

- $id\_seg_c$  current segment selected as "pertinent segment";

- $d_{prevpoint_c}$  distance between candidate corrected point  $c$  and previous corrected point;
- $angle_{line_c}$  bearing of segment  $c$ .

To decide if the segment  $i$  is better than the current segment  $c$  as "pertinent segment" we have to evaluate the different parameters available.

First of all the  $d_{prevpoint}$  is checked:

- if  $d_{prevpoint_i} > d_{prevpoint_c}$  the  $i$  segment is rejected as new "pertinent segment";
- otherwise the the comparison continues considering the bearing.

Then the difference between  $angle_{line_i}$  and  $angle_{point}$  is computed:

$$angle\_diff_i = \text{abs}(angle\_point - angle_{line_i})$$

Finally:

- if  $angle\_diff_i < angle\_diff_c$  the  $id_{seg_i}$  is chosen as new "pertinent segment";
- otherwise the  $i$  segment is rejected.

Then the analysis continues with the  $i+1$  row just to the last row of the list.

---

### Query 3.2

---

```

SELECT pt.coord AS point,
       ST_line_interpolate_point(
         ln.seg, ST_line_locate_point(ln.seg, pt.coord)
       ) AS pointonline,
       ST_Distance(ln.seg, pt.coord) AS dgps,
       ST_Distance(ln.seg, '$prevpoint') AS dprevline,
       ST_Distance(
         ST_line_interpolate_point(
           ln.seg, ST_line_locate_point(ln.seg, pt.coord)
         ), '$prevpoint'
       ) AS dprevpoint,
       ST_Azimuth(ST_StartPoint(ln.seg), ST_EndPoint(ln.seg)) AS angle_line,
       ln.id AS id_seg
FROM gpsdata pt, highway_segment ln
WHERE ST_Distance(ln.seg, pt.coord) < $maxdist and pt.id='$id'
ORDER BY dgps;

/* Where:
   $prevpoint is the previous point id;
   $id is the id of the current point;
   $maxdist is the maximum distance from the point beyond which the
   segment is not taken into account (20 m) */

```

---

In figure 3.10 are shown the same collected data shown in figure 3.6, but this

**Query 3.3**


---

```

SELECT
    ST_Azimuth('$prevgps','$new_point_gps') as angle_point;

/* Where:
    $prevgps is the previous estimated point;
    $new_point_gps is current estimated point;
*/

```

---

time corrected using this new method. You can see that the most of the problems present in 3.6 have been now solved.

The main drawback is that sometimes this method corrects too much. All the estimated points are moved up a segment and the segments, in most of cases, describe a road path. However, users that collect data are usually pedestrian and then they might not follow a road path, therefore it is not right move all the points on the segments always. OSM data describe also some pedestrian paths, but in any case, to assume that every point that is not positioned on a segment is in a wrong position can lead to a overcorrection on the data.

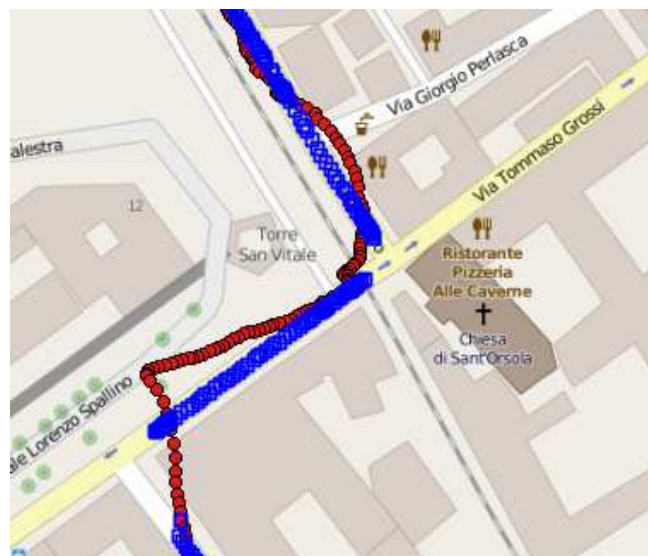


Figure 3.10 – Correction using several indicators in choosing of "pertinent segment": the red circles are positions of the estimated points, the blue squares are the positions after the correction

### 3.3 Correction out of the buildings

To avoid the possible overcorrection problems shown in the previous section, a new approach of data correction was attempted: in this method the point position will be classed as wrong only if the point is inside a building. The point of view is reversed respect the previous approach: now "all the points positions are right, except that satisfy the condition posed (to be inside a building)", when before it

was "all the points positions are wrong, except that satisfy the condition posed (to be on a segment)".

The method uses the buildings (table 3.2, *building.routes*) as reference, not more the segments.

To find if a point is or not inside a building, the buildings should be modeled as polygons but, from a topological point of view, the data in *building.routes* table are simple lines that describe the buildings border. Therefore data were reorganized (query 3.4) in a new database table, named *building* (table 3.4 and figure 3.11) in order to apply the correction procedure. When a point is found inside a building its position is corrected by projecting the point outside the building, along the perimeter. To permit the selection of the perimeter side where project the point the perimeters must be split into single segments (query 3.5), like it was did with the paths in the previous section. These segments are stored in a new database table, named *build\_segment* (table 3.5 and figure 3.12).

---

#### Query 3.4

---

```
CREATE TABLE building AS
  SELECT id, name, ST_Polygon(wkb_geometry, 4326) AS polyg
  FROM building.routes;

/* The ST_Polygon returns a polygon built from the specified
linestring field (wkb_geometry) */
```

---

---

#### Query 3.5

---

```
CREATE TABLE build_segment AS
  SELECT row_number() over(order by segment) AS id,
  name, ogc_fid AS idpoly
  ST_SetSRID(ST_MakeLine(sp,ep),4326) AS seg,
FROM (
  SELECT DISTINCT
  ST_AsText(
  ST_PointN(wkb_geometry, generate_series(1,ST_NPoints(wkb_geometry)-1))
  ) AS sp,
  ST_AsText(
  ST_PointN(wkb_geometry, generate_series(2,ST_NPoints(wkb_geometry)))
  ) AS ep,
  name, ogc_fid
  FROM building.routes
  ) AS segment;

/* The query selects an ordered list of points from the specified
linestring field (wkb_geometry), then uses the couples
of consecutive points to define the segments */
```

---

As in the previous case the correction is applied in real time as soon as a new point is inserted in the database table thank to a trigger that recalls the correction

<b>building</b>		
Column	Type	Description
id	integer	building identifier
polyg	geometry(Polygon,4326)	polygon coordinates: an ordered list of points that define the perimeter
name	character varying	name associated to the building

Table 3.4 – The *building* table

<b>build_segment</b>		
Column	Type	Description
id	integer	building segment identifier
idpoly	integer	building identifier
seg	geometry(LineString,4326)	segment coordinates: start point and end point coordinates

Table 3.5 – The *build\_segment* table

procedure.

The query 3.6 check if the new estimated point is inside a building (function *ST\_Contains* [39]): if it is inside the output of the query is the projected point (*pointonline*) outside the building, using the segments path as reference (*hg.seg*).

Finally the nearest segment to the point (with lower *dpthg* value) is chosen as corrected position (blue circle in figure 3.13). If the estimated point is outside the building the *where condition* in the query is not satisfied, the query output is empty and the point coordinates are not corrected.

This method acts only when position is clearly wrong, doesn't constraint the point position on a defined path. In Figure 3.14 it is shown an example where the application of this method gives good results. However, in other more critical situations, this simple correction may not be sufficient, as shown in figure 3.15, where the corrections applied are not enough to rebuilt the right path.

```

~ : psql - Konsole
File Edit View Bookmarks Settings Help
id | polyg | name
-----
1 | POLYGON((9.0748422 45.8125941,9.074832 45.8126481,9.0749326 45.8126.
.574,9.0749429 45.8126033,9.0748422 45.8125941))
2 | POLYGON((9.0750117 45.8121402,9.074994 45.8122888,9.075257 45.81230.
.41,9.0752748 45.8121554,9.0750117 45.8121402))
3 | POLYGON((9.0752118 45.8125736,9.0752178 45.812574,9.0752284 45.8124.
.984,9.0750338 45.8124851,9.0750234 45.8125596,9.0750703 45.8125628,.
.9.075055 45.8126718,9.0751967 45.8126815,9.0752118 45.8125736))
4 | POLYGON((9.0752305 45.8125559,9.0752333 45.8125561,9.0752201 45.812.
.6687,9.0753387 45.8126754,9.075352 45.8125627,9.0753207 45.8125609,.
.9.0753408 45.8123902,9.0752506 45.812385,9.0752305 45.8125559))
5 | POLYGON((9.0753369 45.8123789,9.0753585 45.8122084,9.0752735 45.812.
.2032,9.0752519 45.8123737,9.0753369 45.8123789))
6 | POLYGON((9.0752828 45.8112999,9.0755243 45.8113793,9.0755731 45.811.
.3051,9.0756637 45.8111672,9.0755126 45.8111193,9.0754478 45.8112178.
.9.0753555 45.8111918,9.0752828 45.8112999))
7 | POLYGON((9.0752848 45.8105191,9.0753686 45.8103961,9.0760503 45.810.
.6219,9.0759665 45.8107448,9.0752848 45.8105191))
8 | POLYGON((9.0753127 45.8121216,9.0753127 45.8121385,9.0754762 45.812.
.1385,9.0754762 45.8121216,9.0753127 45.8121216))
9 | POLYGON((9.0753207 45.8120685,9.0753178 45.8120957,9.0754199 45.812.
.1011,9.0754228 45.812074,9.0753207 45.8120685))
10 | POLYGON((9.0753742 45.812389,9.0754164 45.8123907,9.0754308 45.8122.
.18,9.0753886 45.8122163,9.0753742 45.812389))
11 | POLYGON((9.0754093 45.8138122,9.0753966 45.8138476,9.0754794 45.813.
.8621,9.0754921 45.8138266,9.0754093 45.8138122))
12 | POLYGON((9.0754097 45.8126734,9.0754291 45.8125469,9.0755168 45.812.
lines 1-28
~ : psql
    
```

Figure 3.11 – List of some rows of building table

```

~ : psql - Konsole
File Edit View Bookmarks Settings Help
id | idpoly | segment
-----
1 | 1 | LINESTRING(9.074832 45.8126481,9.0749326 45.8126574)
2 | 1 | LINESTRING(9.0748422 45.8125941,9.074832 45.8126481)
3 | 1 | LINESTRING(9.0749326 45.8126574,9.0749429 45.8126033)
4 | 1 | LINESTRING(9.0749429 45.8126033,9.0748422 45.8125941)
5 | 2 | LINESTRING(9.074994 45.8122888,9.075257 45.8123041)
6 | 2 | LINESTRING(9.0750117 45.8121402,9.074994 45.8122888)
7 | 3 | LINESTRING(9.0750234 45.8125596,9.0750703 45.8125628)
8 | 3 | LINESTRING(9.0750338 45.8124851,9.0750234 45.8125596)
9 | 3 | LINESTRING(9.075055 45.8126718,9.0751967 45.8126815)
10 | 3 | LINESTRING(9.0750703 45.8125628,9.075055 45.8126718)
11 | 3 | LINESTRING(9.0751967 45.8126815,9.0752118 45.8125736)
12 | 3 | LINESTRING(9.0752118 45.8125736,9.0752178 45.812574)
13 | 3 | LINESTRING(9.0752178 45.812574,9.0752284 45.8124984)
14 | 4 | LINESTRING(9.0752201 45.8126687,9.0753387 45.8126754)
15 | 3 | LINESTRING(9.0752284 45.8124984,9.0750338 45.8124851)
16 | 4 | LINESTRING(9.0752305 45.8125559,9.0752333 45.8125561)
17 | 4 | LINESTRING(9.0752333 45.8125561,9.0752201 45.8126687)
18 | 4 | LINESTRING(9.0752506 45.812385,9.0752305 45.8125559)
19 | 5 | LINESTRING(9.0752519 45.8123737,9.0753369 45.8123789)
20 | 2 | LINESTRING(9.075257 45.8123041,9.0752748 45.8121554)
21 | 5 | LINESTRING(9.0752735 45.8122032,9.0752519 45.8123737)
22 | 2 | LINESTRING(9.0752748 45.8121554,9.0750117 45.8121402)
23 | 6 | LINESTRING(9.0752828 45.8112999,9.0755243 45.8113793)
24 | 7 | LINESTRING(9.0752848 45.8105191,9.0753686 45.8103961)
25 | 8 | LINESTRING(9.0753127 45.8121216,9.0753127 45.8121385)
26 | 8 | LINESTRING(9.0753127 45.8121385,9.0754762 45.8121385)
lines 1-28
~ : psql
    
```

Figure 3.12 – List of some rows of build\_segment table



**Query 3.6**


---

```

SELECT ST_Distance(hg.seg, pt.coord) AS dpthg,
       ST_Intersection(
         ST_MakeLine(ST_ClosestPoint(hg.seg, pt.coord),pt.coord),ln.seg
       ) AS pointonline,
       pt.time AS time
FROM gpsdata pt, highway_segment hg, building bl, build_segment ln
WHERE ST_Contains(bl.polyg, pt.coord) and pt.id=$id and
       ST_Distance(hg.seg, pt.coord) < $maxdist and ln.idpoly=bl.id and
       Not ST_IsEmpty(ST_Intersection(
         ST_MakeLine(ST_ClosestPoint(hg.seg, pt.coord),pt.coord),
         ln.seg))
ORDER BY dpthg asc;
/* where:
   $id is the current point id;
   $maxdist is the maximum distance to consider a segment as reference
   to project the point (50 m) */

```

---

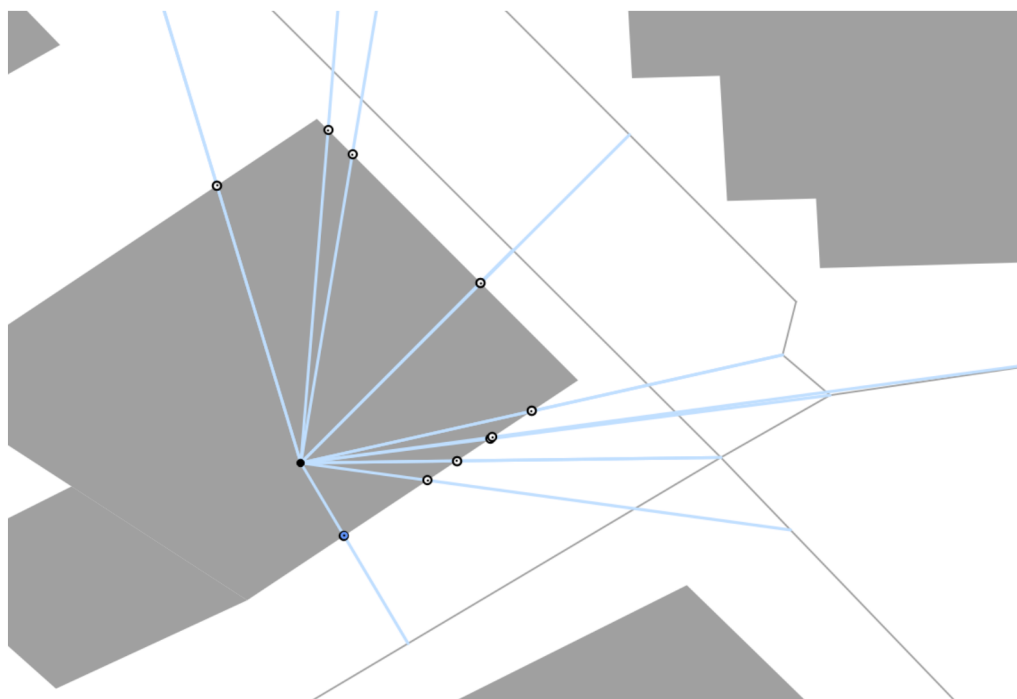


Figure 3.13 – The circle black is the estimated point, the white circles are all the possible corrections, between these, the blue circle is the correction chosen

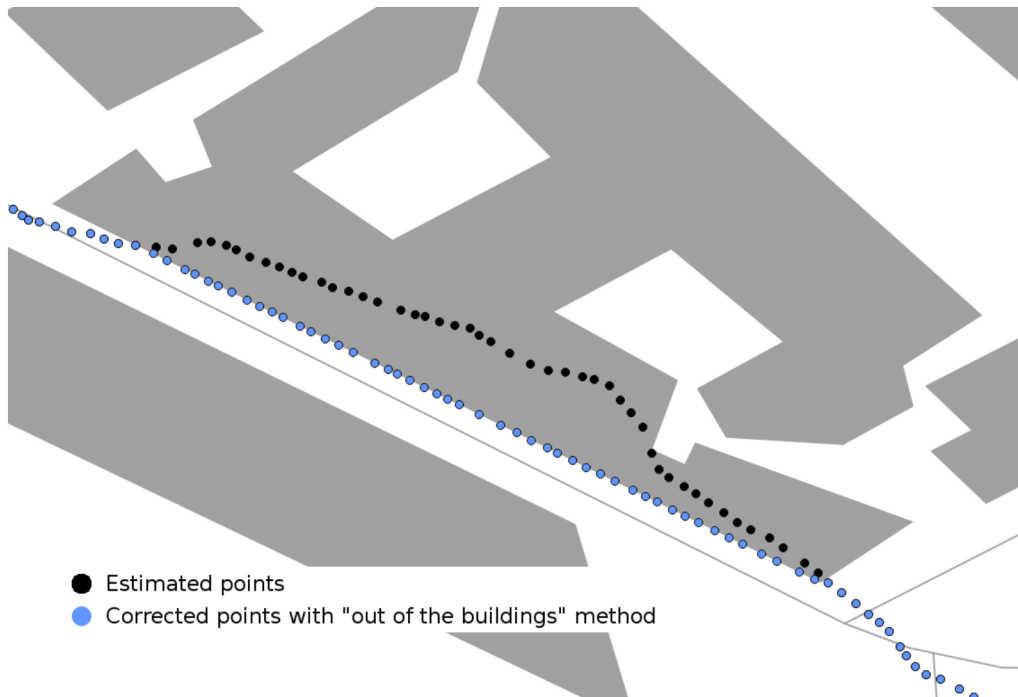


Figure 3.14 – Example of correction "out of the buildings" effective

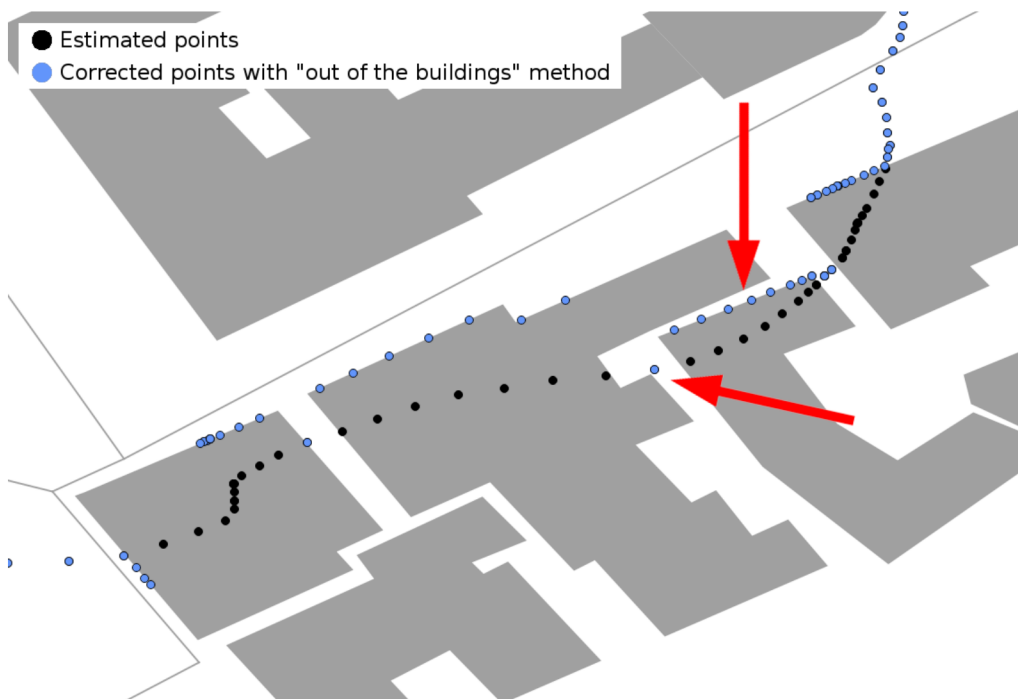


Figure 3.15 – Example of correction "out of the buildings" ineffective, the red arrows show the areas most critical

## 3.4 Correction on buffers

### 3.4.1 Basic principle

The "correction on buffer" method uses as reference the path segments (query 3.1) like the correction method viewed in the section 3.2. However in this case the point position will be considered wrong only if the distance from a segment is greater than a certain threshold, and in this case the point in a wrong place will be moved at the threshold distance, not on the segment as it was done in the correction procedure seen in the section 3.2. The classification criteria is more restrictive than in the section 3.3, because also a point outside a building can be classified in a wrong position, but the points are not forced to be placed on a segment, but it is enough that they are in the neighbourhood.

The core of this correction approach is in establishing a tolerance buffer: if the point is inside the buffer, it will not be correct, if it is outside, the position will be corrected using as new place its projection at the edge of the buffer.

Unlike previous methods the correction procedure is not applied in real-time, but it is executed at the end of mapping process, only when the entire path is completed (post-processing), this because the correction procedure needs all the set of estimated points to work.

The procedure is composed of three phases:

- search and selection of a potential path;
- building of the buffer around of the selected potential path;
- correction of points position that are outside the buffer.

The identification of the potential path [11] is composed of two different steps. In the first step a set of *candidates segment* is chosen to be part of the path (query 3.7): every estimated point assigns a score to the closest  $n$  segments: if the distance between the point and the segment is greater than one meter, the score assigned will be equal to the inverse of the distance; if the distance is equal or less than one meter, the score assigned will be equal to one (equation 3.1).

$$\begin{cases} distance \geq 1 \Rightarrow score = 1/distance \\ distance < 1 \Rightarrow score = 1 \end{cases} \quad (3.1)$$

The total score of each segment is the sum of the scores received by all points. All segments that receive a score high enough (a score equal to one was chosen as threshold) become part of the group of *candidates segment* and are stored in a new database table named *candidate\_path*. It is necessary define a threshold to avoid to include insignificant segments in the *candidates segment* set to avoid problems in the selections process as shown further.

In table 3.6 is shown the structure of the *candidates segment* table: in the field *dist* is storage the score received from the segment; in figure 3.17 is shown as example some data rows of this table. In Figure 3.18 is shown as example a part of the Path 1 with the score received by each segment after this step.

In the next step the segments that describe the path are selected (*segments selected*) from all of the *candidates segment*: every estimated point selects, between the  $n$  nearest *candidates segment*, the one with the highest score (query 3.8), that will be added to the *segments selected* group by setting to one the field *sel*. An

example is reported in figure 3.16, where we have a set of estimated points (P1-P4) and segments (S1-S5), with the associated score (blue numbers), among which to choose those to be selected. In the algorithm implemented every point checks between the two nearest segments ( $n = 2$ ). The two nearest to point P1 are the segments S2 and S3, but the scores of S3 is too low and it doesn't belong to the *segments selected* set. Therefore the segments considered by P1 are S1 and S2. S2 is the nearest segment, but its score is lower than the S1, then S1 is selected by P1. About the P2 point we can do the same considerations: S2 and S4 are the two nearest segments, but the S4 score is too low to be considered. Then, between S2 and S5, P2 selects S5. In this way the intersect between road are managed correctly. This example also explains why, to avoid problems in selection procedure, mainly due to short segments, the segments that receives a low score do not have to be included in *candidates segment* set. Finally, P3 and P4 choose S5 that is the nearest segment and also the segment with the highest score.

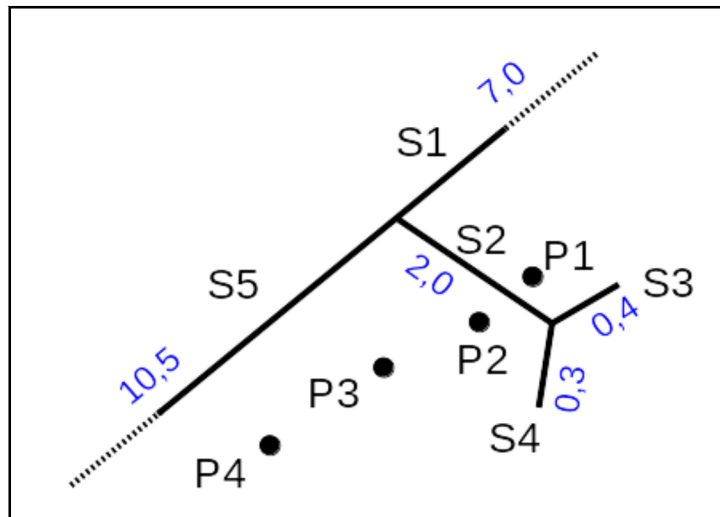


Figure 3.16 – Example of segment selection procedure. P1-P4 estimated points, S1-S5 segments (S1-S5), with the associated score (blue numbers)

Then a buffer is built around the *segments selected* (query 3.9) taking into account the presence of buildings, like shown in figure 3.19. The last step consists in the correction of all the points of which the position is outside the buffer area, by projecting the point on the nearest buffer border. In figure 3.20 it is shown how, following this method, all the points outside the buffer must be corrected, both if they are inside or outside a building. It should be clear how the quality of final results depend strongly from the buffer size chosen and if its dimension are fixed or variable.

**Query 3.7**

---

```
SELECT hg.*,
       ST_Distance(hg.seg, pt.coord) AS dpthg
FROM gpsdata pt, highway_segment hg
WHERE pt.id=$id and ST_Distance(hg.seg, pt.coord)<$dist
ORDER BY dpthg ASC LIMIT $maxsel;

/* where:
   $id is the current point identifier;
   $dist is the maximum distance considered: segments farther
       of $dist will not be considered (10 m);
   $maxsel is the maximum number of segments that can be selected (5);
   $dist and $maxsel are variable and the value is defined inside
       the script that call this query.
```

The 5 (\$maxsel) nearest segments to the estimated point \$id are selected and the distance from the point is computed (dpthg). Segments farther than 10 m (\$dist) are not considered.

```
*/
```

---

candidate_path		
Column	Type	Description
id	integer	segment identifier
seg	geometry	segment coordinates: start point and end point coordinates
name	character varying	name associated to street of which the segment is part
dists	double precision	score assigned to the segment
sel	integer	1 if selected, 0 otherwise

Table 3.6 – structure of candidate\_path DB table

---

### Query 3.8

---

```
SELECT cp.*,
ST_Distance(cp.seg, pt.coord) AS dpthg
FROM gpsdata pt, candidate_path cp
WHERE pt.id=$id and
      ST_Distance(cp.seg, pt.coord) < $maxdist and
      cp.dists >= $minval
ORDER BY dpthg ASC LIMIT $limit;
```

```
/* where:
```

```
  $id is the identifier of the current gps point;
  $maxdist is the maximum distance beyond which the point is
    not considered (10 m);
  $limit is the number of nearest point considered (2).
```

The two (\$limit) nearest candidate segment (table candidate\_path) to the estimated point \$id are selected and the distance from the point is computed (dpthg).

The segment with a score less than 1 (\$minval) and farther than 10 m (\$dist) are not considered.

```
*/
```

---

---

### Query 3.9

---

```
SELECT ST_Buffer(seg, $buffdim, 'endcap=square join=round')
AS buff_path, id
FROM candidate_path
WHERE sel = 1
```

```
/* where:
```

```
  $buffdim is the buffer dimension (7 m);
  $idsel is the id of the selected segment as path component
```

The ST\_Buffer function returns a geometry covering all points within a given distance from the input geometry

```
*/
```

---

id	seg	name	dists	sel
52	LINESTRING(9.0755088 45.8140278,9.0759375 45.8131003, .)	Viale Guglielm. .o Marconi	3.1281197211456	0
85	LINESTRING(9.0759375 45.8131003,9.0760444 45.8128598, .)	Viale Guglielm. .o Marconi	13.056583711079	1
89	LINESTRING(9.0760319 45.813208,9.0759375 45.8131003)		13.626282954727	1
91	LINESTRING(9.0760377 45.8127928,9.0757132 45.8122035, .)		1.6296631081179	0
92	LINESTRING(9.0760377 45.8127928,9.0759621 45.8127866, .)	Viale Fratelli. . Rosselli	1.8791320902796	0
94	LINESTRING(9.0760427 45.8141428,9.0764256 45.8137118, .)	Lungo Lario Ma. .rgherita di Sa. .voia	5.9624716257636	0
95	LINESTRING(9.0760444 45.8128598,9.0758166 45.8128525, .)		5.4542050267588	0
96	LINESTRING(9.0760444 45.8128598,9.0762465 45.812861)		14.144917891318	1
98	LINESTRING(9.0760805 45.8132294,9.0760319 45.813208)		7.0266204868207	1
99	LINESTRING(9.0760805 45.8132294,9.0760333 45.8133341, .)	Viale Guglielm. .o Marconi	4.3018547825476	0
101	LINESTRING(9.0761948 45.8127982,9.0760377 45.8127928, .)	Viale Fratelli. . Rosselli	12.044870776596	1
102	LINESTRING(9.0761948 45.8127982,9.0762056 45.8127711, .)	Viale Felice C. .avallotti	8.9318514386627	0
103	LINESTRING(9.0762021 45.8129595,9.0760805 45.8132294, .)	Viale Guglielm. .o Marconi	5.2264579427631	0
104	LINESTRING(9.0762056 45.8127711,9.07623 45.81271)	Viale Felice C. .avallotti	2.1788252116022	0

Figure 3.17 – List of some rows of candidate\_path table



Figure 3.18 – The red lines are the candidate segments with the score received, black circles are the estimated points - Day 14-259



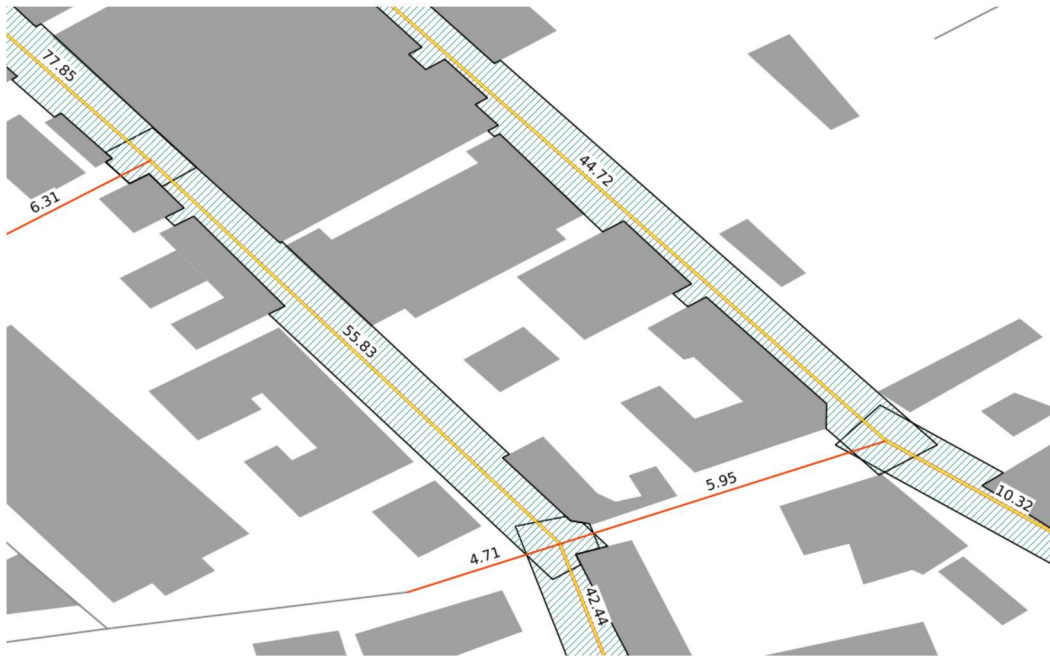


Figure 3.19 – The yellow-red lines are the selected segments; the green hatch areas represent the buffer - Day 14-259

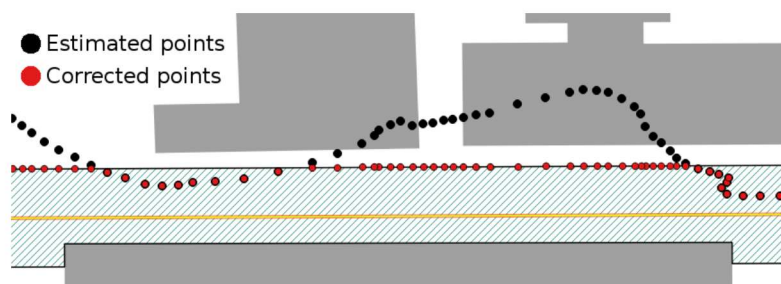


Figure 3.20 – Correction of points positions outside the buffer

### 3.4.2 Buffers with constant dimension

The first implementation of the "correction on buffer" procedure was made fixing a constant value of 7 meters as buffer dimension. This value was chosen taking as reference the width of an urban two-lane road with the sidewalks. In figure 3.21 it is possible to compare the results obtained with the "correction on buffer constant" method with those obtained with the "out of the buildings" method (section 3.3). The buffer method introduces improvements (green arrows) if compared with the other one, but there are still some critical situations in areas between buildings (red arrow).

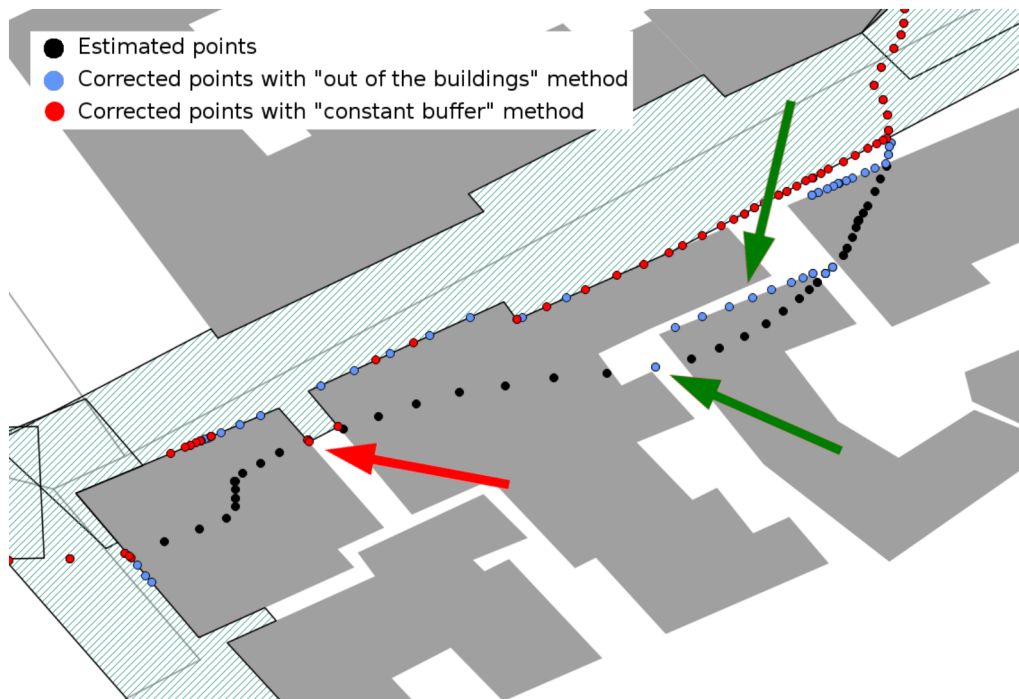


Figure 3.21 – Example of "correction on buffer" compared with the "out of the buildings" method. The green arrows indicate where the method has been more efficient, the red arrow indicates where there are still critics results - Day 14-259

### 3.4.3 Buffers with variable dimension

It is known that in open areas generally the positioning is less affected by errors than in more obstructed places. Starting from this consideration the "correction on buffer" method can be modified to take account of this, defining a variable tolerance buffer, that depends on the road width where it is. The procedure defined in section 3.4.1 is still valid, but now a different buffer dimension is computed for each *selected segments*. The procedure that manages the buffer creation call the query 3.10 that computes the distance between the segment and the nearest building: if it is less than 7 meters, the buffer dimension will be set equal to this distance, otherwise the buffer dimension will be set to 7 meters. To avoid situation with buffers too small the minimum dimension is in any case set to 2 meters. In figure 3.22 it is possible to compare the different results obtained using a fixed

buffer dimension against a variable dimension. It is possible to see how with a buffer less wide the problems saw in previous section are solved (green arrow).

