

ICT for Geomatics

Final report about laboratories' work

Francesco Conforte 277683

A.A. 2019-2020

Contents

1 Evaluation of PVT	2
1.1 Least Mean Square solution	2
1.1.1 Standard deviation of the position error for different datasets and constellations	4
1.2 Weighted Least Mean Square	5
1.2.1 Standard deviation of the position error for different datasets and constellations	7
2 Raw GNSS measurements in Android devices	8
2.1 Analysis of raw measurements and PVT solution	8
2.2 Analysis by applying different data filters	9
3 Planning of GNSS Survey	10
3.1 Planning	10
3.2 Adding obstacles	15
3.3 Azimuth and Elevation angle	16
3.4 DOP estimation	17
4 GNSS data processing	18
4.1 Single point positioning	18
4.2 Relative Static Positioning	18
4.3 Relative Kinematic positioning	21
5 Geodesy	23
5.1 Ellipsoid parameters	23
5.2 Coordinates transformation from (ϕ, λ, h) to (XYZ)	23
5.3 Coordinates transformation from (XYZ) to (ϕ, λ, h)	24
5.4 RS Transformation	25
5.5 Helmert Transformation	26
6 Cartography	27
6.1 Modulus of linear deformation	27
6.2 From geographic (ϕ, λ) to cartographic coordinates (N,E)	27
6.3 From cartographic coordinates (N,E) to geographic (ϕ, λ) coordinates	28
7 Geographical Information System - GIS	28
7.1 Georeferencing an image with ARCMAP	28
7.2 Digital Terrain Model - DTM	30
8 GIS Part 2	32
8.1 Exercise 1	32
8.2 Exercise 2	33
8.3 Exercise 4	33
8.4 Exercise 5	34
8.5 Exercise 6	35

1 Evaluation of PVT

The aim of this lab was to implement the estimation of PVT (*Position, Velocity and Time*) solution having available a set of pseudoranges measured in a range of time of 3600 seconds, from different constellations, such as GPS, GALILEO, GLONASS and BEIDOU. The estimation of PVT for each time instant has been obtained in two different ways, by using two algorithms based on several iterations: firstly by applying the *Least Mean Square* and secondly by applying the *Weighted Least Mean Square*. The final estimated position of a static user was the average of all the estimated positions for each time instant.

1.1 Least Mean Square solution

The first step was to check the visibility of satellites for each constellation. This is a very important step, because in order to estimate the position of the user there is not only the necessity to compute the 3 unknown coordinates x_u, y_u and z_u but there is also a fourth unknown $b_{ut} = c \cdot \delta t_u$, which is the bias due to the fact that the receiver's clock and the satellite's clock are not perfectly synchronized. Thus, in order to compute these 4 unknowns at least 4 different satellites are needed. In Figure 1 are reported the visibilities for each constellation over the whole time¹. As it can be seen, for each time instant at least 4 satellites per constellation were in Line Of Sight of the user.

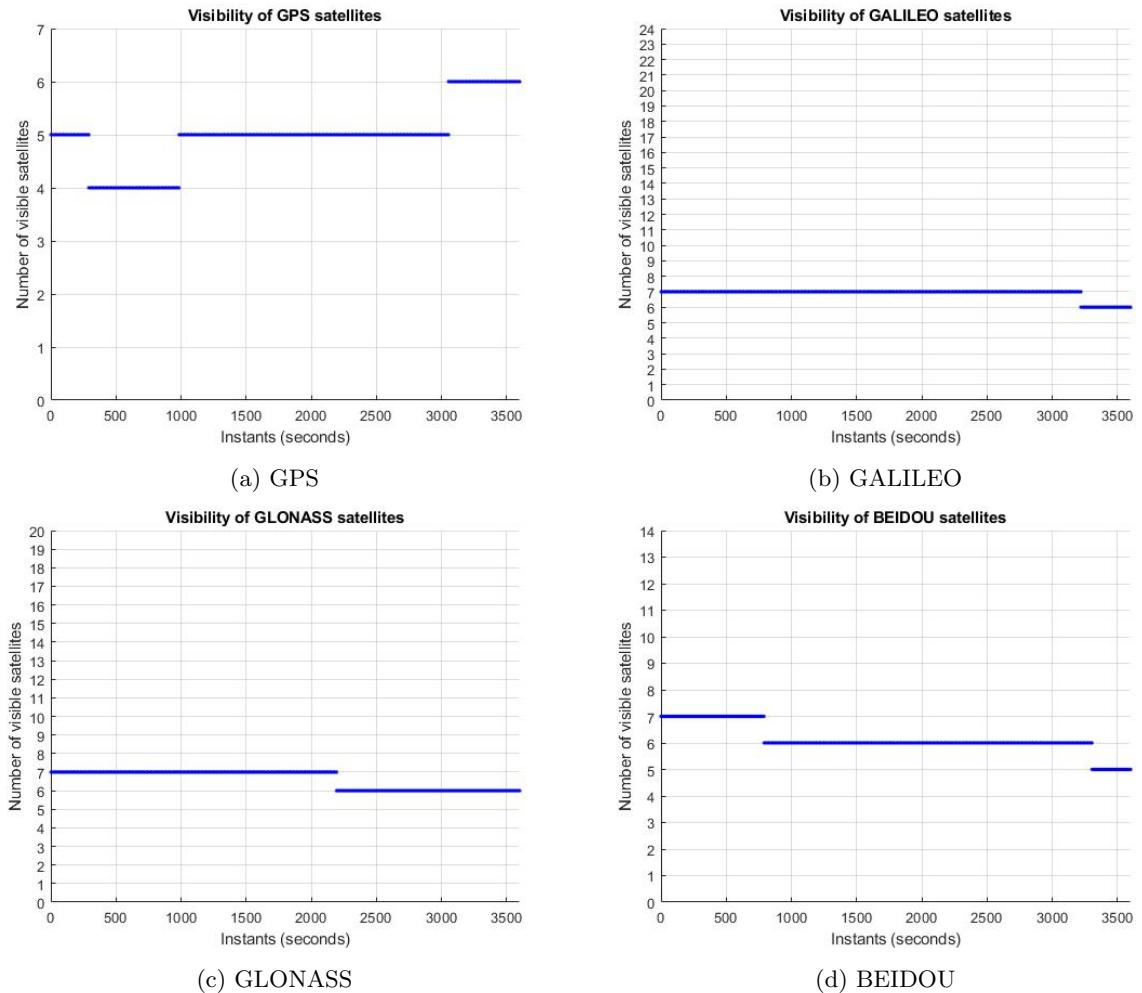


Figure 1: Visibility of each constellation over time

¹dataset_1_20180328T122038 Folder: NominalUERE

Moreover, in Figure 2 are reported the given pseudoranges for each constellation of satellites.

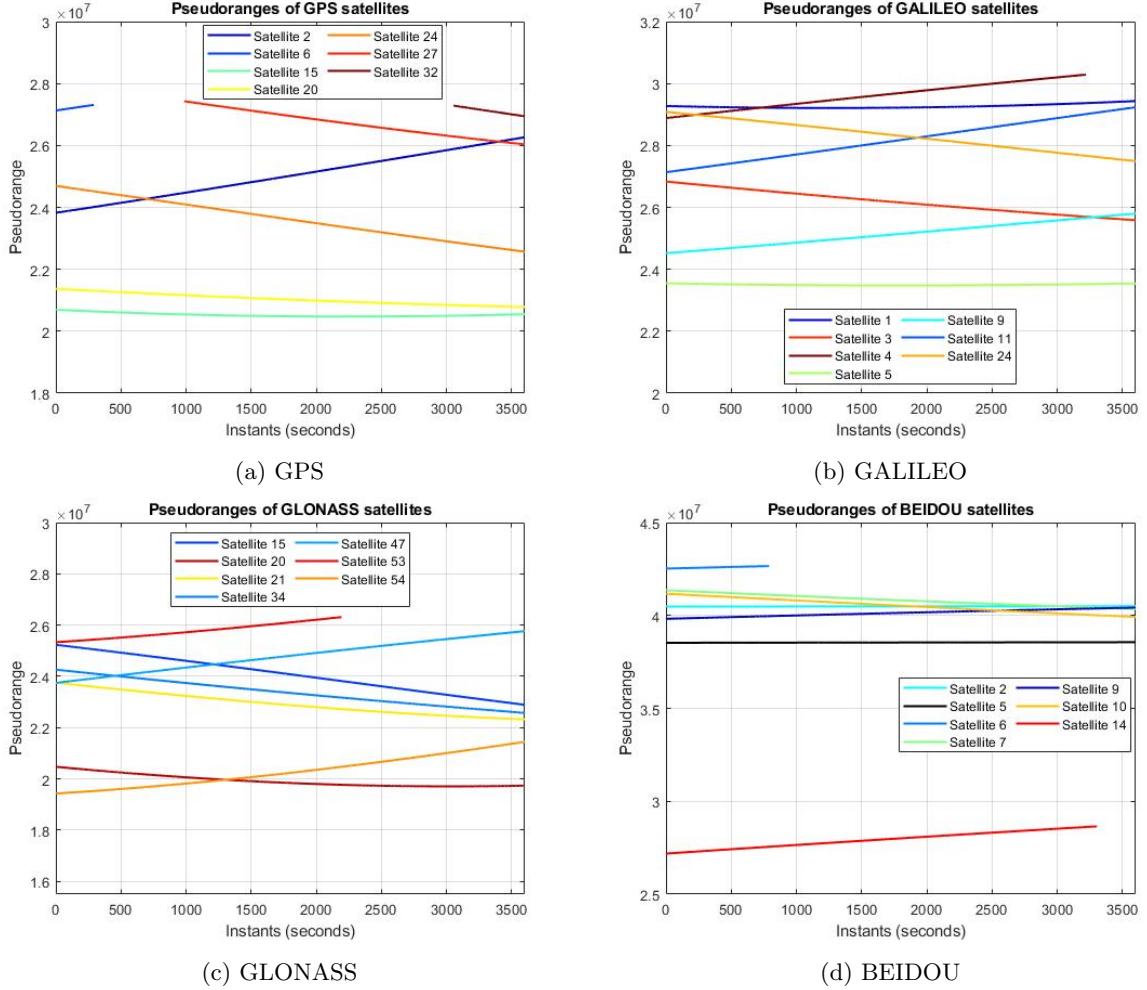


Figure 2: Pseudoranges of each constellation over time

The *Least Mean Square* algorithm is one of the possible ways to compute the PVT solution. The position has been computed according to a set of simulated pseudoranges taken from the visible satellites over time. In order to provide a more precise solution, the algorithm was applied in an iterative way. Doing so, at each iteration, the estimated position of the static user was closer to the real location.

Indeed, it has been iterated for a number of times K equals to 9 and repeated for each time instant from 0 up to 3600.

The estimated position vector $\hat{\mathbf{x}}_n^k$ for the n-th time instant and for the k-th iteration is given by equations 1:

$$\begin{aligned} \hat{\mathbf{x}}_n^k &= \hat{\mathbf{x}}_n^{k-1} + \Delta \hat{\mathbf{x}}_n^k \\ \Delta \hat{\mathbf{x}}_n^k &= ((\mathbf{H}_n^k)^T \mathbf{H}_n^k)^{-1} (\mathbf{H}_n^k)^T \Delta \hat{\rho}_n^k \\ \Delta \hat{\rho}_n^k &= \hat{\rho}_n^k - \rho_n \\ \hat{\rho}_n^k &= \sqrt{(x_{j,n} - \hat{x}_n^k)^2 + (y_{j,n} - \hat{y}_n^k)^2 + (z_{j,n} - \hat{z}_n^k)^2 + \hat{b}_n^k} \end{aligned} \quad (1)$$

where \mathbf{H}_n^k is the geometrical matrix depending on the position of each satellite at n-th time instant:

$$\mathbf{H}_n^k = \begin{bmatrix} a_{x,1} & a_{y,1} & a_{z,1} & 1 \\ a_{x,2} & a_{y,2} & a_{z,2} & 1 \\ a_{x,3} & a_{y,3} & a_{z,3} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ a_{x,J_n} & a_{y,J_n} & a_{z,J_n} & 1 \end{bmatrix} \quad (2)$$

and $\Delta\hat{\rho}_n^k$ is a vector given by the difference between the estimated pseudoranges $\hat{\rho}_n^k$ of visible satellites in that time instant n and the measured pseudoranges ρ_n of visible satellites in that time instant n. Thus, its dimension is $J \times 1$, where J is the number of visible satellites at the n-th time instant.

Each entry of matrix \mathbf{H}_n^k is a coefficient depending on the position of the j-th satellite:

$$a_{x,j} = \frac{x_{j,n} - \hat{x}_n^k}{\hat{r}_{j,n}} \quad a_{y,j} = \frac{y_{j,n} - \hat{y}_n^k}{\hat{r}_{j,n}} \quad a_{z,j} = \frac{z_{j,n} - \hat{z}_n^k}{\hat{r}_{j,n}}$$

where $x_{j,n}, y_{j,n}, z_{j,n}$ are the coordinates of the j-th satellite retrieved by the *ephemeris* parameters broadcast by each satellite and $\hat{r}_{j,n}$ is the Euclidean distance between the approximation point $\hat{\mathbf{x}}_n^k$ and the position point of the j-th satellite that changes for each time instant:

$$\hat{r}_{j,n} = \sqrt{(x_{j,n} - \hat{x}_n^k)^2 + (y_{j,n} - \hat{y}_n^k)^2 + (z_{j,n} - \hat{z}_n^k)^2} \quad (3)$$

Starting from an initial position taken at point $\hat{x}_u = 0, \hat{y}_u = 0, \hat{z}_u = 0, \hat{b}_{ut} = 0$, at each iteration the estimated position vector $\hat{\mathbf{x}}_n^k$ is updated adding to the vector $\hat{\mathbf{x}}_n^{k-1}$, estimated at the previous iteration, the new vector $\Delta\hat{\mathbf{x}}_n^k$. For each iteration k, the matrix \mathbf{H} is obtained from the previous estimated position.

This sequence of iterations is useful to converge the approximated point to the real one. At the end of the process the last approximated point $\hat{\mathbf{x}}_n^9$ has been taken as the user position at the n-th time instant.

With data picked from 'dataset_1_20180328T12203', the estimated coordinates are reported in Table 1, corresponding to a point in Cape Town in South Africa.