**Query 3.10**

```
SELECT ST_Distance(cp.seg, bl.polyg) as dcpbl
FROM candidate_path cp, building bl
WHERE cp.id=$id and ST_Distance(cp.seg, bl.polyg) < $maxbuffdim
ORDER BY dcpbl ASC LIMIT 1

/* where:
    $id is the id of the current segment;
    $maxbuffdim is the maximum buffer dimension (7 m).
The query compute the distance (dcpbl) between the current candidate
path ($id) and the nearest buildings
*/
```

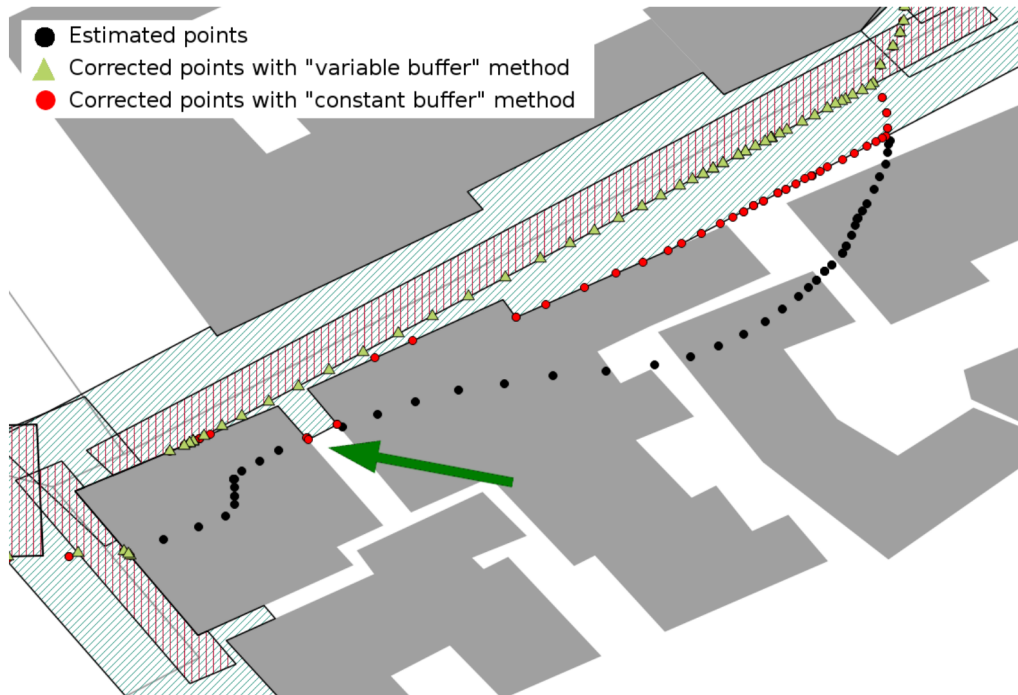


Figure 3.22 – Example of "correction on buffer variable" method compared with the "correction on buffer constant" method. The green arrow indicates where the "correction on buffer variable" method has been more efficient - Day 14-259

**3.4.4 Reclassification: segments selection in two steps**

In some particular situations, where there is a very bad quality of signal the positioning accuracy can be very low. For example, in the old town area (figure 2.14, the red area) there are narrow streets and high buildings: in these urban canyons (figure 3.23) the satellites visibility is very bad. In this condition a segment can receive an high score and therefore can be classified as *selected segment* while

it never was covered during the survey (figure 3.24). But, on the other hand a segment that was covered can receive a very low score and doesn't become a *selected segment* (figure 3.25).

To avoid this kind of problems a new step was added to the segments selection procedure. At the end of the score allocation task, when the *candidate segments* group (see section 3.4.1) is completely defined, a segments reclassification is done. The purpose is to change score to the segments suspects to be erroneously added to the candidate group and also add segments that could be part of candidate group, but they were wrongly discarded.

The reclassification (see figure 3.26):

- reduces the score for handling segments, that are the segments that have only one vertex shared with others candidate segments (segment A) and therefore are suspected to have been erroneously selected (query 3.11);
- cover the holes: adds the segments not selected that have both the vertexes shared with others candidate segments (segment B) and therefore are suspected to have been erroneously discarded (query 3.12).

The reclassification procedure applied to the example in figure 3.24 reduces the score of the handling segment. This avoids that it is added in the *segments selected* group, as it possible to see in figure 3.27. The same procedure adds the segment excluded in figure 3.25 to the group of *selected segment*, figure 3.28.

---

**Query 3.11**

---

```
SELECT b.id
FROM candidate_path AS a, candidate_path AS b
WHERE ST_Touches(a.seg, b.seg) and
      a.id != b.id and a.id=$idsel and b.sel!=0;

/* where
   $idsel is the id of the selected segment as "candidate segment"
```

The ST\_Touches function permits to verify if two segment have at least one point in common \*/

---

---

**Query 3.12**

---

```
SELECT sum(b.dists) AS sumdst, count(b.dists) AS num
FROM candidate_path AS a, candidate_path AS b
WHERE ST_Touches(st_startpoint(a.seg), b.seg) and
      a.id != b.id and a.id=$idsel and b.sel=1";
```

```
SELECT sum(b.dists) AS sumdst, count(b.dists) AS num
FROM candidate_path AS a, candidate_path AS b
WHERE ST_Touches(st_endpoint(a.seg), b.seg) and
      a.id != b.id and a.id=$idsel and b.sel=1";
```

```
/* where
   $idnosel is the id of the segment not selected as
   "candidate segment" */
```

---





Figure 3.23 – An example of urban canyons: Cantù street, where the erroneous segment classification shown in figure 3.24 was verified

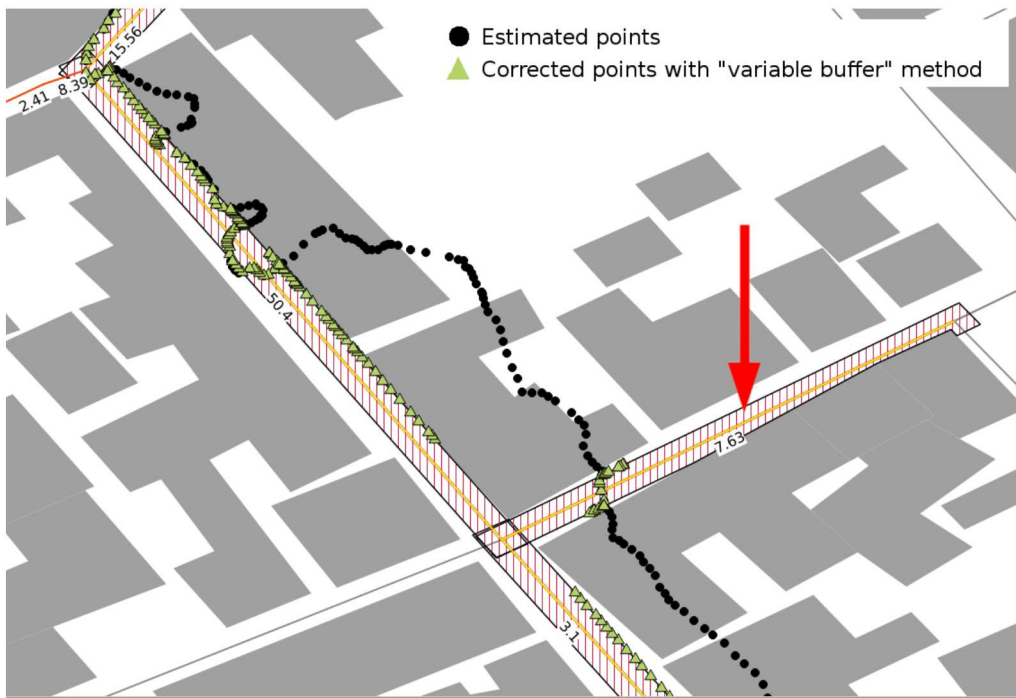


Figure 3.24 – Example of wrong classification for an handling segment: the segment indicated by the red arrow was added to the "selected segments" group, but it shouldn't be added - Day 14-259

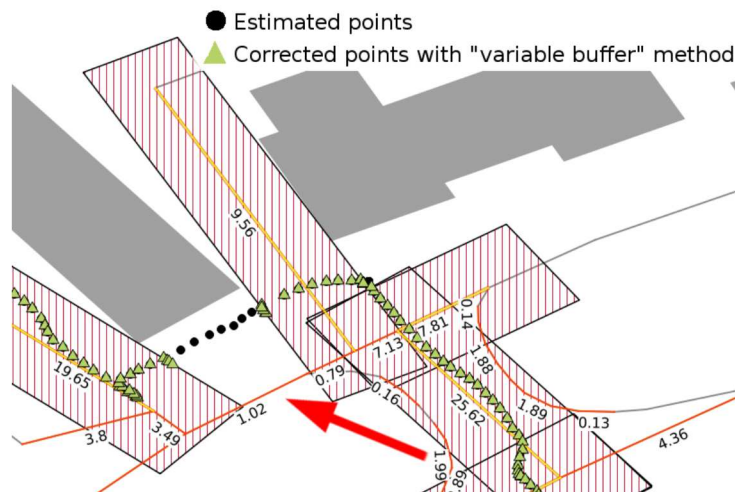


Figure 3.25 – Example of wrong classification: the segment indicated by the red arrow was not added to the "selected segments" group, but it should be added - Day 14-259



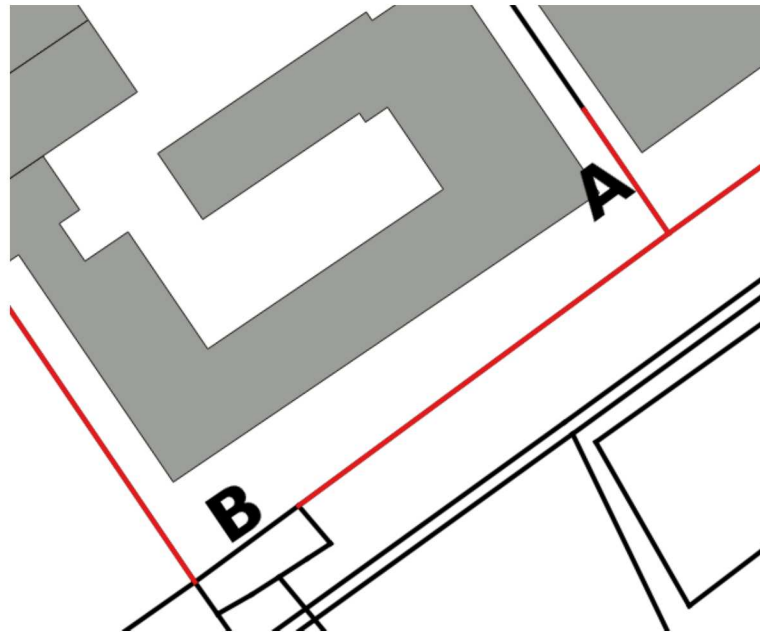


Figure 3.26 – Example of reclassification. After first classification step, the red segments are added to the candidate group, the black segments are the discarded. At the second step the score of segment A will be reduce and the segment B will be added to the candidate group

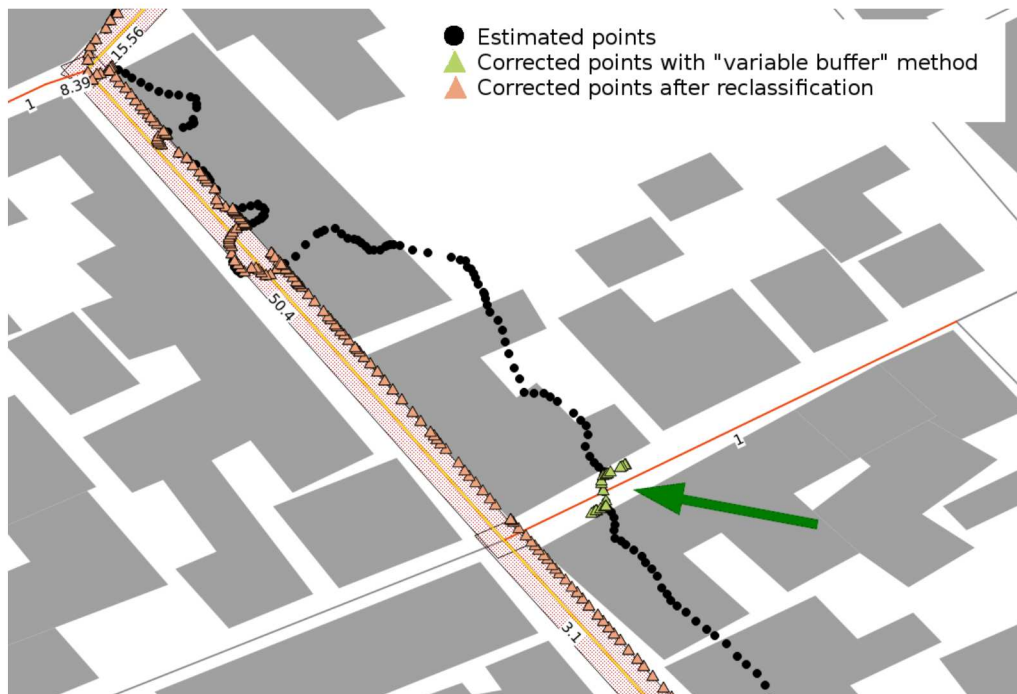


Figure 3.27 – The same path in figure 3.24, but now after the reclassification procedure: the wrong segment is not longer selected - Day 14-259

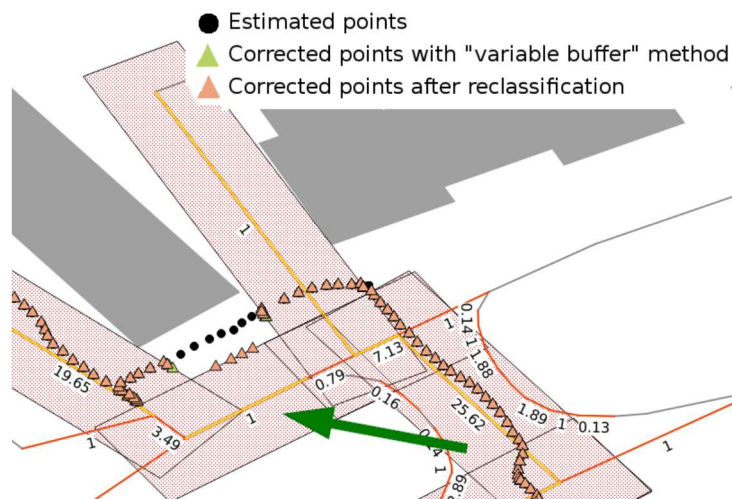


Figure 3.28 – The same path in figure 3.25, but now after the reclassification procedure: the missing segment is now selected - Day 14-259

### 3.5 Comparing results

The different correction procedures seen before were compared checking how many points, after correction procedure, are near to the actual path done, where the actual path is known and specifically defined for these tests. Two tolerance areas were defined around the actual path, wide 2.5 and 5 meters, then it was checking how many points were inside these two buffers.

In table 3.7 the results are reported. Let's consider the GPS day 14-259: the total number of estimated points is 5879, less than 70% of estimated points is a distance less than 5 meters and less than 45% are inside the 2.5 meters. When the correction methods are applied these percentages improve, so generally all the correction methods implemented are valid and improve the positioning accuracy.

If the tolerance area of 5 meters is considered, the best results will be obtained using the "path segments as reference" method (section 3.2), with more than 90% of points inside the tolerance area.

But if it is considered the smallest area (2.5 m), the "path segments as reference" method will become the worst correction method, with less than 50% of points inside the tolerance area when the "correction on buffer variable" method (sections 3.4.3 and 3.4.4) are near to the 60%.

This drastic deterioration, from the best to the worst method, is due precisely to the type of correction applied. As already mentioned, the survey is made by pedestrian users, the type of correction in many cases moves the points on the axis of main road, when the survey was made on the sidewalk or in a pedestrian path. In figure 3.29 is shown an example where this problem is very noticeable. In this figure "path as segments as reference" and "correction on buffer variable" methods are compared. The green line with the area in light blue around is the actual path with the tolerance area of 2.5 meters, the red lines are the roads (DB table highway.routes, see table 3.2) as it has been downloaded from OSM and then the black circles are the estimated points. The "path segments as reference" method moves all the points on the nearest red line (brown circles) when the "variable buffer" method live it at the original position because they are inside the buffer of tolerance (the area identified by the red vertical lines). The final result is that the points corrected on the segment are moved too much, going beyond the tolerance area of 2.5 meters and the "path as segments as reference" in these cases are worse than the "correction on buffer variable". Instead, if a tolerance area of 5 metres is considered (figure 3.30), also the points corrected with "path segments as reference" method will be inside (3.7).

If the other two repetitions (days 14-267 and 14-268) of survey along the Path 1 are considered, the general trend seen before is confirmed. The "path segments as reference" method will get worse more than the others when it is considered a tolerance area of 2.5 meters, also if the impact is less noticeable than seen in the first case: it is still the best correction method for the 14-167 day and the second best for the 14-268 day.

These are clearly a particular cases, in general the choice of correction method is not easy and is based on a delicate compromise and the risk to correct too much or too little is always present. The final results are very dependent on the conditions in which the survey was done, the site characteristic and from the

cartography used as reference. This last aspect is very important: an incomplete or wrong reference datum can lead to consider a point position wrong because far from any element on the map, when in fact the problem can be the map itself. For this reason it is important to have a cartography updated and easily upgradeable, like OSM, that permits to add new paths and correct possible mistakes.

Day & Total points	Tolerance areas		estimated data	correction method applied				
				path segments as reference	out of the buildings	on buffers		
						constant	variable	reclassified variable
14-259	5 m	No. of points inside	4030	5318	4384	4641	4876	4928
		%	68.5	90.5	74.6	78.9	82.9	83.8
5879	2.5 m	No. of points inside	2614	2896	2960	2944	3457	3496
		%	44.5	49.3	50.3	50.1	58.8	59.5
14-267	5 m	No. of points inside	3482	4474	3706	4147	4420	4452
		%	62.5	80.3	66.5	74.4	79.3	79.9
5571	2.5 m	No. of points inside	2004	2566	2166	2259	2552	2560
		%	36.0	46.1	38.9	40.5	45.8	46.0
14-268	5 m	No. of points inside	3381	4488	3668	3973	4189	4197
		%	66.4	88.1	72.0	78.0	82.2	82.4
5094	2.5 m	No. of points inside	2000	2633	2261	2282	2660	2663
		%	39.3	51.7	44.4	44.8	52.2	52.3

Table 3.7 – Compare between the different correction methods tested

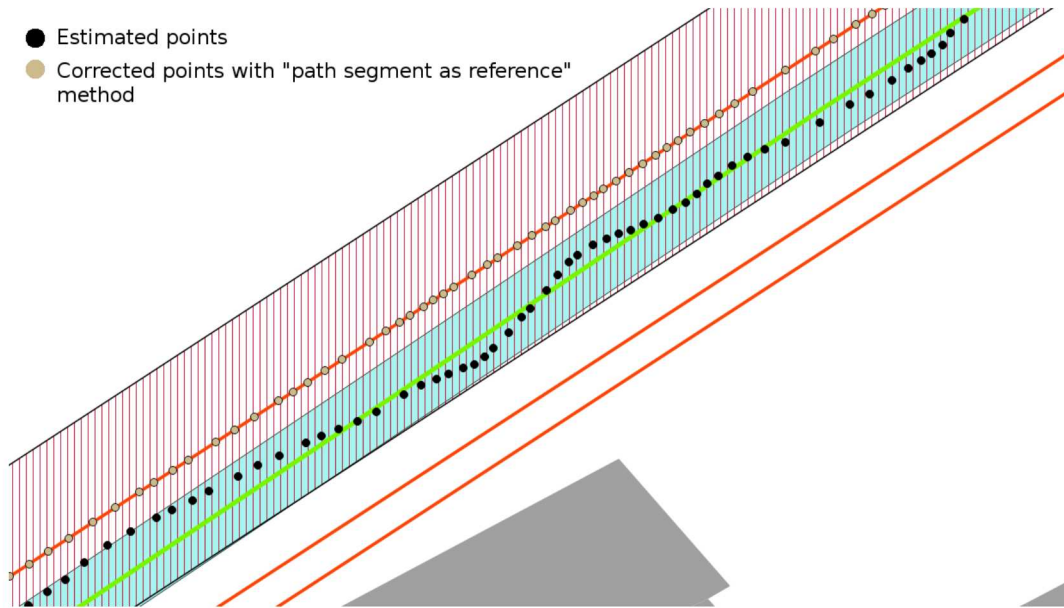


Figure 3.29 – Compare between "path segments as reference" and "correction on buffer variable" methods. The light blue area is the tolerance area of **2.5 meters** around the actual path (green line), while the red area is the buffer built around the segment path

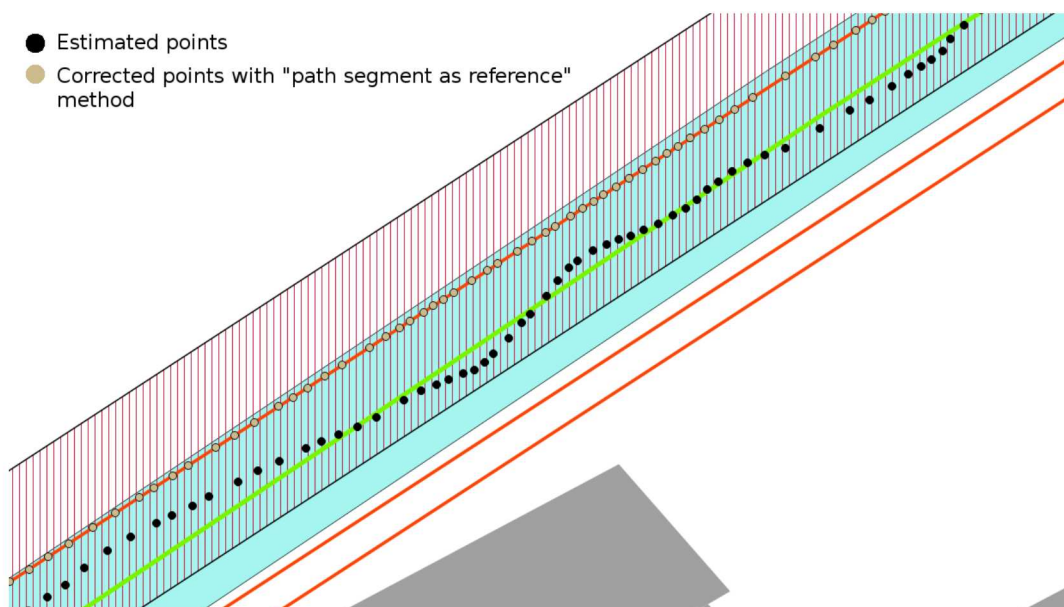


Figure 3.30 – Compare between "path segments as reference" and "correction on buffer variable" methods. The green area is the tolerance area of **5 meters** around the actual path, while the red area is the buffer built around the segment path

## Chapter 4

# Local monitoring: simulation of landslide movement

### 4.1 Introduction

The purpose of local monitoring is to closely model displacements, evolution and deformations of civil engineering structures and natural phenomena like for example landslides. In this chapter the focus will be on the landslides local phenomena, but the proposed methodology and results can be extended to many other applications of local monitoring.

A landslide is defined as "a movement of a mass of rock, earth or debris down a slope" [13]. Landslides are a serious geologic hazard: globally they cause an high number of human casualties and destructions that can be estimated to billions of Euro every year. Because of this an accurate and continuous monitoring of areas prone to landslides is desired. Different techniques are possible to do local monitoring, between these, in the last fifteen years a new technique emerged: the monitoring by Global Navigation Satellite System (GNSS) technology [19] and [31].

Typically in monitoring activities with GNSS technique the relative static processing is applied. As told in section 1.4 this technique exploits data from at least two GNSS receivers using the double differencing method applied to carrier phase observations to estimate the baseline between a reference receiver, in a known position, and a rover receiver, in an unknown position, (section 1.4.2). In local monitoring, one or more reference receivers are installed in stable locations outside the landslide and several rovers are put on the landslide, whose positions are estimated and monitored in time. Using dual frequency receivers in monitoring the final accuracy can be better than 1 cm as constantly observed in many studies (such as [1], [34] and [45]). Procedures to automatically process permanent networks are note and already implemented and tested ([6] and [7]): such approach can be applied also to process local monitoring networks and can provide a very accurate monitoring of the displacements in quasi real time.

Nevertheless, the high cost of dual frequency receivers (several thousand of Euro) can represent a problem because a great number of receivers can be needed, especially if the area must be monitored densely, therefore the initial set up of the control network can be very expensive. Moreover, in the case of a landslide event,

the receivers can be damaged or even lost and this clearly could discourage this monitoring technique in many cases. Accordingly, under an economic point of view, to do monitoring activities with this kind of receivers it is not always possible.

For these reasons, it could be interesting to verify if it is possible to do local monitoring with GNSS techniques, but using low cost receivers ([3] and [9]). The low cost receivers (single frequency<sup>1</sup> receivers, see section 2.3) are cheaper than dual frequency receivers, therefore monitoring should be possible thickly in a large area avoiding the budget problems mentioned just before, also in case of potential risk of loss of entire instrumentation. However their reliability in detecting displacement is not so good as that obtained with dual frequency receivers and then it can be interesting to verify what is the accuracy level achievable by low cost GNSS receivers and their effective use in local monitoring.

A time window of 15 - 60 minutes should be correct for such applications of local monitoring [24]: according to [32] it was decided to work with hourly sessions: this length of the sessions and the adopted sampling rate allowed a high number of available observations under any conditions.

Problems due to use of low cost antennas are expected, in particular regarding a more sensitivity to multipath and the non-availability of information about the Phase Center Variation (PCV).

A first experiment (section 4.2) was done using data from a static survey, followed by two other experiments where controlled displacements were imposed to the rover antenna: in the first case only horizontal displacements were imposed (section 4.4), in the second one both, horizontal and vertical displacements were imposed (section 4.5).

## 4.2 Experiment 1: static survey

The data used in this experiment were acquired during a survey made in Milan<sup>2</sup> from 14-073 to 14-079<sup>3</sup> and are the same used in section 2.5 for the tests with the CORS corrections. In this case raw data from u-blox were considered, instead of the estimated coordinates in real time as done in section 2.5.

The dataset is relevant to a short baseline (approximately 80 m long, see figure 4.1) formed by the permanent station (Topcon Odyssey RS) of Politecnico di Milano, Leonardo Campus and a u-blox EVK-4T<sup>4</sup>, with its standard antenna, used as rover. Both receivers were mounted on the roof of a building owned by Politecnico, at about 24 m above ground level: the acquisition site was already used in several previous geodetic tests and offers an optimal observation environment.

The comparison of two GNSS processing packages was a secondary task: the two chosen programs were Bernese GPS Software Version 5.2 (BSW)<sup>5</sup> from AIUB of University of Berne, and Leica Geo Office Version 8.3 (LGO)<sup>6</sup> from Leica Geosystems.

---

<sup>1</sup>only L1 frequency

<sup>2</sup>Thanks to Dr. Mirko Reguzzoni that provided these data

<sup>3</sup>from March 14 to March 20, 2014

<sup>4</sup>see appendix B.2

<sup>5</sup>see appendix A.1

<sup>6</sup>see appendix A.2



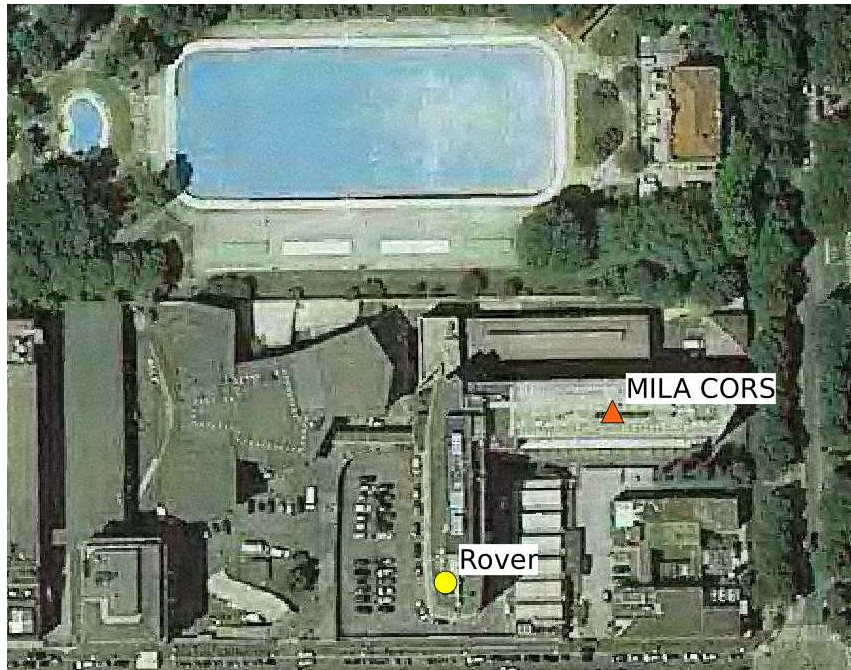


Figure 4.1 – The positions of Milan CORS (Red triangle) and the rover (yellow circle); the distance is less than 80 meters.

BSW [14] is one of the leading software for the international geodetic community. To optimally tune the processing options, the users should possess very good theoretical knowledge and experience in GNSS data processing. BSW was used as benchmark, to compare the results of LGO. The main processing parameters are reported in table 4.1. LGO is a commercial package that supports many surveying sensors and techniques, like for example GNSS, terrestrial positioning system and leveling data [30]. Where a choice is possible, the adopted processing parameters in LGO (table 4.1) were identical to those of BSW: in LGO, the ambiguity resolution algorithm is fixed and can not be chosen by the user.

Parameter	BSW	LGO
Observables	GPS L1 phase observations	GPS L1 phase observations
Differencing level	double	double
Session length	1 hour	1 hour
Troposphere model	Saastamoinen	Saastamoinen
Elevation cut-off	10 degrees	10 degrees
Ambiguity resolution	SIGMA	yes

Table 4.1 – BSW and LGO processing strategy used

Firstly, we processed the whole week with BSW to estimate a reference position of the rover. Then, the hourly sessions were processed to estimate the rover positions: the time series of the residuals relative to the reference position were finally computed (equation 4.1).

$$\delta \mathbf{x}_i = \mathbf{x}_i - \bar{\mathbf{x}} \quad (4.1)$$

where:

- $\bar{\mathbf{x}}$  is the reference position of the rover,
- $\mathbf{x}_i$  is the estimated position for session  $i$ ,
- $\delta\mathbf{x}_i$  is the vector of the residuals.

The residuals are presented as local East, North and Up respect to the reference position.

In total, 168 results were obtained. Five (3%) were clearly blundered: table 4.2 presents their residuals.

The statistics for the other sessions are presented in table 4.3. Overall the results are completely satisfactory: the standard deviation of the solutions is 2 mm for East, 5 mm for North, and 4.5 mm for Up.

Session	3D [cm]
22	24
24	44
30	9
40	19
142	31

Table 4.2 – Experiment 1. BSW residuals of blundered hourly sessions

Statistics	E [mm]	N [mm]	U [mm]
Mean	1.0	-1.5	0.8
Snd Dev	2.1	4.9	4.5
Min	-3.1	-10.8	-10.0
Max	16.8	47.5	14.3

Table 4.3 – Experiment 1. BSW hourly residuals of final solutions. E: East, N: North, U: Up

The same hourly sessions were processed with LGO. In this case, four blunders were identified: for these solutions, ambiguities were not resolved at all and led to large errors, with the maximum reaching 1.30 m (table 4.4). In table 4.5 the basic statistics of the other sessions are reported.

Session	3D [cm]
34	37
58	12
106	130
160	39

Table 4.4 – Experiment 1. LGO residuals of blundered hourly sessions

BSW and LGO results are similar. No bias exists between the two packages. LGO's standard deviation are slightly better in horizontal while vertical results of the two packages are similar. As expected, the vertical statistics are the worst: in this case, beside the usual reason (satellite geometry), this is probably caused also by the lack of antenna PCVs for u-blox.

Statistics	E [mm]	N [mm]	U [mm]
Mean	0.2	-1.3	1.9
Snd Dev	1.8	3.7	4.4
Min	-4.0	-36.3	-10.2
Max	4.1	3.3	14.5

Table 4.5 – Experiment 1. LGO hourly residuals. E: East, N: North, U: Up

For a more detailed comparison between BSW and LGO it was decided to count the sessions with absolute errors in the following classes: 0–5 mm, 5–15 mm, 15–30 mm and 30–50 mm (table 4.6). The results are satisfactory for monitoring applications. All of the residuals are smaller than 50 mm: the horizontal errors are well below 0.5 cm in most of the cases, especially for east with residuals under 5 mm for all sessions processed in LGO, and almost all sessions (99.4%) processed in BSW (see columns East in table 4.6). As expected, vertical residuals have worse statistics compared to the horizontal residuals, but are still satisfactory: 74.2% of the BSW results present residuals smaller than 5 mm, and the remaining 25.8% are between 5 and 15 mm. For LGO, 69.7% of the vertical residuals are smaller than 5 mm, and the remaining 30.3% are between 5 and 15 mm (see columns Up in table 4.6).

Moreover, in the processing of our dataset, a commercial and user-friendly software, like LGO, provides results that are completely consistent with those of BSW, with a slight degradation in the height estimates. In table 4.6 it can be seen that both horizontal and vertical residuals do not pass 15 mm for almost all session in the case of both software. There are only four exemptions: for BSW one session (0.6%) for east and two session for north, and for LGO only one session for north.

Finally, a graphical comparison between LGO and BSW residuals is presented in figures 4.2, fig:estRes and fig:hRes. For a better visual inspection, four residuals that exceed the interval [-2.0 cm, +2.0 cm] have been removed from the graphs: two for BSW and one for LGO. All the other residuals show similar magnitude and sign for almost all the sessions and a daily period appears, in particular for the height: this is probably caused by multipath, particularly significant for low cost antennas.

In conclusion, we have five (BSW) plus four (LGO) blundered results, which seems to be a relatively big number. These blunders are mainly caused by occasional problems in the data but probably also because of another reason: packages for post-processing of GNSS static sessions are optimized for data acquired by geodetic receivers and not for low cost receivers Other and more detailed results relevant to Milano test are discussed in [22].

Error class	BSW			LGO		
	E [%]	N [%]	U [%]	E [%]	N [%]	U [%]
0 – 5 mm	99.4	84.9	74.2	100	93.9	69.7
5 – 15 mm	0	13.9	25.8	0	5.5	30.3
15 – 30 mm	(*) 0.6	(*) 0.6	0	0	0	0
30 – 50 mm	0	(*) 0.6	0	0	(*) 0.6	0

Table 4.6 – Experiment 1. Classification of hourly residuals by percentages. E: East, N: North, U: Up.

Note (\*): only one session

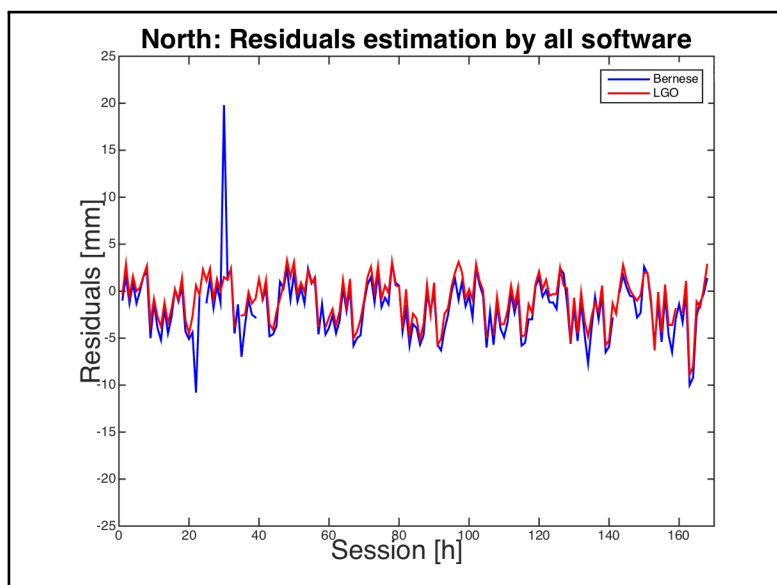


Figure 4.2 – Experiment 1. North residuals of hourly sessions. BSW in blue, LGO in red

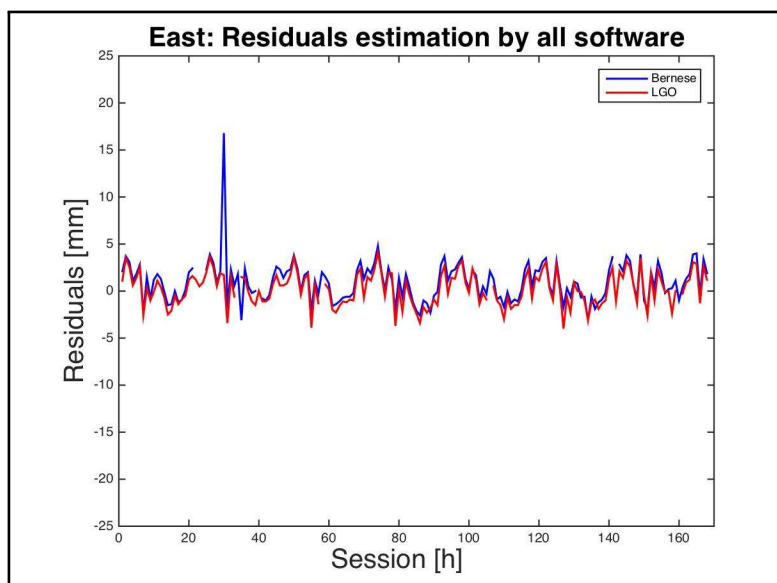


Figure 4.3 – Experiment 1. East residuals of hourly sessions. BSW in blue, LGO in red

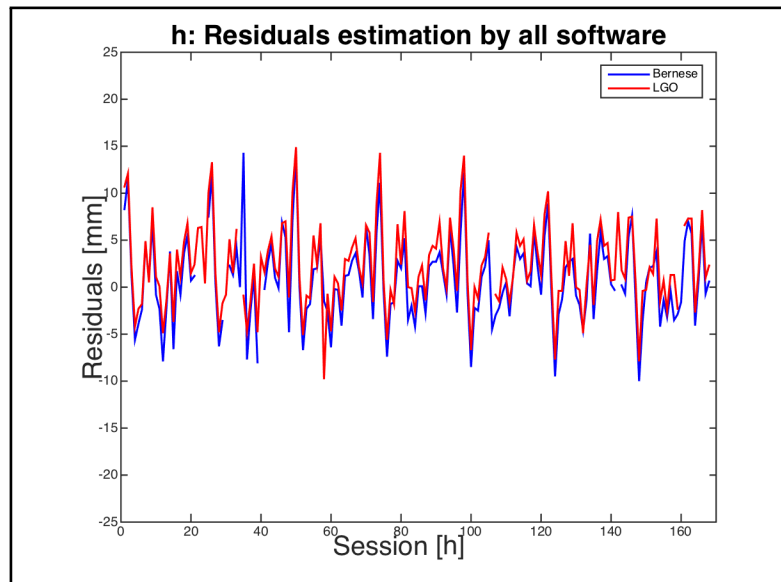


Figure 4.4 – Experiment 1. Vertical residuals of hourly sessions. BSW in blue, LGO in red

### 4.3 Analysis task in monitoring dynamic displacements experiments

Two different experiments were planned in order to assess the precision and the accuracy of low-cost GNSS receivers in monitoring dynamic displacements with low latency (Experiment 2, section 4.4 and Experiment 3, section 4.5). The tests were done using two tools to impose controlled displacements to the rover antenna. The aim was to evaluate which is the precision level reachable, comparing scheduled displacements with detected displacements obtained using low-cost GNSS receivers.