CONSTELLATION	LATITUDE [degrees]	LONGITUDE [degrees]	ALTITUDE [meters]
GPS	-33.9249	18.4241	13.0199
GLONASS	-33.9249	18.4241	13.0467
GALILEO	-33.9249	18.4241	12.8089
BEIDOU	-33.9249	18.4241	12.8779

Table 1: Estimated point by using different GNSS systems

1.1.1 Standard deviation of the position error for different datasets and constellations

Generally, the raw measurements of the pseudoranges are affected by errors which must be corrected otherwise they will affect the estimation of the user position. Unfortunately, after having applied corrections, there is still a *residual contribution* that can be modelled as a random variable with Gaussian distribution having zero mean and variance $\sigma_{UERE}^2: \mathcal{N}(0, \sigma_{UERE}^2)$. Thus, the estimated positions were affected by imprecision.

The relation between the error of the measurements and the imprecision in the estimations can be expressed by equation 4:

$$\sigma_x = \text{GDOP} \cdot \sigma_{UERE} \quad (4)$$

representing the standard deviation of the position error, that is the uncertainty in meters in the estimation of the position.

The *Geometric Dilution Of Precision*(GDOP) is a parameter depending on the \mathbf{H} matrix and thus depending on the displacement of the satellites. For this reasons, it tells the impact of the geometry on the estimation of the position. It is a sort of amplifier of the uncertainty, hence its value should not be too high.

Several values of standard deviation were computed, for different datasets and different constellations. They are reported in Table 2:

	GPS [m]	GLONASS [m]	GALILEO [m]	BEIDOU [m]
Cape Town	23.3412	11.5485	11.3334	67.2225
Helsinki	21.5000	12.9449	13.1683	31.8889
Longyearbyen (Svalbard)	20.9955	12.5768	16.1833	32.0456
Shanghai	10.1989	12.1351	12.7355	20.8272
Stanford	11.6838	19.0128	15.3793	-
Turin	13.6218	25.6998	19.0069	50.7954

Table 2: Standard deviations of the position error

	GPS	GLONASS	GALILEO	BEIDOU
Cape Town	3.5733	1.8231	1.8269	12.0214
Helsinki	3.1653	2.0853	2.1347	5.7004
Longyearbyen (Svalbard)	2.8200	2.0999	2.6403	5.7097
Shanghai	1.6434	1.9080	2.0183	3.6176
Stanford	1.8462	3.0201	2.5848	-
Turin	2.0948	3.3604	3.0886	9.6932

Table 3: Values of GDOP

In Table 3 are reported values of GDOP computed for different constellations and different datasets over time. As one could expect, there is a relation between their values and the ones of the standard deviations: when the value of the GDOP is much higher than 1, the uncertainty in the estimation point becomes an unacceptable value. As it can be seen, for example, for Beidou system all the GDOP values are much higher than 1, reaching values around 12. That's the reason why the estimation point has high values of uncertainty, around 67 meters for 'dataset 1'. This high values of GDOP were due to the fact that during the time the satellites had a bad displacement in the sky with respect to the static user position. The '-' marked values are due to software limitation.

1.2 Weighted Least Mean Square

An optimized version of the Least Mean Square is the *Weighted Least Mean Square*. The ordinary LMS algorithm is implemented in such a way that it attributes to all measured pseudoranges the same variance σ_{UERE}^2 , meaning that the residual error of each measured pseudorange can range into the same interval of values. In real cases, this is not true because each pseudorange may be characterized by a different value of standard deviation σ_{UERE} , but the errors of each pseudorange continues to be independent each other. For this reason, the covariance matrix \mathbf{R} of the error of pseudoranges is always a diagonal matrix:

$$\mathbf{R} = \begin{bmatrix} \sigma_{1,UERE}^2 & 0 & 0 & \dots & 0 \\ 0 & \sigma_{2,UERE}^2 & 0 & \dots & 0 \\ 0 & 0 & \sigma_{3,UERE}^2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & \dots & \sigma_{J,UERE}^2 \end{bmatrix}$$

Thus, a new version of this algorithm has been implemented to be used on a different set of data².

This version as well was applied in an iterative way for a number of times equal to 9 and repeated for each time instant. It consists of introducing, during the estimation of the user position, a weighting matrix \mathbf{W} , which is a positive definite matrix, in order to consider more the measurements affected by a minor error and less those with a major error. Indeed, it has been set to the inverse of the covariance matrix \mathbf{R} :

$$\mathbf{W} = \mathbf{R}^{-1}$$

The estimated position vector $\hat{\mathbf{x}}_n^k$ for the n-th time instant and for the k-th iteration was estimated in a slightly different way:

$$\begin{aligned} \hat{\mathbf{x}}_n^k &= \hat{\mathbf{x}}_n^{k-1} + \Delta\hat{\mathbf{x}}_n^k \\ \Delta\hat{\mathbf{x}}_n^k &= \bar{\mathbf{H}}_n^k \Delta\hat{\rho}_n^k \\ \bar{\mathbf{H}}_n^k &= ((\mathbf{H}_n^k)^T \mathbf{W} \mathbf{H}_n^k)^{-1} (\mathbf{H}_n^k)^T \mathbf{W} \\ \Delta\hat{\rho}_n^k &= \hat{\rho}_n^k - \rho_n \\ \hat{\rho}_n^k &= \sqrt{(x_{j,n} - \hat{x}_n^k)^2 + (y_{j,n} - \hat{y}_n^k)^2 + (z_{j,n} - \hat{z}_n^k)^2} + \hat{b}_n^k \end{aligned} \tag{5}$$

Nevertheless, the direct computation of the covariance matrix \mathbf{R} were not possible because the measured pseudoranges were time dependent and multiple measurements for each time instant were not available, as it is in real cases because the receiver cannot take several measurements at the same time.

²data contained in folder RealisticUERE

It has been however estimated by analyzing the error on the pseudorange ρ_j itself along the time: after removing the time dependency from each measured pseudorange, its variance over time has been computed³:

$$\sigma_{j,UERE}^2 = \text{var}\left(\frac{d^2\rho_j}{dt^2}\right)$$

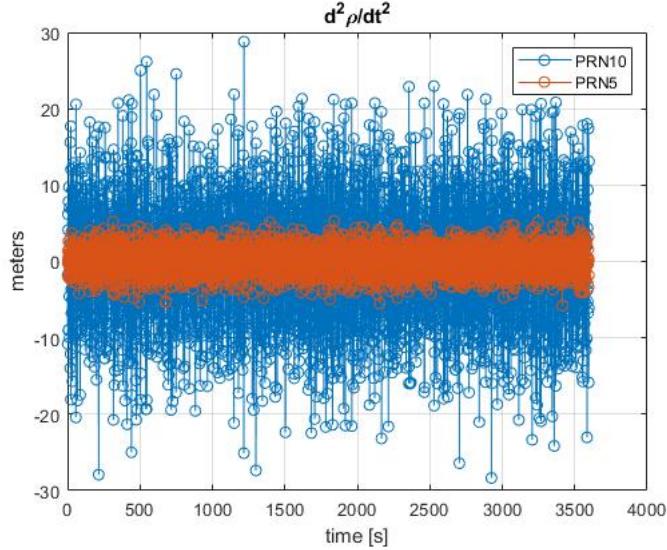


Figure 3: Variances of pseudoranges from satellites 10 and 5 of GPS

In Figure 3 are reported the trends of the second order derivative of the pseudoranges of satellites 10 and 5 of GPS. It can be also seen the variance $\sigma_{10,UERE}^2$ and $\sigma_{5,UERE}^2$, that indicate how much the error of the measurement of the pseudoranges taken from satellites 10 and 5 of GPS system can vary over time⁴.

It's clear that the variance of pseudoranges of satellite 10 is much higher than the variance of pseudoranges of satellite 5, meaning that the pseudoranges of satellite 10 are affected by much more uncertainty than the pseudoranges of satellite 5. Thus, the former must be considered with a lower weight with respect to the latter during the estimation of user position.

The values of standard deviations $\sigma_{j,UERE}$ for each satellite of GLONASS, GPS, GALILEO and BEIDOU are reported in Table 4.

³Since pseudorange measurement follows a quadratic trend, the second order derivative can be applied to remove the variation along the time

⁴dataset 1 RealisticUERE

GLONASS		GPS		GALILEO	
SATELLITE	σ_{UERE} [m]	SATELLITE	σ_{UERE} [m]	SATELLITE	σ_{UERE} [m]
15	7.043	2	6.025	1	14.878
19	6.312	5	3.380	3	3.737
20	2.126	10	14.107	5	1.851
21	4.645	13	2.157	9	2.823
34	5.3566	15	1.665	11	7.686
37	6.616	20	2.109	18	3.393
42	2.752	21	2.781	24	7.902
44	2.752	24	3.966		
47	10.112	29	2.857		
51	6.920	30	8.020		
54	2.371				

BEIDOU	
SATELLITE	σ_{UERE} [m]
2	7.128
5	3.568
7	9.908
9	6.132
10	7.890

Table 4: $\sigma_{j,UERE}$ of visible satellites

1.2.1 Standard deviation of the position error for different datasets and constellations

Several values of standard deviation of the position error over time have been computed, for different datasets and different constellations, both after applying LMS and after applying WLMS. They are reported in Tables 5 and 6.

	GPS [m]	GLONASS [m]	GALILEO [m]	BEIDOU [m]
Cape Town	4.1314	2.8659	6.2554	278.2836
Helsinki	10.9906	3.2424	6.6339	51.2724
Longyearbyen (Svalbard)	10.2516	-	10.3920	38.2669
Shanghai	12.4954	4.0310	-	6.9851
Stanford	10.3482	5.4981	7.1818	-
Turin	5.8626	6.2365	3.7853	20.9141

Table 5: Standard deviations of the position error after LMS application

	GPS [m]	GLONASS [m]	GALILEO [m]	BEIDOU [m]
Cape Town	2.2128	2.4069	3.7599	273.3864
Helsinki	10.5041	2.9863	3.7655	19.7764
Longyearbyen (Svalbard)	9.7427	-	9.4172	22.1540
Shanghai	11.1081	3.6532	-	4.5441
Stanford	8.6679	3.8261	7.1818	-
Turin	4.6301	5.8569	3.6209	20.0079

Table 6: Standard deviations of the position error after WLMS application

By comparing the two tables, it can easily be observed how the values of standard deviations obtained with the WLMS algorithm were all lower than the ones obtained with the ordinary LMS. This is due to the fact that the WLMS, during the estimation of the user point, takes into account the different contributions of residual errors of the pseudoranges.

The '-' marked values are due to software limitation.

Furthermore, the standard deviations of the BeiDou system were always positions affected by a high uncertainty because of a worse displacement of satellites in the sky with respect to the static position of the user.

2 Raw GNSS measurements in Android devices

The aim of this lab was to perform the analysis of raw measurements and PVT solution obtained from data collected by an Android device running OS version Android 7 Nougat or newer, which were able to receive two GNSS signals at different frequencies from a satellite.

Due to the impossibility of collecting own data since an Android phone was missing, data provided by older collections have been exploit to perform the tasks. Two different datasets were used: one containing data collected in urban environment and another containing data collected in open space.

2.1 Analysis of raw measurements and PVT solution

The data have been processed by a Matlab code provided by Google to compute the PVT solutions and some parameters, such as the Carrier-to-Noise ratio, which refers to the ratio of the carrier power C of the received signal and the noise power per unit bandwidth N_0 : $CNR = \frac{C}{N_0}$. It is expressed in dB-Hz.

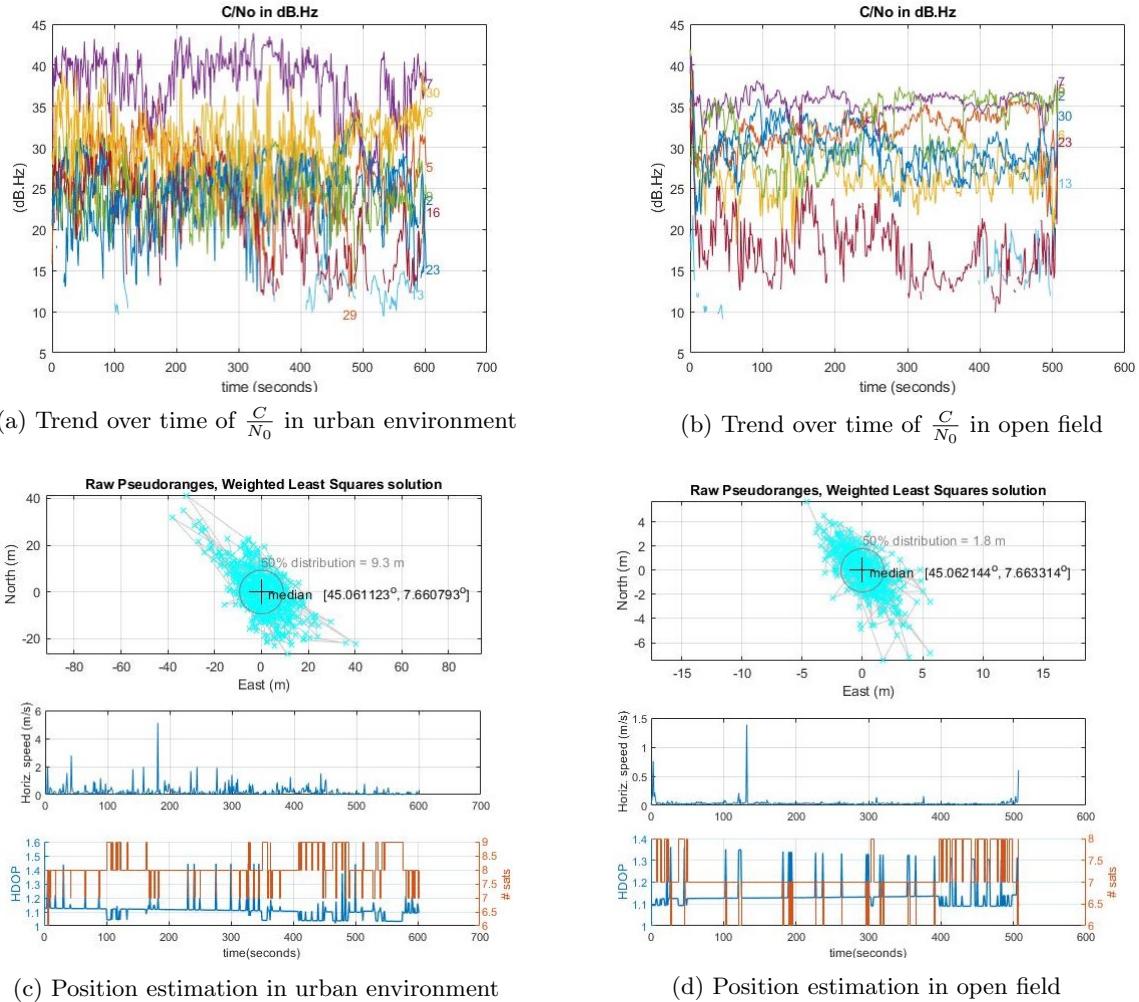


Figure 4: Comparison results between urban environment and open field collection

Figures 4a and 4b show the evolution of ratio $\frac{C}{N_0}$ over the data acquisition time. In urban environment, the signal power of the various satellites being tracked by the receiver varies in relation to the satellite visibility and it's needed also taking into account the noise environment. For these reasons, the carrier-power-to-noise-density ratio in urban environment varies in a range of values greater wrt the one estimated in open field. That's why the navigation solution estimated in urban environment has a *Circular Error Probable*⁵ of 9.3 meters (Figure 4c), larger than the one estimated in open field measurements, where its value is 1.8 meters (Figure 4d).

⁵CEP: value within which there is the 50% of the solution

2.2 Analysis by applying different data filters

The Matlab code provided by Google allows to filter the raw data by applying filters and see how the new computed PVT solutions vary.

A filter about the $\frac{C}{N_0}$ has been applied with a threshold of 25 dB-Hz, but as it can be seen from Figure 5a a lot of trends related to signals of some satellites are cut out. In this way, the estimated PVT solution had a CEP of 16 meters, a value pretty high. The same happens for measurements in open field: by applying this filter, since the threshold is too high, some satellites are no more tracked by the receiver. Indeed, the number of satellites in Figure 4d is higher than in Figure 5d and the HDOP reaches values very high (closer to 20) if the filter is applied. Nevertheless, the estimated position had a CEP of 1.9 meters, slightly higher than when this kind of filter was not applied.

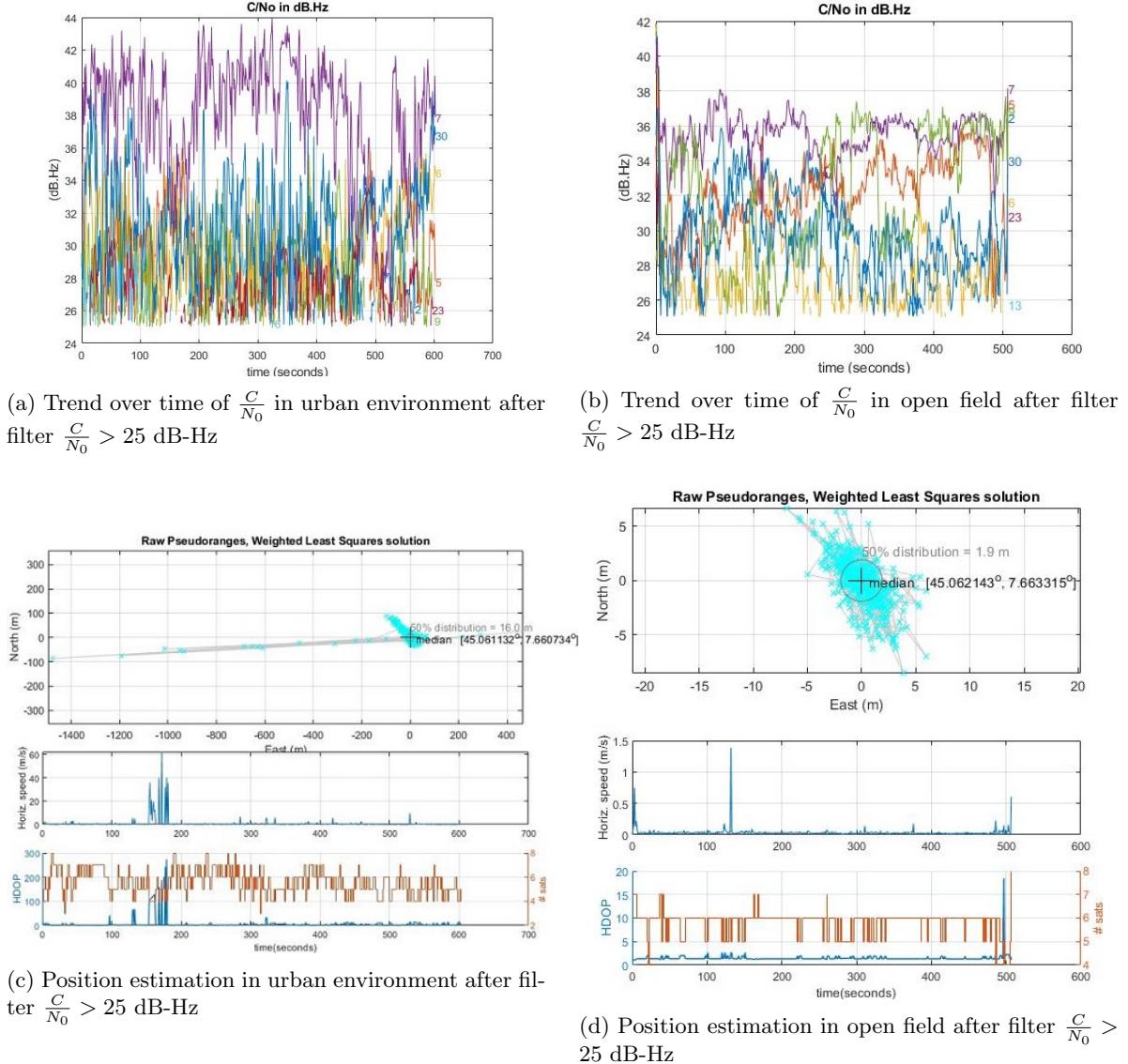
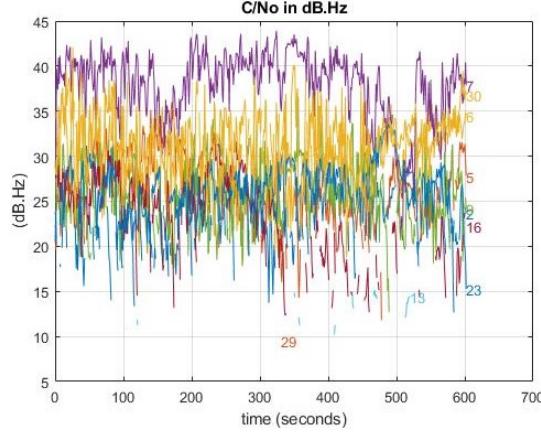


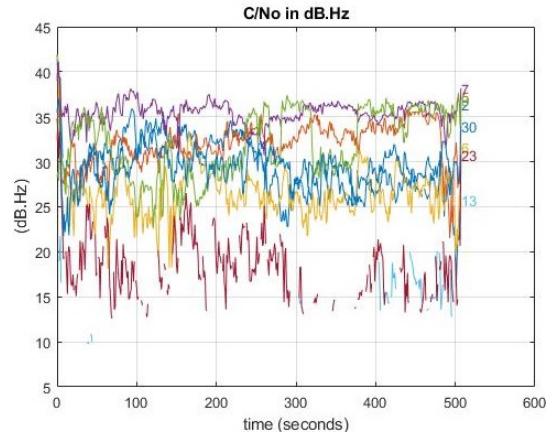
Figure 5: Comparison results between urban environment and open field collection after filter $\frac{C}{N_0} > 25$ dB-Hz

Another filter that has been applied is the one regarding the multipath effect, that occurs when the GNSS receiver antenna is not receiving directly the satellite signals but it receives reflections of these signals. Because of this phenomenon, a position error may occur since the computation of the distances between the receiver and the satellites may be biased. Of course, this phenomenon is more present in urban environment rather than in open field, since in urban environment there are more reflective surfaces. Indeed, by considering the effects of the multipath on raw measurements, a less accurate PVT solution was estimated in urban environment (CEP: 10.4 meters, Figure 6c), whereas the CEP for the PVT solution for the open field measurements was always of 1.8 meters (Figure 6d). This because the

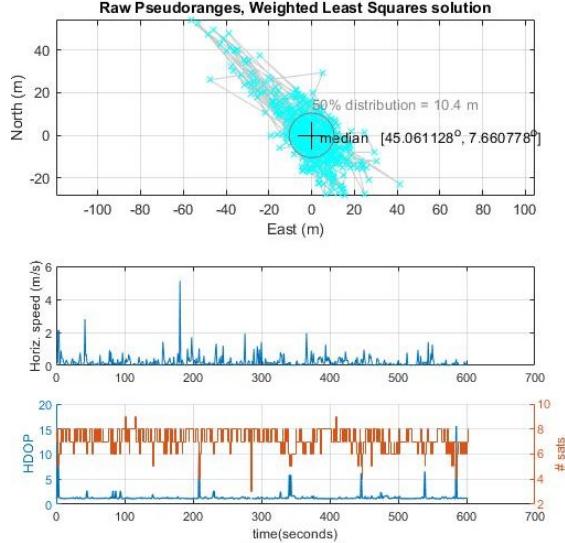
multipath is less present in open field.



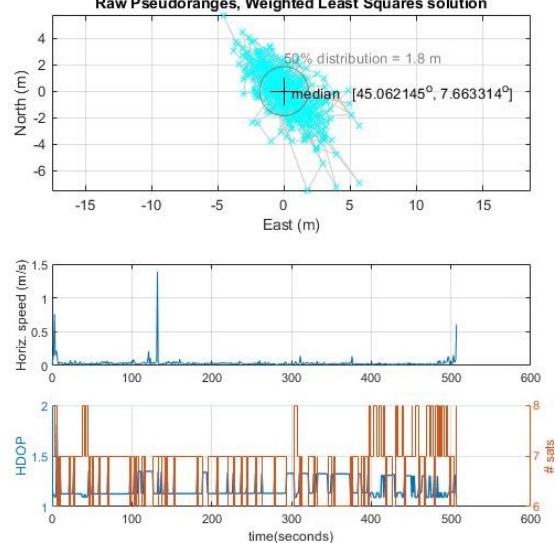
(a) Trend over time of $\frac{C}{N_0}$ in urban environment after multipath filter



(b) Trend over time of $\frac{C}{N_0}$ in open field after multipath filter



(c) Position estimation in urban environment after multipath filter



(d) Position estimation in open field after multipath filter

Figure 6: Comparison results between urban environment and open field collection after multipath filter

3 Planning of GNSS Survey

Realizing a GNSS survey requires some preliminary tasks, such as on-the-field investigation in order to physically evaluate the territory, the selection of appropriate instruments to obtain the materialization of points, the analysis of satellite configuration, the determination of the observation window. Once the survey has been set up, the measurements session must be performed.

3.1 Planning

The goal of the first exercise of this lab was to realize a planning in four different cities. The chosen ones were: Carpino (IT), Longyearbyen (Svalbard Islands), Punta Arenas (CH) and Quito (EC).

During a planning there are some factors to monitor, such as the *number of visible satellites* over time because it says when moving from a site to another one where one wants to do the planning and the elevation mask, i.e. the *cut-off angle* under which the receiver is no more able to receive correctly the GNSS signal because the satellite is too low in the sky. To be noted the fact that using the navigation file gives predictions and not very reliable information, indeed it's possible, for example, that when one goes on-the-field, these information are slightly different from the predicted ones.

Moreover, another interesting plot is the *Sky Plot*, where it's possible to see the trajectories of satellites with respect to the receiver position. Then there is also a parameter to keep track of: the

Dilution of Precision: DOP, which depends on the number of satellites and on their distributions in the sky. In particular, during a planning it's important to observe the PDOP, i.e. the Position DOP, because it provides assessments on the geometric configurations and thus it helps in evaluating the goodness of the survey.

By using a GNSS online tool⁶, for each city, plannings in different conditions have been performed: by considering only GPS system, then only GPS and GLONASS systems and finally all the GNSS constellations. The above-mentioned cities were chosen because of they very different positions on the globe: one near to the Equator (Quito), one on the Northern Hemisphere (Longyearbyen) and one on the Southern Hemisphere (Punta Arenas). The obtained results are referred to the 23th of May 2020 for 24 hours from the midnight of the respective cities.

To reduce the measurements error due to a small elevation angle of the satellite, the cutoff has been set equal to 10°.

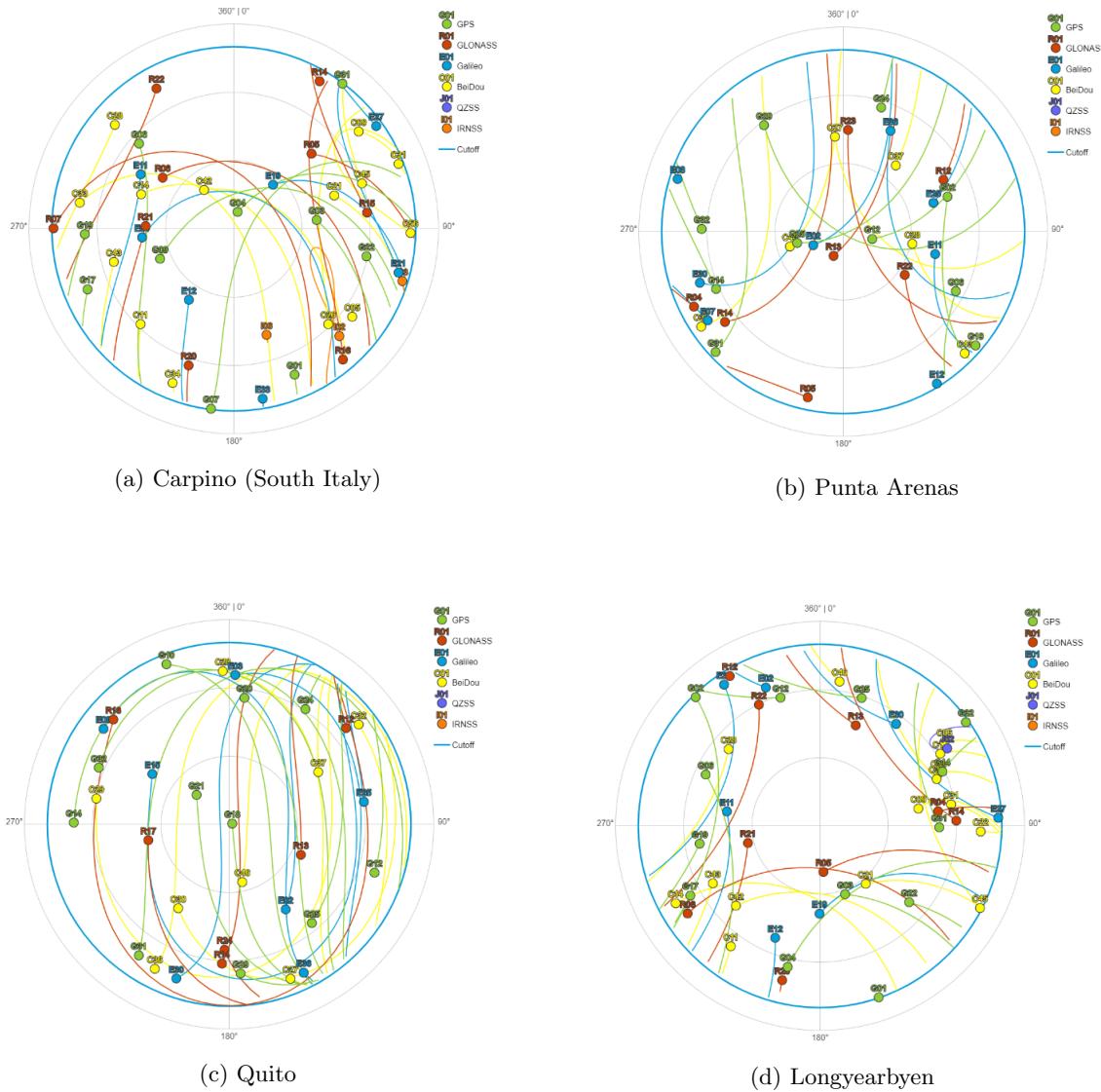
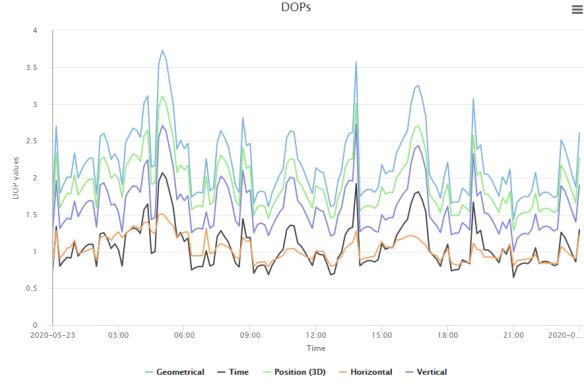