For each experiment was scheduled a sequence of antenna displacements in time and then imposed and estimated displacement were compared to evaluate the low cost receiver. Two type of analysis were done:

- **significance analysis:** how reliable are the results obtained or rather, how many false or missed alarms are expected per unit of time;
- **congruence analysis:** the ability to identify at least the order of magnitude of a displacement (congruence).

#### 4.3.1 Significance analysis

The aim of significance analysis is to assess how many imposed displacements are identified by a significance test on the estimated displacements. A time series of coordinates is usually checked for displacements by a Fisher test [38]. The estimated coordinates of different sessions are assumed independent and that follow a Gaussian distribution. Given two sessions  $i$  and  $j$ , the null hypothesis is:

$$H_0 : \mathbf{x}_i = \mathbf{x}_j \quad (4.2)$$

If  $H_0$  is true, then

$$\frac{\delta \mathbf{x}_{ij}^T C_{\delta\delta_{ij}}^{-1} \delta \mathbf{x}_{ij}}{m} = F_{exp} \sim F_{m,(r_i+r_j)} \quad (4.3)$$

where  $C_{\delta\delta}$  is the covariance matrix of the differences,  $r_i$  and  $r_j$  are the redundancies of the two data adjustments,  $m$  the dimension of the coordinates vector.

A threshold  $F_t(\alpha)$  is computed for the Fisher function, based on  $m, (r_i + r_j)$  and a given significance level  $\alpha$ .

If  $F_{exp} \leq F_t$  the null hypothesis is accepted as true: the coordinates did not change between the two sessions. In the contrary case, the null hypothesis is not accepted: the coordinates changed, therefore a displacement is identified.

#### 4.3.2 Congruence analysis

An another very important step is to evaluate if with a low cost GNSS receivers it will be possible to detect correctly the shift trend during the test, therefore if the estimated coordinates in every session will be congruent with the imposed displacements.

The congruence analysis is used just to verify how many estimated displacements between couples of sessions are congruent with the imposed ones.

This analysis uses a Bayesian approach [5] to define classification errors  $\epsilon_\omega$  of estimated displacements. Let's define  $\omega_k$  the class corresponding to imposed displacement  $\delta x_k$  where in the experiments that follow the single displacement is equal to 5 mm, then:  $\delta x_k = k \times 5mm, k = 0, \dots, N_{max}$ . Let's suppose that the probability distribution of estimated  $\delta x_{ij}$  conditioned to class  $\omega_k$  is normal with constant average  $\overline{\delta x_k}$  and covariance  $\mathbf{C}$ : with the Bayesian approach it is straightforward to classify each estimated  $\delta x_{ij}$  in the most probable class  $\omega_k$ .

The analysis is performed in three steps:

1. for each class  $\omega_k$  the conditional probabilities  $P(\omega_k/\delta x_{ij})$  are computed;
2.  $\delta x_{ij}$  is attributed to the class  $\omega_k$  with the maximum  $P(\omega_k/\delta x_{ij})$ ;
3. the class error is computed according to:

$$\epsilon = abs(\omega_k - \overline{\omega_{ij}}) \quad (4.4)$$

where  $\overline{\omega_{ij}}$  is the true class corresponding to the imposed  $\overline{\delta x_{ij}}$ .

## 4.4 Experiment 2: only horizontal displacements

### 4.4.1 Set up and survey

For the following experiments (Experiment 2 and Experiment 3, section 4.5) we chose to use LGO<sup>7</sup>, for which the results reliability was verified in the Experiment 1 (section 4.2) where it was compared with BSW, benchmark in this kind of applications, and another software, rtkLIB<sup>8</sup>, an open source software.

In the Experiment 2 only horizontal controlled displacements were imposed. Two point of survey, distant just under 100 meters, were set: a reference station and a controlled point where the low cost receiver was placed (Figure 4.5).

As reference station (figure 4.6) COMO TEST CORS was used with geodetic instrumentation: antenna Leica AR25 and receiver Leica GR10.

As low cost receiver it was used a u-blox NEO 7P (evaluation kit)<sup>9</sup> with its standard antenna (the yellow circle in figure 4.7) mounted on a simple device (figure 4.8) that permits to impose controlled movement in horizontal direction. The u-blox was prepared as shown in section 2.5, inside a waterproof box (figure 2.11) and connected a laptop via USB to be powered and to save data into PC. The laptop was put inside a rack to be powered and protected (figure 2.9).

A geodetic receiver<sup>10</sup> was placed near the rover to verify that there weren't other effects to affect the survey (the red triangle in figure 4.7).

A two days of static acquisition, with a geodetic receiver, was done to compute the rover start position, then the experiment started on GPS Day 15-152<sup>11</sup>.

The antenna was shifted horizontally 5 mm for each session along the baseline direction. Three sessions per day were done, with one hour per session, for a total displacement aimed of 10 cm, but the test was stopped at 5.5 cm due to an instrumental problem.

<sup>7</sup>see appendix A.2

<sup>8</sup>see appendix A.4

<sup>9</sup>see appendix B.2

<sup>10</sup>Leica GX 1230, see appendix B.1

<sup>11</sup>June 1th, 2015



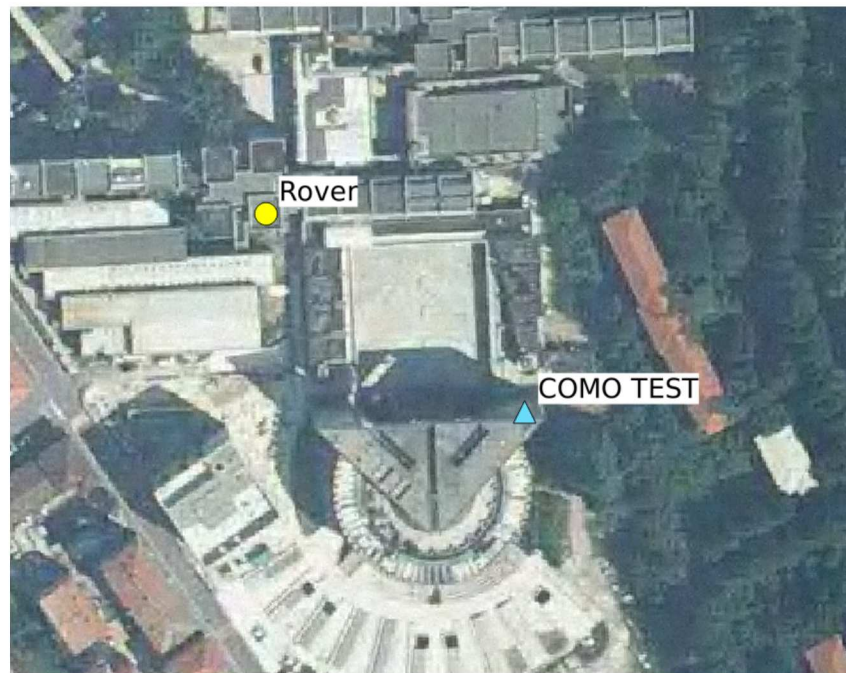


Figure 4.5 – Experiment 2: the positions of reference station (light blue triangle) and the rover (yellow circle)



Figure 4.6 – The reference stations in Experiment 2: Como Test CORS (light blue triangle)

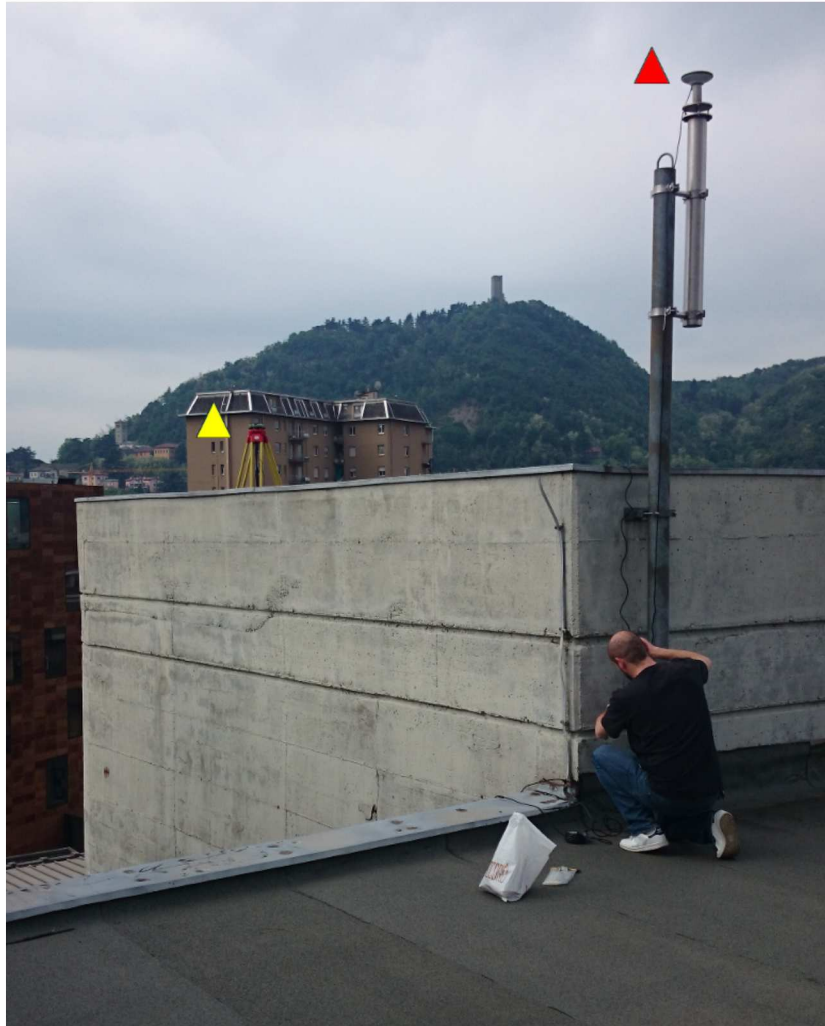


Figure 4.7 – The rover antenna in Experiment 2



Figure 4.8 – The horizontal sliding device used in Experiment 2

#### 4.4.2 Processing and data analysis

Data were processed with LGO<sup>12</sup> and rtkLIB<sup>13</sup>, to compare two different software, developed using two different approaches, one close source, developed inside a private enterprise and the other one on an open environment, with widespread support from many developers and open source. Data were processed using double difference combination, the strategy used in both software for this experiment is reported in table 4.7.

In figures 4.9 and 4.10 are plotted the results obtained after the processing using both software, where it is shown how the imposed displacements (blue line) along the horizontal component are generally recognizable by the low cost GPS receiver (red line).

Then the congruence analysis (section 4.3.2) was applied: in tables 4.8 and 4.9 are reported the results obtained with LGO and RTKLIB respectively. Regarding the horizontal displacement, the most couples are in the first three classes, with slightly less than half in class 0, that means that in these cases the movement was correctly detected, with no errors. If are considered the displacement along the North and East components the results are a little worse, with the north component that presents the worst results.

In any case the final results are good and satisfactory as general indication that can be possible detect the general trend of displacements with low cost receivers: this encourage us to the set up a new experiment to do more deep analysis about the effective use of low cost receivers in local monitoring.

Parameter	Value
Observables	GPS L1
Ephemeris	broadcast
Ionospere model	Klobuchar
Troposphere model	Saastamoinen
Elevation cut-off	10 degrees

Table 4.7 – Processing strategy used

LGO	East		North		horizontal	
class 0	180	32.1%	104	18.5%	246	43.9%
class 1	277	49.4%	174	31.0%	274	48.8%
class 2	86	15.3%	155	27.6%	36	6.4%
class 3	18	3.2%	71	12.7%	5	0.9%
class 4	0	0	44	7.8%	0	0
class 5	0	0	10	1.8%	0	0
class 6	0	0	3	0.5%	0	0

Table 4.8 – Congruence analysis: data processed with LGO

<sup>12</sup>appendix A.2

<sup>13</sup>appendix A.4

RTKLIB	East		North		horizontal	
class 0	215	38.3%	140	25.0%	261	46.5%
class 1	279	49.3%	226	40.3%	275	49.0%
class 2	65	11.6%	110	19.6%	25	4.5%
class 3	2	0.4%	61	10.9%	0	0
class 4	0	0	19	3.4%	0	0
class 5	0	0	3	0.5%	0	0
class 6	0	0	2	0.4%	0	0

Table 4.9 – Congruence analysis: data processed with rtkLIB

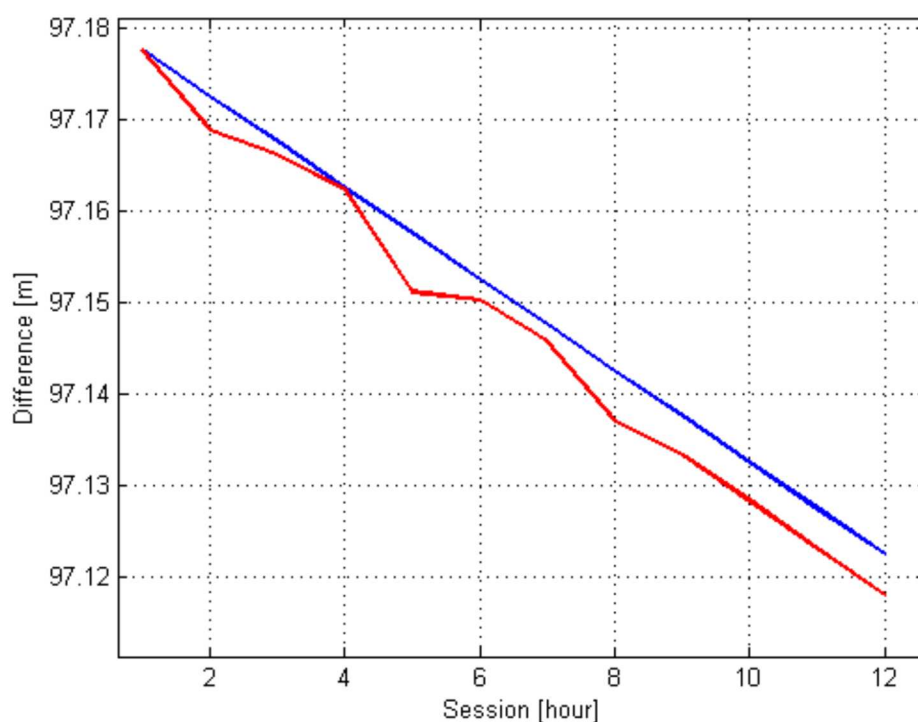


Figure 4.9 – Data processed with LGO software: comparison between imposed (blue) and estimated (red)

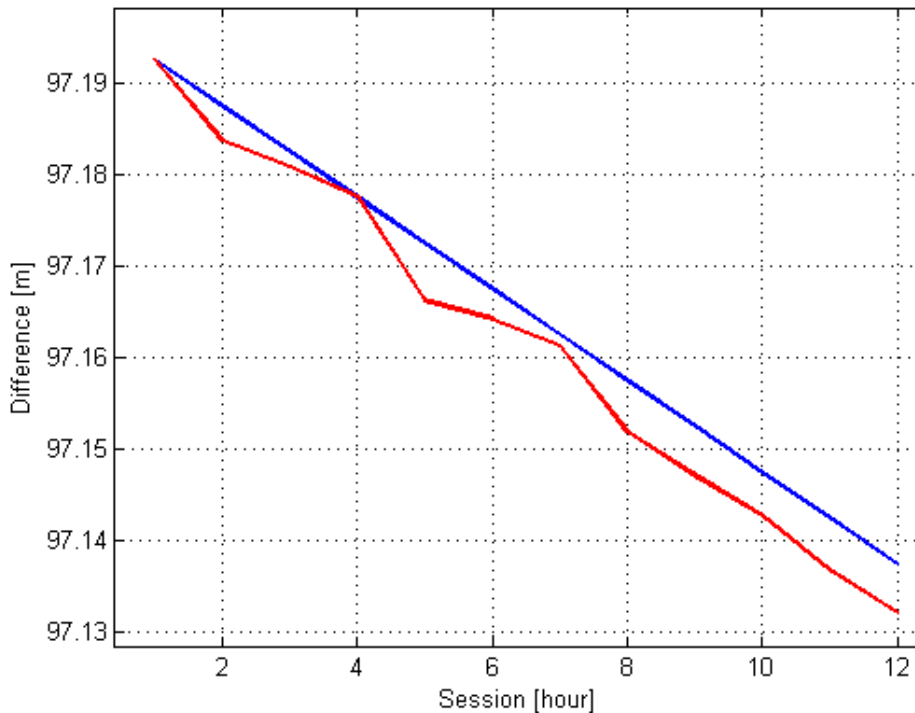


Figure 4.10 – Data processed with RTKLIB software: comparison between imposed (blue) and estimated (red)

## 4.5 Experiment 3: horizontal and vertical displacements

### 4.5.1 Set up and survey

First experiment gave promising result and then a new experiment was set up to take into account two more features:

- vertical displacement: a different sliding device was used to impose displacements also in this direction;
- use of single frequency receivers as reference station.

In the Experiment 2 (section 4.4) the monitoring was done using geodetic receivers as reference stations while low cost receivers are installed on the monitored area: this it is already cost effective, because one reference station could suffice and it does not risk any damage or loss. Another possible schema implies the installation of low cost receivers both in reference and in monitored sites and is even more cost effective. Therefore, a comparison between these two alternatives is interesting and for that in this second experiment it was chosen to set up a second reference station using an u-blox.

Three points of survey were set up: two reference stations (just a few meters from each other) on the roof of Valleggio building and a monitored point on the roof of Castelnuovo building, where the rover was put on a sliding device. The distance between reference stations and rover was about 130 meters (see figure 4.11).

The reference stations were:



- the CORS COMO<sup>14</sup>, the green triangle, named Valleggio 1 in figure 4.11, used also in the experiment 1 (section 4.4), with the geodetic instrumentation already mentioned: antenna TPSCR3 GGD and receiver Topcon Odyssey RS;
- the low cost reference station composed by a receiver u-blox NEO 7P<sup>15</sup> with the antenna Tallysman TW3152<sup>16</sup>, distant about ten meters from COMO EUREF, the blue triangle named Valleggio 2 in figure 4.11.

To control horizontal and vertical displacements of rover on the survey point it was used a special instrument (see in figure 4.12) that allows to impose accurate displacements of the receiver antenna. This sliding device was designed and built at "Dipartimento di Ingegneria dell'Ambiente, del Territorio e delle Infrastrutture (DIATI)<sup>17</sup>", Politecnico di Torino<sup>18</sup> and it is composed of a calibrated hardened steel bar (X axis), with a special support for the antenna, on a vertical steel bar (Y axis). The device has two graduated rulers, one for horizontal and one for vertical movements, which allow the user to control the imposed displacements. The maximum displacements are 1.30 m in horizontal and 30 cm in vertical direction. In the figure 4.12 the red arrows indicate the directions of movement during the test. To measure the controlled movements on the sliding device (the brown circle in figure 4.11) it was used a u-blox NEO 7P like in the second reference station: in this case its default antenna was used.

A total station was used to orient the sliding device in North-East direction (figure 4.13), to permit to easily evaluate the effects not only along the movement direction, but also along the North and East components.

In figure 4.14 it is possible to see the antennas of the the two reference stations placed on Valleggio roof; on background it is marked with a brown circle the point where it was placed the sliding device. In figure 4.15 instead it is shown the Castelnuovo roof, with the sliding device with the rover antenna mounted on top and the box with inside the u-blox; on background it is possible to see the Valleggio building with marked (with two triangles) the positions of the two reference stations. The u-blox was prepared as shown in previous section, inside a waterproof box and connected to a laptop to be powered and to collect data.

Initial position of the rover was determined using double difference observables (section 1.4.2) by a 24 hour session with a geodetic receiver<sup>19</sup>, estimated respect COMO EUREF station. At the end of the experiment the antenna was placed again in the start position and another 24 hours session was made to verify possible movements. Indeed, even if only slightly, the first and the second estimates were different between that estimated at the beginning: 2 mm level in horizontal plane and 6 mm in height. As result of this, the start position was computed as mathematical average of these two sets of coordinates.

---

<sup>14</sup>It is part of the EUREF network from 2004:  
[http://epncb.oma.be/\\_networkdata/siteinfo4onestation.php?station=COMO00ITA](http://epncb.oma.be/_networkdata/siteinfo4onestation.php?station=COMO00ITA)

<sup>15</sup>Appendix B.2

<sup>16</sup>Many thanks to Stefano Caldera of Geomatics Research & Development (GR&D) that provided this enhanced antenna

<sup>17</sup>Departement of Environment, Land and Infrastructure Engineering

<sup>18</sup>Many thanks to the colleagues Professors Manzino, Piras and Engineer Bendea

<sup>19</sup>Leica GX 1230 - Appendix B.1



Figure 4.11 – Experiment 3: the positions of reference stations (green and blue triangles) and the rover (red circle)

The test was performed from 28-16 to 42-16<sup>20</sup>: from the start position the antenna was moved, alternately, of 5 mm in horizontal and 5 mm down in vertical at leaving at least 2 hours between consecutive movements. The total movement reached 10 cm, both in vertical and horizontal directions; the data were acquired at 1 Hz of frequency<sup>21</sup>. During the test a geodetic receiver (figure 4.16) was used to verify that the measures were not affected by other events: data from this receiver were processed at the begin and at the end of the experiment to verify eventual displacements of the roof and no difference was detected.



Figure 4.12 – The sliding device used in Experiment 3

<sup>20</sup>from 28th of January to 11th of February 2016

<sup>21</sup>one new datum every second



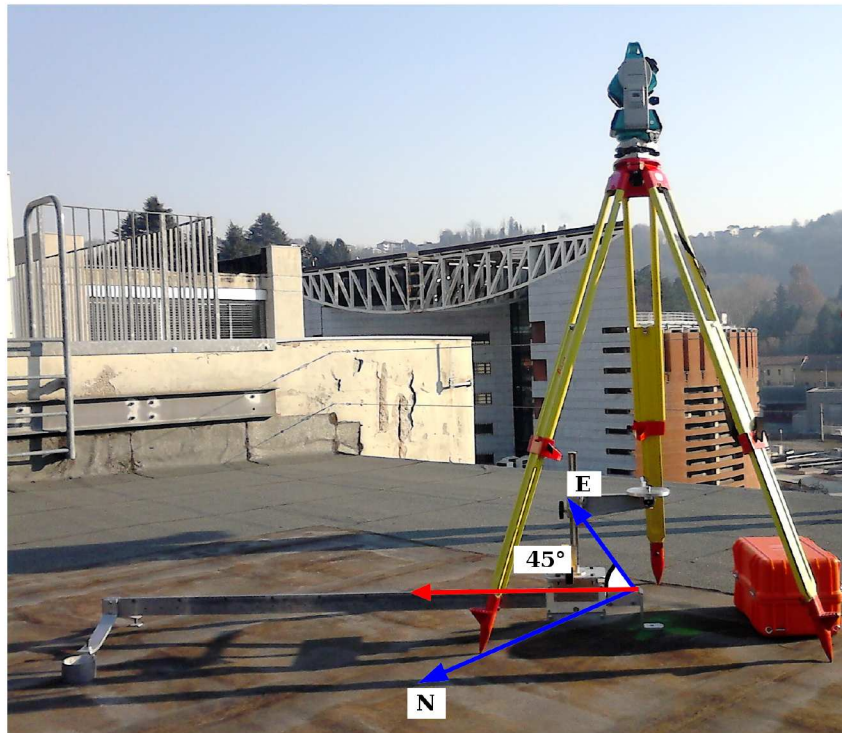


Figure 4.13 – Orientation task with total station



Figure 4.14 – The reference stations in Experiment 3: Como EUREF (green triangle) and secondary reference station (blu triangle); the orange circle is in correspondence of rover position



Figure 4.15 – The sliding device on Castelnuovo roof. In the upper left corner it is possible to see the position of reference stations (blue and green triangles)



Figure 4.16 – Monitoring the roof stability with another geodetic receiver

### 4.5.2 Processing and data analysis

The data from the rover were processed with LGO<sup>22</sup> and rtkLIB<sup>23</sup> two times: once using *Valleggio 1* as reference station, the other one using *Valleggio 2*. With rtkLIB we had some problems with a low number of fixed ambiguities, therefore the results obtained were not comparable with results obtained with LGO and for that they will not be reported.

In figure 4.10 are reported the processing parameters set in LGO during the processing phase on the double difference observation.

Parameter	Value
Observables	GPS L1 phase observations
Fix Ambiguities	yes
Differencing level	double
Ephemeris	broadcast
Ionosphere	Klobuchar
Troposphere	Saastamoinen
Elevation cut-off	10 degrees

Table 4.10 – Processing parameters chosen in LGO

A good session was one in which the ambiguity fixing was carried out successfully and the residuals for each of the component of the 3-D position was no more than 5 cm. This value was arbitrarily chosen based on expected accuracy of low-cost GNSS sensors, on accuracy required by landslide monitoring and lack of antenna calibration information. The antenna was moved every two hours, alternatively in horizontal and vertical directions: for every horizontal/vertical position there are four different measurement sessions. In all 80 sessions of one hour each were processed. Data were processed two times, one using COMO Euref (*Valleggio 1*) station as reference and another time using the low cost receiver (*Valleggio 2*) as reference station: in the figures from 4.17 to 4.24 the graphs with the comparisons between actual and estimated displacement are shown. The general displacement trend is followed in all the cases: along the horizontal direction (figures 4.17 and 4.21) the results are the best, with the estimated points very close to the actuals.

If the displacements are decomposed along North and East directions (figures 4.18, 4.19 and figures 4.22, 4.23), it will be possible to see some greater oscillation, but in any case, each single step is quite clearly recognizable. The worse results are obtained along the vertical direction (figures 4.20 and 4.24): the general trend is recognizable, but the single movements are almost impossible to be identified. This is what was expected: working with a single frequency receiver, with a very cheap antenna, leads a general problem of accuracy, that it is more evident along the vertical direction that is not well recognized.

In table 4.11 are shown the statistic of the processed data with both the reference stations, *Valleggio 1* and *Valleggio 2*. Values obtained with *Valleggio 1* (geodetic receiver), as expected, are better, but in any case also the results after the processing with *Valleggio 2* are good: the mean of errors is under 0.5 cm, only the

<sup>22</sup>Appendix A.2

<sup>23</sup>Appendix A.4

vertical component with Valleggio 2 it is slightly above, but the problems with vertical component are note. Horizontal standard deviations are around 0.5 cm on Valleggio 1 and a bit worse on Valleggio 2, but in any case under 1 cm. These results are good and they suggest the effective application with successful of low cost receivers to monitoring activities.

However the statistics in table 4.11 are not enough to full evaluate the low cost receivers goodness: significance analysis and congruence analysis have been applied to the data to verify if individual displacements were detected by the survey system.

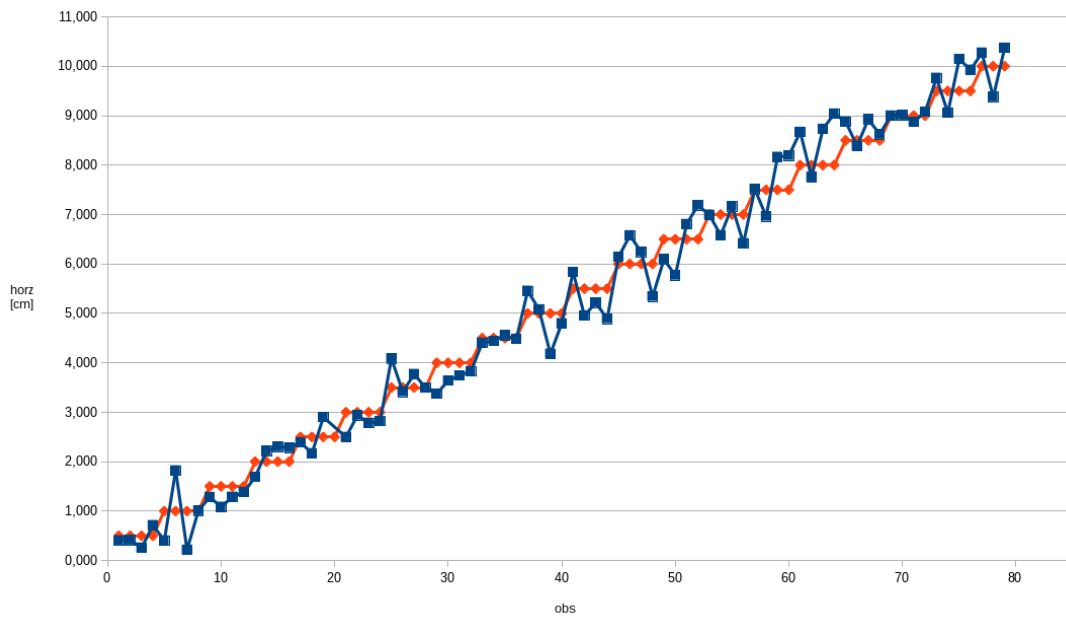


Figure 4.17 – Horizontal displacements: comparison between actual (red) and estimated (blue), Valleggio 1 as reference

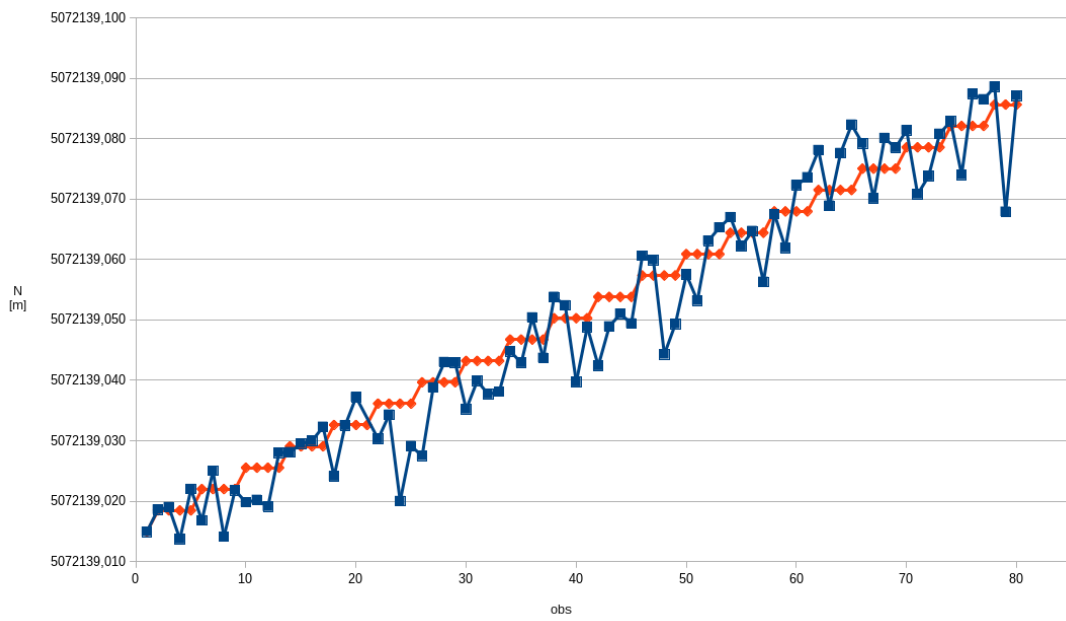


Figure 4.18 – North component displacements: comparison between actual (red) and estimated (blue), Valleggio 1 as reference

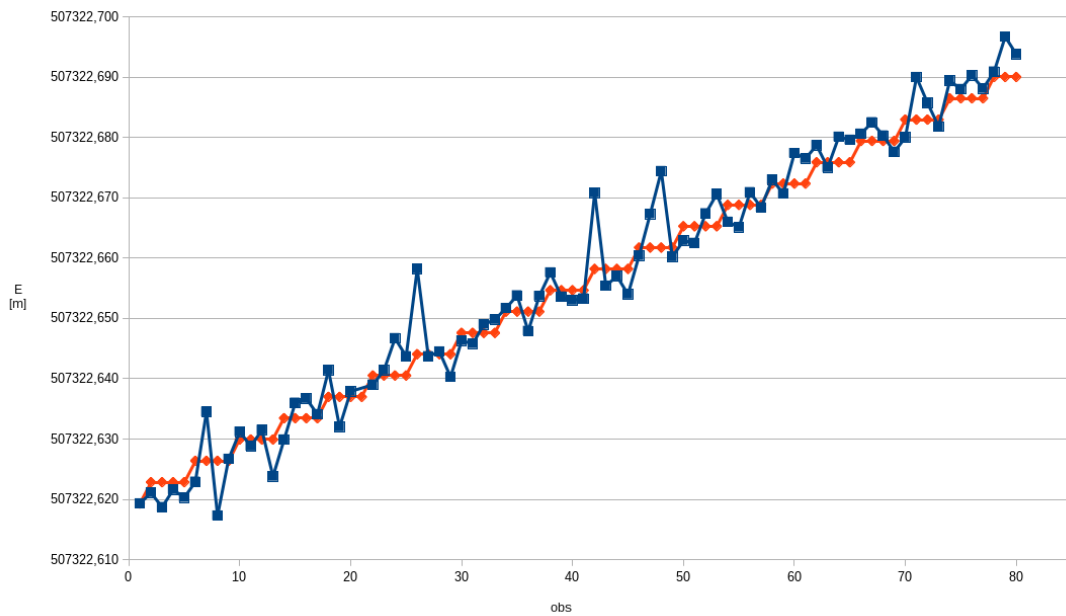


Figure 4.19 – East component displacements: comparison between actual (red) and estimated (blue), Valleggio 1 as reference



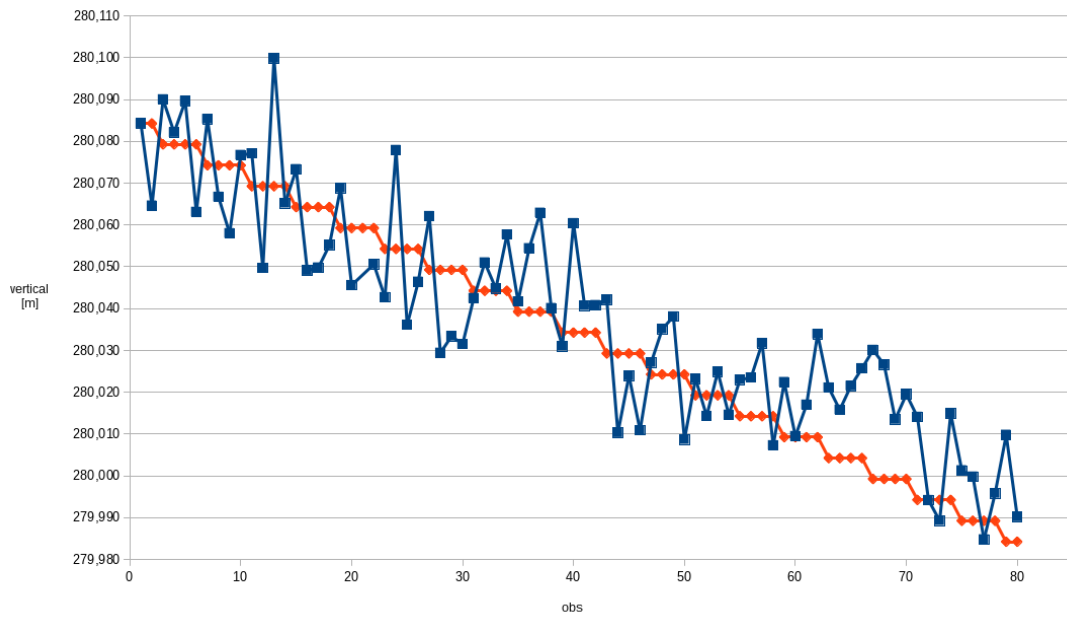


Figure 4.20 – Vertical displacements: comparison between actual (red) and estimated (blue), Valleggio 1 as reference

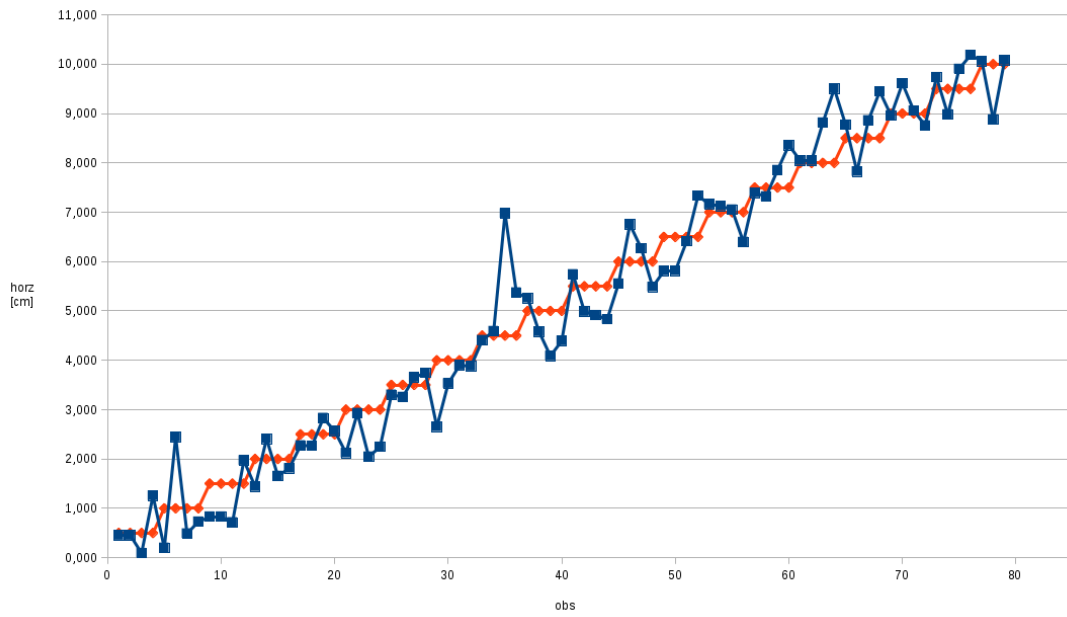


Figure 4.21 – Horizontal displacements: comparison between actual (red) and estimated (blue), Valleggio 2 as reference