Figure 7: SkyPlots of each city considering full constellations

From Figure 7 it can be seen that for each city the gap within the plot is always different: in Longyearbyen (Northern Hemisphere near the North Pole) the gap is on the head of the observer, that is in the center of the plot, whereas in Quito (near to the Equator) there is no gap, meaning that at or near the Equator and at or near the Poles the sky view is symmetric. Moving from South to North (in sequence: Punta Arenas, Quito, Carpino, Longyearbyen) the constellations shift towards the South, so that in Carpino there are many satellites on the southern horizon but fewer on the northern horizon whereas the opposite situation is in Punta Arenas.

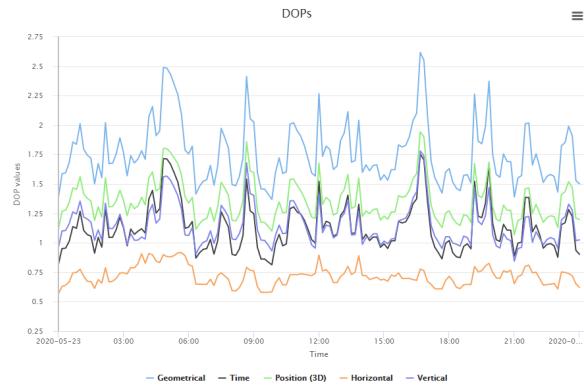
⁶<https://www.gnssplanning.com/#/settings>



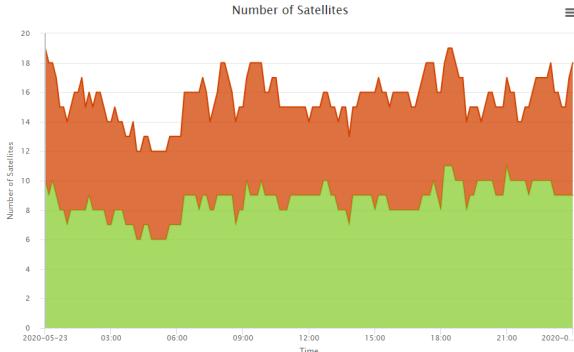
(a) DOPs with only GPS in Carpino



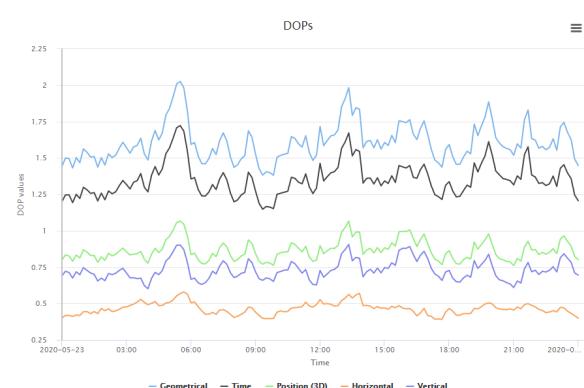
(b) Number of visible satellites of GPS in Carpino



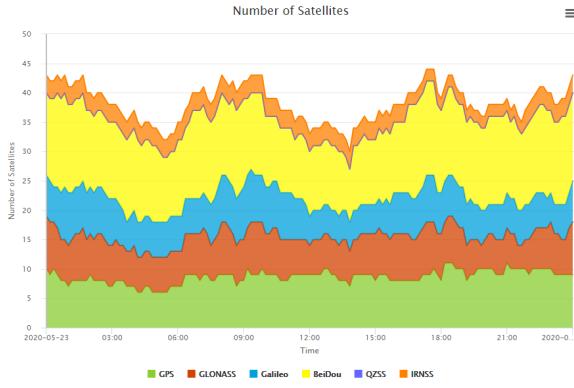
(c) DOPs with GPS and GLONASS in Carpino



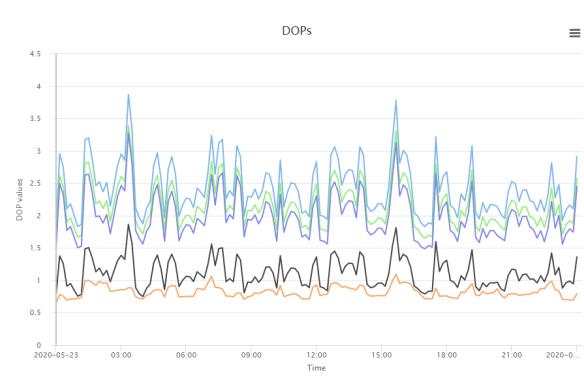
(d) Number of visible satellites of GPS and GLONASS in Carpino



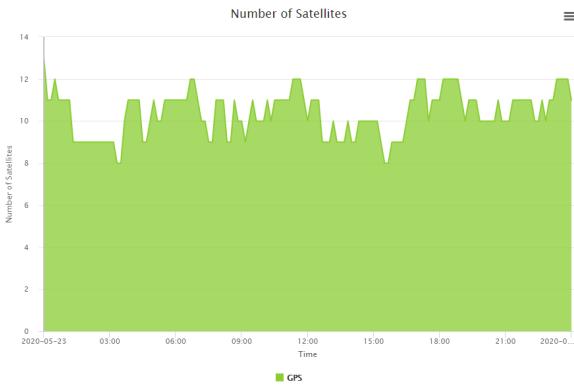
(e) DOPs with all constellations in Carpino



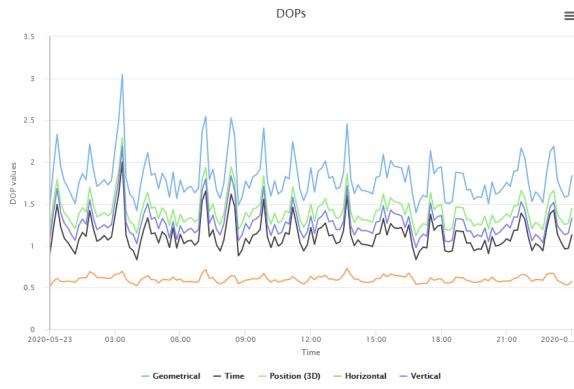
(f) Number of visible satellites of all constellations in Carpino



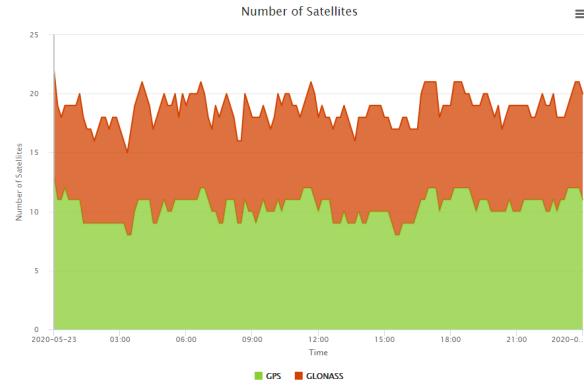
(g) DOPs with GPS in Longyearbyen



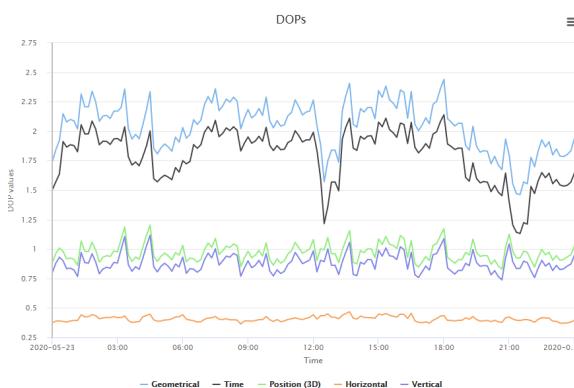
(h) Number of visible satellites of GPS in Longyearbyen



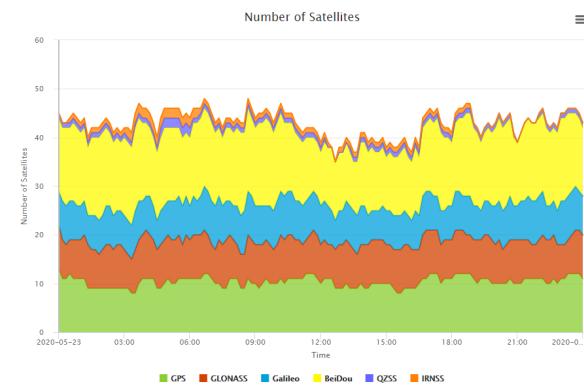
(i) DOPs with GPS and GLONASS in Longyearbyen



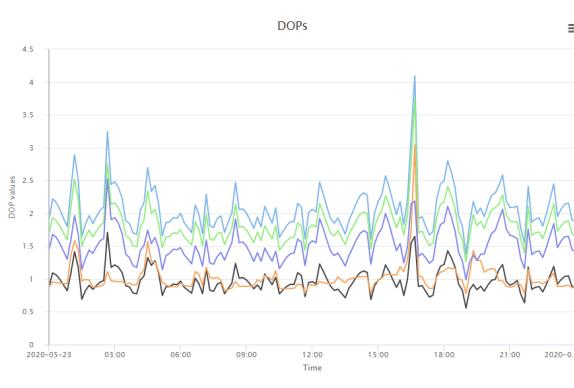
(j) Number of visible satellites of GPS and GLONASS in Longyearbyen



(k) DOPs with all constellations in Longyearbyen



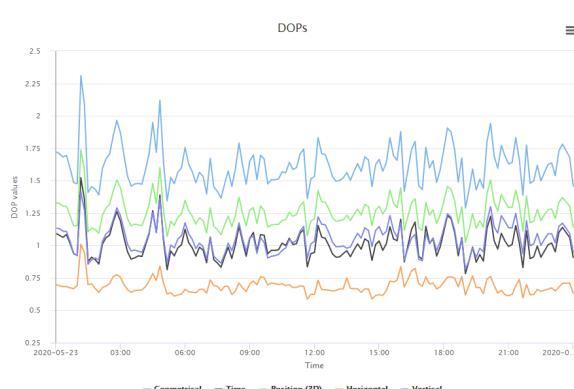
(l) Number of visible satellites of all constellations in Longyearbyen



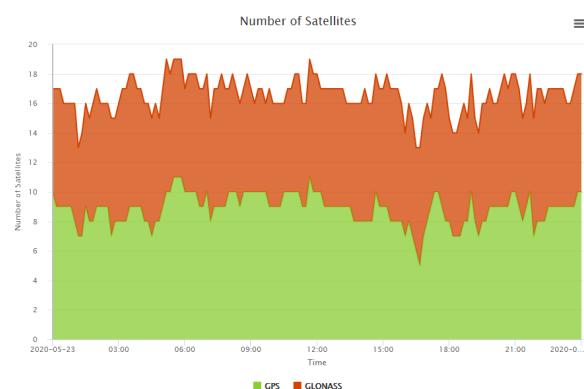
(m) DOPs with only GPS in Punta Arenas



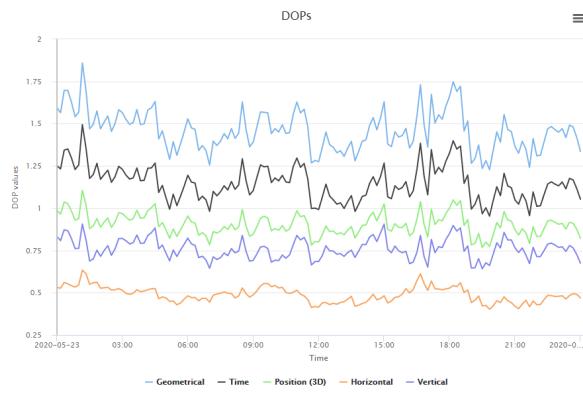
(n) Number of visible satellites of GPS in Punta Arenas



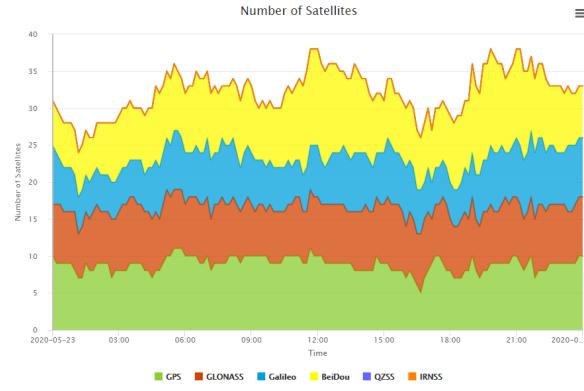
(o) DOPs with GPS and GLONASS in Punta Arenas



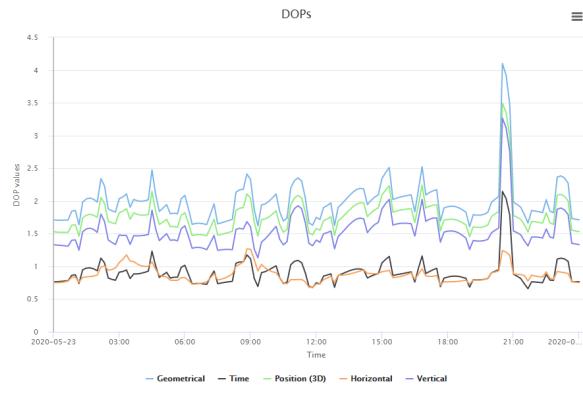
(p) Number of visible satellites of GPS and GLONASS in Punta Arenas



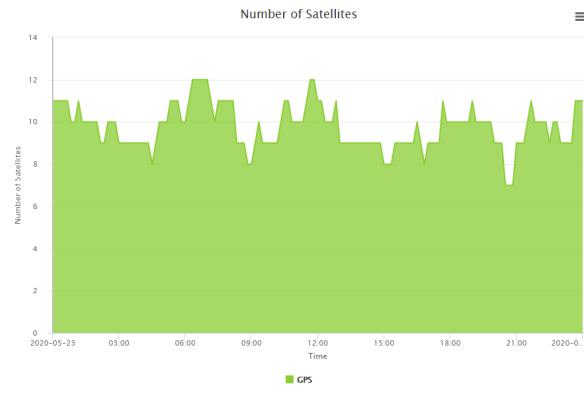
(q) DOPs with all constellations in Punta Arenas



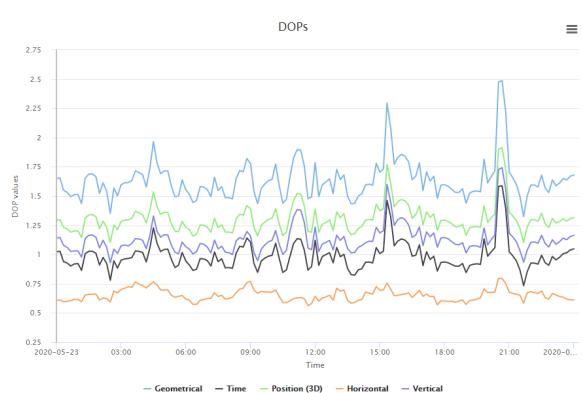
(r) Number of visible satellites of all constellations in Punta Arenas



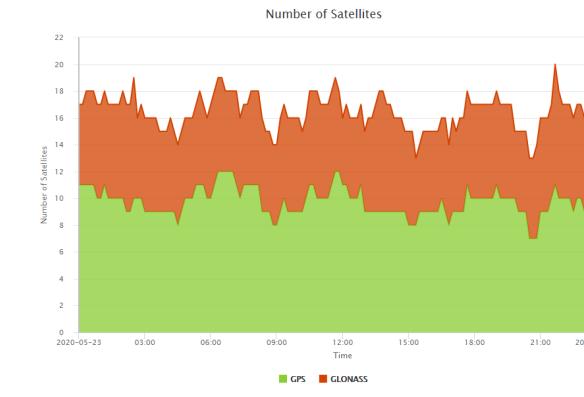
(s) DOPs with only GPS in Quito



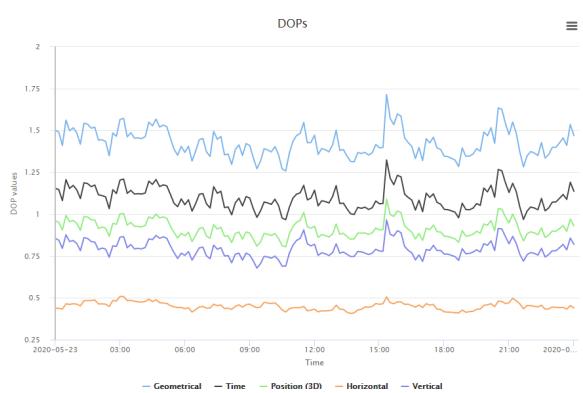
(t) Number of visible satellites of GPS in Quito



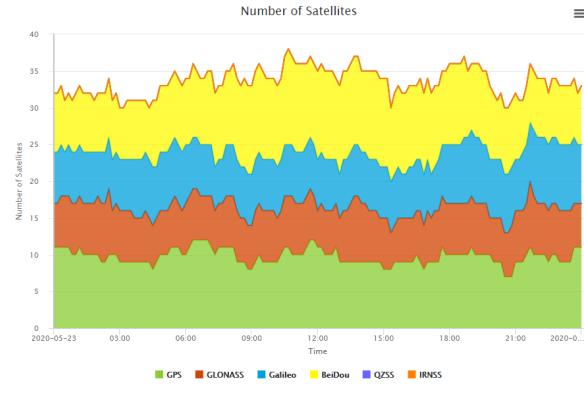
(u) DOPs with GPS and GLONASS in Quito



(v) Number of visible satellites of GPS and GLONASS in Quito



(w) DOPs with all constellations in Quito



(x) Number of visible satellites of all constellations in Quito

Figure 8: DOPs and number of visible satellites in each city over 24 hours

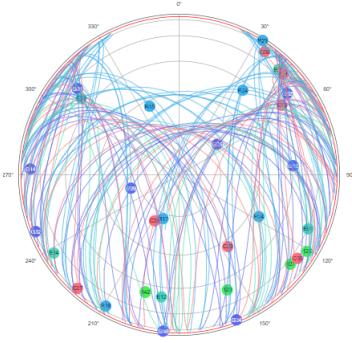
In Figure 8 are reported all trends over the planning period of the parameters Dilution of Precision for each city. Moreover, the trends of the number of visible satellites over time in each site are reported. By focusing on the values of the PDOP (trends in green), it can be seen that in some hours of the day, the value of this parameter has a spike in correspondence of a lower number of visible satellites. For example, by looking at Figure 8s and Figure 8t around 21:00 the value of PDOP is 3.5 and the number of visible satellites is only 7. This means that performing a survey based only on the GPS system at that time is not a good idea because it will be very poor, whereas the results would get better if all the constellations were considered, because the value of the PDOP at the same hour is around 1.

The same reasoning can be done for all the other cities.

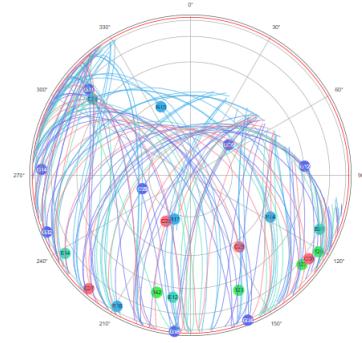
3.2 Adding obstacles

Plannings described in Section 2.1 are referred to an open space area without obstacles. By using another online tool⁷ allowing to design and display some objects to make the signal interfere with them, the planning has been repeated. In order to highlight differences in terms of visibility, only the site placed in Carpino (IT) has been used. The obstacle has been placed with an Azimuth angle of 35°, at a distance of 20 meters from the observer and starting directly from ground (offset 0 meter). Its dimensions were: height: 30 m; width: 50 m; depth: 20 m.

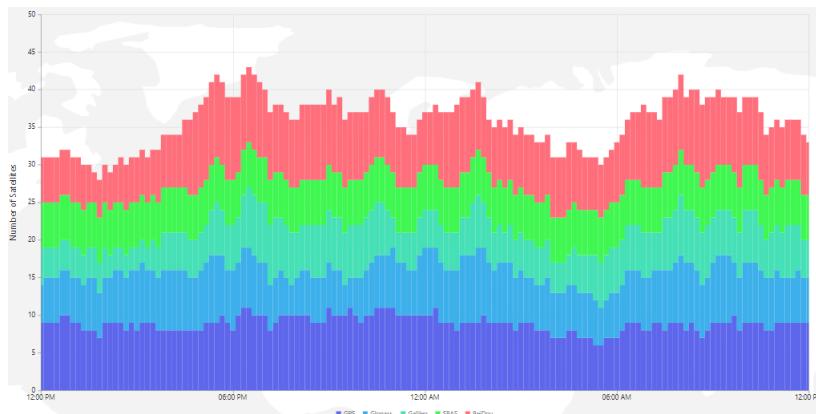
As it can be seen from Figure 9a and Figure 9b, the obstacle causes a decrease in visibility of satellites. Indeed, if the obstacle is not present, it's possible to follow all the trajectories of all the visible satellites, whereas a lost of trajectories is caused from the presence of the obstacles. Moreover, as it can be seen from the different scale on y-axes of Figure 9c and Figure 9d, also the number of visible satellites decreases over time if the obstacle is placed.



(a) SkyPlot without obstacles

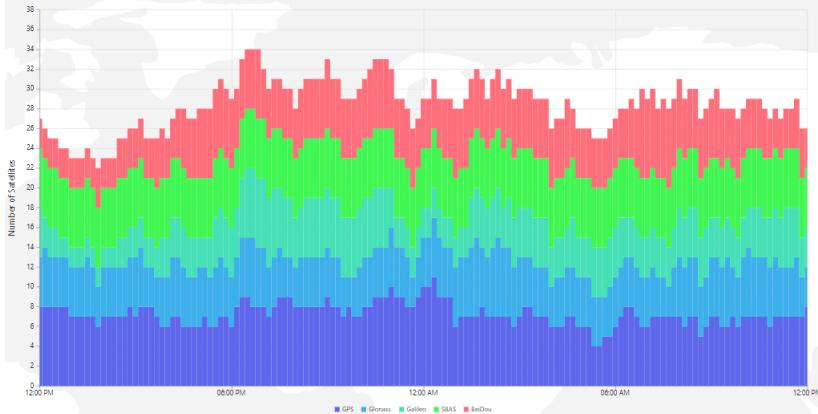


(b) SkyPlot with obstacles



(c) Number of visible satellites without obstacles

⁷<http://gnssmissionplanning.com/App/Settings>



(d) Number of visible satellites with obstacles

Figure 9: Comparison of results in Carpino (IT)

3.3 Azimuth and Elevation angle

The goal of the second exercise was to compute, by using Matlab, the Azimuth angle and the Elevation angle between a satellite and a GPS station.

Azimuth and Elevation angle are the two coordinates of the *Horizontal Coordinates* system: the former indicates the angle that the object in the sky (e.g. satellite) forms toward the horizontal direction with respect to the True North; the latter indicates the angle that the object in the sky forms with the horizon. They are indicated as follows:

$$\text{azimuth} = \arctan \frac{u}{\sqrt{n^2 + e^2}} \quad \text{elevation} = \arctan \frac{e}{n} \quad (6)$$

where n , u and e are the *Local coordinates*.

The input data of the exercise were the ECEF (*Earth-Centered Earth-Fixed*) coordinates X, Y and Z of the satellite position and the *Geographic* coordinates latitude and longitude of the station:

$$X_{sat} = 15487292.829 \text{ m}$$

$$Y_{sat} = 6543538.932 \text{ m}$$

$$Z_{sat} = 20727274.429 \text{ m}$$

$$\text{Latitude (station)} = 45^\circ 3' 48.114''$$

$$\text{Longitude (station)} = 7^\circ 39' 40.605''$$

By considering the Geographic coordinates of the station and the ECEF coordinates of both the satellite and the station, it has been possible to determine the Local coordinates of the satellite by applying the relation 7.

$$\begin{bmatrix} e \\ n \\ u \end{bmatrix} = \begin{bmatrix} -\sin \alpha & \cos \alpha & 0 \\ -\sin \phi \cos \alpha & -\sin \phi \sin \alpha & \cos \phi \\ \cos \phi \cos \alpha & \cos \phi \sin \alpha & \sin \phi \end{bmatrix} \cdot \begin{bmatrix} X_{sat} - X_{sta} \\ Y_{sat} - Y_{sta} \\ Z_{sat} - Z_{sta} \end{bmatrix} \quad (7)$$

where ϕ is the latitude in radians and α is the longitude in radians.

First of all, the provided geographic coordinates of the station were transformed from degree to radians, in order to compute the ECEF coordinates of the station. Then, $X_{sta}, Y_{sta}, Z_{sta}$ have been computed according to Equations 8.

$$\begin{cases} X_{sta} = \frac{a \cos \phi \cos \alpha}{W} \\ Y_{sta} = \frac{a \cos \phi \sin \alpha}{W} \\ Z_{sta} = \frac{a(1-e^2) \sin \phi}{W} \end{cases} \quad (8)$$

where $W = \sqrt{1 - e^2 \sin^2 \phi}$, a is the semi-major axis, $e^2 = 2f - f^2$ is the first eccentricity and f is the flattening. By considering the **WGS84** ellipsoid, the semi-major axis was 6378137.0 m and the flattening was $\frac{1}{298.257223}$.

In table 7 are reported the final obtained results.

Station Coord. [m]	Sat. Local Coord. [m]	Azimuth and Elevation [degree]
$X_{sta} = 4472327.04$	$e = 4420417.39$	$\text{azimuth} = 74.5972052$
$Y_{sta} = 601604.94$	$n = 3178617.82$	$\text{elevation} = 54.28097947$
$Z_{sta} = 4492325.03$	$u = 19762740.99$	

Table 7: Obtained results

3.4 DOP estimation

The aim of the third exercise was to estimate the GDOP, PDOP and HDOP and the \mathbf{Q}_{uu} matrix defined as follows:

$$\mathbf{Q}_{uu} = \begin{bmatrix} XDOP^2 & & & \\ & YDOP^2 & & \\ & & VDOP^2 & \\ & & & TDOP^2 \end{bmatrix} \quad (9)$$

$$GDOP = \sqrt{XDOP^2 + YDOP^2 + VDOP^2 + TDOP^2}$$

$$PDOP = \sqrt{XDOP^2 + YDOP^2 + VDOP^2}$$

$$HDOP = \sqrt{XDOP^2 + YDOP^2}$$

To determine the DOPs it is needed a first computation of the Euclidean Distance ρ_{sat} between the station and each visible satellite:

$$\rho_{sat} = \sqrt{(X_{sat} - X_{stat})^2 + (Y_{sat} - Y_{stat})^2 + (Z_{sat} - Z_{stat})^2} \quad (10)$$

After that, the unitary vector \mathbf{d}_{sat} pointing from the station towards the satellite position is evaluated as:

$$d_{x,J} = \frac{X_{satJ} - X_{sta}}{\rho_{sat}} \quad d_{y,J} = \frac{Y_{satJ} - Y_{sta}}{\rho_{sat}} \quad d_{z,J} = \frac{Z_{satJ} - Z_{sta}}{\rho_{sat}}$$

The \mathbf{D} matrix is defined as:

$$\mathbf{D} = \begin{bmatrix} d_{x,1} & d_{y,1} & d_{z,1} & -1 \\ d_{x,2} & d_{y,2} & d_{z,2} & -1 \\ d_{x,3} & d_{y,3} & d_{z,3} & -1 \\ \vdots & \vdots & \vdots & \vdots \\ d_{x,J} & d_{y,J} & d_{z,J} & -1 \end{bmatrix} \quad (11)$$

By exploiting Equation 11, the matrix $\mathbf{Q}_{xx} = (\mathbf{D}^T \mathbf{D})^{-1}$ has been computed. Then, matrix \mathbf{Q}_{xx}^* has been defined as $\mathbf{Q}_{xx}^* = \mathbf{Q}_{xx}(1 : 3, 1 : 3)$ and it has been exploited to compute $\mathbf{Q}_{uu}^* = \mathbf{R} \mathbf{Q}_{xx}^* \mathbf{R}^T$, where \mathbf{R} is the rotation matrix defined in Equation 7. Finally, the last row and the last column of matrix \mathbf{Q}_{uu} were set as the last row and the last column of \mathbf{Q}_{xx} .