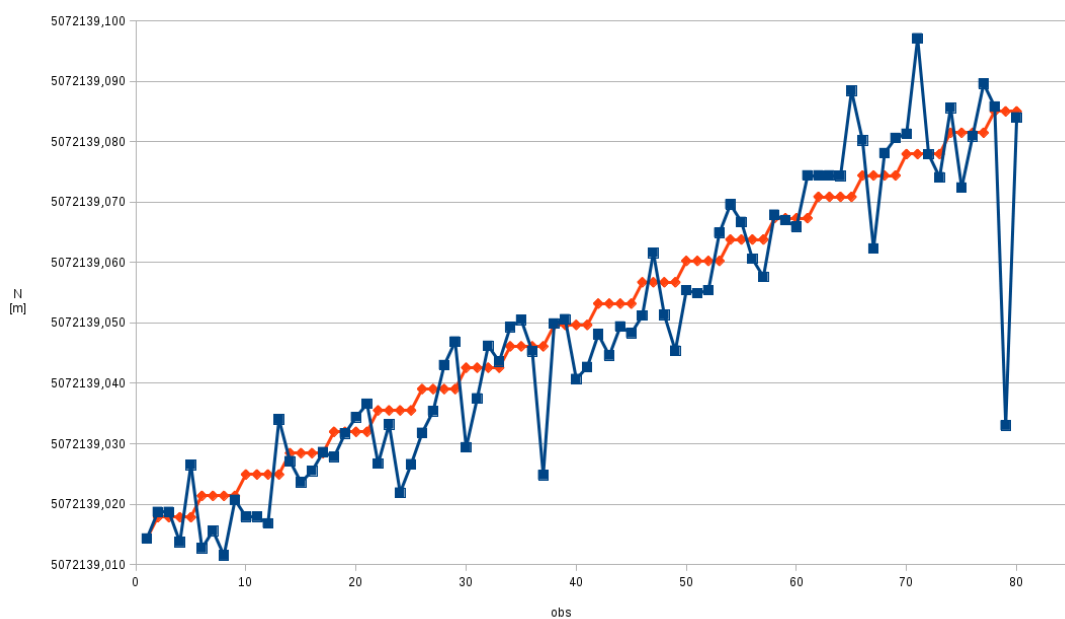


Figure 4.22 – North component displacements: comparison between actual (red) and estimated (blue), Valleggio 2 as reference

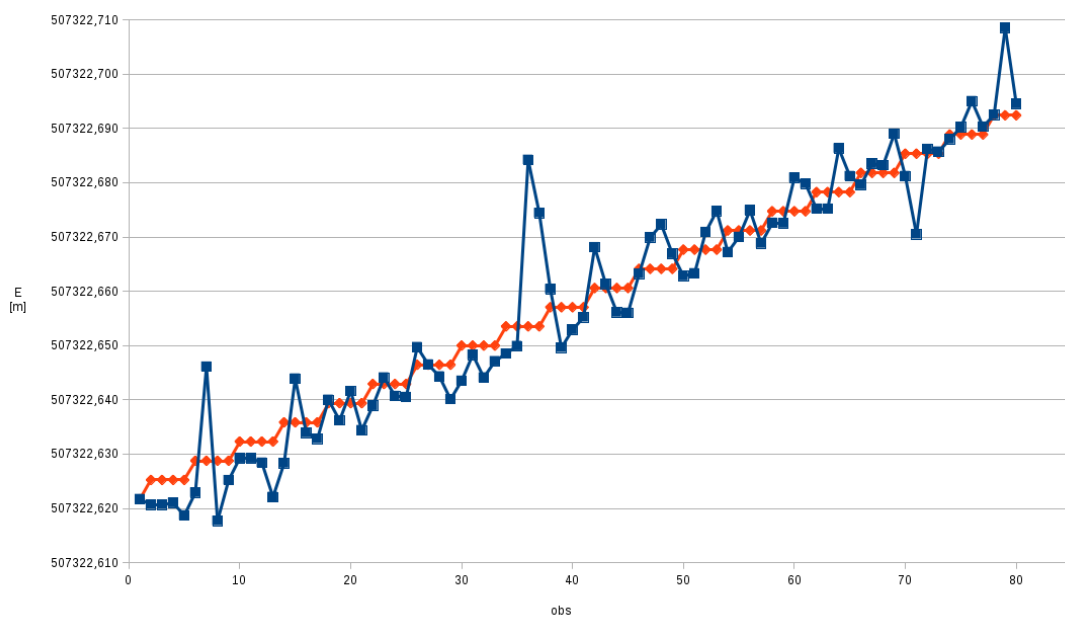


Figure 4.23 – East component displacements: comparison between actual (red) and estimated (blue), Valleggio 2 as reference



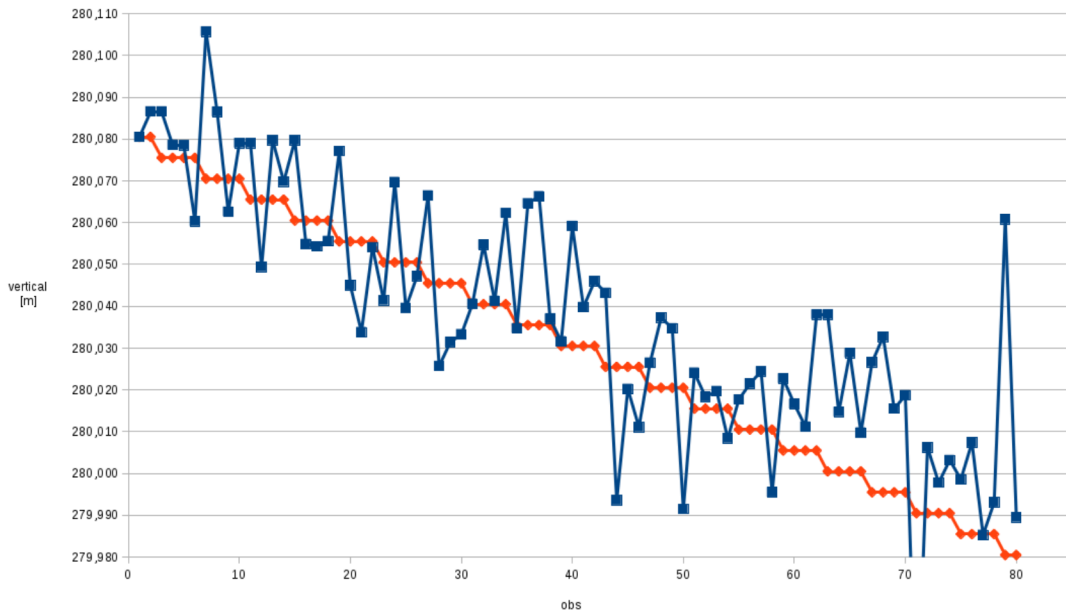


Figure 4.24 – Vertical displacements: comparison between actual (red) and estimated (blue), Valleggio 2 as reference

	Valleggio 1				Valleggio 2			
[cm]	E	N	vert	horz	E	N	vert	horz
mean	0.1	-0.2	0.3	0.0	0.0	-0.2	-0.6	-0.1
std dev	0.4	0.6	1.3	0.4	0.7	0.9	1.8	0.7
max	1.4	1.1	2.0	1.0	3.1	1.9	5.7	2.4
min	-0.9	-1.8	-2.1	-0.8	-1.4	-5.2	-7.5	-2.5
median	0.1	-0.1	0.4	0.0	-0.2	-0.1	-0.7	-0.1

Table 4.11 – Processed data: statistics and comparison between results obtained using two different reference station, Valleggio 1 and Valleggio 2

### Significance analysis

Significance analysis (section 4.3.1) has been applied to the data collected putting  $\alpha = 0.05$ . Let be  $i$  and  $j$  two sessions that are compared, let be  $n$  is the number of movements between  $i$  and  $j$ . First of all, the test is performed for the horizontal (2D,  $m=2$ : E, N) and vertical displacements ( $m=1$ : U): in these cases,  $i$  and  $j$  are separated by imposed displacements of  $n \times 5mm$ . Then, the test is performed for the three dimensional displacements (3D,  $m=3$ : E, N, U): in this case, the imposed displacement between  $i$  and  $j$  is  $\delta = n \times \sqrt{2} \times 5mm$  (the composition of  $n$  horizontal and vertical displacements of 5 mm). The test was applied to couples of imposed displacements up to  $n = 5$  movements. The percentages of right answers are shown in table 4.12 as it regards the data processed using Valleggio 1 as reference station and in table 4.13 as it regards the data processed using

Valleggio 2. Note that in case 0 (no movement imposed) the right hypothesis is "no displacement". In the other cases (one or more movements imposed to the rover) the right hypothesis is "yes displacement".

Let's consider horizontal movement in table 4.12: after 3 steps the majority of displacements are identified; after 5 movements, the displacement is identified almost in the whole dataset. Similar statistics are provided by the analysis of three dimensional displacements.

Let's consider now data in table 4.13: these results are worse than those in table 4.12, in this case are needed 4 step to identify the majority of displacement in 2D and 3D instead than three, but in any case they are still satisfactory.

As expected, vertical results are worse. Firstly this is due to the worse accuracy of GNSS vertical estimates; moreover, the significance test in 4.2 strengthens at the increase of  $m$ : therefore, the displacement test on one component is intrinsically weaker than the 2D and 3D cases.

Movements between couples	2D	UP	3D
0	78 %	85 %	80 %
1	20 %	14 %	28 %
2	29 %	17 %	46 %
3	63 %	28 %	68 %
4	88 %	31 %	90 %
5	99 %	38 %	100 %

Table 4.12 – Significance analysis on estimated movement, data processed using Valleggio 1 as reference station. Percentage of right answers

Movements between couples	2D	UP	3D
0	88 %	84 %	92 %
1	14 %	14 %	14 %
2	23 %	18 %	21 %
3	31 %	27 %	31 %
4	64 %	33 %	61 %
5	84 %	36 %	88 %

Table 4.13 – Significance analysis on estimated movement, data processed using Valleggio 2 as reference station. Percentage of right answers

### Congruence analysis

Congruence analysis (section 4.3.2) has been applied to verified if the difference between estimated displacement and actual displacement are significant and such as to determined the trend of the displacement made:

1. displacements are estimated considering pairs of sessions;
2. comparison between estimated and actual displacements;
3. assignment to each pair of a class based on the error, following the formula 4.4 where  $\Delta_{step}$  is:

- equal to 5 mm for displacement along horizontal and vertical components;
- equal to  $5/\sqrt{2} = 3.5mm$  for displacement in north, east components.

The analysis has been done considering all possible pairs. In the figures 4.25 and 4.26 the graphs with the results of are shown: it is clearly visible how the most of couples are classified in the lower three classes for horizontal (purple bar), north (red bar) and east (blue bar) displacements, instead couples regarding vertical displacement (green bar) are over the 10th class. This confirm the first results viewed in table 4.11, where biggest errors were on vertical component. Analytic results can be viewed in the tables 5.9 and 5.10, where the numeric values for each class are reported. In the first classes there are more than 70% of couple of data processing with Valleggio 1 (table 4.16) and Valleggio 2 (table 4.17) as reference station. In these classes the errors are less than 7 mm regarding the North and East components and less than 10 mm regarding the horizontal component. This mean that generally the estimated displacements follow the trend of the actual displacements. The vertical component is not so well recognized even if the global trend is identified.

Considering the results obtained using Valleggio 2 as reference it is possible to use a low-cost receiver also as a reference station: the results are not so good as the results obtained with a geodetic receiver, but in any case satisfactory.

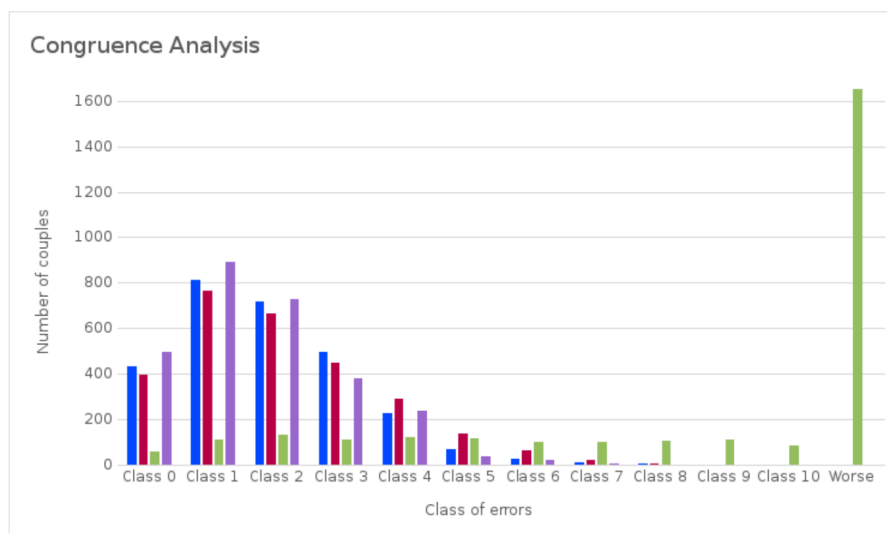


Figure 4.25 – Graph of frequency of couples for each class of errors for data processed with Valleggio 1 reference station. Horizontal (Purple bar), north (red bar), east (blue bar) and vertical (green bar) displacement

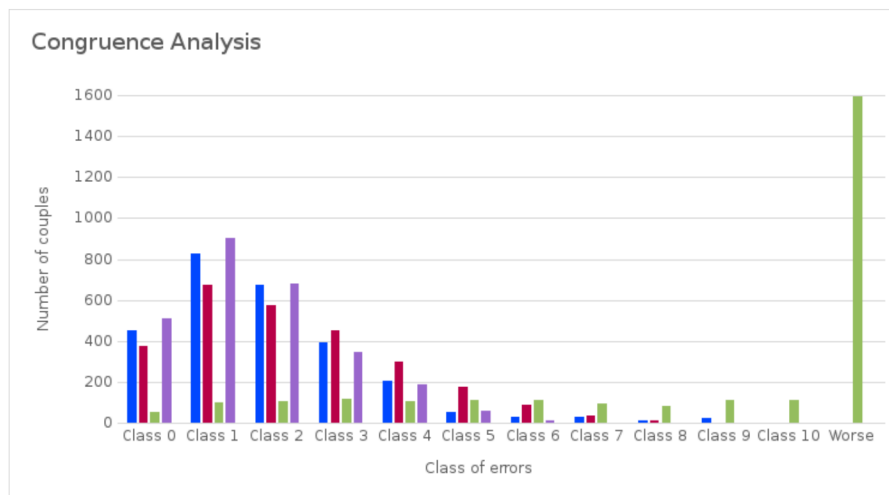


Figure 4.26 – Graph of frequency of couples for each class of errors for data processed with Valleggio 2 reference station. Horizontal (Purple bar), north (red bar), east (blue bar) and vertical (green bar) displacement

Valleggio 1	East		North		vertical		horizontal	
class 0	432	15.6%	393	14.2%	53	1.9%	495	17.8%
class 1	810	29.2%	765	27.6%	107	3.9%	893	32.2%
class 2	717	25.8%	663	23.9%	131	4.7%	725	26.1%
class 3	495	17.8%	447	16.1%	106	3.8%	376	13.5%
class 4	225	8.1%	290	10.5%	117	4.2%	234	8.4%
class 5	64	2.3%	136	4.9%	114	4.1%	33	1.2%
class 6	25	0.9%	61	2.2%	100	3.6%	17	0.6%
class 7	6	0.2%	19	0.7%	99	3.6%	2	0.1%
class 8	1	0.0%	1	0.0%	103	3.7%	0	0.0%
class 9	0	0.0%	0	0.0%	111	4.0%	0	0.0%
class 10	0	0.0%	0	0.0%	81	2.9%	0	0.0%
over	0	0.0%	0	0.0%	1653	59.6%	0	0.0%

Table 4.14 – Numbers and frequencies of couples for each class of error for data processed with Valleggio 1 as reference station

Valleggio 2	East		North		vertical		horizontal	
class 0	451	16.7%	374	13.8%	55	2.0%	511	18.9%
class 1	824	29.7%	677	24.4%	100	3.6%	900	32.4%
class 2	673	24.3%	576	20.8%	108	3.9%	678	24.4%
class 3	393	14.2%	453	16.3%	116	4.2%	346	12.5%
class 4	207	7.5%	302	10.9%	109	3.9%	188	6.8%
class 5	53	1.9%	177	6.4%	111	4.0%	61	2.2%
class 6	30	1.1%	87	3.1%	110	4.0%	13	0.5%
class 7	29	1.0%	38	1.4%	94	3.4%	4	0.1%
class 8	16	0.6%	12	0.4%	81	2.9%	0	0.0%
class 9	22	0.8%	4	0.1%	112	4.0%	0	0.0%
class 10	3	0.1%	1	0.0%	115	4.1%	0	0.0%
over	0	0.0%	0	0.0%	1590	57.3%	0	0.0%

Table 4.15 – Numbers and frequencies of couples for each class of error for data processed with Valleggio 2 as reference station

Valleggio 1	East		North		vertical		horizontal	
class 0	432	15.6%	393	14.2%	53	1.9%	495	17.8%
class 1-2	1527	55.0%	1428	51.5%	238	8.6%	1618	58.3%
over	816	29.4%	954	34.4%	2484	89.5%	662	23.9%

Table 4.16 – In evidence class 0, class 1-2 (data processed with Valleggio 1 as reference station)

Valleggio 2	East		North		vertical		horizontal	
class 0	451	16.7%	374	13.8%	55	2.0%	511	18.9%
class 1-2	1497	53.9%	1253	45.2%	208	7.5%	1578	56.9%
over	753	27.1%	1074	38.7%	2438	87.9%	612	22.1%

Table 4.17 – In evidence class 0, class 1-2 (data processed with Valleggio 2 as reference station)

## Chapter 5

# Conclusions

The main purpose of this PhD thesis was to test if it is possible to increase the accuracy of low cost GNSS receivers with the aim to expand the fields of applications for these devices. In particular two different application fields have been explored: **navigation** and **monitoring activities**, with two different targets about the final accuracy improvement:

- point positioning in real time, with the goal to reach an accuracy the order of the meter regarding navigation applications, in particular about mapping obstacles in path ways;
- data post processing, using geodetic technique, with the goal to reach an accuracy of few centimeters, for static sessions, in particular about local landslide monitoring.

In navigation applications, differently than in monitoring activities, the low cost GNSS receivers are already used, like in tablets and smartphones, car navigation systems, smart watches and so on: the accuracy of autonomous navigation can degrade up to tens meters, and is clearly not enough for the above mentioned application.

Two different procedures were implemented and tested with the aim to increase the usually accuracy obtained with these devices: correction using data from Continuously Operating Reference Stations (Chapter 2) and correction using digital cartography as reference (Chapter 3).

The first case uses the CORS corrections estimated epoch by epoch to improve the estimated positions from low cost receivers: the difference between the actual CORS position and the CORS position at epoch  $i$  can be supposed due to particular conditions present at this epoch that affects the estimation process. If the rover is near to the CORS these temporary effects should be more or less the same that afflict the rover, than it should be possible to apply the CORS correction to correct the rover estimated positions.

At first the corrections were applied to data collected during statics occupations, with a good satellites visibility. In this case was verified how the CORS corrections generally slightly improve accuracy (tables 2.1, 2.2 and graphs reported in appendix C; in tables 5.1 and 5.2 are reported extracts of these tables with the standard deviations before and after the application of CORS corrections). These results were obtained in good visibility conditions both for the CORS and for the

rover and were in some way expected.

[m]	before corrections		after corrections	
day	East	North	East	North
14-073	1.4	2.1	1.4	1.9
14-074	1.5	2.1	1.4	2.0
14-075	1.5	2.1	1.4	2.0
14-076	1.5	2.3	1.4	2.0
14-077	1.4	2.4	1.4	2.0
14-078	1.5	2.6	1.4	2.0
14-079	1.6	2.7	1.5	2.1

Table 5.1 – Milan test: standard deviations before and after the application of CORS corrections

[m]	before corrections		after corrections	
day	East	North	East	North
15-128	0.8	1.2	0.6	0.8
15-129	0.8	1.4	0.6	0.8
15-130	0.8	1.2	0.6	0.8
15-131	0.9	1.1	0.7	0.8
15-132	0.9	1.1	0.6	0.8
15-133	0.9	1.2	0.7	0.9
15-134	0.9	1.2	0.6	0.8
15-135	0.8	1.3	0.6	0.8

Table 5.2 – Como test: standard deviations before and after the application of CORS corrections

These good results pushed us to pass at the following step and to test the correction procedure on a more general case, in a typical urban scenario. In this case data were collected during several repetitions of two urban test paths in Como town (figures 2.14 and 2.15) defined in order to simulate the possible different conditions of satellite visibility that can occur in a urban context: urban canyons, areas of fair visibility and open areas. The results obtained in this second experiment were not so good (tables 2.4, 2.6 and 2.7; in tables 5.3, 5.4 and 5.5 are reported extracts of these tables with the standard deviations before and after the the CORS correction were applied). Also in condition of discrete visibility (for example in Cavour square, see figure 2.31) the CORS corrections did not enough to improve final positioning. This happens because generally the positioning errors were mainly affected by the local effects, probably due to multipath and the CORS corrections could not correct for such effects.

Therefore the obtained results didn't leave space to further improvement of the correction method and showed the scarce efficacy of this method to improve the accuracy of navigation at the meter level. The tests didn't leave many possibility to use effectively CORS corrections, therefore others possible way to correct the estimated positions in autonomous navigation were explored.



[m]	before corrections		after corrections	
day	East	North	East	North
14-259	2.4	2.5	2.5	2.5
14-267	3.7	5.3	3.4	5.2
14-268	3.9	5.0	4.0	5.1

Table 5.3 – Path 1: standard deviations before and after the application of CORS corrections; survey with smartphone

[m]	before corrections		after corrections	
day	East	North	East	North
16-139	2.0	2.4	2.0	2.4
16-207	4.0	4.4	4.0	4.5
16-208	4.9	4.0	4.9	4.0

Table 5.4 – Path 2: standard deviations before and after the application of CORS corrections; survey with smartphone

[m]	before corrections		after corrections	
day	East	North	East	North
16-139	0.9	1.3	1.1	1.4
16-207	1.4	2.4	1.4	2.8
16-208	1.2	1.1	1.2	1.2

Table 5.5 – Path 2: standard deviations before and after the application of CORS corrections; survey with u-blox

In the second tested procedure a digital, globally available by web cartography was used as reference to correct the estimated positions in autonomous navigation. Different correction algorithms were tested with different levels of correction, from a minimum level of correction to more significant levels:

- correction out of the buildings;
- correction on buffers;
- correction path segments as reference.

Different results were obtained, depending on the particular local conditions, but generally the positioning was improved in all the cases (table 3.7). The best results were obtained with the *path segments as reference* method (section 3.2) and with the *variable buffer* method (section 3.4). The *path segments as reference* method is a very constraining method and in one particular day (14-259) it didn't give us good results as in the others cases.

In any case the use of cartography as reference it promising, in all cases, to improve positioning: in the future it will be interesting a more deeply investigation using also non parametric methods to correct positioning, like fuzzy logic and neural networks.

The testing capabilities of low cost receivers for local monitoring (Chapter 4) was focused in post-processing techniques, using different GNSS processing packages. The goal of these experiments was to verify the accuracy achievable using low cost receivers to collect and post-process raw data with the same technique usually used with geodetic instrumentation.

Different surveys were done: a first experiment (Experiment 1: section 4.2) was done using data from a static survey. The data used in this experiment were acquired during one week of survey made in Milan: the dataset is relevant to a short baseline (approximately 80 m long) formed by the permanent station of Politecnico di Milano and a low cost receiver (u-blox) used as rover.

To processing data we chose Bernese GPS Software (BSW), one of the leading software for the international geodetic community, and Leica Geo Office (LGO), a commercial package and a comparison between these software was done. Firstly, we processed the whole week with BSW to estimate a reference position of the rover. Then, the hourly sessions were processed to estimate the rover positions using both, BSW and LGO.

The results obtained are completely satisfactory: the standard deviation is less than 5 mm in all the cases (see table 5.6; completely results are reported in tables 4.3 and 4.5).

BSW and LGO results are similar and no bias exists between the two packages. LGO's standard deviation are slightly better in horizontal while vertical results of the two packages are similar. The vertical statistics are the worst probably caused by the lack of antenna PCVs for u-blox.

Statistics	E [mm]	N [mm]	U [mm]
BSW Snd Dev	2.1	4.9	4.5
LGO Snd Dev	1.8	3.7	4.4

Table 5.6 – Experiment 1. BSW and LGO: hourly residuals of final solutions. E: East, N: North, U: Up

The good results obtained in the first test push us to define a new experiment (Experiment 2: section 4.4): in this case the survey was done in Como. As in the previous case a permanent station<sup>1</sup> was used as reference and a low cost receiver (u-blox) was used as rover.

Differently from the first experiment, in this case controlled displacements were imposed to the rover antenna to verify if it was possible recognize these displacements using a low cost receiver. The antenna was shifted horizontally 5 mm for each session along the baseline direction, three sessions per day, with one hour per session, for a total displacement of 5.5 cm.

After the satisfactory results obtained in the previous experiment, we decided to use LGO again to process data and, in order to compare different software, data were processed also with another software, rtkLIB, a free and open source software. These software are developed using two different approaches, one close source, inside a private enterprise and the other one on an open environment, with widespread support from many developers and open source.

---

<sup>1</sup>CORS COMO TEST

Data were processed using double difference combination and the congruence analysis was applied to verify if the displacement were correctly identified. As reported in table 5.7 (in tables 4.8 and 4.9 are reported the comprehensive data) regarding the horizontal displacement slightly less than half is in class 0, that means that in these cases the movement was correctly detected, with no errors. If we consider also the first class more the 90% of cases are included. Along the north and east components the results are a little worse, with the north component that presents the worst results, but the final results are good and satisfactory as the general indication is that can be possible detect the general trend of displacements with low cost receivers.

LGO	East		North		horizontal	
class 0	180	32.1%	104	18.5%	246	43.9%
class 1	277	49.4%	174	31.0%	274	48.8%
over	104	18.5%	283	50.5%	41	7.3%
RTKLIB						
class 0	215	38.3%	140	25.0%	261	46.5%
class 1	279	49.3%	226	40.3%	275	49.0%
over	67	11.9%	195	34.8%	25	4.5%

Table 5.7 – Experiment 2. Congruence analysis: in evidence class 0 and class 1

The Experiment 3 (section 4.5) was set up to verify also:

- if also the imposed displacements in vertical direction can be detected;
- if it is possible to use of single frequency receivers as reference station.

Three points of survey were set up: two reference stations, one with geodetic double frequency receiver (Valleggio 1) and an other a single frequency receiver (Valleggio 2), just a few meters from each other, and the usual low cost receiver as rover. The distance between reference stations and rover was about 130 meters.

The sliding device with top-mounted the rover antenna was oriented in North-East direction to permit to easily evaluate also along the North and East components. From the start position the antenna was moved, alternately, of 5 mm in horizontal and 5 mm down in vertical at leaving at least 2 hours between consecutive movements (data were acquired at 1 Hz of frequency); the total movement reached 10 cm in both directions.

The data from the rover were processed with LGO and rtkLIB two times, using both the reference stations prepared. In this test with rtkLIB some problems occurred, with a low number of fixed ambiguities, therefore the results obtained were not comparable with results obtained with LGO.

The obtained results showed that the general displacement trend is recognizable: along the horizontal direction the results are satisfactory, with the estimated displacements close to the imposed ones. Along the vertical direction the results are worse: the general trend is still recognizable, but the individual movements are almost impossible to identify (table 5.8; comprehensive data in table 4.11). The singles movements are generally recognizable along north, east and horizontal components; about the vertical component the results are slight worse (tables 5.9 and 5.10; comprehensive data in tables 4.16 and 4.17).

The main problems that affect the vertical estimation can be probably due to the missing estimates of *Phase Center Variations* (PCV) parameters for the low cost antennas and a generally weakness u-blox antenna that seemed very sensible to the local environment quality.

Generally the results testify that it could be possible to use low cost receivers in landslides monitoring activities, especially when centimeter accuracy is sufficient and the horizontal component is prevalent in the landslide movement.

Further works could improve the comparison of different processing software. A very interesting thing to verify in the future is the test of better antennas than the standard u-blox antenna, in order to to increase the accuracy, in particular for of vertical movement and evaluate the contribution of PCV parameters in the final accuracy.

It would be also interesting to move the test in a realistic environment, going to make surveys on active landslides and compare the results obtained with low cost receivers with the currently tools used in monitoring.

Finally, for future tests, can be worthwhile to consider single frequency multi-constellation receivers, to evaluate the contribution of the others GNSS services, with particular focus on Galileo potentialities, that on 15 December 2016 reached the Early Operational Capability (EOC) [18].

	Valleggio 1				Valleggio 2			
[cm]	E	N	vert	horz	E	N	vert	horz
snd dev	0.4	0.6	1.3	0.4	0.7	0.9	1.8	0.7

Table 5.8 – Experiment 3. Processed data: comparison between standard deviation obtained using two different reference station, Valleggio 1 and Valleggio 2

Valleggio 1	East		North		vertical		horizontal	
class 0	432	15.6%	393	14.2%	53	1.9%	495	17.8%
class 1-2	1527	55.0%	1428	51.5%	238	8.6%	1618	58.3%
over	816	29.4%	954	34.4%	2484	89.5%	662	23.9%

Table 5.9 – Experiment 3. Congruence analysis: data processed with Valleggio 1 as reference station

Valleggio 2	East		North		vertical		horizontal	
class 0	451	16.7%	374	13.8%	55	2.0%	511	18.9%
class 1-2	1497	53.9%	1253	45.2%	208	7.5%	1578	56.9%
over	753	27.1%	1074	38.7%	2438	87.9%	612	22.1%

Table 5.10 – Experiment 3. Congruence analysis: data processed with Valleggio 2 as reference station

# Appendix A

## Processing software

### A.1 Bernese GPS Software

The Bernese GNSS software<sup>1</sup> is a scientific, high-precision, multi-GNSS data processing software developed at the Astronomical Institute of the University of Bern (AIUB)<sup>2</sup>. It is a sophisticated tool meeting highest quality standards for geodetic and further applications using GNSS [15]. For example it is used by Center for Orbit Determination in Europe<sup>3</sup> (CODE) for its international (IGS) and European (EUREF/EPN) activities.

This software is a multi-platform software, available on Unix/Linux, Mac and Windows platforms, in a permanent process of development and improvement.

### A.2 Leica Geo Office (LGO)

Leica Geo Office<sup>4</sup> (LGO) is a software produced by Leica Geosystems<sup>5</sup>, a Switzerland enterprise that produces instruments for surveying and measurement and the relative software to process the data acquired.

LGO is dedicated to process data from GNSS survey and is able to handle many different processing scenarios: static, rapid static, stop-and-go, kinematic. It can process data from multi constellations, GPS, Glonass and Galileo and can process both single frequency and dual frequency data. Many tools to import data from different formats (for example raw data, cad and GIS data), to export data and create report are available.

In this thesis the version 8.3 was used to process data from GNSS receivers, both single and double frequency, using the double differences observations.

---

<sup>1</sup><http://www.bernese.unibe.ch/>

<sup>2</sup><http://www.aiub.unibe.ch/>

<sup>3</sup><http://www.aiub.unibe.ch/code>

<sup>4</sup><http://leica-geosystems.com/products/gnss-systems/software/leica-geo-office>

<sup>5</sup><http://leica-geosystems.com/>

### A.3 goGPS

goGPS<sup>6</sup> is a open source software expressly designed to improve the positioning accuracy of GNSS single-frequency receiver by relative positioning ([42], and [43]). The development started in 2007 at the Geomatics Laboratory<sup>7</sup> of Politecnico di Milano, in Como Campus as a Master thesis project and then continued during two other Master theses. First release was published in August 2009. Its development is now carried out by various teams in Italy and Japan. The goGPS code works under MATLAB environment: it is possible to use it both in UNIX/Linux and Windows system. There exists also a Java version originally developed by Cryms<sup>8</sup> in collaboration with Politecnico di Milano and Osaka City University<sup>9</sup>. Now, both the MATLAB and Java versions of goGPS are developed at GReD<sup>10</sup> and Politecnico di Milano, with various contributors at international level.

Currently, the main developer is Eugenio Realini (GReD), with the support of Mirko Reguzzoni (Politecnico di Milano). The software is distributed under the terms of the GNU General Public License, version 3<sup>11</sup>

In this thesis the version 0.4.3, installed under Linux Gentoo, was used to post-processing single frequency data in point positioning mode (Chapter 2).

### A.4 RTKLIB

RTKLIB<sup>12</sup> is an open source software written in C for standard and precise positioning with GNSS [47]: it can be used both in Windows and in UNIX/Linux environment. The main author is Tomoji Takasu.

It supports data from many different system, like GPS, GLONASS, Galileo, QZSS, BeiDou and SBAS; variuos positioning modes are available for both real-time and post-processing: Single, Differential GNSS, Kinematic, Static, Moving-Baseline, Fixed, PPP-Kinematic, PPP-Static and PPP-Fixed. RTKLIB is distributed under the BSD 2-clause license<sup>13</sup>.

In this thesis the version 2.4.3 installed under Linux Gentoo was used to post-process from the single frequency data, in point positioning mode (Chapter 2) and in double difference mode (Chapter 4).

---

<sup>6</sup><http://www.gogps-project.org/>

<sup>7</sup><http://geomatica.como.polimi.it/>

<sup>8</sup><http://www.cryms.com/>

<sup>9</sup><http://www.osaka-cu.ac.jp/en/>

<sup>10</sup><http://www.g-red.eu/>

<sup>11</sup><https://www.gnu.org/licenses/gpl-3.0.en.html>

<sup>12</sup><http://www.rtklib.com/>

<sup>13</sup><http://opensource.org/licenses/BSD-2-Clause>

# Appendix B

## GNSS receivers used

### B.1 Leica GX 1230

Leica GX1230 is a receiver used during the survey along the path to get out a reference to evaluate the goodness of low cost receivers (section 2) and to control the condition of survey in chapter 4. It is a geodetic receiver, multi-constellation, used with its standard survey antenna AX 1203.

The main characteristic are [29]:

- GNSS receiver, measuring mode: static, rapid static, kinematic, code, phase Real-time RTK, Post processing, DGPS/RTCM standard, Survey, geodetic and real-time RTK applications;
- high precision GNSS: it can reach in kinematic mode, after initialization, an accuracy<sup>1</sup> of 10 mm in horizontal and 20 mm in vertical; when only code solutions are available the accuracy is typical 25 cm.
- 120 channels: GPS L1/L2/L5 GPS; L1/L2 GLONASS; E1/E5a/E5b/Alt-BOC Galileo; 4 SBAS;
- raw measurement data.

### B.2 u-blox

u-blox<sup>2</sup> is a Swiss company leading provider of wireless and positioning sensors and modules for the automotive, industrial and consumer markets. u-blox created "evaluation kits" to make easier the evaluation of its GNSS chips: these devices permit to setting, powering the GNSS receiver and keep data by a simple USB connection with a notebook or a tablet/smartphone. By connecting the evaluation kit to a PC, using the dedicated software *u-center*, it is possible to set the output, for example choose the constellation (GLONASS or GPS), choose the infos inserted in the NMEA file, or get as output also the raw data (figure B.1).

---

<sup>1</sup>in root mean square (rms)

<sup>2</sup><https://www.u-blox.com>



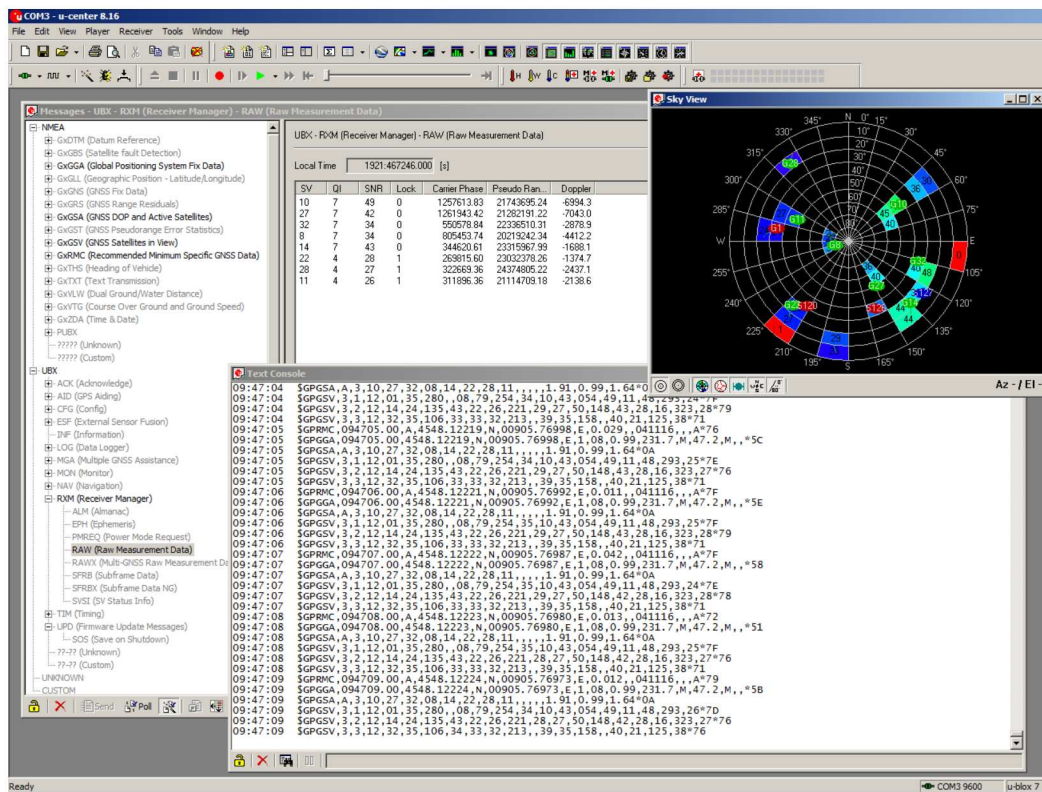


Figure B.1 – U-Center at work

### B.2.1 u-blox EVK-7P

In this thesis as single frequency low cost receiver was used mainly the u-blox evaluation kit EVK-7<sup>3</sup>. The EVK-7<sup>4</sup> has inside the NEO-7P<sup>5</sup> module: the precise point positioning GNSS u-blox module. The main technical characteristics of u-blox NEO-7P are [49]:

- GNSS receiver: GPS, GLONASS, Quasi Zenith Satellite System (QZSS) and Galileo;
- high precision GPS, less 1 m, in the configuration Satellite Based Augmentation System (SBAS) and Precise Point Positioning (PPP);
- 56 channels: GPS L1 C/A, GLONASS L1 FDMA, QZSS L1 C/A, SBAS;
- differential GPS by SBAS or Radio Technical Commission for Maritime Service (RTCM);
- raw measurement data (GPS).