Since the coordinates of the receiver were provided in geographic coordinates: Latitude = $45^\circ 3' 48.114''$; Longitude = $7^\circ 39' 40.605''$, they have been transformed in Horizontal Coordinates X, Y, Z by considering the ellipsoid WGS84.

The obtained \mathbf{Q}_{uu} matrix were:

$$\mathbf{Q}_{uu} = \begin{bmatrix} 0.2594 & & & \\ & 0.2943 & & \\ & & 1.1329 & \\ & & & 0.3705 \end{bmatrix}$$

and the computed parameters were: GDOP = 1.434225, PDOP = 1.298673 and HDOP = 0.744066.

4 GNSS data processing

The goal of this lab was to process GNSS data obtained considering two different techniques of *Relative Positioning*: **Static positioning** and **Kinematic positioning**. Moreover, another goal was to estimate the Single point positioning of the known point (IGM⁸ point). The used tool was RTKLIB.

4.1 Single point positioning

To estimate the Single point positioning, the data has been processed with "RTKPost" tool twice: once considering only the GPS system and then by using both GPS and GLONASS. Before starting the process with the tool, some options have been set: for this kind of positioning, in "Setting 1" has been set the "Positioning mode" on *single* and in output has been set as "Solution Format" Latitude/Longitude/Height. Moreover, the "observation file" containing pseudoranges and carrier phases collected by the rover has been uploaded in section "ROVER OBS" and the two "navigation files" of GPS system (*.18n) and of GLONASS system (*.18g), containing parameters to compute the Keplero's Laws and Ionosphere corrections have been uploaded in section "RINEX NAV".

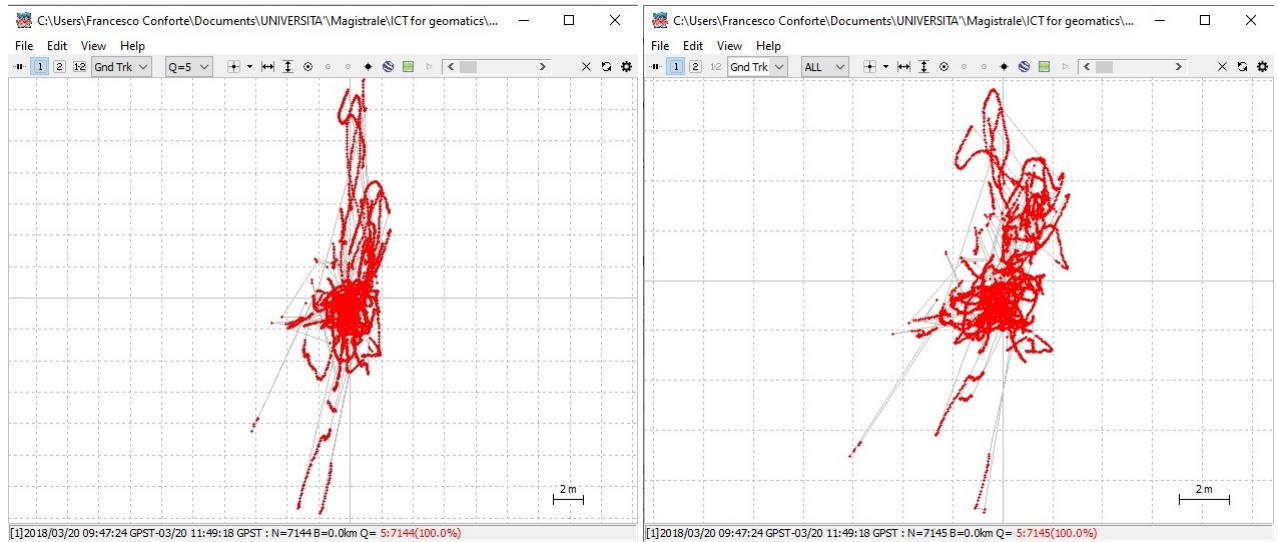


Figure 10: Different tracks Single point positioning

In Figure 10 are reported the different tracks of position considering first only the GPS and then GPS and GLONASS systems. At the bottom of both the figures, it can be read the time range of the data collection, the number of solution epochs (N=7144), the baseline length (B=0 because no master stations were considered for this kind of processing), the number and percentage of each quality solutions (Q=5:7144(100%)). These last values mean that 100% of the epochs have quality equal to 5, that is a solution by single point positioning with no phase ambiguity because a master receiver is not considered.

Moreover, from the output file .POS that the RTKPost tool output after the process, it can be seen the number of visible satellites for each second of observation. It's clear that, when the GLONASS system has been also included, the number of visible satellites was greater than the first case. This observation affects the values of the standard deviations for each components as well: horizontal ones and vertical ones. In both of files, the greatest value of standard deviation was for the vertical components, reaching values of around 10 meters for GPS only and 9 meters for GPS and GLONASS. Other components had lower values, around 2 and 3 meters. Thus, the obtained results were very poor.

4.2 Relative Static Positioning

The *Static Positioning* is a technique that foresees the placement of a receiver on a known point for a long time and collecting data. Then, such data are processed a-posteriori and allow to reach a precision

⁸Istituto Geografico Militare (ITALY)

of $10^{-6} \div 10^{-8}$ multiplied by the length of the baseline between the receiver and the Master station in estimating the precise point.

During the lab, two different master stations were considered: one was the Master station of the Politecnico di Torino, called "TURI" and one was a "VIRTUAL STATION".

First of all, the post-processing action has been done by considering TURI, whose reference coordinates are [45.063° 7.661° 310.8177 meters].

Before starting the process with the tool, some options have been set: for this kind of positioning, in "Setting 1" the "Positioning mode" has been set on *Static* and the Filter type on "Forward", meaning that it sets the Kalman filter forward in time. In "Setting 2", the "Integer Ambiguity Resolution" has been set on "Continuous" for GPS, meaning that continuously, static integer ambiguities were estimated and resolved and on "ON" for GLONASS, meaning that the ambiguities were fixed. The output has been always set on "Solution Format" = Latitude/Longitude/Height.

It has been also re-processed by setting the type of the Filter on "Combined", meaning that smoother combined solution with forward and backward filter solutions was obtained.

In the first case, when the filter was set on "Forward" and considering only the GPS, it has been obtained the result: "Q=1:4804(67.2%) 2:2340(32.8%)" (Figure 11a). It means that the 67.2% of the epochs had quality equals to 1, i.e. for the 67.2% of the epochs, the tool could fix the phase ambiguity, that is considering the phase ambiguity as an integer; whereas the 32.8% of the epochs had quality equals to 2, i.e. the phase ambiguity was *float*. On the other hand, when the filter type was "Combined" the result was: "Q=1:5638(78.9%) 2:1506(21.1%)". It was a better result from the previous case, because a combined solution between Forward and Backward has been used.

Figure 11 shows how the Ratio Factor (Ratio Test) varies over the data collection time. When the trend is green, it means that the the phase ambiguity can be fixed by the tool, because the result of the Ratio Test is clear. Indeed, the values in green are all over 3. When the trend is yellow, it means that the tool is not able to fix the ambiguity, thus it leaves the phase as a float value. Indeed, the values of the Ratio Test are lower than 3, meaning that there are at least 2 values of phase ambiguity which give the same performance in terms of precision.

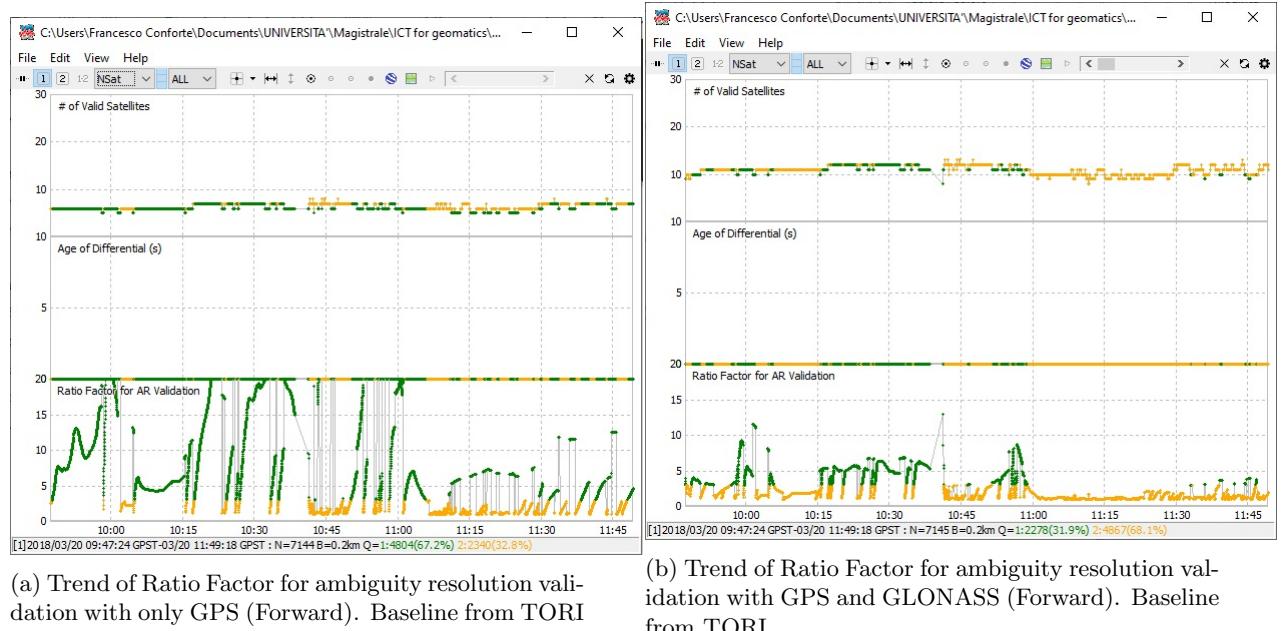
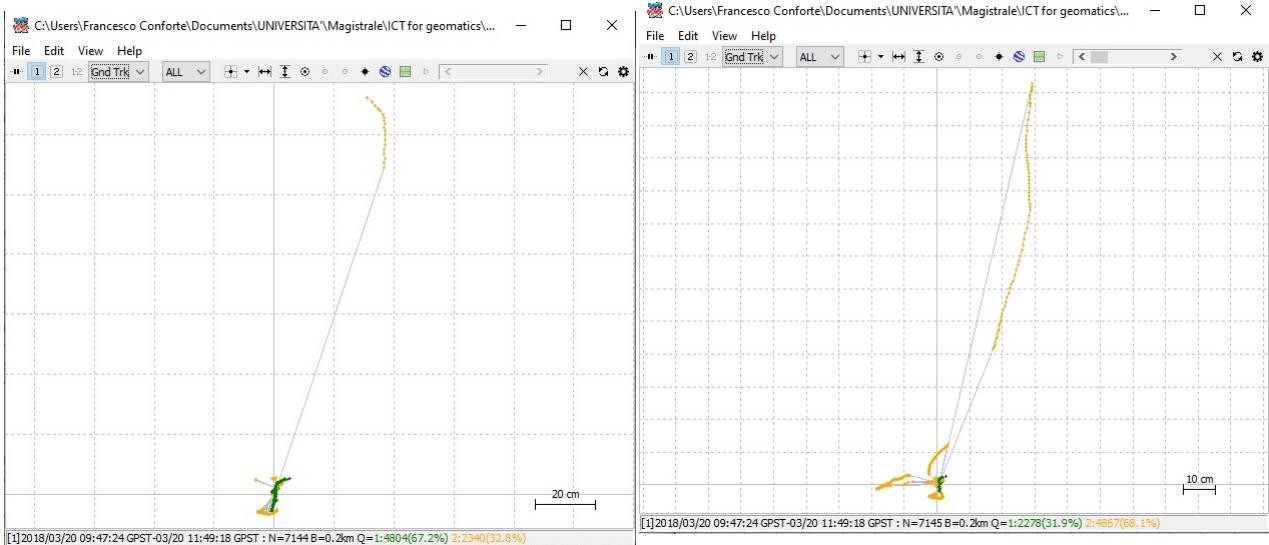


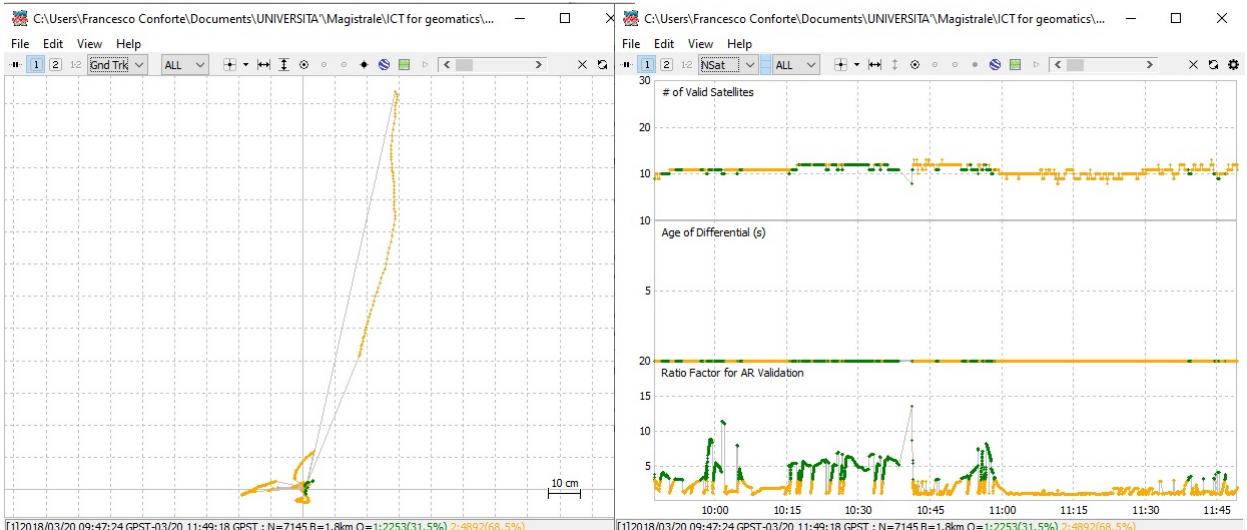
Figure 11: Trend of Ratio Factor for ambiguity resolution validation and number of visible satellites over time



(a) Tracking of positions estimations over epochs considering GPS. Baseline from TORI

(b) Tracking of positions estimations over epochs considering both GPS and GLONASS. Baseline from TORI

Figure 12: Tracking of estimated position over epochs. Baseline from TORI



(a) Tracking of estimated positions over epochs considering both GPS and GLONASS. Baseline from the Virtual Station

(b) Trend of Ratio Factor for ambiguity resolution validation with GPS and GLONASS (Forward). Baseline from the Virtual Station

Figure 13: Tracking of estimated position over epochs and Ration Factor. Baseline from the Virtual Station

Figure 12 shows the difference between the tracking obtained by considering only the GPS system and the tracking obtained by considering both GPS and GLONASS. At the bottom of the figures are reported the length of the baseline from TORI station, which was $B=0.2$ Km from the Rover. Moreover, when both GNSS systems were considered, the majority of the epochs had a quality equal to 2, i.e. the phase ambiguity was not fixed, on the contrary of when the GPS only was considered.

In Figure 13, instead, the results obtained by using VIRTUAL RINEX file are reported.

The Master station's coordinates were: [45.05° 7.65° 320.0791 meters].

At the bottom of both Figures 13a and 13b is reported the length of the baseline between the Rover and the Virtual Station: $B=1.8$ km. Moreover, as when the Master Station was TORI, the majority of the epochs had quality $Q=2$.

From the Solution Files got out by the RTKPost, it has been possible to see the difference in standard deviation values of each components of the position between when $Q=1$ and $Q=2$. When the phase

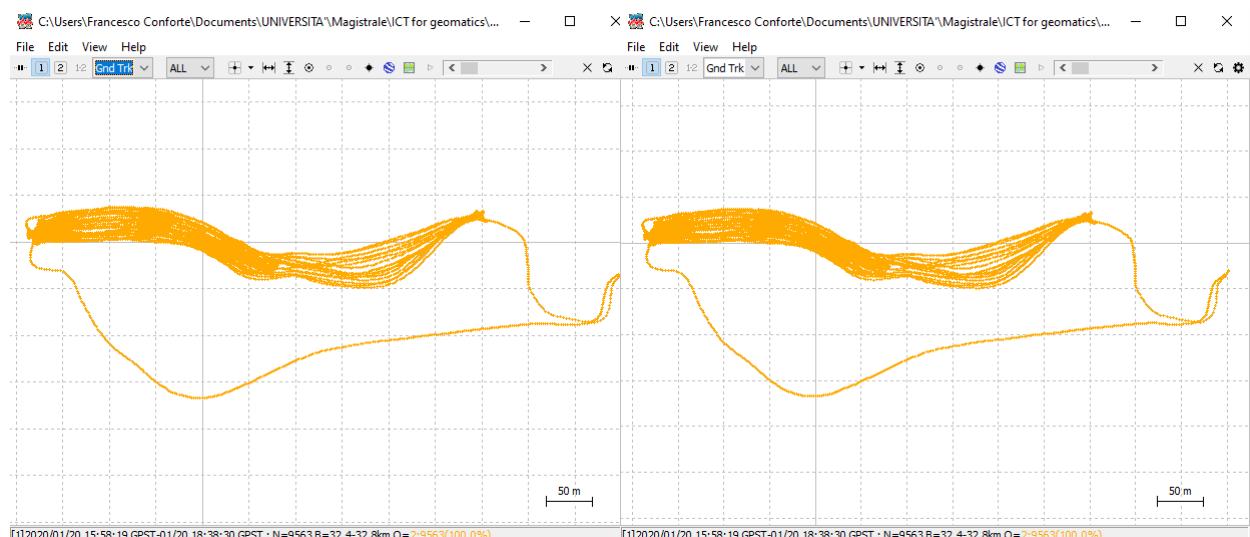
ambiguity was fixed ($Q=1$), the standard deviation of each component, for the vertical one as well, was of the order of millimeter, whereas when the phase ambiguity was floating ($Q=2$), the std was of the order of centimeters.

4.3 Relative Kinematic positioning

The *Kinematic Positioning* is a technique that foresees a continuous collection of data from a moving rover, like a vehicle or a drone. Then, such a data are processed in PPK (*Post Processing Kinematic*) a-posteriori, by considering epoch-by-epoch a Master Station as reference from which computing the baseline. This technique allows to achieve a precision of the order of the centimeters.

The goal for this lab was to process data collected by a snow groomer using the Master Station MONV as reference from which computing the baseline. Its geographic coordinates were: 44.390356851° , 7.828955133° , 637.6965 meters. The target was to compute continuously the trajectory of the snow groomer using two different strategies about the phase ambiguity estimation: *FIX and HOLD* and *Continuous*. The first one, Fix and Hold, means that static integer phase ambiguities are continuously estimated and resolved and if the ratio test is positive, the ambiguities are kept fixed at the values estimated at the previous time, otherwise they are recomputed again⁹. The second one, Continuous, means that static integer phase ambiguities are continuously estimated and resolved for each epoch.

First of all, some option in "Setting 1" has been selected: the "Positioning mode" has been set on *Kinematic* and the Filter type on "Forward". Moreover, both GPS and GLONASS have been selected. In "Setting 2", the "Integer Ambiguity Resolution" has been set on "FIX and HOLD" for GPS and on "ON" for GLONASS. Subsequently, the process has been repeated by selecting "Continuous" for the "Integer Ambiguity Resolution".



(a) Tracking of estimated positions over epochs considering both GPS and GLONASS with FIX and HOLD. Baseline from the MONV station

(b) Tracking of estimated positions over epochs considering both GPS and GLONASS with Continuous. Baseline from the MONV station

⁹This happens for example when there is a cycle slip due to a loss in tracking the satellites



(c) Tracking of estimated positions over epochs on Google Earth



(d) Trend of Ratio Factor over time

Figure 14: Tracking of estimated position over epochs and Ration Factor. Baseline from the Virtual Station

In Figures 14a and 14b are reported the estimated trajectories of the vehicle for both cases: "Fix and Hold" and "Continuous". Moreover, in Figure 14c the computed trajectories is reported on Google Earth.

At the bottom of the figures are reported the length of the baseline B, varying from 32.4 Km to 32.8 Km. There is also the quality of the obtained results: $Q=2.9563(100\%)$, meaning that 100% of the epochs have quality equal to 2, i.e. the tool hasn't been able to Fix the phase ambiguities and it has considered them as *float*. Indeed, by looking at the trend of Ratio Factor, the values are always around 1 and they don't overcome the value 3. This means that there were at least 2 values of N which gave the same performances in terms of precision. Hence, the tool didn't fix the ambiguity value.

The same results in terms of estimated trajectories and same number of float phase ambiguities were obtained both with "Continuous" and "Fix and Hold". Even in terms of standard deviation of each component have been obtained the same results: std of orders of centimeters for the first epochs and millimeters then.

From the output file it can be seen that for some seconds no available data there were and this could also be seen from the fact that the standard deviation passed instantaneously from the order of millimeters to the order of meters, but after some epochs, the order returned on centimeters.

5 Geodesy

5.1 Ellipsoid parameters

The goal of the first exercise of this lab was to estimate several parameters for different Ellipsoids.

The *ellipsoid* is a reference surface defined by a simple mathematical function and used to represent an approximation of the Earth's form. Its equation is:

$$\frac{X^2 + Y^2}{a^2} + \frac{Z^2}{c^2} = 1 \quad (12)$$

where a is the semi-major axis and c is the semi-minor axis. X, Y, Z are the ECEF coordinates, because the used model of ellipsoid is the Geocentric one. From the two semi-axes, other parameters can be obtained: the flattening α , the first eccentricity e^2 and the second eccentricity e'^2 .

$$c = a(1 - \alpha); \quad e^2 = \frac{a^2 - c^2}{a^2}; \quad e'^2 = \frac{a^2 - c^2}{c^2} \quad (13)$$

With a Matlab code, semi-minor axes c , the first eccentricity e^2 and the second eccentricity e'^2 have been computed. The results are reported in Table 8.

Ellipsoid	a [m]	α	c [m]	e^2	e'^2
Delambre	6376985	1/308.6	6356320.76	0.0064703	0.0065125
Everest	6377276	1/300.8	6356074.95	0.0066378	0.0066822
Bessel	6377397	1/299.1528128	6356078.81	0.0066743	0.0067192
Fisher	6378338	1/288.5	6356229.38	0.0069203	0.0069686
Clarke	6378249	1/293.5	6356517.32	0.0068027	0.0068492
Helmhert	6378140	1/298.3	6356758.37	0.0066934	0.0067385
Hayford	6378388	1/297.0	6356911.95	0.0067226	0.0067681
Krassovsky	6378245	1/298.3	6356863.09	0.0066934	0.0067385
WGS84	6378137	1/298.257223563	6356752.32	0.0066943	0.0067394

Table 8: Ellipsoids' parameters

5.2 Coordinates transformation from (ϕ, λ, h) to (XYZ)

The goal of the second exercise was to represent with their *Geocentric Coordinates* two different points provided with their *Geographic Coordinates*. Thus, the target was to pass from a Geographic Coordinates system to a Geocentric Coordinates System (ECEF), by considering two different ellipsoids: **WGS84** and **Hayford**.

The transformation can be done applying directed equations reported in Eq. 14.

$$\begin{cases} X = (\frac{a}{W} + h) \cos \phi \cos \lambda \\ Y = (\frac{a}{W} + h) \cos \phi \sin \lambda \\ Z = [\frac{a}{W}(1 - e^2) + h] \sin \phi \end{cases} \quad (14)$$

where $W = \sqrt{1 - e^2 \sin^2 \phi}$, a is the semi-major axis, e^2 is the first eccentricity, ϕ is the latitude in radians, λ is the longitude in radians and h is the height in meters.

In Table 9 are reported the input points in Geographic coordinates to be transformed. In Table 10 and Table 11 are reported the computed ECEF coordinates.

	Latitudine (ϕ)	Longitudine (λ)	h (m)
Point 1	44°45' 01.03930"	7°24' 29.20335"	322.4909
Point 2	44°47' 10.90505"	7°30' 26.53939"	305.7367

Table 9: Input points in degrees, minutes, seconds

	X [m]	Y [m]	Z [m]
Point 1	4496896.47942384	604909.956067010	4467910.35953867
Point 2	4495694.26953271	592457.860452754	4470744.77809804

Table 10: Points in ECEF considering WGS84

	X [m]	Y [m]	Z [m]
Point 1	4497105.06973958	604938.01505376	4467990.35663277
Point 2	4495902.84493447	592485.347227624	4470824.86623251

Table 11: Points in ECEF considering Hayford

Finally, the height of the Point 1 has been increased by 2000 meters and the transformation has been performed again, by only considering WGS84 as reference ellipsoid. The results are reported in Table 12. Moreover, the difference for each component between the point estimated in Table 10 and the one estimated considering an increased height is reported as well.

Such differences explain the fact that a variation of the height in Geographic Coordinates cannot be clearly identified in ECEF coordinates, because the changing in height is distributed, is projected over the 3 Geocentric components x,y and z.

	X [m]	Y [m]	Z [m]
Point 1	4498304.16423566	605099.313896698	4469318.39614434
	$\Delta X = 1407.68481182$	$\Delta Y = 189.357829688$	$\Delta Z = 1408.03660567$

Table 12: Point 1 in ECEF with h increased of 2000 m

5.3 Coordinates transformation from (XYZ) to (ϕ, λ, h)

The goal of the third exercise was to perform the reverse transformation from ECEF coordinates to Geographic coordinates of the points reported in Table 13, by considering the same ellipsoids of Section 5.2.

Points	X [m]	Y [m]	Z [m]
1	4499525.4271	585034.1293	4467910.3596
2	4495694.2695	592457.8605	4470744.7781
3	4503484.7172	578160.7507	4465024.3002
4	4498329.3715	562840.7651	4472537.6125

Table 13: Input points in ECEF

If obtaining the longitude λ can be done applying a direct equation (Eq. 15), obtaining the other two components, latitude ϕ and height h , needs to perform a loop (Algorithm 1):

$$\lambda = \arctan \frac{Y}{X} \quad (15)$$

Loop:

- Estimate a value of ϕ by approximating the radius r of a sphere: $r = \sqrt{X^2 + Y^2}$. $\phi = \arctan \frac{Z}{r}$
- Compute an approximation of $N = \frac{a}{\sqrt{1 - e^2 \sin^2 \phi}}$
- Compute an approximation of $h = \frac{X}{\cos \phi \cos \lambda} - N$
- Estimate a new value of $\phi = \arctan \frac{Z}{r(1 - \frac{e^2 N}{N+h})}$
- check if $\phi_n - \phi_{n-1} < \epsilon$ and if $h_n - h_{n-1} < \epsilon$ where n indicates the current loop and n-1 the previous one.

If *true*, then end the loop. If *false*, then restart from point 2

At the end of the process, the obtained values have been transformed in degree. In Table 14 and Table 15 are reported the computed values of each components.

Points	ϕ [degree]	λ [degree]	h [m]
1	44.7502886948632	7.40811204154704	322.490937582217
2	44.7863625140679	7.50737205342363	305.736680135131
3	44.7125504912773	7.31565904880963	455.195298180915
4	44.8051624043273	7.13190879187592	745.962230150588

Table 14: Points in Geographic Coordinates with WGS84

Points	ϕ [degree]	λ [degree]	h [m]
1	44.7511107909513	7.40811204154704	116.700870447792
2	44.7871846189175	7.50737205342363	100.004135328345
3	44.7133725619345	7.31565904880963	249.345055660233
4	44.8059844552651	7.13190879187592	540.259663403034

Table 15: Points in Geographic Coordinates with Hayford

As it can be seen, the main difference between the two results reported in Tables 14 and 15 is about the height. This fact is of course due to the difference in the shape between WGS84 ellipsoid and Hayford ellipsoid. Whereas, the difference in the angular components in not so visible.

5.4 RS Transformation

The goal of the fourth exercise was to perform transformations of reference systems from *Local* reference systems ETRF¹⁰ to *Global* reference systems ITRF¹¹, for the ECEF points in Section 5.2 using a product online provided by EUREF consortium¹²:

- ETRF89 \Rightarrow ITRF89
- ETRF2000 \Rightarrow ITRF2000

In Table 16 and 17 are reported the ECEF coordinates of the two points in different reference systems. For Point 1, the transformation from ETRF89 to ITRF89 has a difference of $\Delta X = 15.9$ cm, $\Delta Y = 19.6$ cm, $\Delta Z = 13.3$ cm. The differences for Point 2 have the same order of magnitude. The transformation from ETRF2000 to ITRF2000 has a difference of $\Delta X = 19.6$ cm, $\Delta Y = 15.8$ cm, $\Delta Z = 16.3$ cm. Such differences are due to natural phenomenon that affect the Earth, like the tectonic plate movements.