The evaluation kit EVK-7P was used as rover with its default antenna: Taoglas Dominator AA.161<sup>6</sup>: gain 26 dB, noise 2.6 dB.

The EVK-7P was also as reference station (Valleggio 2 in section 4.5), but in this case was paired with another antenna: Tallysman TW3150<sup>7</sup>. It is always a low cost device as its retail price is approximately 80 Euro but has higher gain

<sup>3</sup><https://www.u-blox.com/en/product/evk-7>

<sup>4</sup><https://www.u-blox.com/en/product/evk-7>

<sup>5</sup><https://www.u-blox.com/en/product/neo-7p-module>

<sup>6</sup><http://www.taoglas.com/product/dominator-aa161-magnetic-mount-gps-glonass-antenna/>

<sup>7</sup><http://www.tallysman.com/index.php/gnss/products/antennas-gps-l1/tw3152-timing-antenna/>

(50 dB), lower noise ( $\approx 1.5$  dB), better multipath suppression and a more stable mean phase center. The antenna was courteously provided by Stefano Caldera of G-Red<sup>8</sup>.

### **B.2.2 u-blox EVK-4T**

In section 2.5 and 4.2 were used data acquired and courteously provided by Prof. Riccardo Barzaghi and Ing. Mirko Reguzzoni, Department of Civil and Environmental Engineering, Politecnico di Milano. In this case the survey was done using the u-blox evaluation kit EVK-4.

The u-blox evaluation kit EVK-4T has inside the LEA-4T module. The main technical characteristics of this module are [48]:

- Stationary mode for GPS timing operation
- 15 ns timing accuracy (error compensated)
- Single Satellite GPS timing
- 10 Hz raw measurement data output
- Ultra low 39 mA power consumption
- Supports SBAS: WAAS, EGNOS, MSAS

---

<sup>8</sup><http://www.g-red.eu/>



## Appendix C

# Correct Rover with CORS correction: graphs of Como and Milan surveys

In the following figures are shown the results of comparison of positioning data of Como (figures C.1 - C.16) and Milan (figures C.17 - C.30) test sites, before and after the application of CORS corrections (see section 2.5).

R<sup>1</sup> software was used to compare data and give errors statistic and relative graphs.

---

<sup>1</sup>R is a free software environment for statistical computing and graphics, it is licensed under GPL-2 — GPL-3. Web site: <https://www.r-project.org/>

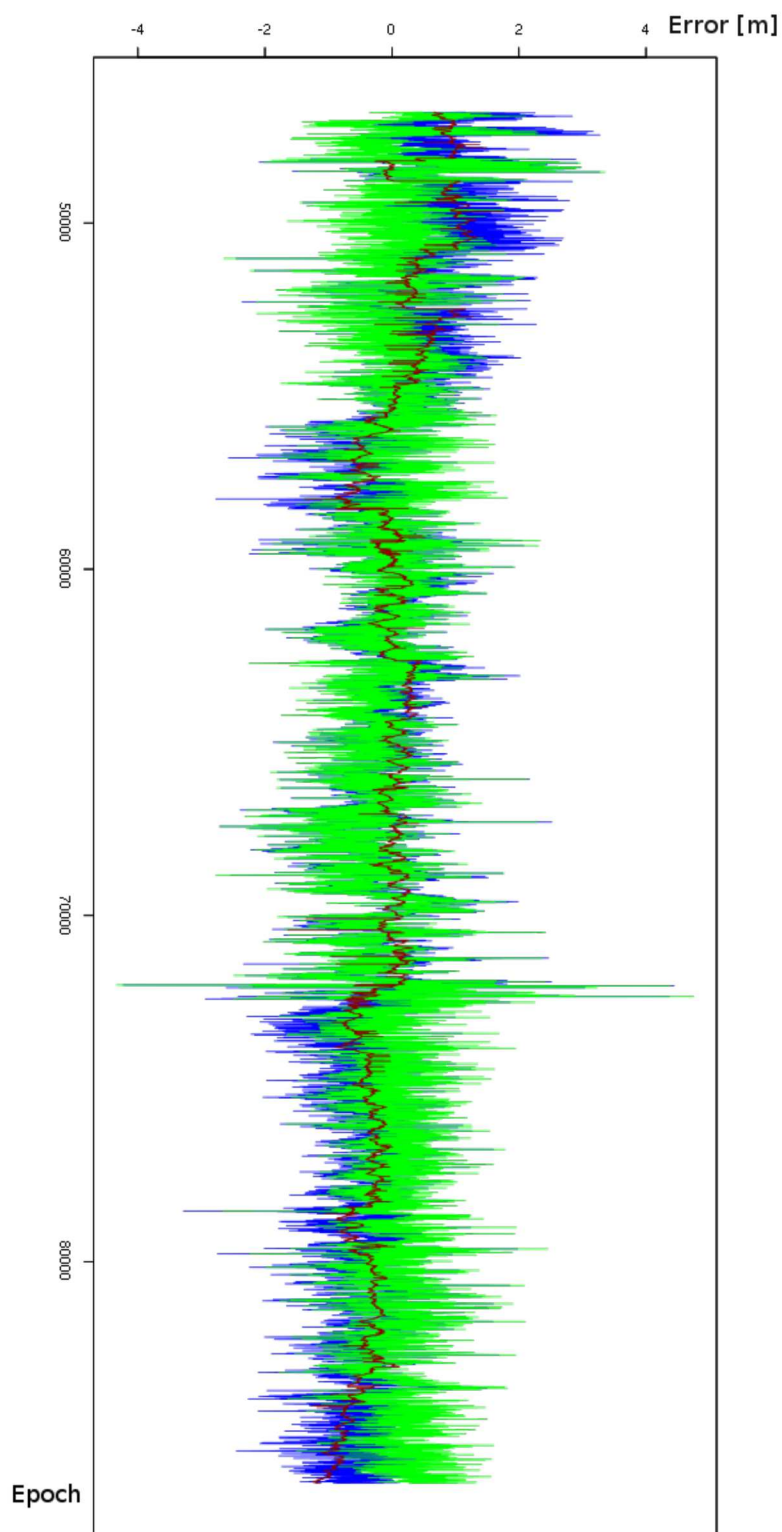


Figure C.1 – Statistical results for the **day 15-128** for the **east coordinate**: the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections

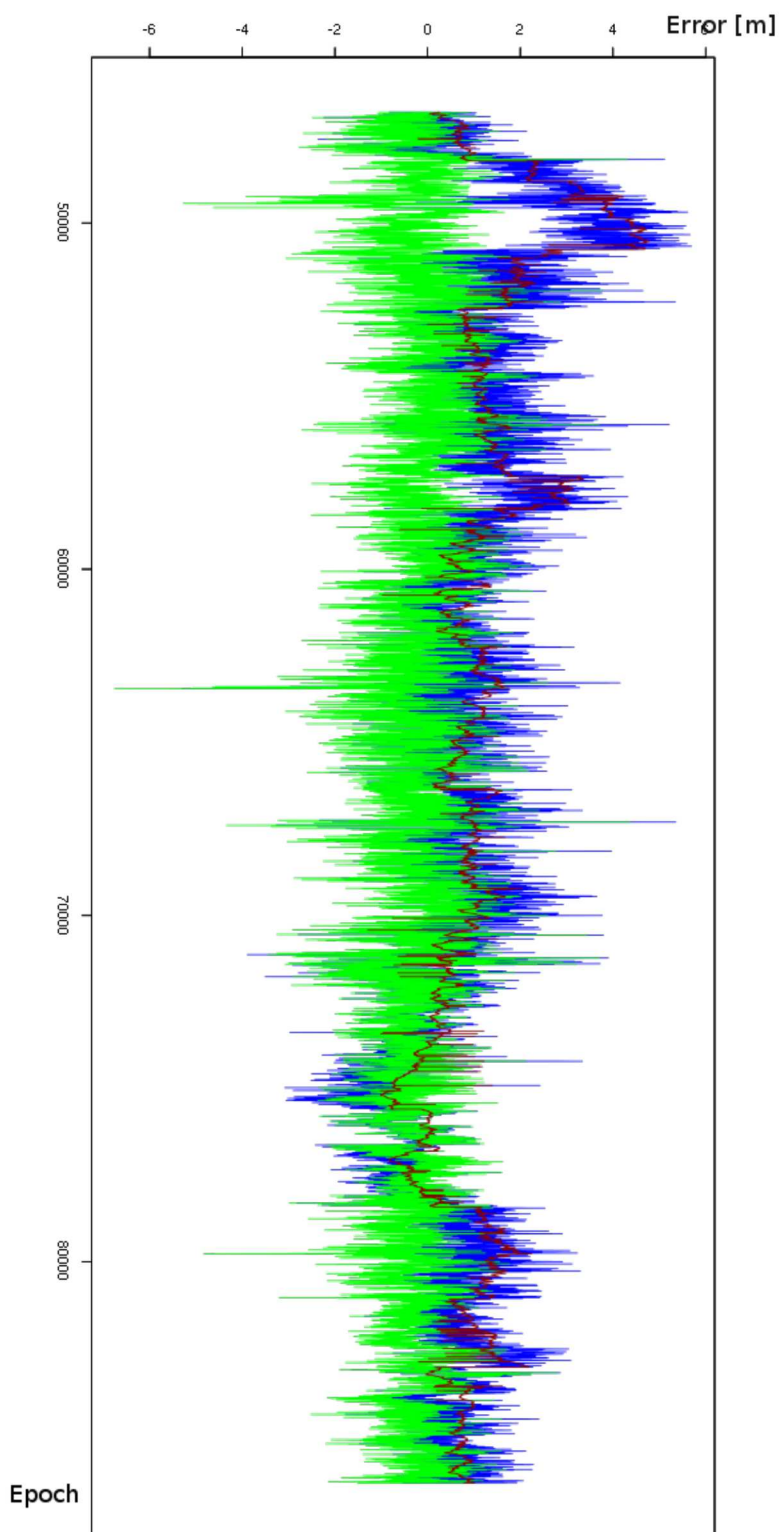


Figure C.2 – Statistical results for the **day 15-128** for the **north coordinate**: the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections

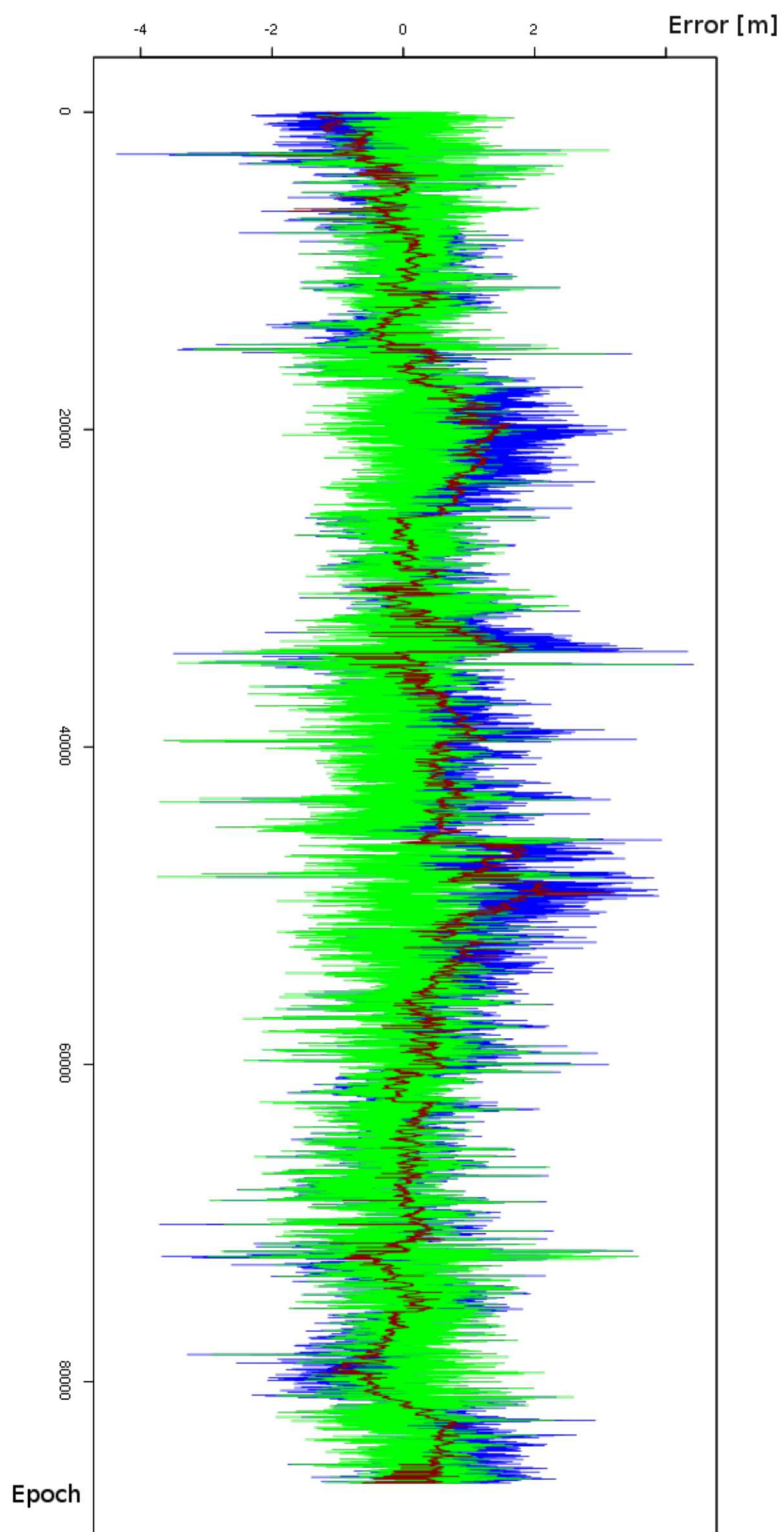


Figure C.3 – Statistical results for the **day 15-129** for the **east coordinate**: the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections



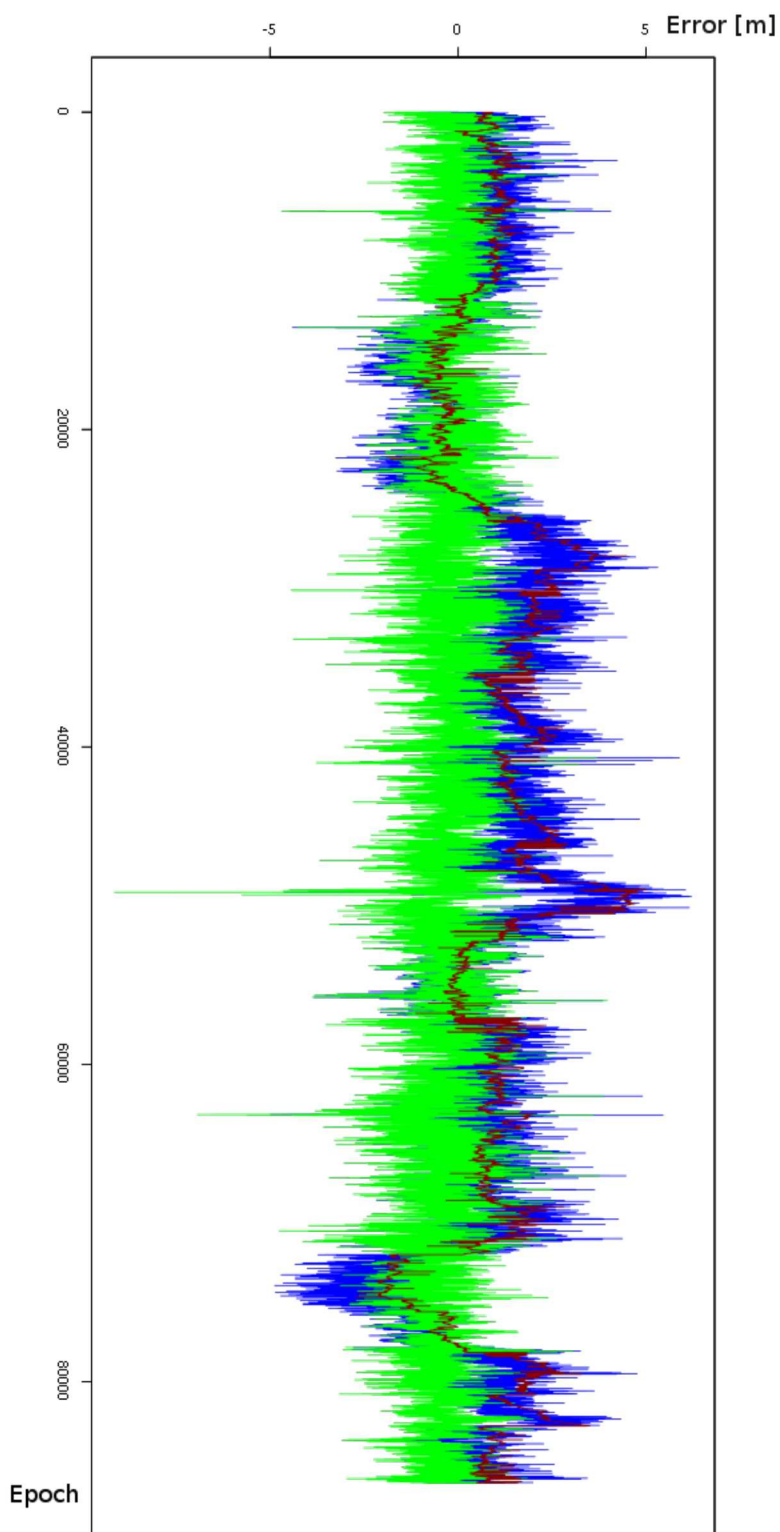


Figure C.4 – Statistical results for the **day 15-129** for the **north coordinate**: the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections

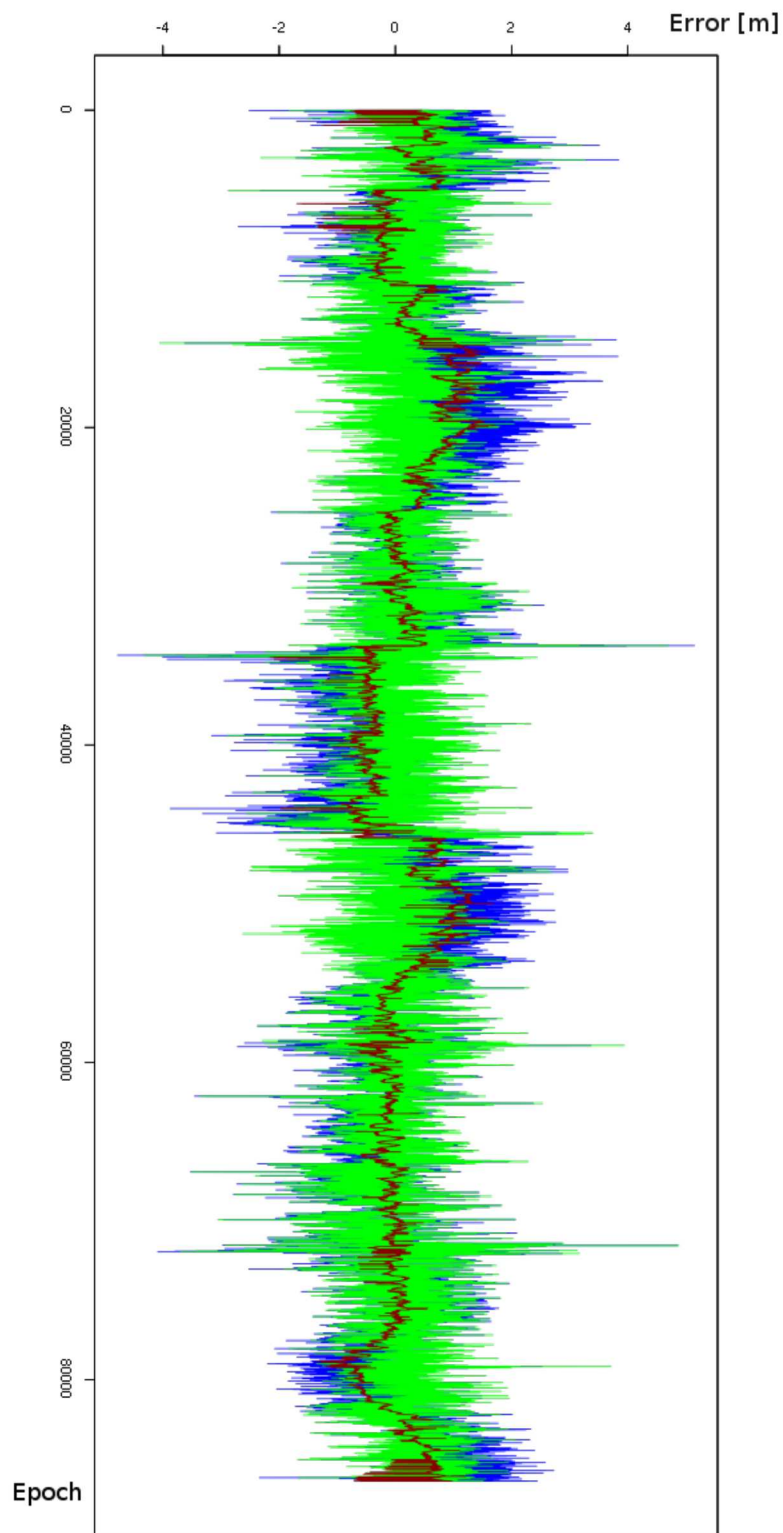


Figure C.5 – Statistical results for the **day 15-130** for the **east coordinate**: the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections

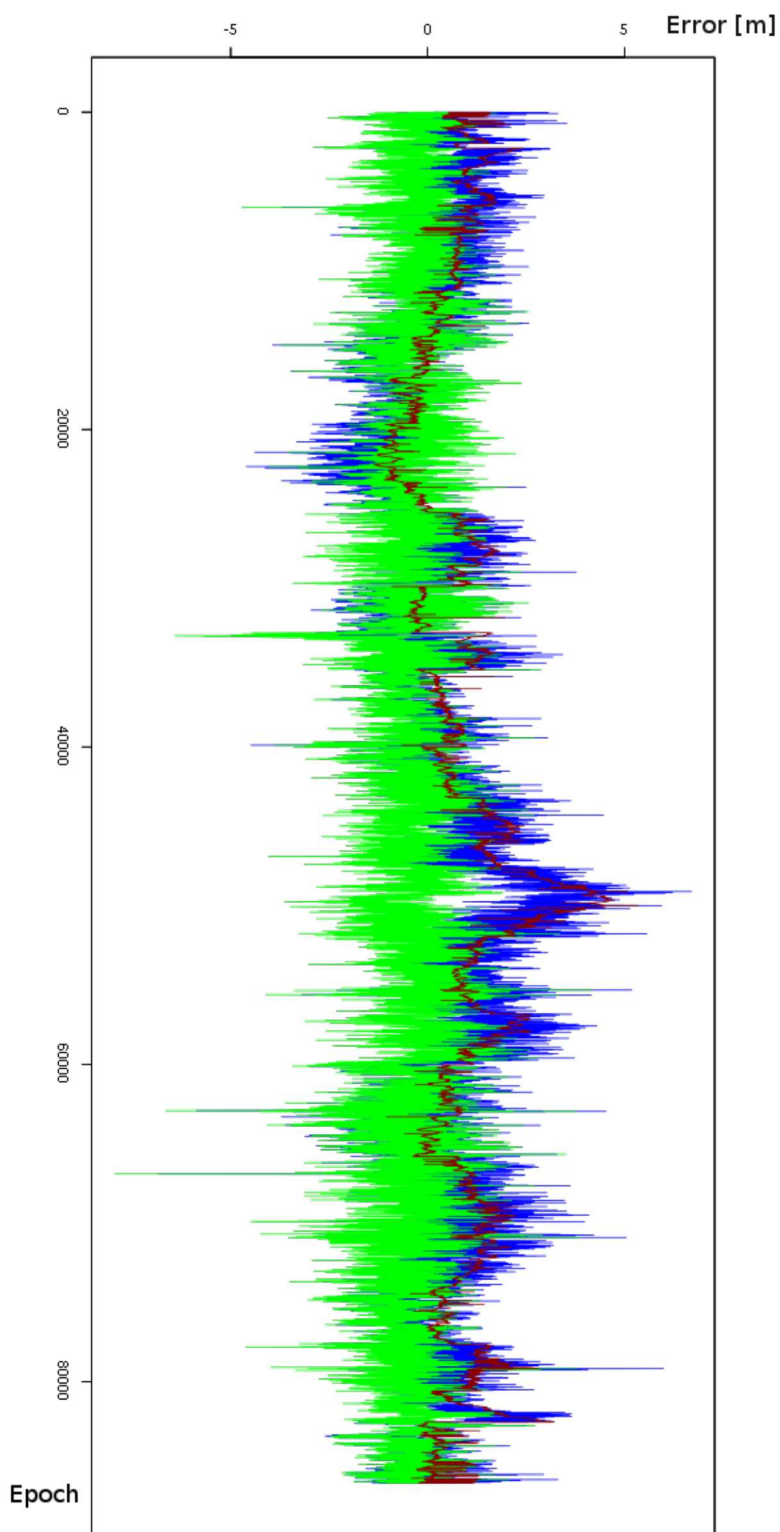


Figure C.6 – Statistical results for the **day 15-130** for the **north coordinate**: the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections

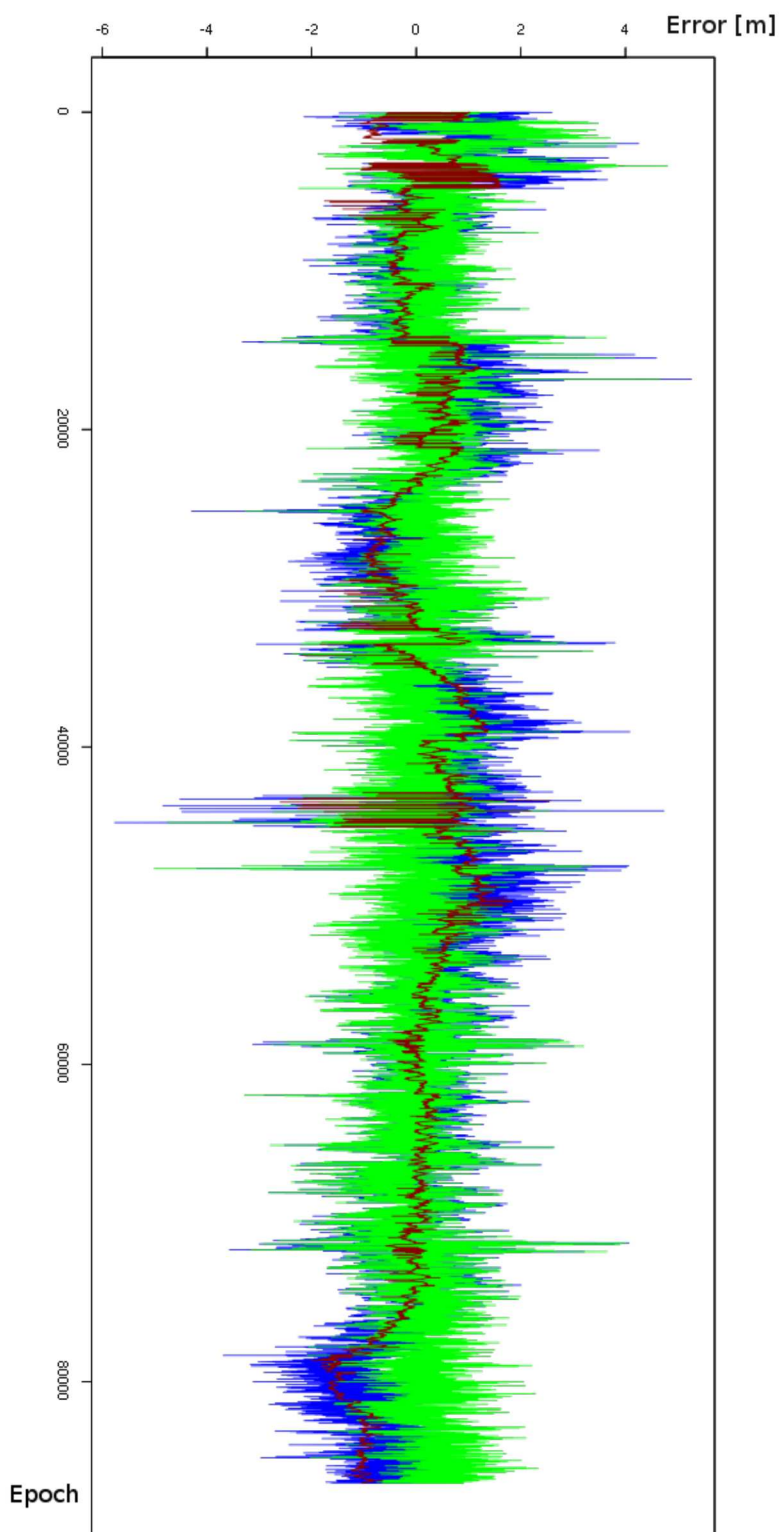


Figure C.7 – Statistical results for the **day 15-131** for the **east coordinate**: the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections

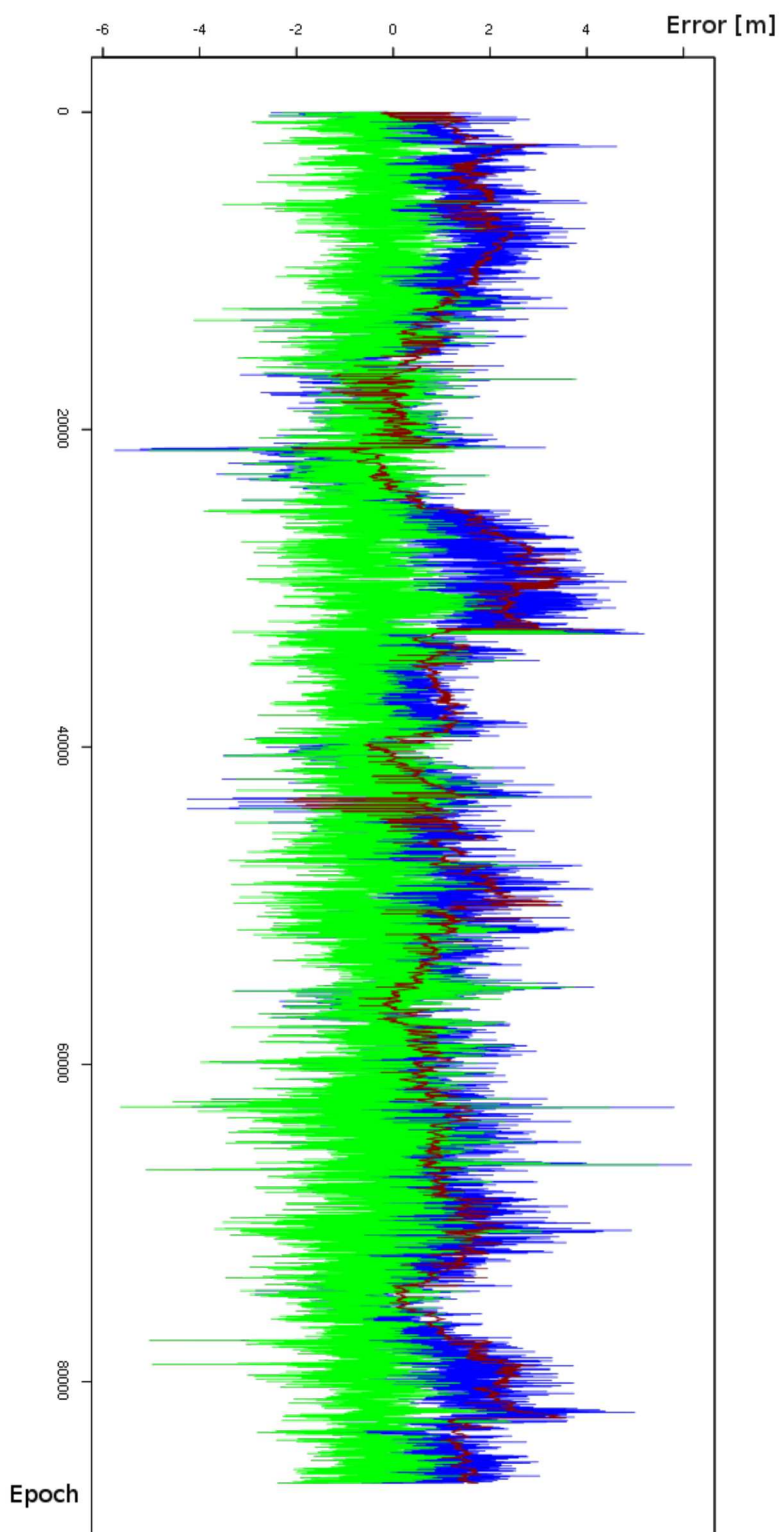


Figure C.8 – Statistical results for the **day 15-131** for the **north coordinate**: the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections

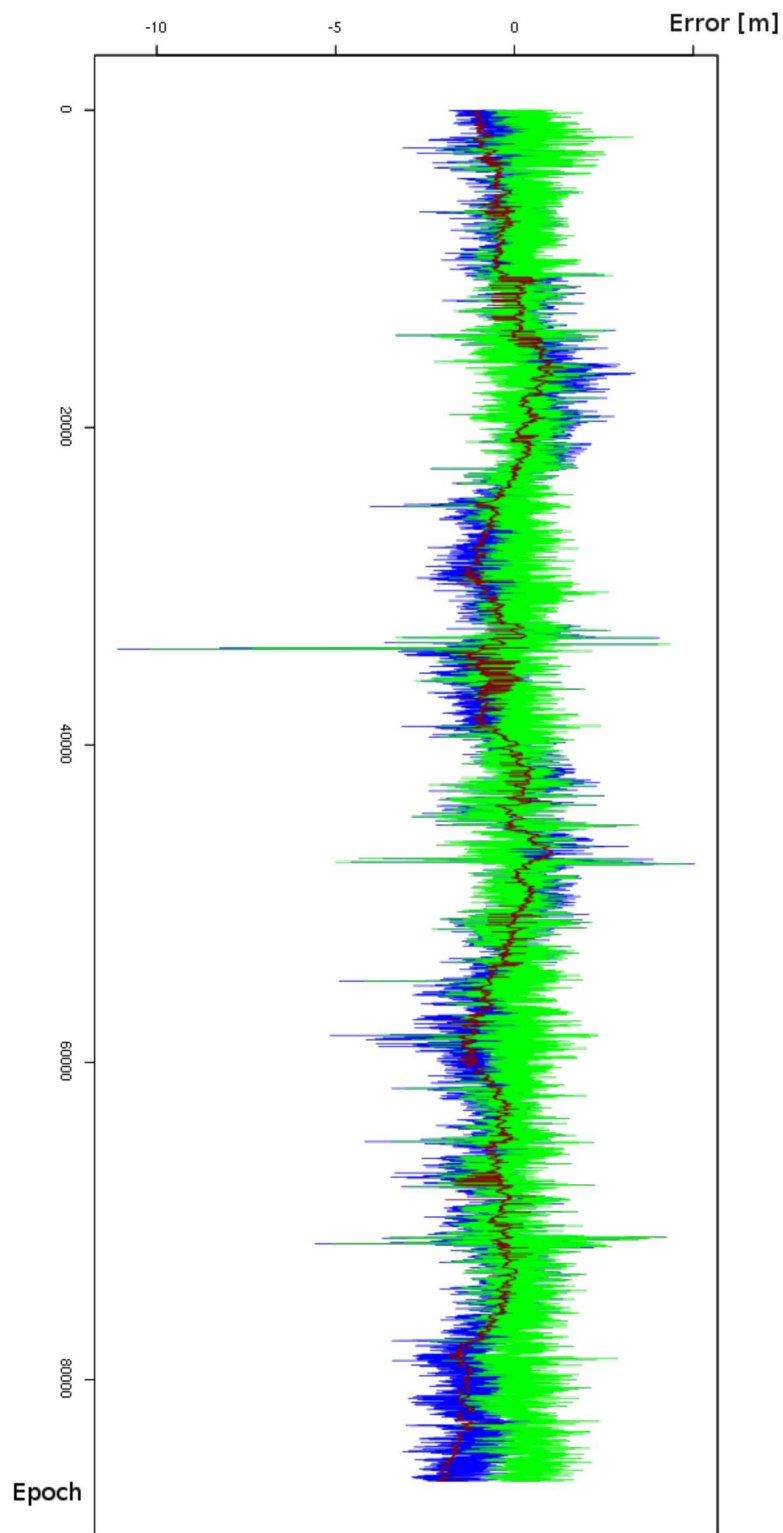


Figure C.9 – Statistical results for the **day 15-132** for the **east coordinate**: the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections



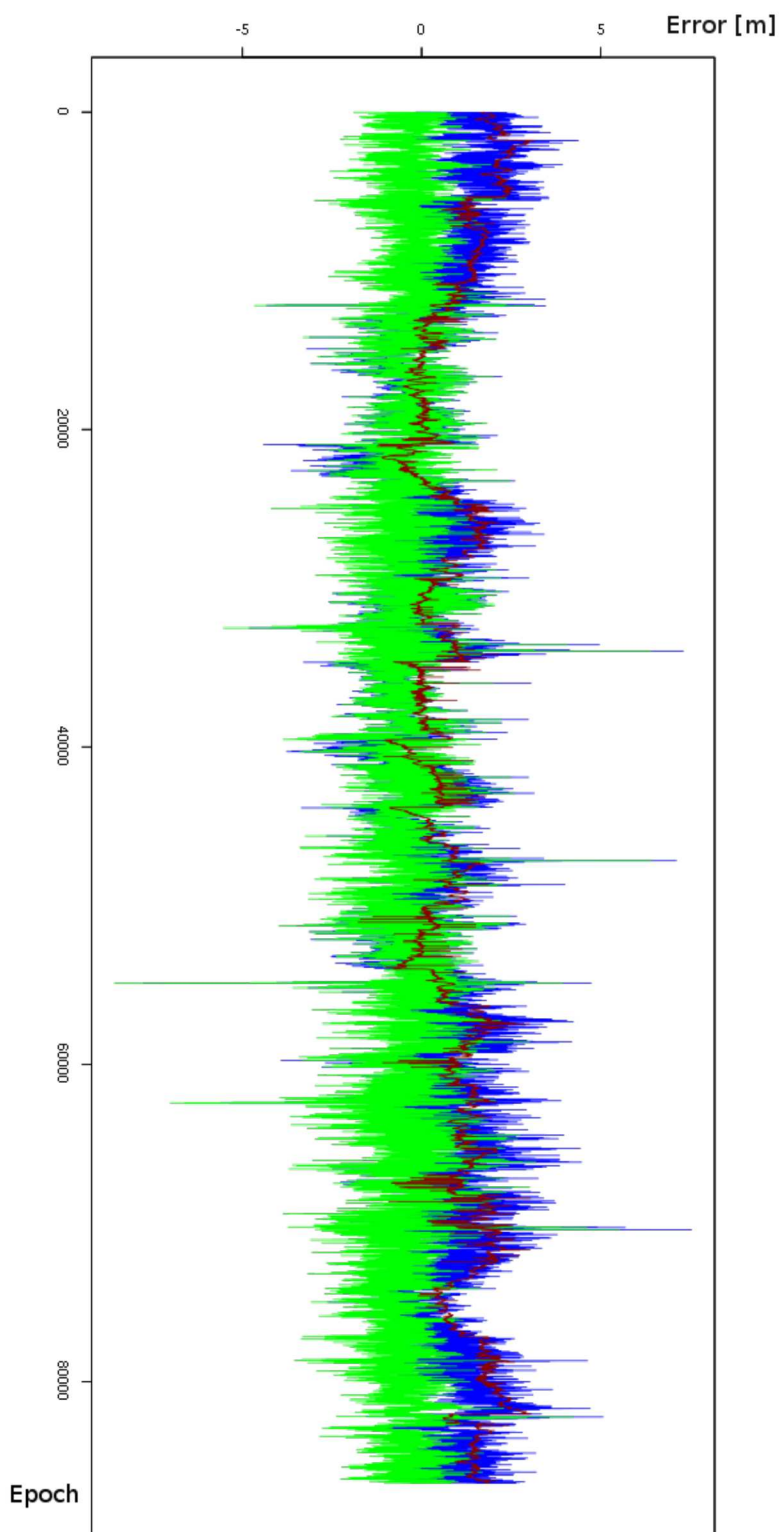


Figure C.10 – Statistical results for the **day 15-132** for the **north coordinate**: the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections



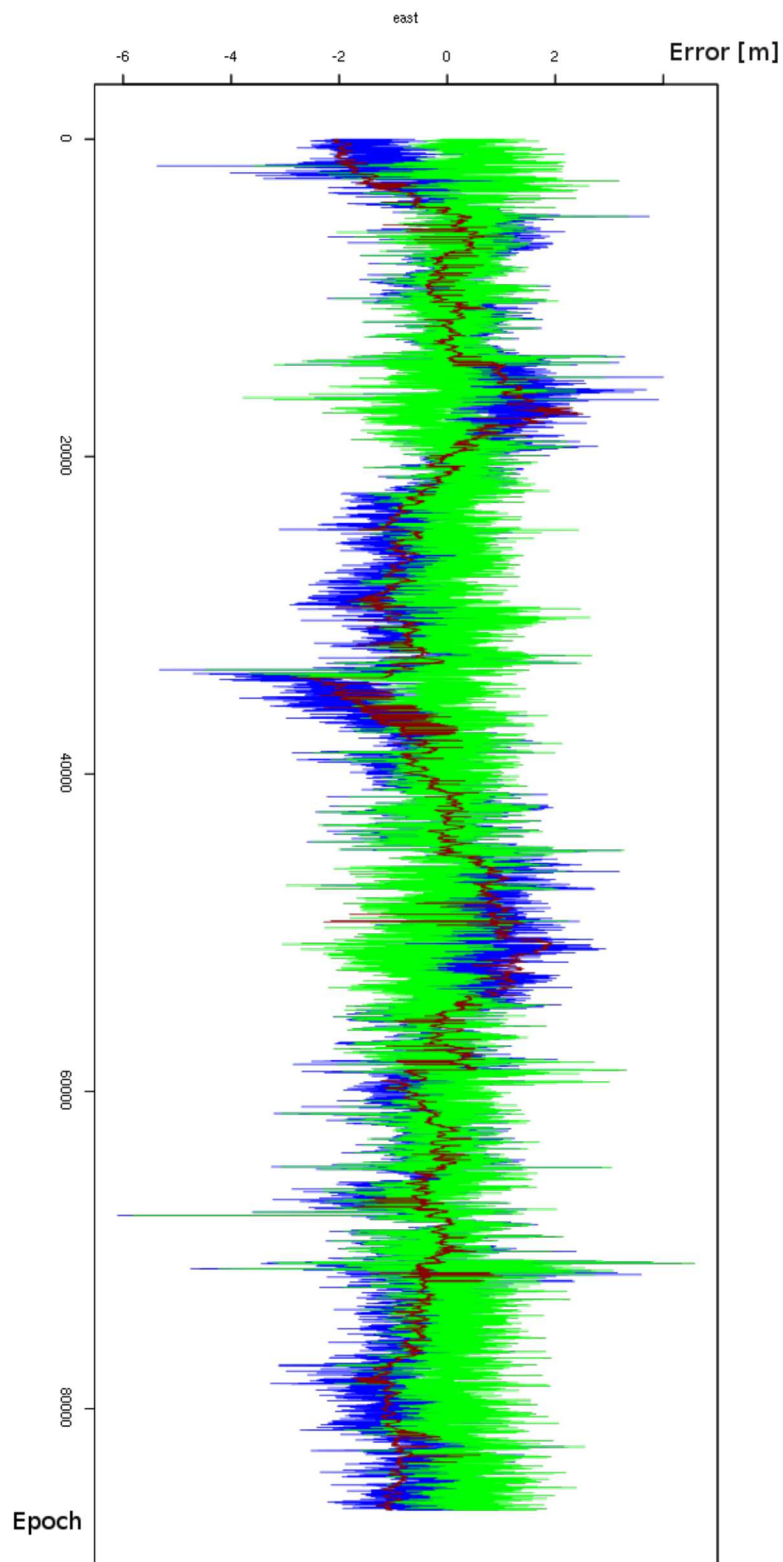


Figure C.11 – Statistical results for the **day 15-133** for the **east coordinate**: the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections

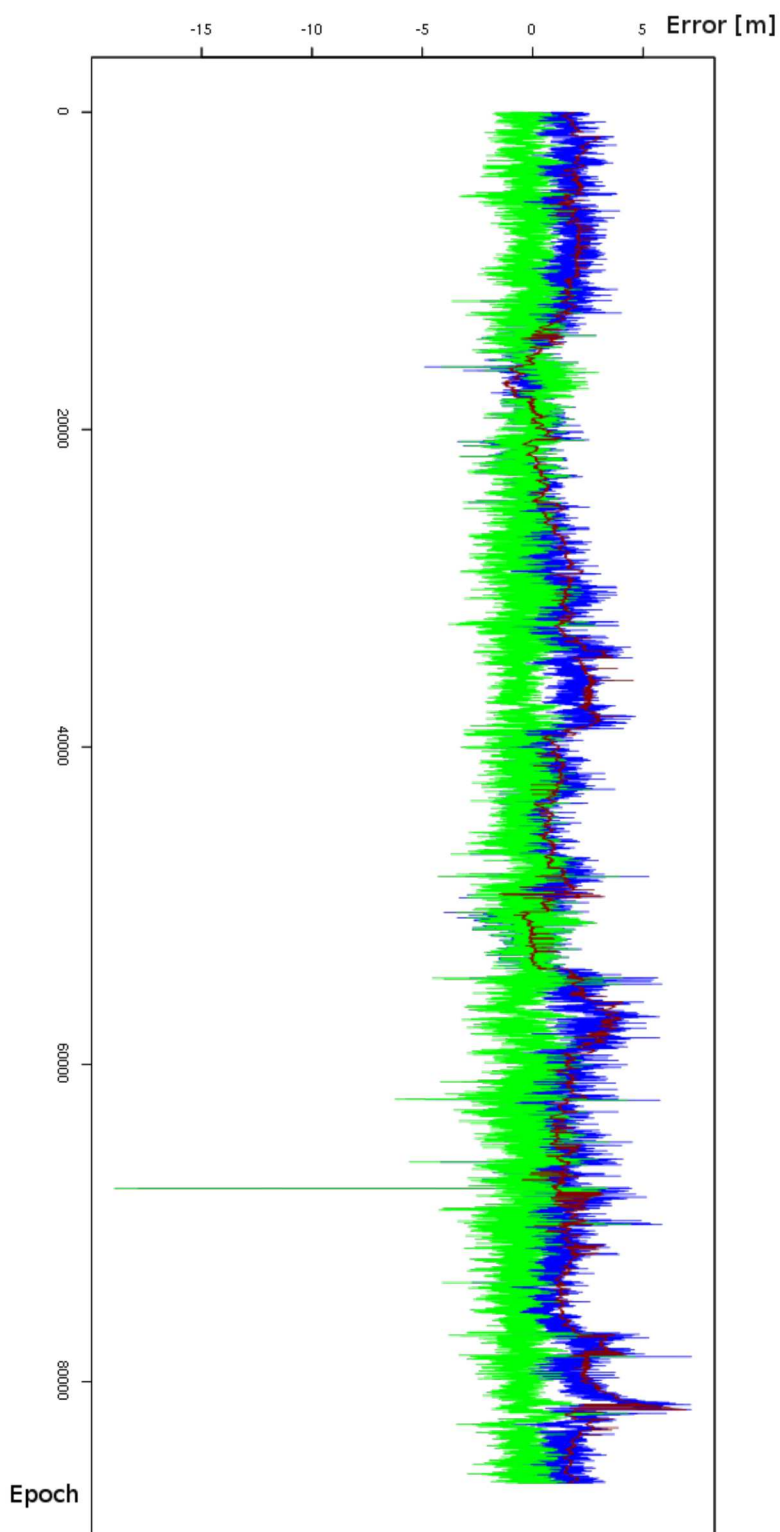


Figure C.12 – Statistical results for the **day 15-133** for the **north coordinate**: the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections

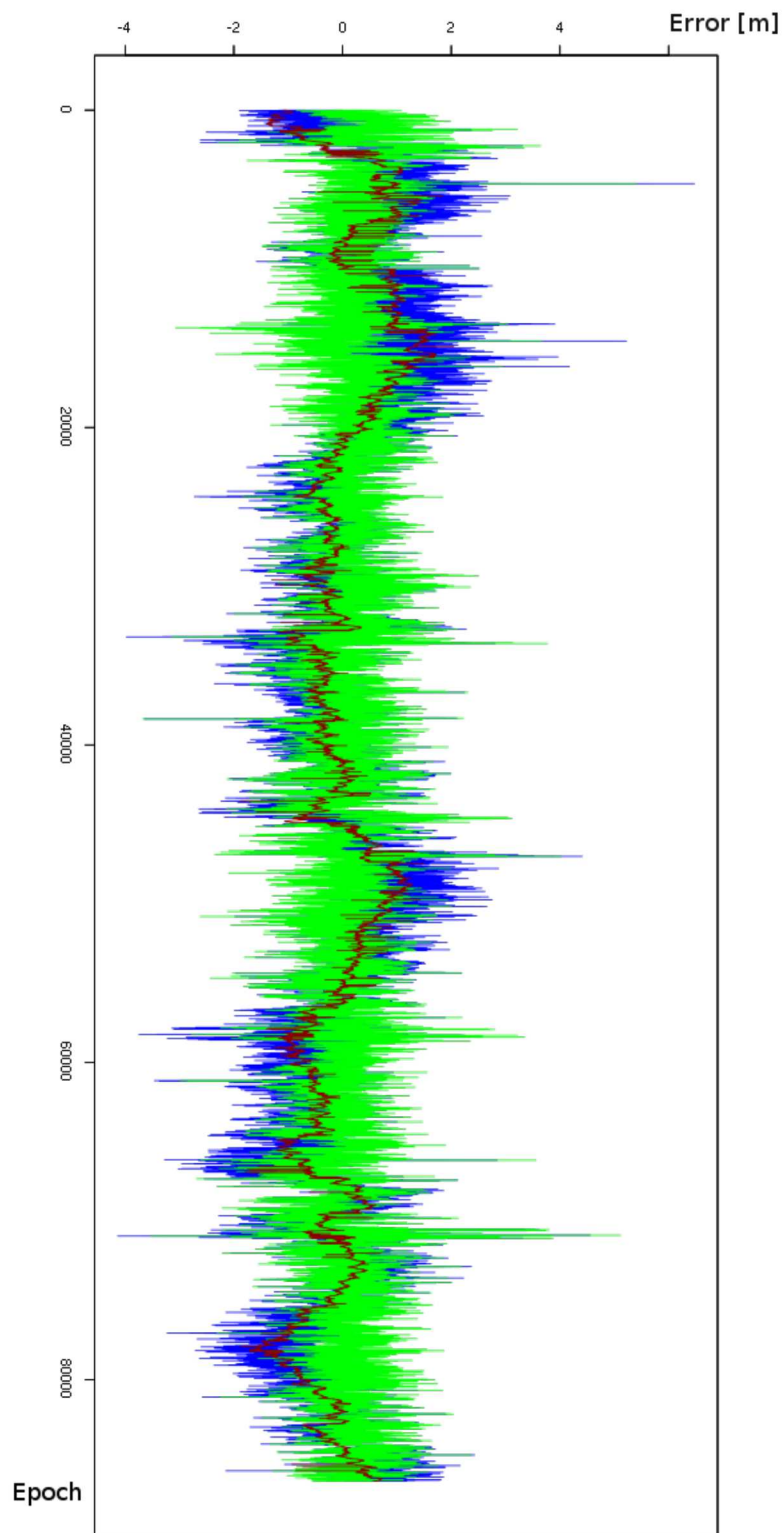


Figure C.13 – Statistical results for the **day 15-134** for the **east coordinate**: the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections

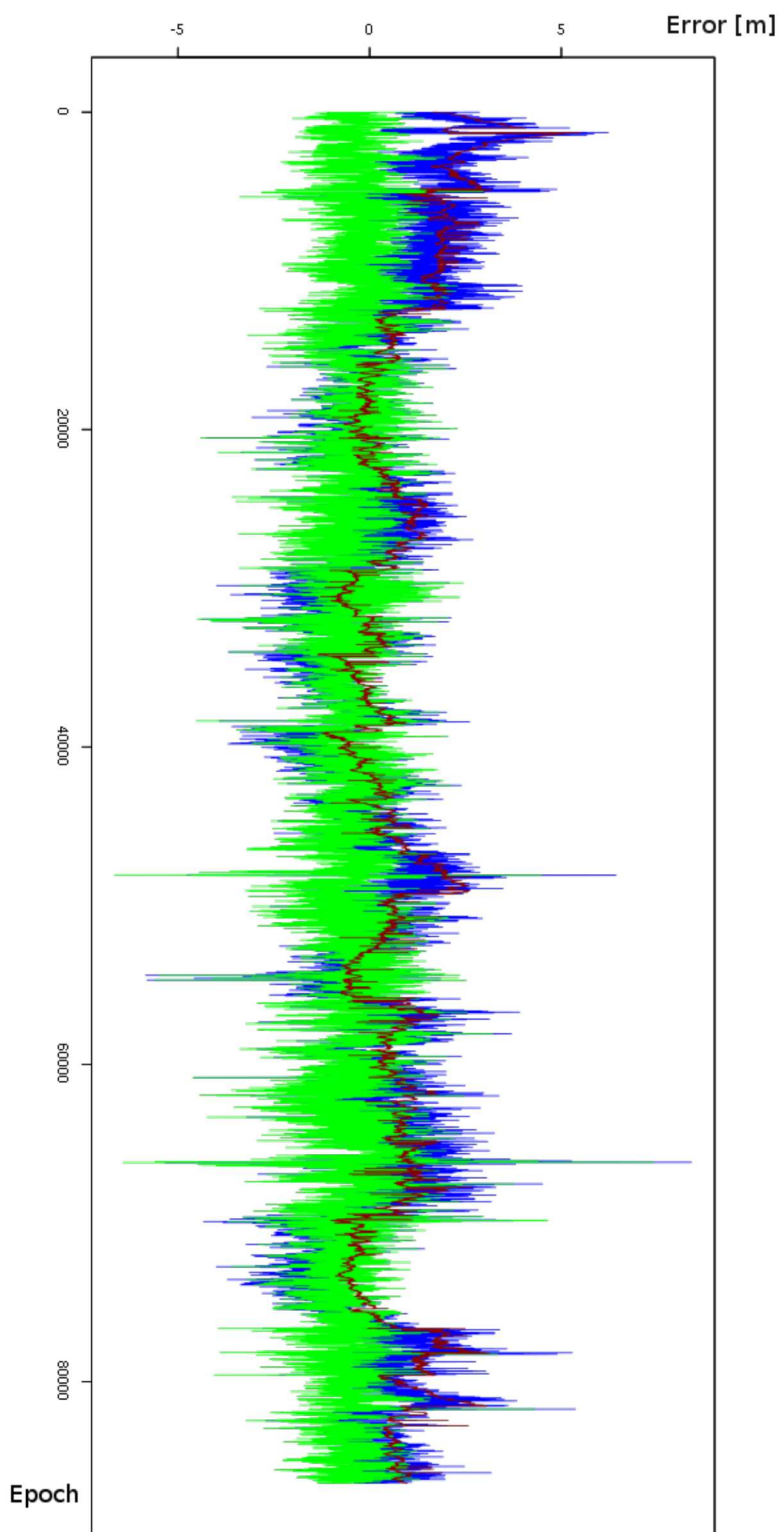


Figure C.14 – Statistical results for the **day 15-134** for the **north coordinate**: the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections

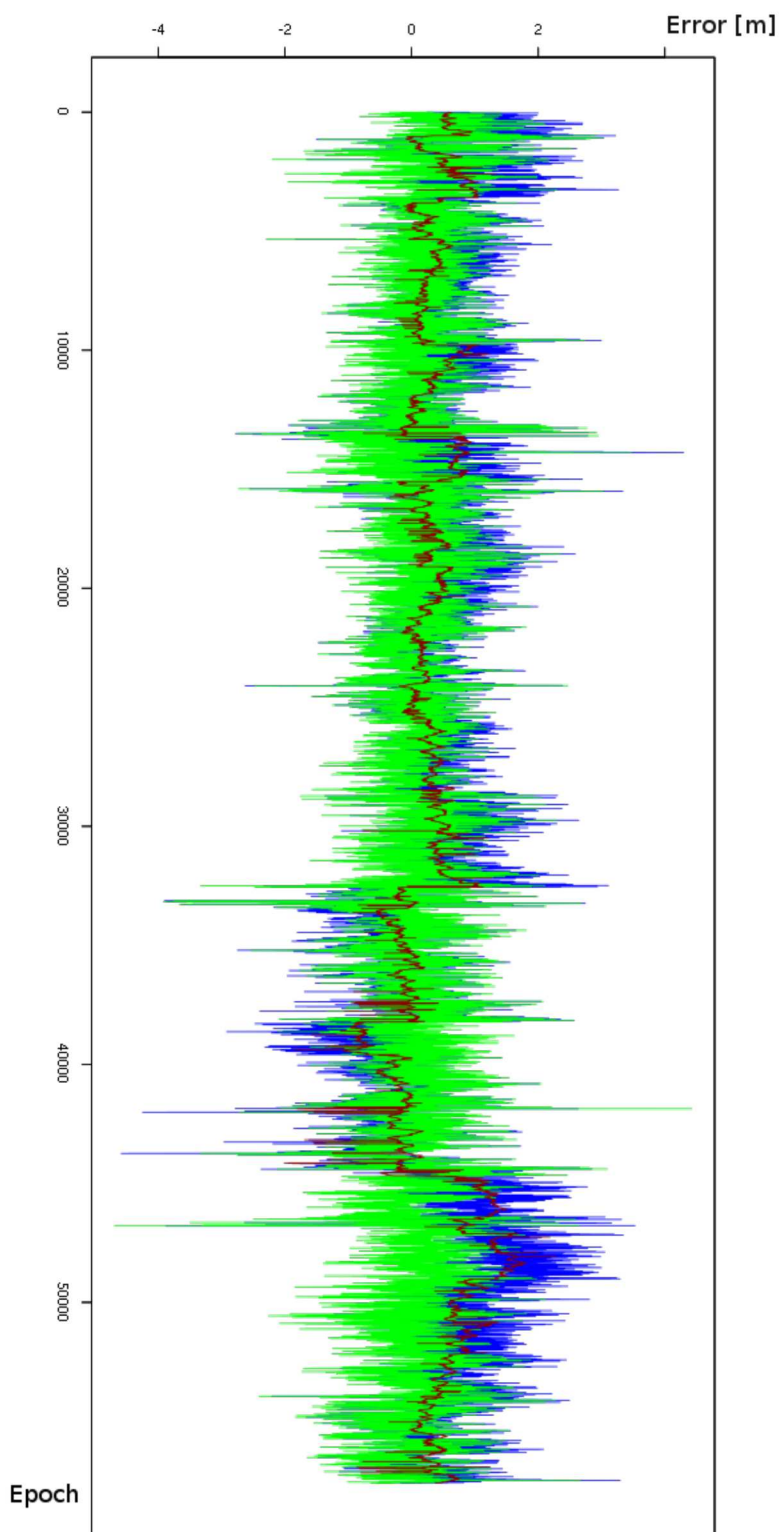


Figure C.15 – Statistical results for the **day 15-135** for the **east coordinate**: the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections

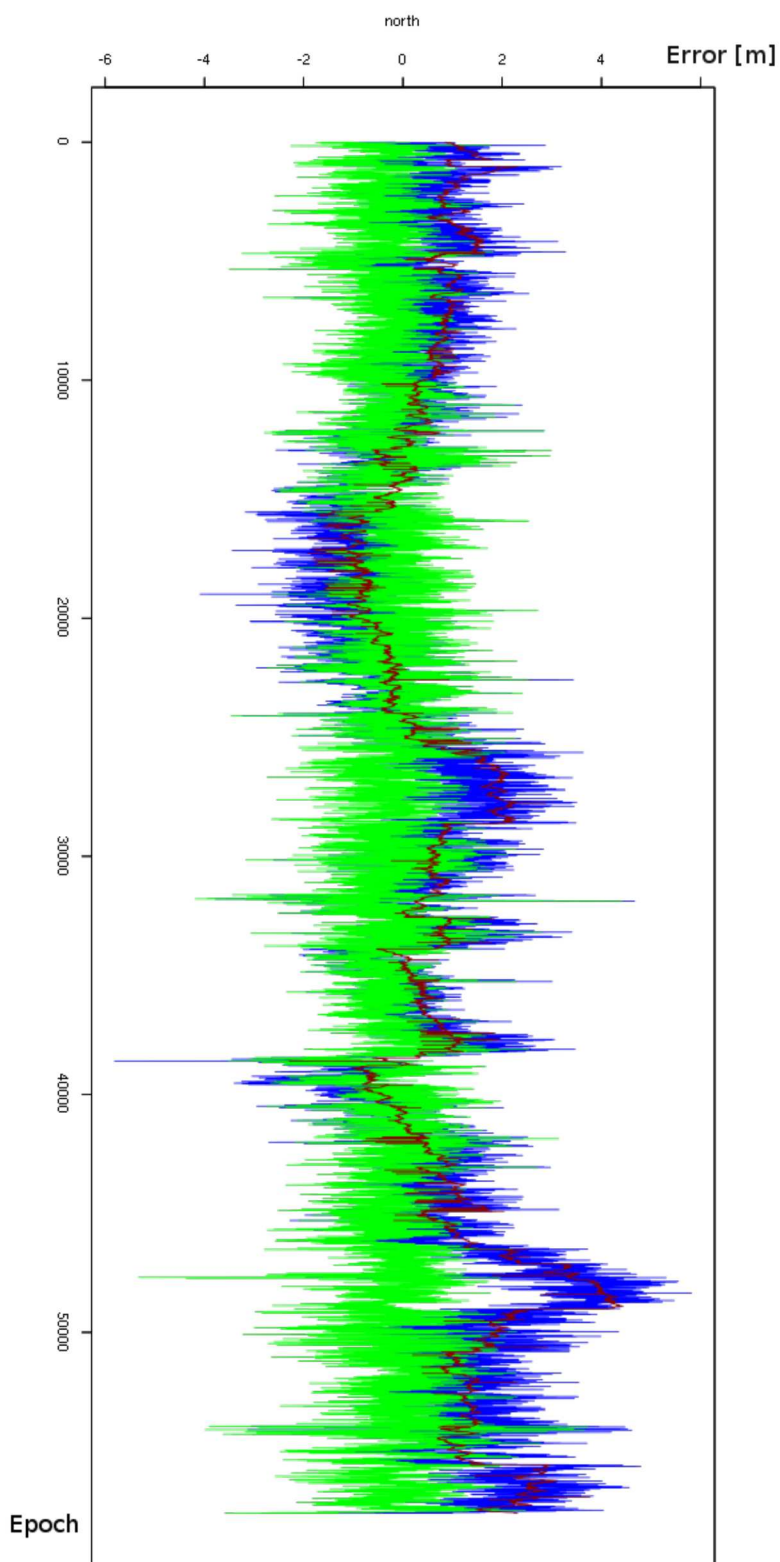


Figure C.16 – Statistical results for the **day 15-135** for the **north coordinate**: the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections



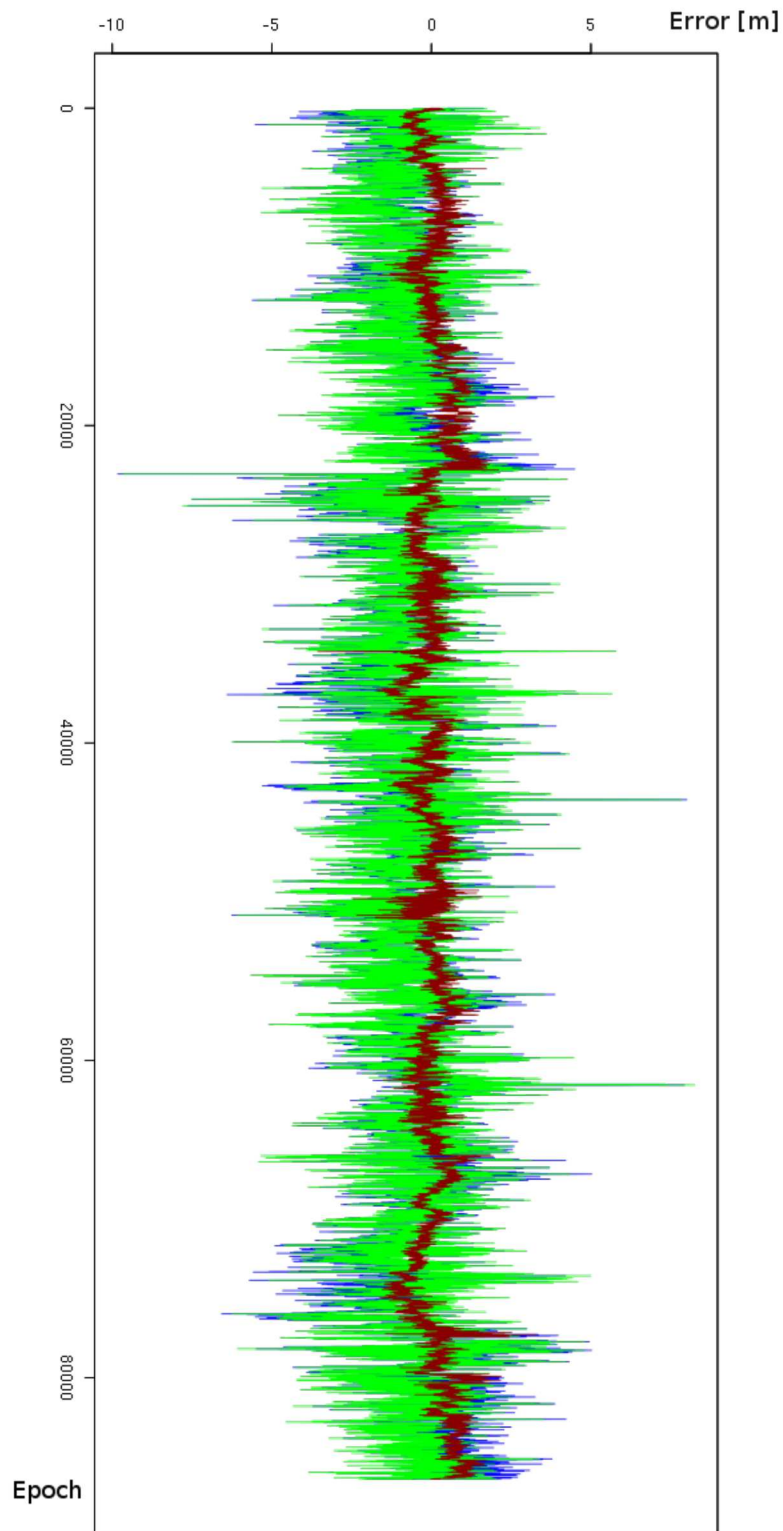


Figure C.17 – Statistical results for the **day 14-073** for the **east coordinate**: the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections



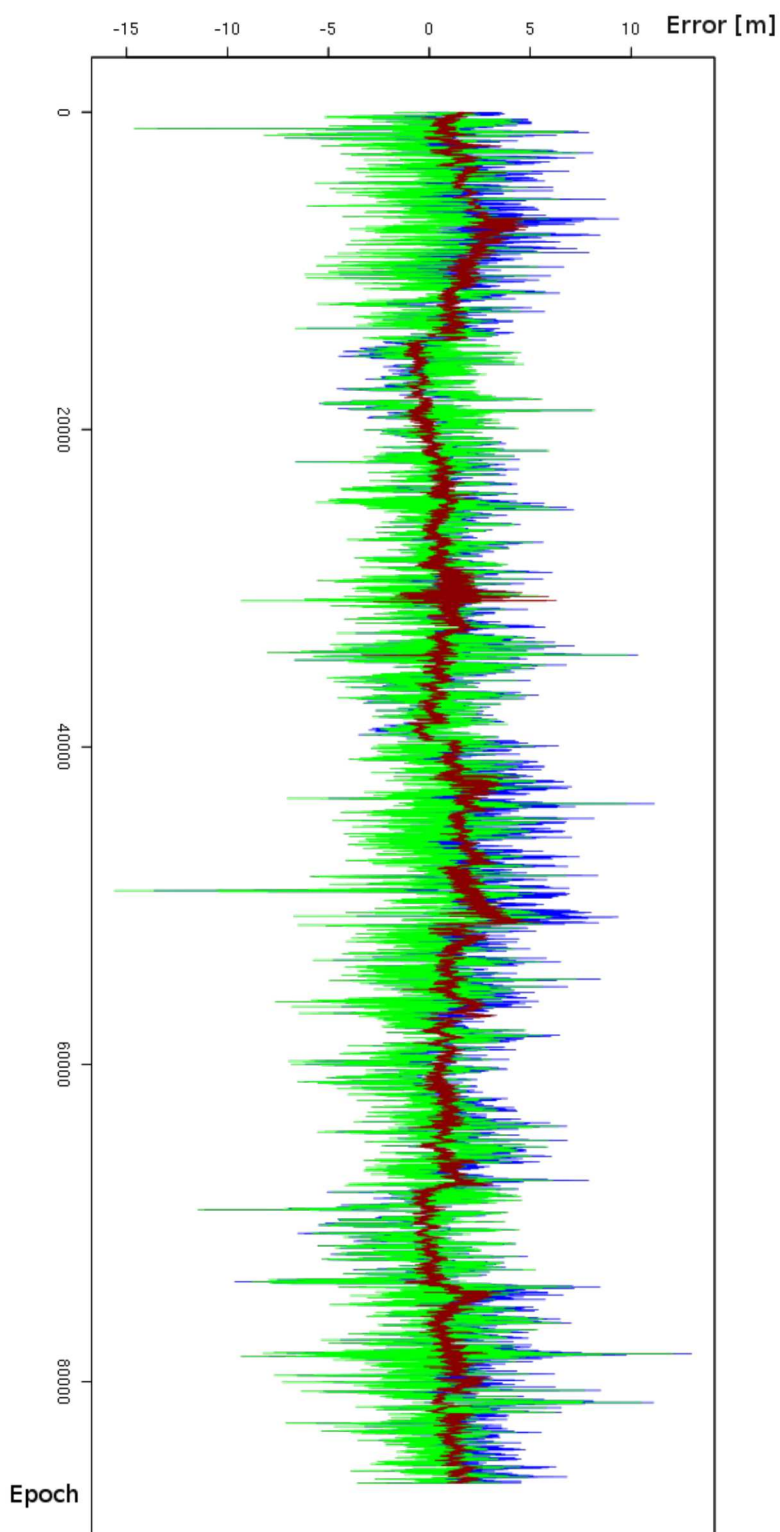


Figure C.18 – Statistical results for the **day 14-073** for the **north coordinate**: the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections

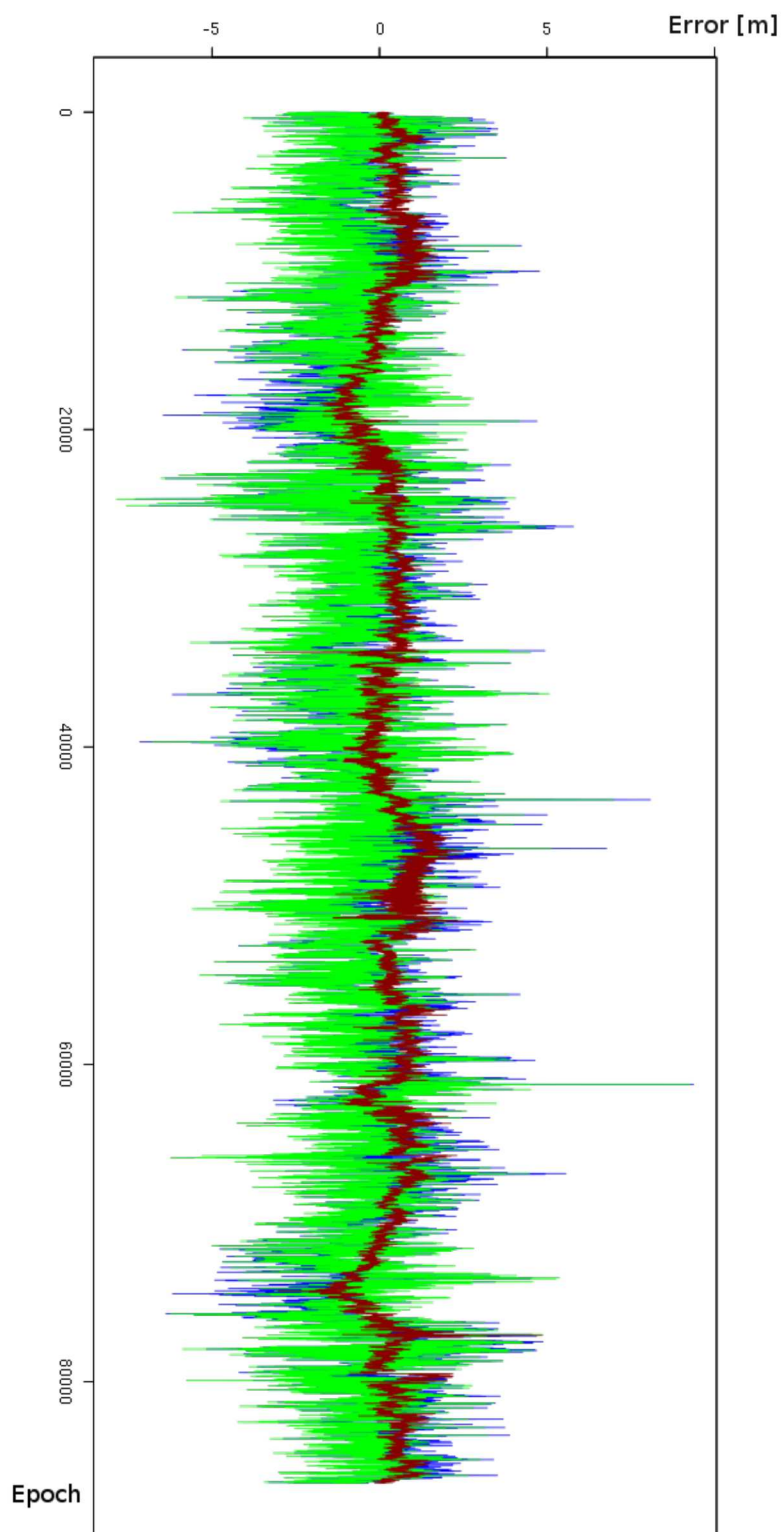


Figure C.19 – Statistical results for the **day 14-074** for the **east coordinate**: the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections

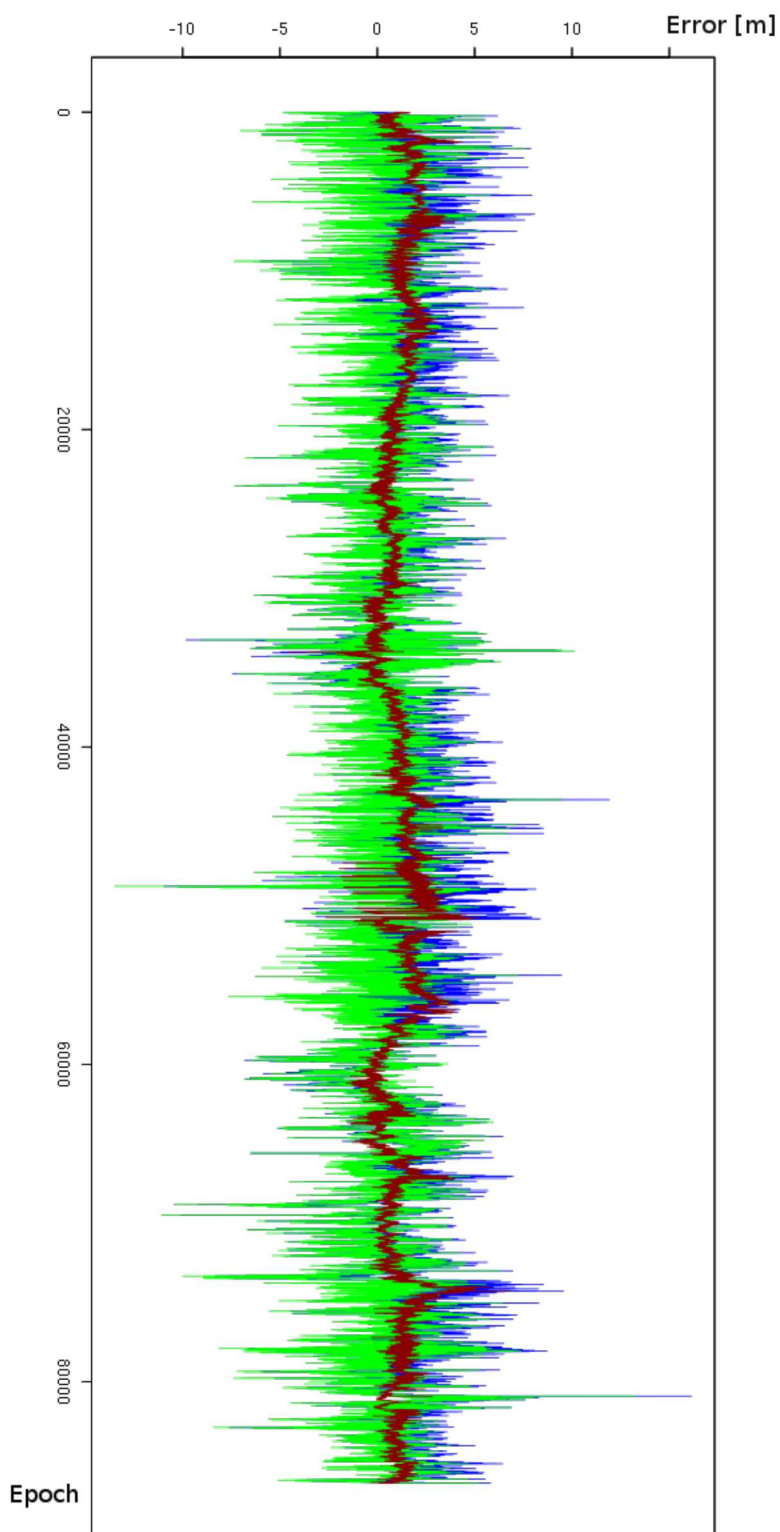


Figure C.20 – Statistical results for the **day 14-074** for the **north coordinate**: the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections

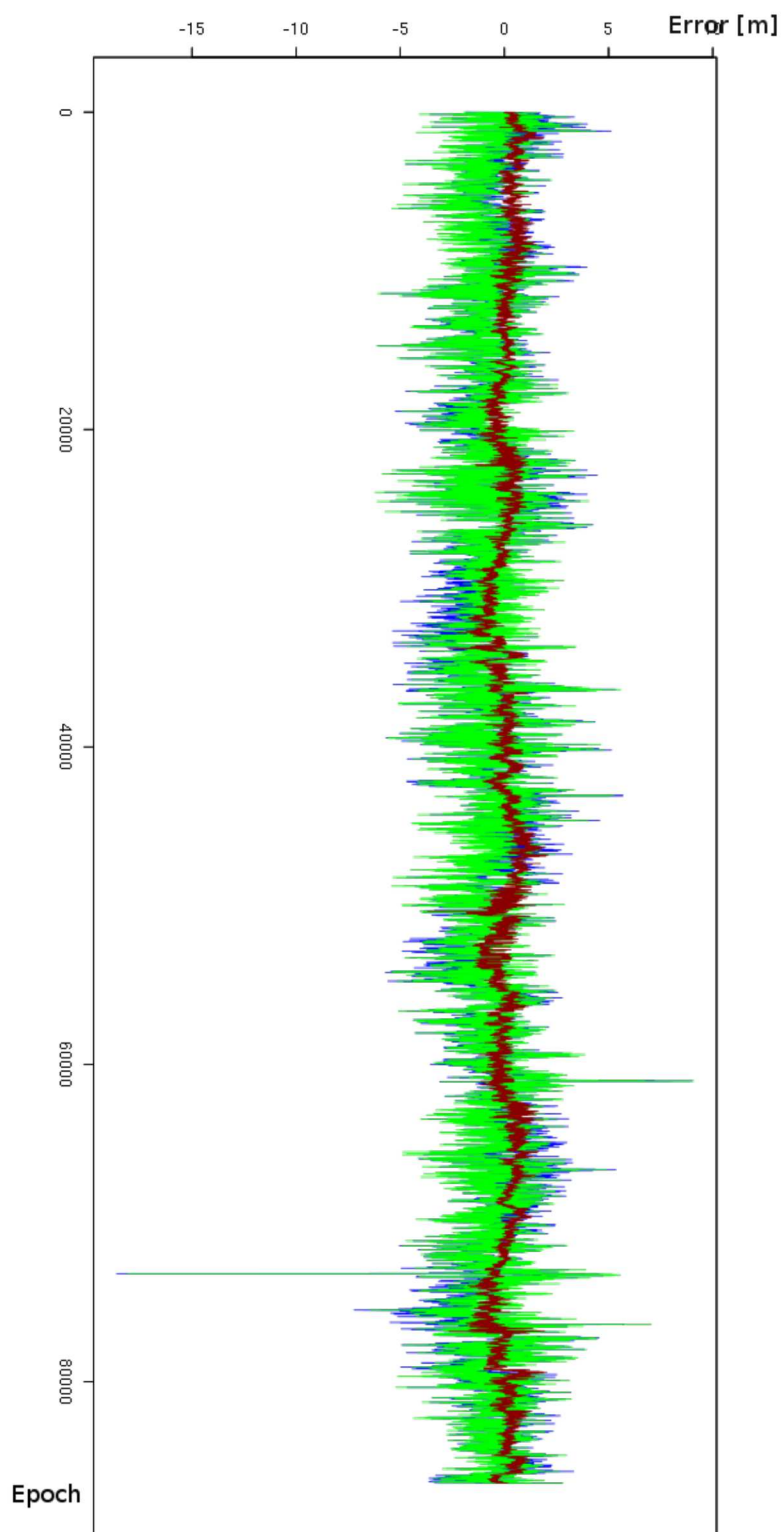


Figure C.21 – Statistical results for the **day 14-075** for the **east coordinate**: the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections

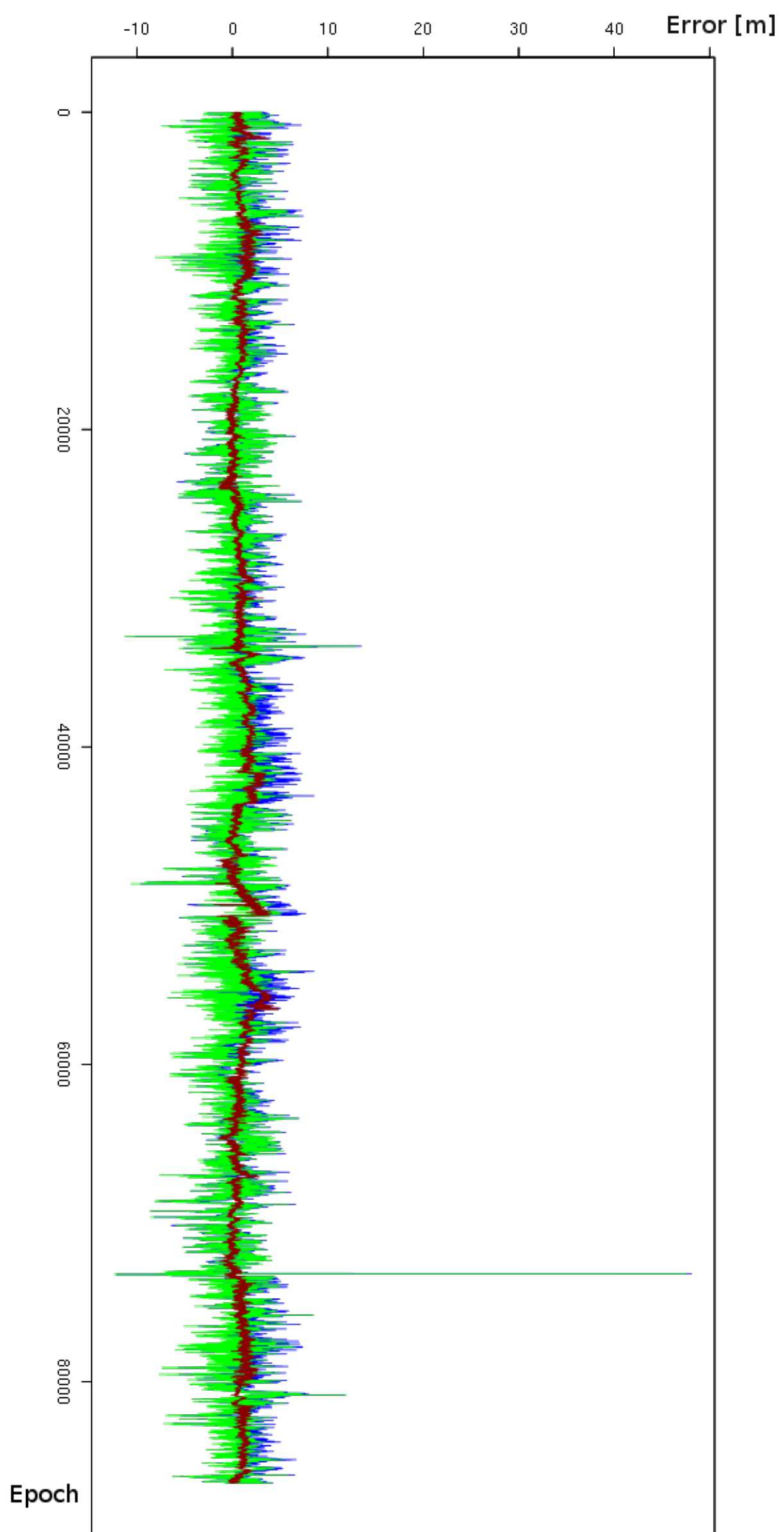


Figure C.22 – Statistical results for the **day 14-075** for the **north coordinate**: the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections

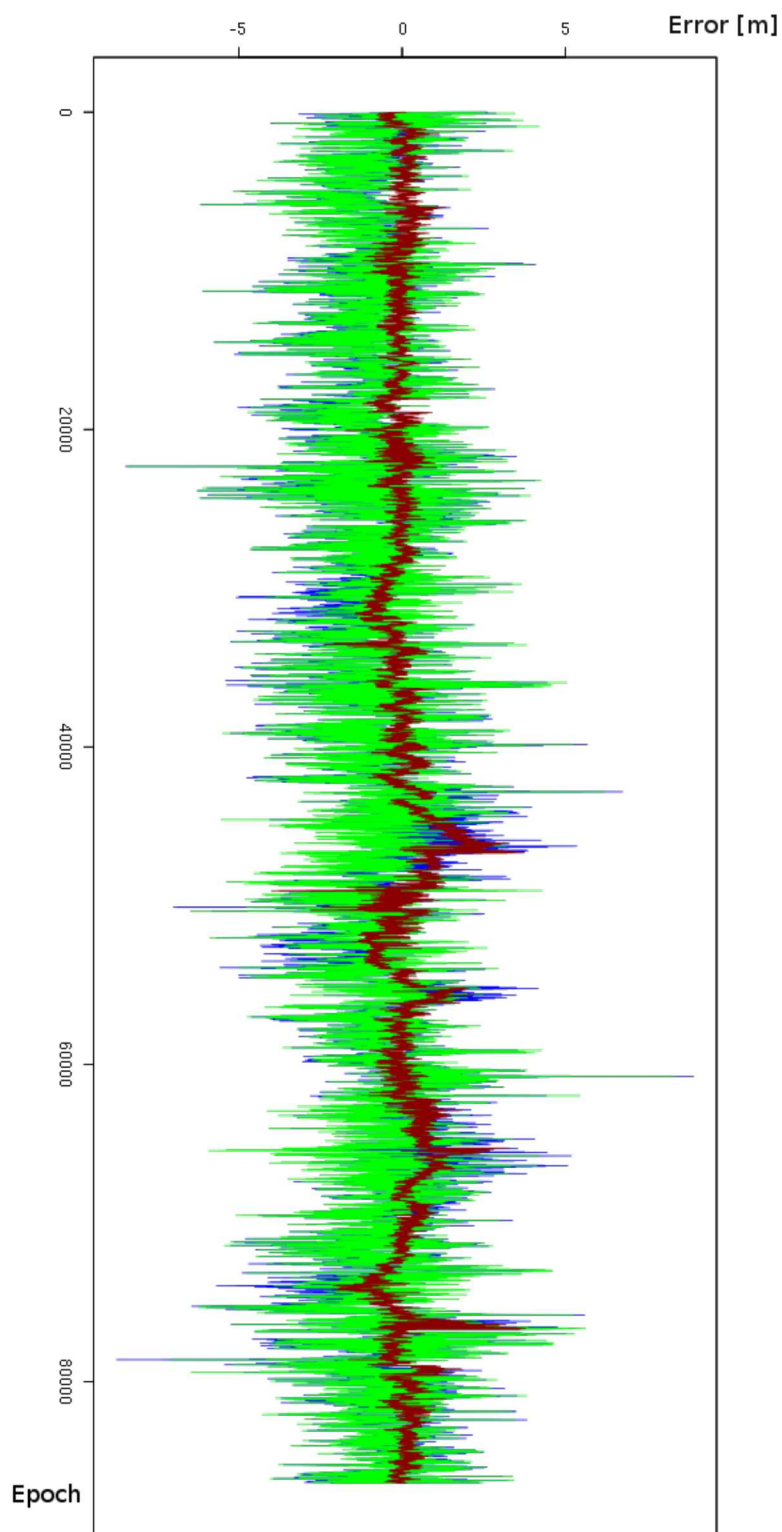


Figure C.23 – Statistical results for the **day 14-076** for the **east coordinate**: the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections



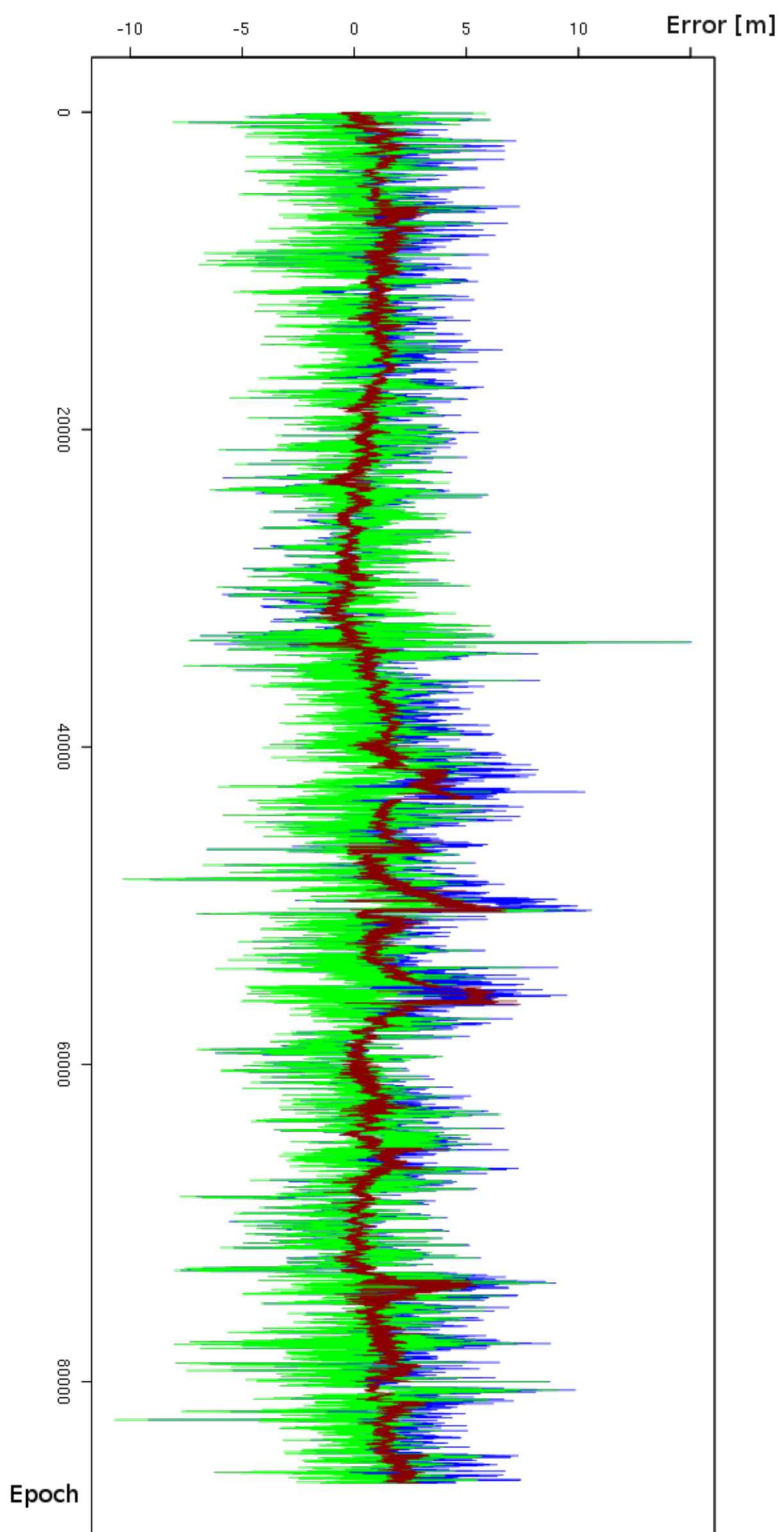


Figure C.24 – Statistical results for the **day 14-076** for the **north coordinate**: the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections



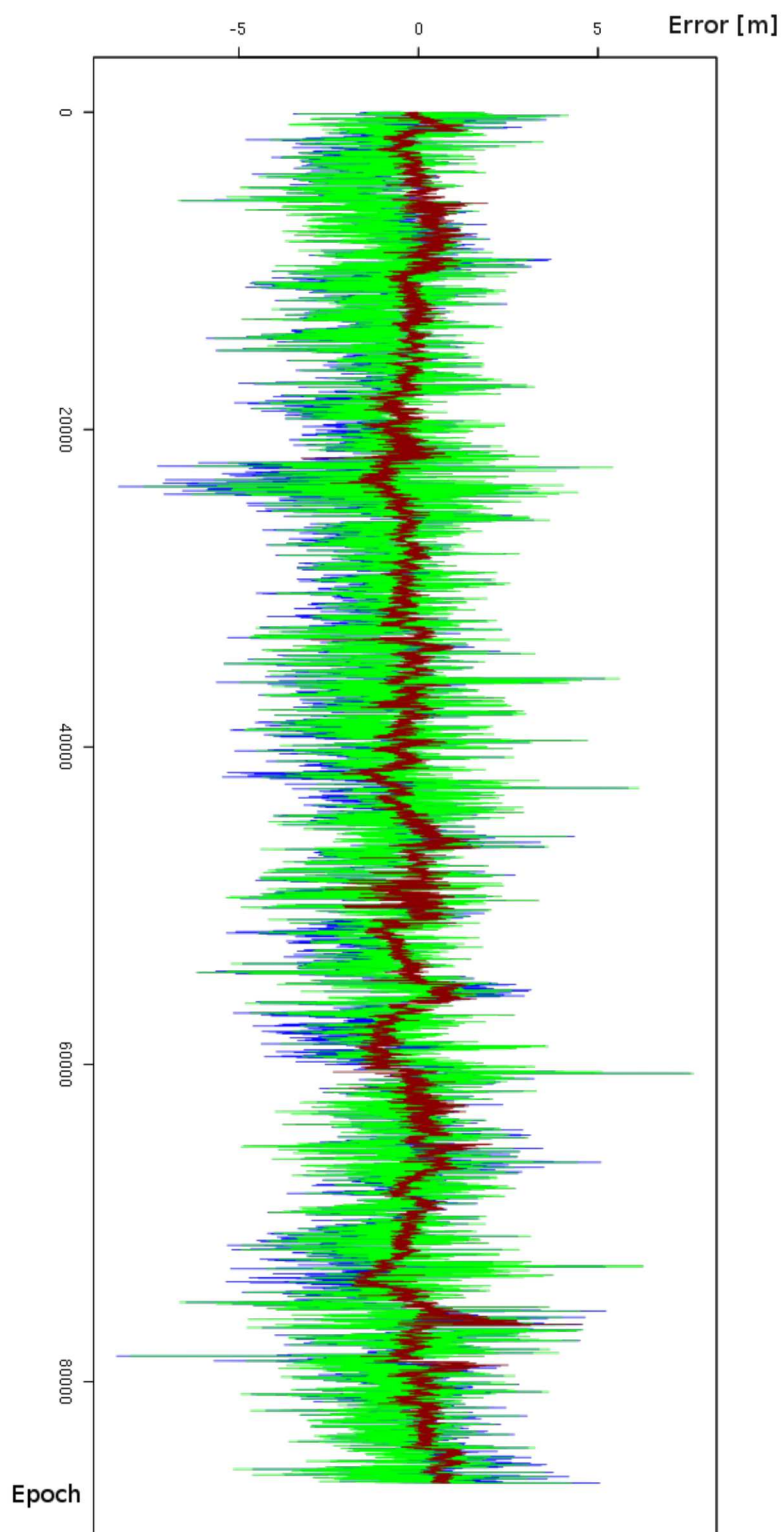


Figure C.25 – Statistical results for the **day 14-077** for the **east coordinate**: the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections

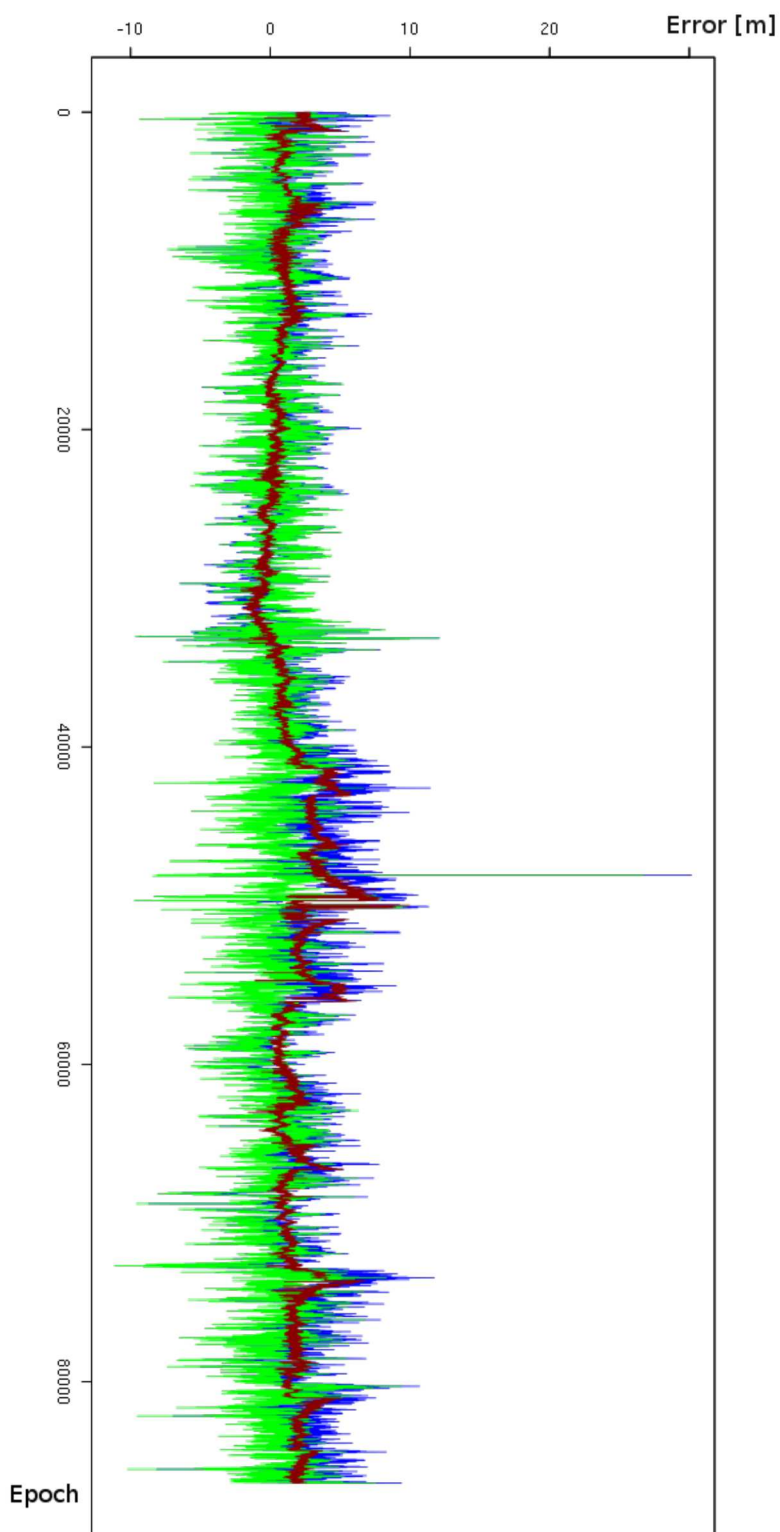


Figure C.26 – Statistical results for the **day 14-077** for the **north coordinate**: the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections

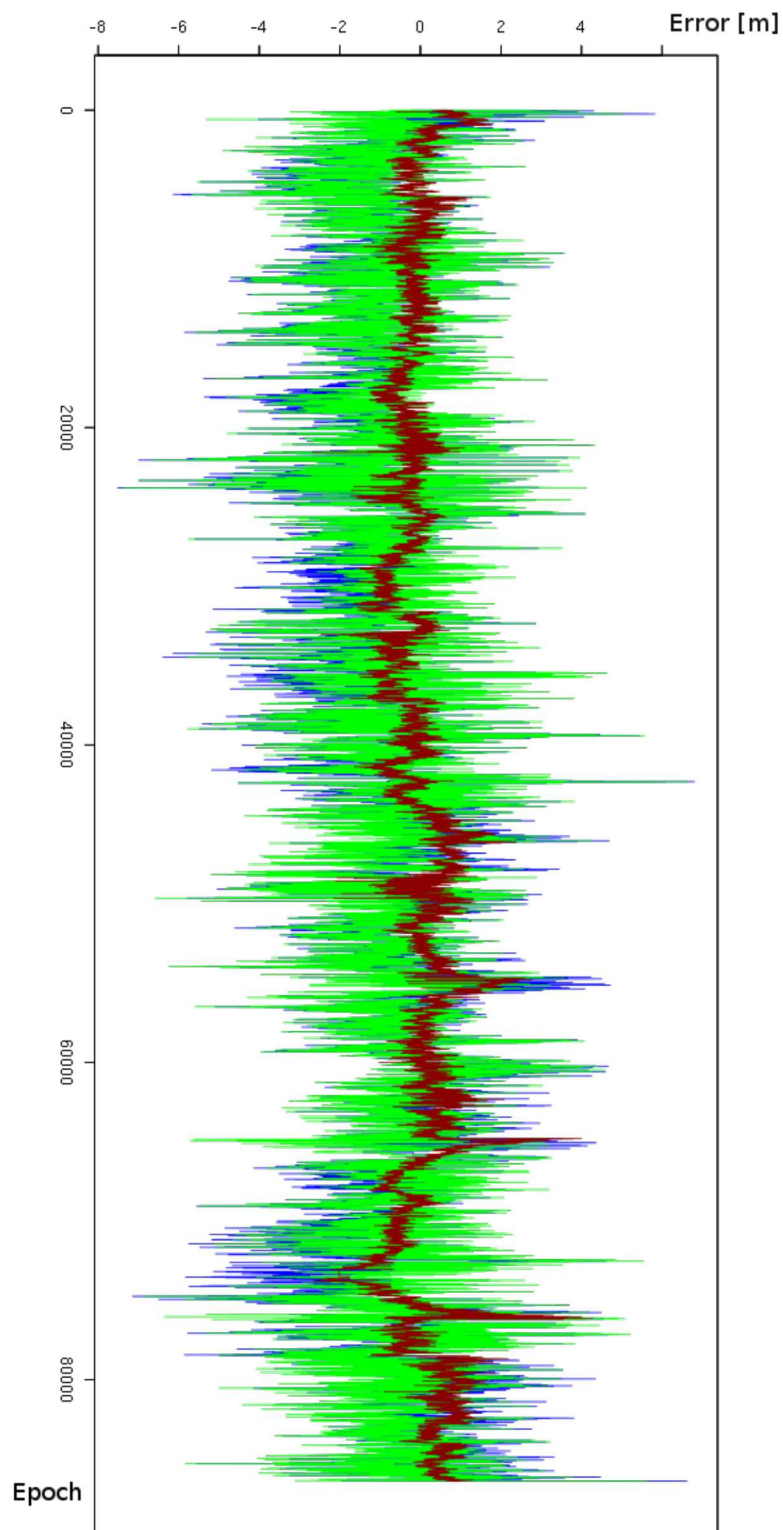


Figure C.27 – Statistical results for the **day 14-078** for the **east coordinate**: the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections

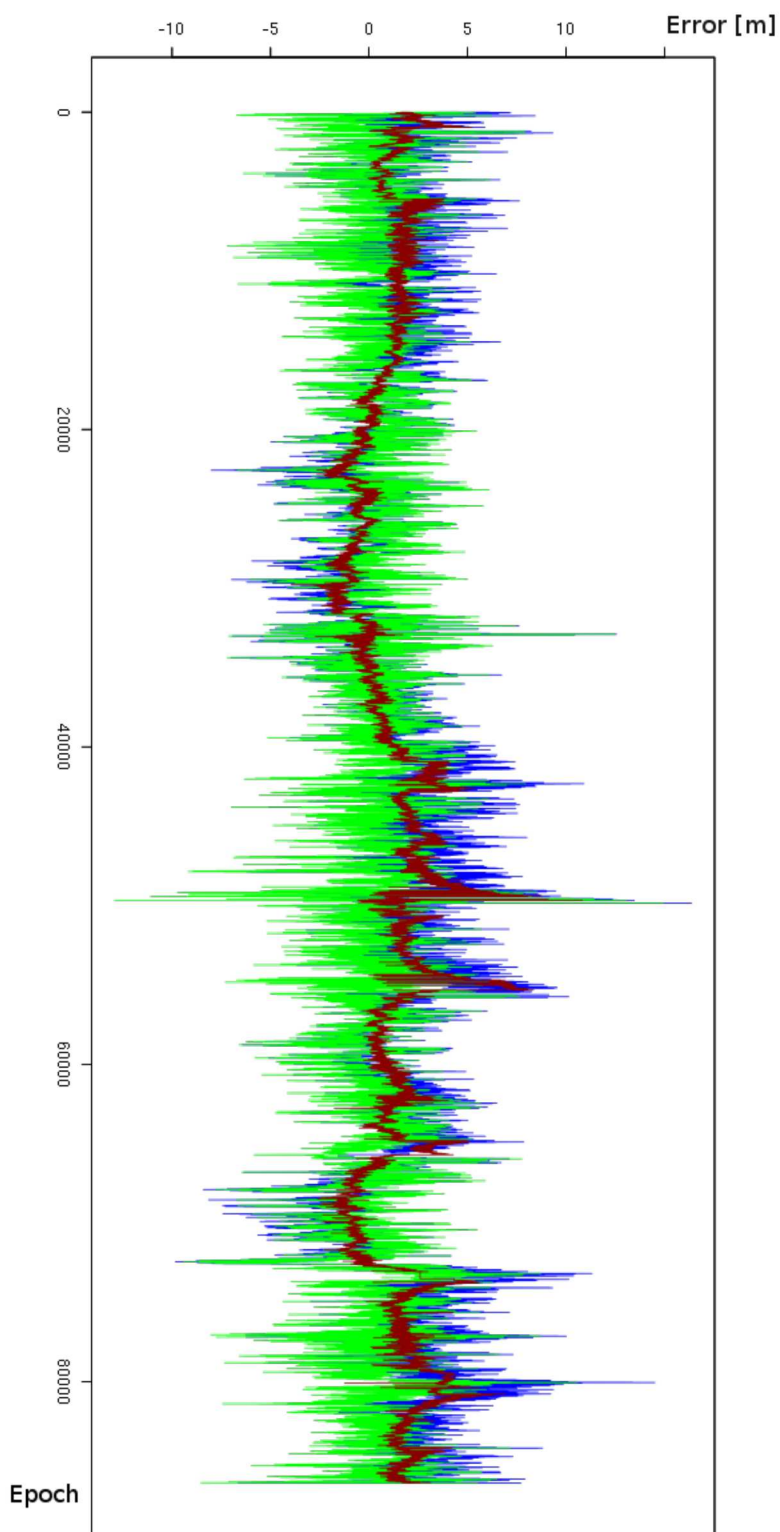


Figure C.28 – Statistical results for the **day 14-078** for the **north coordinate**: the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections

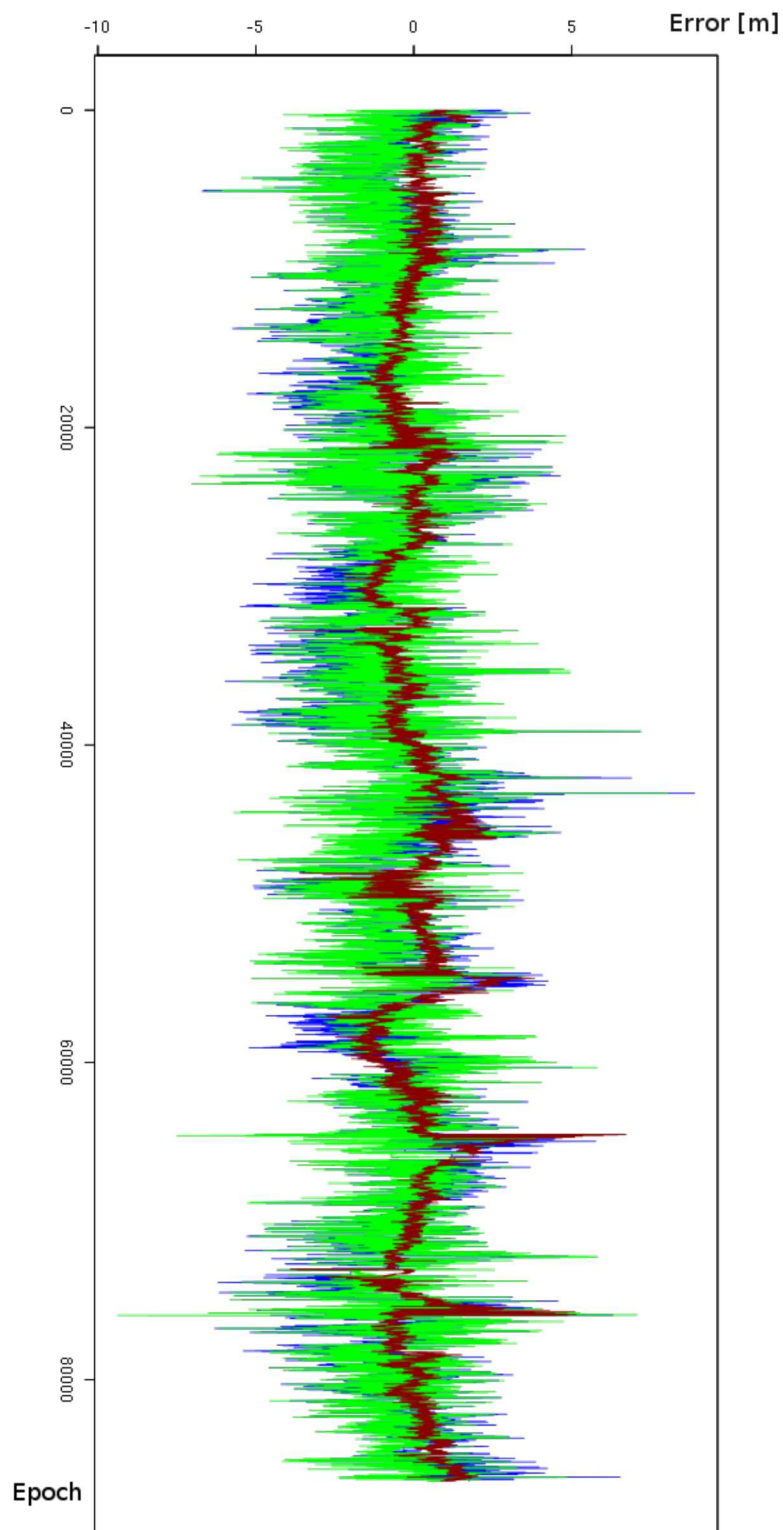


Figure C.29 – Statistical results for the **day 14-079** for the **east coordinate**: the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections

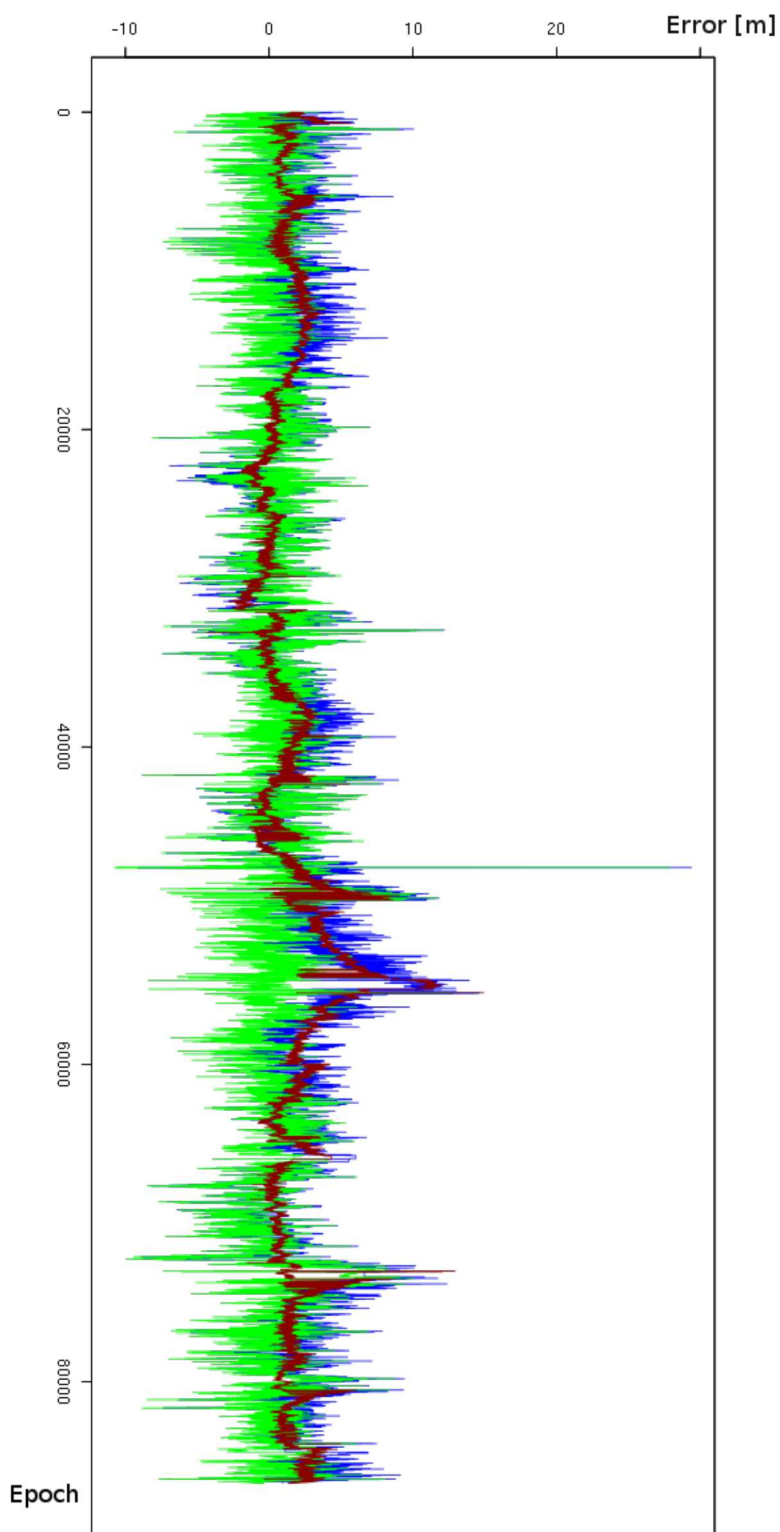


Figure C.30 – Statistical results for the **day 14-079** for the **north coordinate**: the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections





# References

## Bibliography

- [1] P. Baldi, N. Cenni, M. Fabris, and A. Zanutta. “Kinematics of a landslide derived from archival photogrammetry and GPS data”. In: *Geomorphology* 102 (3-4 2008), pp. 435–444. ISSN: 0169-555X.
- [2] BeiDou Navigation Satellite System Team. *System Introduction*. 2016. URL: <http://en.beidou.gov.cn/introduction.html> (visited on 2016-09-06).
- [3] T. Bellone, P. Dabove, A.M. Manzano, and C. Taglioretti. “Real-time monitoring for fast deformations using GNSS low-cost receivers”. In: *Geomatics, Natural Hazards and Risk* 7.2 (2016), pp. 458–470. DOI: 10.1080/19475705.2014.966867. URL: <http://dx.doi.org/10.1080/19475705.2014.966867>.
- [4] L. Biagi. *I fondamentali del GPS*. Italian. 2009. URL: <http://geomatrica.comopolimi.it/workbooks/n8/> (visited on 2016-10-04).
- [5] L. Biagi, B. Betti, M. Crespi, and F. Riguzzi. “GPS sensitivity analysis applied to non-permanent deformation control networks”. In: *Journal of Geodesy* 73 (1999), pp. 158–167.
- [6] L. Biagi and S. Caldera. “The automation of permanent networks monitoring: remarks and case studies”. In: *Applied Geomatics* 3 (3 2011), pp. 137–152.
- [7] L. Biagi, S. Caldera, and M.G. Visconti. “Data Quality and Coordinate Monitoring for a Permanent Network: Proposals and Experiences”. In: *20th International Technical Meeting of the Satellite Division of The Institute of Navigation* (2007), pp. 687–698.
- [8] L. Biagi, S. Comai, F.C. Grec, R. Mangiarotti, M. Matteucci, M. Negretti, C. Ugur, and M.G. Visconti. “The MEP project: Map for Easy Paths”. In: *Geomatics Workbooks*. 12. 2015, pp. 417–420.
- [9] L. Biagi, F.C. Grec, and M. Negretti. “Low-Cost GNSS Receivers for Local Monitoring: Experimental Simulation, and Analysis of Displacements”. In: *Sensors* 16 (12 2016). ISSN: 1424-8220. DOI: 10.3390/s16122140. URL: <http://www.mdpi.com/1424-8220/16/12/2140/html>.
- [10] L. Biagi, M. Negretti, and M.G. Visconti. “Contributi geomatici al progetto MEP (Map for Easy Paths)”. In: *Atti della XIX Conferenza Nazionale ASITA*. 2015.
- [11] M. Bierlaire, J. Chen, and J. Newman. “A probabilistic map matching method for smartphone GPS data”. In: *Transportation Research Part C: Emerging Technologies* 26 (2013), pp. 78–98.
- [12] P. Corti, T.J. Kraft, S.V. Mather, and B. Park. *PostGIS Cookbook*. 2014.

- [13] D.M. Cruden. “A Simple Definition of a Landslide”. In: *Bulletin International Association for Engineering Geology* 43 (1991), pp. 27–29. DOI: 10.1007/BF02590167.
- [14] R. Dach, E. Brockmann, S. Schaer, G. Beutler, M. Meindl, L. Prange, H. Bock, A. Jäggi, and L. Ostini. “GNSS Processing at CODE: Status Report”. In: *Journal of Geodesy* 83 (3 2009), pp. 353–365.
- [15] R. Dach, U. Hugentobler, P. Fridez, and M. Meindl. *Bernese GPS Software Version 5.0. Documentation*. Astronomical Institute, University of Bern. Bern, Switzerland, 2007.
- [16] ESA, European Commission. *Galileo Mission High Level Definition*. 2002. URL: [http://ec.europa.eu/dgs/energy\\_transport/galileo/doc/galileo\\_hld\\_v3\\_23\\_09\\_02.pdf](http://ec.europa.eu/dgs/energy_transport/galileo/doc/galileo_hld_v3_23_09_02.pdf) (visited on 2016-09-06).
- [17] European GSA. *GNSS market report*. Version 4. 2015. URL: [http://www.gsa.europa.eu/system/files/reports/GNSS-Market-Report-2015-issue4\\_0.pdf](http://www.gsa.europa.eu/system/files/reports/GNSS-Market-Report-2015-issue4_0.pdf) (visited on 2016-06-30).
- [18] Galileo GNSS Team. *Galileo initial services declaration*. 2016-12-18. URL: <http://galileognss.eu/galileo-initial-services-declaration/> (visited on 2017-01-19).
- [19] J.A. Gili, J. Corominas, and J. Rius. “Using Global Positioning System techniques in landslide monitoring”. In: *Engineering Geology* 55 (3 2000), pp. 167–192.
- [20] M.F. Goodchild. “Citizens as sensors: the world of volunteered geography”. In: *GeoJournal* 69 (4 2007), pp. 211–221. DOI: 10.1007/s10708-007-9111-y.
- [21] GPS World staff. *Galileo Achieves In-Orbit Validation*. 2014-10-02. URL: <http://gpsworld.com/galileo-achieves-in-orbit-validation/> (visited on 2016-09-06).
- [22] F.C. Grec. “Low Cost GNSS Receivers for Geodetic Monitoring”. Master. Politecnico di Milano, 2015. URL: [http://geomatica.como.polimi.it/tesi/MS\\_C\\_Grec.pdf](http://geomatica.como.polimi.it/tesi/MS_C_Grec.pdf) (visited on 2016-12-31).
- [23] W.H. Guier and G.C. Weiffenbach. “Genesis of satellite navigation”. In: *John Hopkins Technical Digest* 19.1 (1998). URL: <http://www.jhuapl.edu/techdigest/TD/t1901/> (visited on 2016-10-04).
- [24] J. Günther, O. Heunecke, S. Pink, and S. Schuhbäck. “Developments towards a low cost GNSS based sensor network for the monitoring of landslides”. In: (2008).
- [25] Information and Analysis Center for Positioning, Navigation and Timing, Koryolov, Russia. *GLONASS history*. 2016. URL: <https://www.glonass-iac.ru/en/guide/> (visited on 2016-08-16).
- [26] International Association of Oil & Gas Producers. *Surveying and Positioning Guidance Note. Using the EPSG geodetic parameter dataset*. 7. 2012. URL: <http://www.iogp.org/pubs/373-07-1.pdf>.
- [27] J. Klobuchar. “Ionospheric Time-Delay Algorithms for Single-Frequency GPS Users”. In: *IEEE Transactions on Aerospace and Electronic Systems* (3 1987-05), pp. 325–331.
- [28] R.B. Langley. “The GPS Observables”. In: *GPS World* 4.4 (1993), pp. 52–59.

- [29] Leica Geosystem AG. *Leica GPS 1200+ Series. Technical data*. 2008. URL: [http://w3.leica-geosystems.com/downloads123/zz/gps/general/brochures-datasheet/GPS1200\\_TechnicalData\\_en.pdf](http://w3.leica-geosystems.com/downloads123/zz/gps/general/brochures-datasheet/GPS1200_TechnicalData_en.pdf) (visited on 2016-11-03).
- [30] Leica Geosystems. *Leica Geo Office. Datasheet*. 2010. URL: [http://leica-geosystems.com/-/media/files/products/datasheets/leica\\_geo\\_office\\_ds.ashx?la=en](http://leica-geosystems.com/-/media/files/products/datasheets/leica_geo_office_ds.ashx?la=en) (visited on 2016-12-31).
- [31] A. Leick, L. Rapoport, and D. Tatarnikov. *GPS Satellite Surveying*. 4th ed. Wiley, 2015. ISBN: 978-1-118-67557-1.
- [32] J.P. Malet, O. Maquaire, and E. Calais. “The use of Global Positioning System techniques for the continuous monitoring of landslides: application to the Super-Sauze earthflow (Alpes-de-Haute-Provence, France)”. In: *Geomorphology* 43 (2002), pp. 33–54.
- [33] K.D. McDonald. “The Modernization of GPS: Plans, New Capabilities and the Future Relationship to Galileo”. In: *Journal of Global Positioning Systems* 1.1 (2002), pp. 1–17. URL: <http://www.gnss.com.au/JoGPS/v1n1/v1n1pA.pdf> (visited on 2016-10-03).
- [34] P. Mora, P. Baldi, G. Casula, M. Fabris, M. Ghirotti, E. Mazzini, and A. Pesci. “Global Positioning Systems and digital photogrammetry for the monitoring of mass movements: application to the Ca’ di Malta landslide (northern Apennines, Italy)”. In: *Engineering Geology* 68 (1 2003), pp. 103–121.
- [35] National Marine Electronics Association. *NMEA 0183 Standard*. 2008. URL: [http://www.nmea.org/content/nmea\\_standards/nmea\\_0183\\_v\\_410.asp](http://www.nmea.org/content/nmea_standards/nmea_0183_v_410.asp) (visited on 2016-12-13).
- [36] OpenStreetMap Foundation. *Licence*. 2016. URL: <http://wiki.osmfoundation.org/wiki/Licence> (visited on 2016-09-12).
- [37] S. Pace, G.P. Frost, I. Lachow, D.R. Frelinger, D. Fossum, D. Wassem, and M.M. Pinto. “GPS History, Chronology, and Budgets”. In: *The Global Positioning System. Assessing National Policies*. RAND Corporation, 1995. Chap. Appendix B, pp. 237–270. ISBN: 0-8330-2349-7. URL: [https://www.rand.org/content/dam/rand/pubs/monograph\\_reports/MR614/MR614.appb.pdf](https://www.rand.org/content/dam/rand/pubs/monograph_reports/MR614/MR614.appb.pdf).
- [38] K. Pearson. *Karl Pearson’s Early Statistical Papers*. Cambridge University Press, 1956, pp. 339–357.
- [39] PostGIS developpers. *PostGIS 2.2.5dev Manual*. 2016. URL: <http://postgis.net/stuff/postgis-2.2.pdf>.
- [40] M.A. Quddusa, W.Y. Ochiengb, and R.B. Nolandb. “Current map-matching algorithms for transport applications: State-of-the art and future research directions”. In: *Transportation Research Part C: Emerging Technologies* 15 (2007), pp. 312–328.
- [41] F. Ramm, J. Topf, and S. Chilton. *OpenStreetMap: Using and Enhancing the Free Map of the World*. 2010.
- [42] E. Realini. “goGPS—free and constrained relative kinematic positioning with low cost receivers”. Ph.D. Politecnico di Milano, 2009. URL: [http://geomatica.como.polimi.it/tesi/E\\_Realini.tar](http://geomatica.como.polimi.it/tesi/E_Realini.tar) (visited on 2016-05-13).

- [43] E. Realini and M. Reguzzoni. “goGPS: open source software for enhancing the accuracy of low-cost receivers by single-frequency relative kinematic positioning”. In: *Measurement Science and technology* 24.11 (2013).
- [44] S. Riggs, H. Krosing, G. Bartolini, and G. Ciolli. *PostgreSQL 9 Administration Cookbook*. 2015.
- [45] V. Rizzo. “GPS monitoring and new data on slope movements in the Maratea Valley (Potenza, Basilicata)”. In: *Physics and Chemistry of the Earth* 27 (36 2002), pp. 1535–1544.
- [46] J. Saastamoinen. “Contributions to the theory of atmospheric refraction. Part II. Refraction corrections in satellite geodesy”. In: *Bulletin Géodésique* 107 (1973-05), pp. 13–34.
- [47] T. Takasu and A. Yasuda. “Development of the Low-cost RTK-GPS Receiver with an Open Source Program Package RTKLIB”. In: *Proceedings of International Symposium on GPS/GNSS*. Ed. by International Convention Centre. 2009.
- [48] u-blox Team. *LEA-4T. Antaris (R)4 GPS module with Precision Timing*. Data Sheet. 2009. URL: [http://www.datasheetlib.com/datasheet/1263862/lea-4t\\_u-blox.html](http://www.datasheetlib.com/datasheet/1263862/lea-4t_u-blox.html) (visited on 2016-12-31).
- [49] u-blox Team. *NEO-7P. u-blox 7 Precise Point Positioning GNSS module*. Data Sheet. Version 4. 2014. URL: [https://www.u-blox.com/sites/default/files/products/documents/NEO-7P\\_DataSheet\\_\(UBX-13003787\).pdf](https://www.u-blox.com/sites/default/files/products/documents/NEO-7P_DataSheet_(UBX-13003787).pdf) (visited on 2016-09-26).
- [50] US Government. *GPS Precision Positioning Service (PPS) Performance Standard*. Version 1. 2007. URL: <http://www.gps.gov/technical/ps/2007-PPS-performance-standard.pdf> (visited on 2016-09-08).
- [51] US Government. *GPS Standard Positioning Service (SPS) Performance Standard*. Version 4. 2008. URL: <http://www.gps.gov/technical/ps/2008-SPS-performance-standard.pdf> (visited on 2016-09-08).
- [52] US Government. *Interface Specification IS-GPS-200*. Version H. 2013. URL: <http://www.gps.gov/technical/icwg/IS-GPS-200H.pdf> (visited on 2016-07-11).
- [53] US Government. *Interface Specification IS-GPS-705*. Version D. 2013. URL: <http://www.gps.gov/technical/icwg/IS-GPS-705D.pdf> (visited on 2016-07-11).

## Cartography

In many figures they were used as background cartographic data from two main sources:

- Orthophoto AGEA 2012 from WMS service of Regione Lombardia:  
[http://www.cartografia.regione.lombardia.it/ArcGIS10/services/wms/ortofoto2012\\_wms/MapServer/WMSServer](http://www.cartografia.regione.lombardia.it/ArcGIS10/services/wms/ortofoto2012_wms/MapServer/WMSServer)
  - Chapter 2: figures 2.6, 2.7, 2.21, 2.24, 2.25, 2.26, 2.27, 2.28, 2.29, 2.30, 2.31;
  - Chapter 4: figure 4.5 and 4.11.
- OSM data, ©OpenStreetMap contributors:
  - Chapter 1: figures 1.2, 1.3, 1.5, 1.6, 1.8;

- Chapter 2: figures 2.14, 2.15, 2.22.
- Chapter 3: figures 3.6, 3.10, 3.13, 3.14, 3.15, 3.18, 3.19, 3.20, 3.21, 3.22, 3.24, 3.25, 3.26, 3.27, 3.28, 3.29, 3.30.



# List of figures

1.1	The generation of GPS signals . . . . .	9
1.2	Coverage extension of BeiDou-1 navigation system . . . . .	12
1.3	Coverage extension of BeiDou-2 navigation system (2012) . . . . .	13
1.4	Comparison between PRN receiver code and PRN satellite code . . . . .	14
1.5	The red line represents the sphere where can be a GNSS receiver distant $\rho$ from the satellite in figure. . . . .	15
1.6	Positioning a GNSS receiver . . . . .	16
1.7	comparison between receiver and satellite phases . . . . .	18
1.8	integer ambiguity . . . . .	18
1.9	Single Differences method . . . . .	20
1.10	Double Differences method . . . . .	22
1.11	Multipath: example of overestimated path . . . . .	23
1.12	Example of NMEA produced during the survey done . . . . .	25
2.1	Organization schema of MEP project . . . . .	29
2.2	An example of erroneous positioning: the green diamond is the actual position of the obstacle, but the position detected is not at the right side of the street (red diamond) . . . . .	29
2.3	The green triangle is the actual position of CORS, the yellow circles are its estimated positions epoch by epoch. It can be assumed that a similar trend affect the rover: the orange triangle is the unknown actual position, than the orange circles are its estimated positions epoch by epoch . . . . .	32
2.4	Example of different visibility of the satellites between CORS and rover: the circles are the satellites viewed by CORS, the orange circles are the subset of satellites viewed by the rover . . . . .	32
2.5	Flow chart followed to implement <i>ExtractRinex</i> software . . . . .	33
2.6	The positions of MILA CORS (Red triangle) and the rover (yellow circle); the distance of the two points is less than 80 meters. . . . .	35
2.7	The positions of COMO CORS (Green triangle) and the rover (yellow circle); the distance of the two points is less than 150 meters. . . . .	35
2.8	The setup of test point in Como . . . . .	37
2.9	The rack used to power the devices using during the test . . . . .	38
2.10	A detail, inside the rack: a PC was used to power u-blox and collect data from it . . . . .	39
2.11	The waterproof box used to protect the u-blox during the survey . . . . .	39



2.12	Statistical results for the day 15-130 for the north coordinate: the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections . . . . .	40
2.13	Parameters used in LGO software to process data from geodetic receiver (extract of LGO report) . . . . .	44
2.14	Path 1: urban canyon (red), area of fair visibility (yellow), open areas (green) . . . . .	45
2.15	Path 2 . . . . .	46
2.16	Example of urban canyon: Luini Street, in the old town . . . . .	47
2.17	Example of area with fair visibility: XX settembre street . . . . .	48
2.18	Example of open area: breakwater Caldirola . . . . .	48
2.19	The pack setup with the instruments used for the tests . . . . .	49
2.20	The position of geodetic antenna (red circle) and u-blox antenna (red circle) during the test; smartphones was taken by hand . . . . .	49
2.21	Yellow triangles are the points estimated with the geodetic receiver, red circles are the positions collected with the low cost receiver during the same survey. The different of positioning between points at the same epoch is the error (the orange dotted line) . . . . .	50
2.22	Systematic error that occurred in every repetition (14-259 red line, 14-267 green line, 14-268 blue line) of survey along Path 1 The interested area is Gerbetto square . . . . .	52
2.23	The building in Gerbetto square that can be the possible source of the systematic error . . . . .	53
2.24	Path 2 detected with smartphone - Day 16-208 . . . . .	56
2.25	Path 2 detected with u-blox - Day 16-208 . . . . .	56
2.26	Comparison between data collected with smartphone in different days in a open area: 16-139 (green circles), 16-207 (red circles), 16-208 (yellow circles). The actual path are the blue triangles . . . . .	58
2.27	Comparison between data collected with smartphone in different days in a area partially covered by trees: 16-139 (green circles), 16-207 (red circles), 16-208 (yellow circles). The actual path are the blue triangles . . . . .	58
2.28	Detail of the area in front of the lake with good exposition: points collected with geodetic receiver (green triangles), u-blox (blue circles) and smartphone (yellow circles) - Day 16-208 . . . . .	59
2.29	Detail of the area near War Memorial: points collected with geodetic receiver (green triangles), u-blox (blue circles) and smartphone (yellow circles) - Day 16-208 . . . . .	59
2.30	Detail of an area with bad exposition: points collected with geodetic receiver (green triangles when the ambiguity solution was fixed, red triangles when it was not fixed), u-blox (blue circles) and smartphone (yellow circles) - Day 16-208 . . . . .	60
2.31	Data collected with u-blox: before (light blue circles) and after the CORS corrections (brown square) and actual point (green triangles) - Day 16-208 . . . . .	60

3.1	Example of wrong point positioning: the green circles are the points correctly positioned, the red one is inside a building (grey rectangle) and so in wrong position. The circles with question mark are two possible right positions . . . . .	62
3.2	List of some rows of <i>gpsdata</i> table . . . . .	64
3.3	List of some rows of <i>highway_routes</i> table . . . . .	65
3.4	List of some rows of <i>building_routes</i> table . . . . .	65
3.5	The implemented architecture scheme . . . . .	66
3.6	Correction on the nearest segment: the reds circles are positions of the points collected, the green circles are the positions after the correction to the nearest path . . . . .	67
3.7	Data organization in table <i>highway.routes</i> : each vector feature (lines with the same color) is composed by one or more segment . . . . .	68
3.8	Data organization in table <i>highway.segment</i> : each vector feature is composed by only one segment . . . . .	68
3.9	List of some rows of <i>highway.segment</i> table . . . . .	70
3.10	Correction using several indicators in choosing of "pertinent segment": the red circles are positions of the estimated points, the blue squares are the positions after the correction . . . . .	72
3.11	List of some rows of <i>building</i> table . . . . .	75
3.12	List of some rows of <i>build.segment</i> table . . . . .	75
3.13	The circle black is the estimated point, the white circles are all the possible corrections, between these, the blue circle is the correction chosen . . . . .	76
3.14	Example of correction "out of the buildings" effective . . . . .	77
3.15	Example of correction "out of the buildings" ineffective, the red arrows show the areas most critical . . . . .	77
3.16	Example of segment selection procedure. P1-P4 estimated points, S1-S5 segments (S1-S5), with the associated score (blue numbers) . . . . .	79
3.17	List of some rows of <i>candidate_path</i> table . . . . .	82
3.18	The red lines are the candidate segments with the score received, black circles are the estimated points - Day 14-259 . . . . .	82
3.19	The yellow-red lines are the selected segments; the green hatch areas represent the buffer - Day 14-259 . . . . .	83
3.20	Correction of points positions outside the buffer . . . . .	83
3.21	Example of "correction on buffer" compared with the "out of the buildings" method. The green arrows indicate where the method has been more efficient, the red arrow indicates where there are still critics results - Day 14-259 . . . . .	84
3.22	Example of "correction on buffer variable" method compared with the "correction on buffer constant" method. The green arrow indicates where the "correction on buffer variable" method has been more efficient - Day 14-259 . . . . .	85
3.23	An example of urban canyons: Cantù street, where the erroneous segment classification shown in figure 3.24 was verified . . . . .	88

3.24	Example of wrong classification for an handling segment: the segment indicated by the red arrow was added to the "selected segments" group, but it shouldn't be added - Day 14-259 . . . . .	89
3.25	Example of wrong classification: the segment indicated by the red arrow was not added to the "selected segments" group, but it should be added - Day 14-259 . . . . .	89
3.26	Example of reclassification. After first classification step, the red segments are added to the candidate group, the black segments are the discarded. At the second step the score of segment A will be reduce and the segment B will be added to the candidate group . .	90
3.27	The same path in figure 3.24, but now after the reclassification procedure: the wrong segment is not longer selected - Day 14-259 . .	90
3.28	The same path in figure 3.25, but now after the reclassification procedure: the missing segment is now selected - Day 14-259 . . . .	91
3.29	Compare between "path segments as reference" and "correction on buffer variable" methods. The light blue area is the tolerance area of <b>2.5 meters</b> around the actual path (green line), while the red area is the buffer built around the segment path . . . . .	94
3.30	Compare between "path segments as reference" and "correction on buffer variable" methods. The green area is the tolerance area of <b>5 meters</b> around the actual path, while the red area is the buffer built around the segment path . . . . .	94
4.1	The positions of Milan CORS (Red triangle) and the rover (yellow circle); the distance is less than 80 meters. . . . .	97
4.2	Experiment 1. North residuals of hourly sessions. BSW in blue, LGO in red . . . . .	100
4.3	Experiment 1. East residuals of hourly sessions. BSW in blue, LGO in red . . . . .	100
4.4	Experiment 1. Vertical residuals of hourly sessions. BSW in blue, LGO in red . . . . .	101
4.5	Experiment 2: the positions of reference station (light blue triangle) and the rover (yellow circle) . . . . .	104
4.6	The reference stations in Experiment 2: Como Test CORS (light blue triangle) . . . . .	104
4.7	The rover antenna in Experiment 2 . . . . .	105
4.8	The horizontal sliding device used in Experiment 2 . . . . .	105
4.9	Data processed with LGO software: comparison between imposed (blue) and estimated (red) . . . . .	107
4.10	Data processed with RTKLIB software: comparison between imposed (blue) and estimated (red) . . . . .	108
4.11	Experiment 3: the positions of reference stations (green and blue triangles) and the rover (red circle) . . . . .	110
4.12	The sliding device used in Experiment 3 . . . . .	110
4.13	Orientation task with total station . . . . .	111

4.14	The reference stations in Experiment 3: Como EUREF (green triangle) and secondary reference station (blue triangle); the orange circle is in correspondence of rover position . . . . .	111
4.15	The sliding device on Castelnuovo roof. In the upper left corner it is possible to see the position of reference stations (blue and green triangles) . . . . .	112
4.16	Monitoring the roof stability with another geodetic receiver . . . . .	112
4.17	Horizontal displacements: comparison between actual (red) and estimated (blue), Valleggio 1 as reference . . . . .	114
4.18	North component displacements: comparison between actual (red) and estimated (blue), Valleggio 1 as reference . . . . .	115
4.19	East component displacements: comparison between actual (red) and estimated (blue), Valleggio 1 as reference . . . . .	115
4.20	Vertical displacements: comparison between actual (red) and estimated (blue), Valleggio 1 as reference . . . . .	116
4.21	Horizontal displacements: comparison between actual (red) and estimated (blue), Valleggio 2 as reference . . . . .	116
4.22	North component displacements: comparison between actual (red) and estimated (blue), Valleggio 2 as reference . . . . .	117
4.23	East component displacements: comparison between actual (red) and estimated (blue), Valleggio 2 as reference . . . . .	117
4.24	Vertical displacements: comparison between actual (red) and estimated (blue), Valleggio 2 as reference . . . . .	118
4.25	Graph of frequency of couples for each class of errors for data processed with Valleggio 1 reference station. Horizontal (Purple bar), north (red bar), east (blue bar) and vertical (green bar) displacement	120
4.26	Graph of frequency of couples for each class of errors for data processed with Valleggio 2 reference station. Horizontal (Purple bar), north (red bar), east (blue bar) and vertical (green bar) displacement	121
B.1	U-Center at work . . . . .	132
C.1	Statistical results for the <b>day 15-128</b> for the <b>east coordinate</b> : the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections . . . . .	136
C.2	Statistical results for the <b>day 15-128</b> for the <b>north coordinate</b> : the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections . . . . .	137
C.3	Statistical results for the <b>day 15-129</b> for the <b>east coordinate</b> : the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections . . . . .	138
C.4	Statistical results for the <b>day 15-129</b> for the <b>north coordinate</b> : the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections . . . . .	139

C.5	Statistical results for the <b>day 15-130</b> for the <b>east coordinate</b> : the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections . . . . .	140
C.6	Statistical results for the <b>day 15-130</b> for the <b>north coordinate</b> : the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections . . . . .	141
C.7	Statistical results for the <b>day 15-131</b> for the <b>east coordinate</b> : the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections . . . . .	142
C.8	Statistical results for the <b>day 15-131</b> for the <b>north coordinate</b> : the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections . . . . .	143
C.9	Statistical results for the <b>day 15-132</b> for the <b>east coordinate</b> : the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections . . . . .	144
C.10	Statistical results for the <b>day 15-132</b> for the <b>north coordinate</b> : the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections . . . . .	145
C.11	Statistical results for the <b>day 15-133</b> for the <b>east coordinate</b> : the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections . . . . .	146
C.12	Statistical results for the <b>day 15-133</b> for the <b>north coordinate</b> : the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections . . . . .	147
C.13	Statistical results for the <b>day 15-134</b> for the <b>east coordinate</b> : the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections . . . . .	148
C.14	Statistical results for the <b>day 15-134</b> for the <b>north coordinate</b> : the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections . . . . .	149
C.15	Statistical results for the <b>day 15-135</b> for the <b>east coordinate</b> : the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections . . . . .	150

C.16	Statistical results for the <b>day 15-135</b> for the <b>north coordinate</b> : the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections . . . . .	151
C.17	Statistical results for the <b>day 14-073</b> for the <b>east coordinate</b> : the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections . . . . .	152
C.18	Statistical results for the <b>day 14-073</b> for the <b>north coordinate</b> : the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections . . . . .	153
C.19	Statistical results for the <b>day 14-074</b> for the <b>east coordinate</b> : the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections . . . . .	154
C.20	Statistical results for the <b>day 14-074</b> for the <b>north coordinate</b> : the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections . . . . .	155
C.21	Statistical results for the <b>day 14-075</b> for the <b>east coordinate</b> : the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections . . . . .	156
C.22	Statistical results for the <b>day 14-075</b> for the <b>north coordinate</b> : the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections . . . . .	157
C.23	Statistical results for the <b>day 14-076</b> for the <b>east coordinate</b> : the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections . . . . .	158
C.24	Statistical results for the <b>day 14-076</b> for the <b>north coordinate</b> : the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections . . . . .	159
C.25	Statistical results for the <b>day 14-077</b> for the <b>east coordinate</b> : the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections . . . . .	160
C.26	Statistical results for the <b>day 14-077</b> for the <b>north coordinate</b> : the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections . . . . .	161

C.27	Statistical results for the <b>day 14-078</b> for the <b>east coordinate</b> : the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections . . . . .	162
C.28	Statistical results for the <b>day 14-078</b> for the <b>north coordinate</b> : the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections . . . . .	163
C.29	Statistical results for the <b>day 14-079</b> for the <b>east coordinate</b> : the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections . . . . .	164
C.30	Statistical results for the <b>day 14-079</b> for the <b>north coordinate</b> : the red line represents the CORS data, the blue line represents the rover data before correction and the green line represents the rover data after the corrections . . . . .	165



# Tables

1.1	Past, current and future generations of GPS satellites . . . . .	8
1.2	Past, current and future generations of GLONASS satellites. The access to SF and SC signals is restricted . . . . .	10
1.3	BeiDou satellites launch sequence . . . . .	12
1.4	Double Difference method: error in function of baseline length . . . . .	21
2.1	Como test point: statistical errors . . . . .	41
2.2	Milan test point: statistical errors . . . . .	42
2.3	Survey with geodetic receiver along the Path 1: number of points with fixed phase solution compared to total points collected . . . . .	51
2.4	Statistical errors before and after the correction with CORS data along the Path 1 . . . . .	52
2.5	Survey with geodetic receiver along the Path 2: number of points with fixed phase solution compared to total points collected . . . . .	55
2.6	Smartphone statistical errors before and after the correction with CORS data along the Path 2 . . . . .	57
2.7	u-blox statistical errors before and after the correction with CORS data along the Path 2 . . . . .	57
3.1	GPS data table . . . . .	63
3.2	OSM tables: two tables with the same structure, where data downloaded from OSM are storage: <i>highway.routes</i> is the table with the path data and <i>building.routes</i> is the table with the building data . . . . .	64
3.3	The <i>highway.segment</i> table . . . . .	69
3.4	The <i>building</i> table . . . . .	74
3.5	The <i>build.segment</i> table . . . . .	74
3.6	structure of candidate_path DB table . . . . .	80
3.7	Compare between the different correction methods tested . . . . .	93
4.1	BSW and LGO processing strategy used . . . . .	97
4.2	Experiment 1. BSW residuals of blundered hourly sessions . . . . .	98
4.3	Experiment 1. BSW hourly residuals of final solutions. E: East, N: North, U: Up . . . . .	98
4.4	Experiment 1. LGO residuals of blundered hourly sessions . . . . .	98
4.5	Experiment 1. LGO hourly residuals. E: East, N: North, U: Up . . . . .	99
4.6	Experiment 1. Classification of hourly residuals by percentages. E: East, N: North, U: Up. Note (*): only one session . . . . .	100

4.7	Processing strategy used . . . . .	106
4.8	Congruence analysis: data processed with LGO . . . . .	106
4.9	Congruence analysis: data processed with rtkLIB . . . . .	107
4.10	Processing parameters chosen in LGO . . . . .	113
4.11	Processed data: statistics and comparison between results obtained using two different reference station, Valleggio 1 and Valleggio 2 . . .	118
4.12	Significance analysis on estimated movement, data processed using Valleggio 1 as reference station. Percentage of right answers . . . .	119
4.13	Significance analysis on estimated movement, data processed using Valleggio 2 as reference station. Percentage of right answers . . . .	119
4.14	Numbers and frequencies of couples for each class of error for data processed with Valleggio 1 as reference station . . . . .	121
4.15	Numbers and frequencies of couples for each class of error for data processed with Valleggio 2 as reference station . . . . .	122
4.16	In evidence class 0, class 1-2 (data processed with Valleggio 1 as reference station) . . . . .	122
4.17	In evidence class 0, class 1-2 (data processed with Valleggio 2 as reference station) . . . . .	122
5.1	Milan test: standard deviations before and after the application of CORS corrections . . . . .	124
5.2	Como test: standard deviations before and after the application of CORS corrections . . . . .	124
5.3	Path 1: standard deviations before and after the application of CORS corrections; survey with smartphone . . . . .	125
5.4	Path 2: standard deviations before and after the application of CORS corrections; survey with smartphone . . . . .	125
5.5	Path 2: standard deviations before and after the application of CORS corrections; survey with u-blox . . . . .	125
5.6	Experiment 1. BSW and LGO: hourly residuals of final solutions. E: East, N: North, U: Up . . . . .	126
5.7	Experiment 2. Congruence analysis: in evidence class 0 and class 1	127
5.8	Experiment 3. Processed data: comparison between standard de- viation obtained using two different reference station, Valleggio 1 and Valleggio 2 . . . . .	128
5.9	Experiment 3. Congruence analysis: data processed with Valleggio 1 as reference station . . . . .	128
5.10	Experiment 3. Congruence analysis: data processed with Valleggio 2 as reference station . . . . .	128

# Acknowledgements

First of all I would like to express a great thank to my supervisor, Professor Ludovico Biagi: he supported and led me throughout this experience.

A part of this thesis concerned topics of the MEP project: thank to all the team, it was a pleasure to work together.

Thanks to my evaluators, Prof. Ambrogio Manzino (Politecnico di Torino) and Ing. Augusto Mazzoni (Università degli Studi di Roma "La Sapienza"), for their suggestions, notes and corrections that improved this thesis and were much appreciated.

Also thanks to Politecnico di Milano that allowed me to come back student: three years of training hard, but very interesting that I hope will enhance my contribution to the growth of Politecnico.

In questi tre anni di tesi ho lavorato, pranzato, discusso, scherzato e incontrato molte persone, ma un saluto particolare va ad Alessandro, Florin, Grazia, Lorenzo e Ludovico: ci siamo trovati e ora ci lasciamo, *ognuno col suo viaggio, ognuno diverso e ognuno in fondo perso dentro i fatti suoi*, un abbraccio a tutti e grazie per la compagnia.