		X [m]	Y [m]	Z [m]
Point 1	ETRF89	4496896.47942	604909.95606	4467910.35953
	ITRF89	4496896.32070	604910.15250	4467910.49270
Point 2	ETRF89	4495694.26953	592457.86045	4470744.77809
	ITRF89	4495694.11120	592458.05690	4470744.91130

Table 16: Transformation from ETRF89 to ITRF89

		X [m]	Y [m]	Z [m]
Point 1	ETRF2000	4496896.47942	604909.95606	4467910.35953
	ITRF2000	4496896.28310	604910.11430	4467910.52240
Point 2	ETRF2000	4495694.26953	592457.86045	4470744.77809
	ITRF2000	4495694.07370	592458.01860	4470744.94100

Table 17: Transformation from ETRF2000 to ITRF2000

¹⁰EUROPEAN TERRESTRIAL REFERENCE SYSTEM

¹¹INTERNATIONAL TERRESTRIAL REFERENCE SYSTEM

¹²http://www.epncb.oma.be/_productsservices/coord_trans/index.php

5.5 Helmert Transformation

The *Helmert Transformation* is an equation allowing to pass from a reference system A to a reference system B. It combines 7 parameters: 3 for the translation of the origin of the axis ($\Delta T_x, \Delta T_y, \Delta T_z$), 3 for the rotation of each axis (R_x, R_y, R_z) and 1 for the scale factor ($\Delta\lambda$).

A way to find these parameters is by using at least 3 points that are known for both the reference systems: if the coordinates of 3 points for both the reference systems are known, it's possible to retrieve them.

The goal of this exercise was to retrieve the *Helmert's parameters* applying the Least Square solution, by starting from 12 points whose ECEF coordinates in reference system A and in reference system B were known.

The transformation equation is reported in Eq 16.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_B = \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix} + \lambda \cdot \begin{bmatrix} 1 & R_z & -R_y \\ -R_z & 1 & R_x \\ R_y & -R_x & 1 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_A \quad (16)$$

First of all, in order to avoid singular matrix, the coordinates of the known points (X, Y, Z) have been divided by 10^6 .

Then, the Least Squares method has been applied to estimate the Helmert's parameters:

$$\mathbf{x} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{l}_0 = \begin{bmatrix} \Delta T_x \\ \Delta T_y \\ \Delta T_z \\ \Delta \lambda \\ R_x \\ R_y \\ R_z \end{bmatrix} \quad (17)$$

where

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & X_A & 0 & -Z_A & Y_A \\ 0 & 1 & 0 & Y_A & -Z_A & 0 & -X_A \\ 0 & 0 & 1 & Z_A & Y_A & X_A & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \quad (18)$$

and

$$\mathbf{l}_0 = \begin{bmatrix} X_B - X_A \\ Y_B - Y_A \\ Z_B - Z_A \\ \vdots \end{bmatrix} \quad (19)$$

Now, the residuals of the transformation can be determined by:

$$\nu = \mathbf{Ax} - \mathbf{l}_0$$

The obtained parameters are reported in Table 18. The computed residuals are smaller than 0.0143 and this means that the parameters estimation has been accurately performed, since the maximum value of the residual error is not high.

ΔT_x	ΔT_y	ΔT_z	$\Delta \lambda$	R_x	R_y	R_z
1.9937	0.8960	0.4532	-0.2830	0.3255	0.1950	-0.2444

Table 18: Estimated Helmert's parameters

6 Cartography

6.1 Modulus of linear deformation

The goal of the first exercise of this lab was to see how the *modulus of linear deformation* varies, changing different values of latitude and changing the longitude from 0° to 3° with a pace of 0.5° .

The modulus of linear deformation m_l depends also on the *contraction modulus* $m_c = 0.9996$:

$$m_l = 0.9996 \left(1 + \frac{\lambda^2}{2} \cos^2 \phi\right)$$

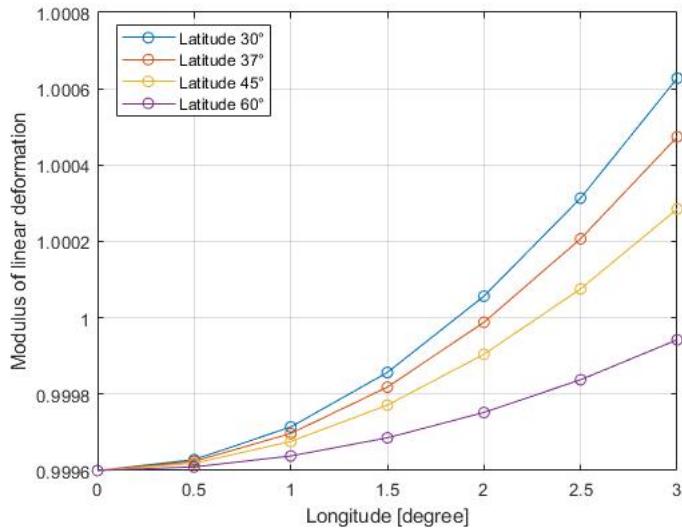


Figure 15: Trend of m_l for different latitude and longitude

Figure 15 shows the trend of the modulus of linear deformation for different values of latitude and for the longitude starting from 0° up to 3° . It can be seen how not all the points of the fuse are deformed fairly, but the deformation depends strongly from latitude and longitude. Focusing, for example, on latitude 30° , it can be seen that for points between 0° and 1.8° of longitude, the modulus of linear deformation is lower than 1. This means that those points on maps are represented in a smaller way with respect the reality. Instead, the value of m_l is bigger than 1 for values of longitude from 1.8° and 3° . This means that those points are represented in a bigger way on maps with respect the reality.

The same cannot be said for greater latitudes. For example, for latitude of 60° , the values of m_l doesn't overcome 1. This means that every point of the fuse at 60° is represented in a smaller way on maps with respect the reality.

6.2 From geographic (ϕ, λ) to cartographic coordinates (N,E)

The goal of this exercise was to use the *Hirvonen's formulas* to represent two points, known with their geographic coordinates in UTM system, with their cartographic coordinates North and East.

The two given points are reported in Table 19. To perform the transformation, the WGS84 ellipsoid has been considered. Moreover, since the points were given in 2 different UTM zones (32 and 33), two different central meridians were considered, 9° and 15° respectively for UTM zone 32 and 33; the False East was 500 Km.

Points	UTM zone	Central Meridian [deg]	Latitude [dms]	Longitude [dms]
1	32	9 E	45°3' 45,717"	7°47' 26,292"
2	33	15 E	38°32' 34,649"	16°50' 06,493"

Table 19: Points in geographic coordinates

After computing the coordinates x and y of both the points, the cartographic coordinates North and East were computed according to Eq. 20.

$$\begin{cases} East &= x \cdot m_c + FalseEast \\ North &= y \cdot m_c \end{cases} \quad (20)$$

where m_c is the contraction modulus 0.9996.

In Table 20 are reported the new estimated coordinates.

Points	UTM zone	Central Meridian [deg]	North [m]	East [m]
1	32	9 E	4990626,881	404787,687
2	33	15 E	4267656,350	659931,128

Table 20: Points in cartographic coordinates

The North points are of the order of near to $5 \cdot 10^6$ meters, whereas the East is of the order of $4 \cdot 10^5$ meters.

6.3 From cartographic coordinates (N,E) to geographic (ϕ, λ) coordinates

The last part of the lab aimed to perform the inverse of the previous exercise, i.e. transforming a point given in cartographic coordinates to geographic coordinates, using the Hirvonen's equations. The given point was:

- East = 470139,66 m
- North = 5031468,37 m

The considered ellipsoid was always WGS84 and the UTM zone was 32, thus the considered central meridian was 9°E.

The obtained point in geographic coordinates was:

- latitude = $45^\circ 26' 9'',9617$
- longitude = $8^\circ 37' 5'',6209$

All the equations have not been reported for brevity, but only the most important ones are listed below.

- $\phi = \arctan [\tan(\xi) \cos(\nu\lambda')]$
- $\lambda' = \arctan \frac{\nu \sinh \frac{x}{R_p}}{\cos \xi}$
- $\lambda = \lambda' + \lambda_{m_c}$, where λ_{m_c} is the central meridian of the fuse
- $R_p = \frac{a^2}{c}$, where a is the semi-major axis and c is the semi-minor axis of WGS84 ellipsoid

7 Geographical Information System - GIS

7.1 Georeferencing an image with ARCGMap

The goal of this exercise was to transform a raster image of Caselle (TO) in a Georeferenced map. A *Georeferenced image*, in general, is a particular image for which each pixel is associated to coordinates.

The target has been realized by using a tool of ARCGIS called *ARCMAP*.

The first step was to upload on the tool the raster image *CASELLE.TIF* and realizing the pyramid of the image, i.e. creating several level of resolution in order to apply the zoom on the image.

After this basic step, the actual Georeferencing step started. By using the tool *Georeferencing* of ArcMap and thanks to the fact that the uploaded image had already a grid drowned on it, with the cartographic coordinates reported on each edge, 5 intersection points have been selected, for which the cartographic coordinates were known: 2 for the upper part, 2 for the lower part and 1 in the middle of the image. For each of them, the coordinates X and Y have been modified and set to the ones reported on the grid. In Table 21 are reported the chosen points: the X/Y sources are the coordinates of the chosen point before applying georeferencing, the X/Y map (East a North coordinates) are the coordinates after applying the georeferencing and the Residuals are the errors due to the fact that the

points have been chosen manually and this doesn't allow to reach a maximum precision with respect the estimated transformation. The last column is the standard deviation of the Residual x and y in meters. Their values are not too high, meaning that the points have been chosen properly and not too far away from the estimated ones.

X source	Y source	X map[m]	Y map [m]	Residual x [m]	Residual y [m]	Residual [m]
26.19881095	6.34004566	395000	5001000	-0.51935761	-0.74016989	0.90420340
2.57979844	26.05716163	389000	5006000	-0.74056780	-0.77937063	1.07510894
2.60744637	6.32840962	389000	5001000	-0.36268946	0.58228676	0.68600403
14.37742090	18.16483459	392000	5004000	-0.57790465	0.39373442	0.69928576
26.17610889	26.05730872	395000	5006000	-0.58185101	0.54351933	0.79621848

Table 21: Link Table

Then, the georeferencing has been updated and a file of format .TFwx was created. It is the Georeferencing file, a text file in which are reported:

1. In 1-st and 4-th lines are reported the sizes of the pixels in meters;
2. In 2-nd and 3-rd lines are reported the rotations;
3. In 5-th and 6-th lines are reported the cartographic coordinates of top-left angle of the georeferenced image.

In Table 22 are reported their values.

Size of pixel [m]	0,6357589
Rotation	-0,000158275
Rotation	-0,000838615
Size of pixel [m]	0,6337854
East [m]	388344,962
North [m]	5006279,084

Table 22: Results reported in CASELLE.TFwx file

Finally, to test if the Georeferencing action has been performed correctly, a file containing the map of the roads has been uploaded and overlapped on the new Georeferenced Image. In Figure 16 it can be seen how the official map is coincident with the one that has been georeferenced.

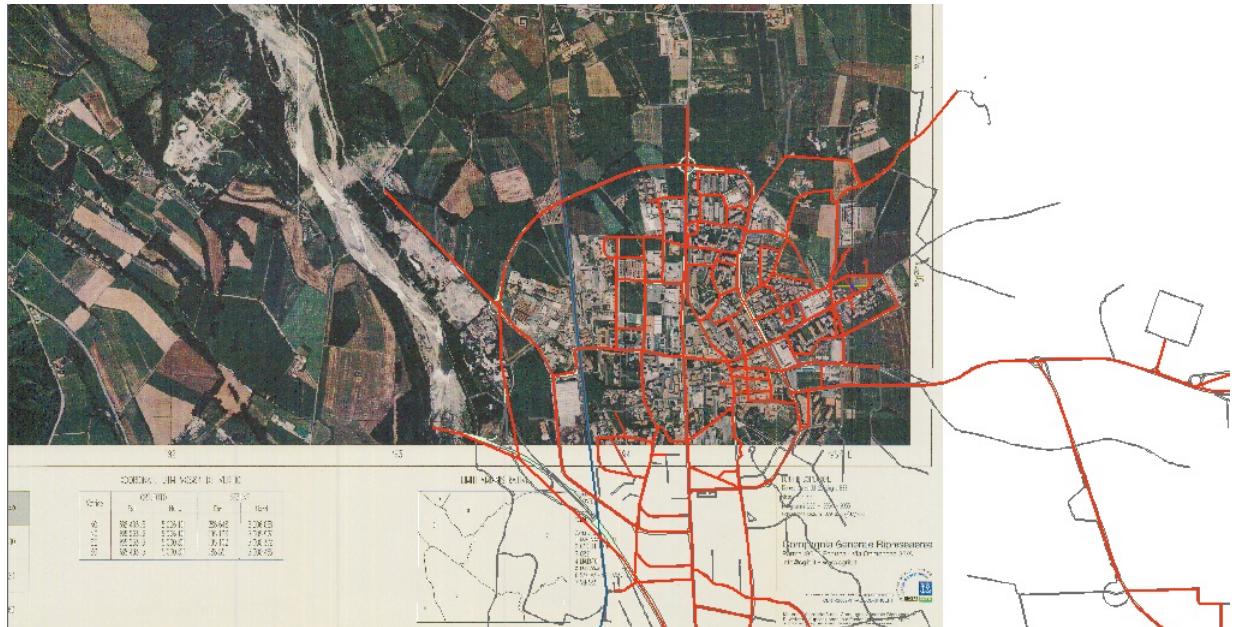


Figure 16: Road map and Georeferenced Image overlapped

7.2 Digital Terrain Model - DTM

In the second part of this lab, it has been used always the tool ArcMap. The goal was to use the vector data of TURIN Scale 1:1000, to extract the points with known elevation and to create a DTM.

Firstly, by using the tool *select by attributes* of ArcMap, it has been made possible to extract from a big database contained in a CAD file .DWG three different attributes:

- Layer S_08_06 = ground point with elevation (point)
- Layer S_08_07= point on the eaves (figure) (point)
- Layer A_02_01= building (polygon)

Each of this feature was extracted singularly from the database making a query by considering their own "*layer*" (name); for example, in order to extract all the ground points with elevation, the query "Layer" = 'S_08_06' has been applied. The same has been done with the other two attributes. Furthermore, all of these points were exported and in Figures 17 are reported the exported points.

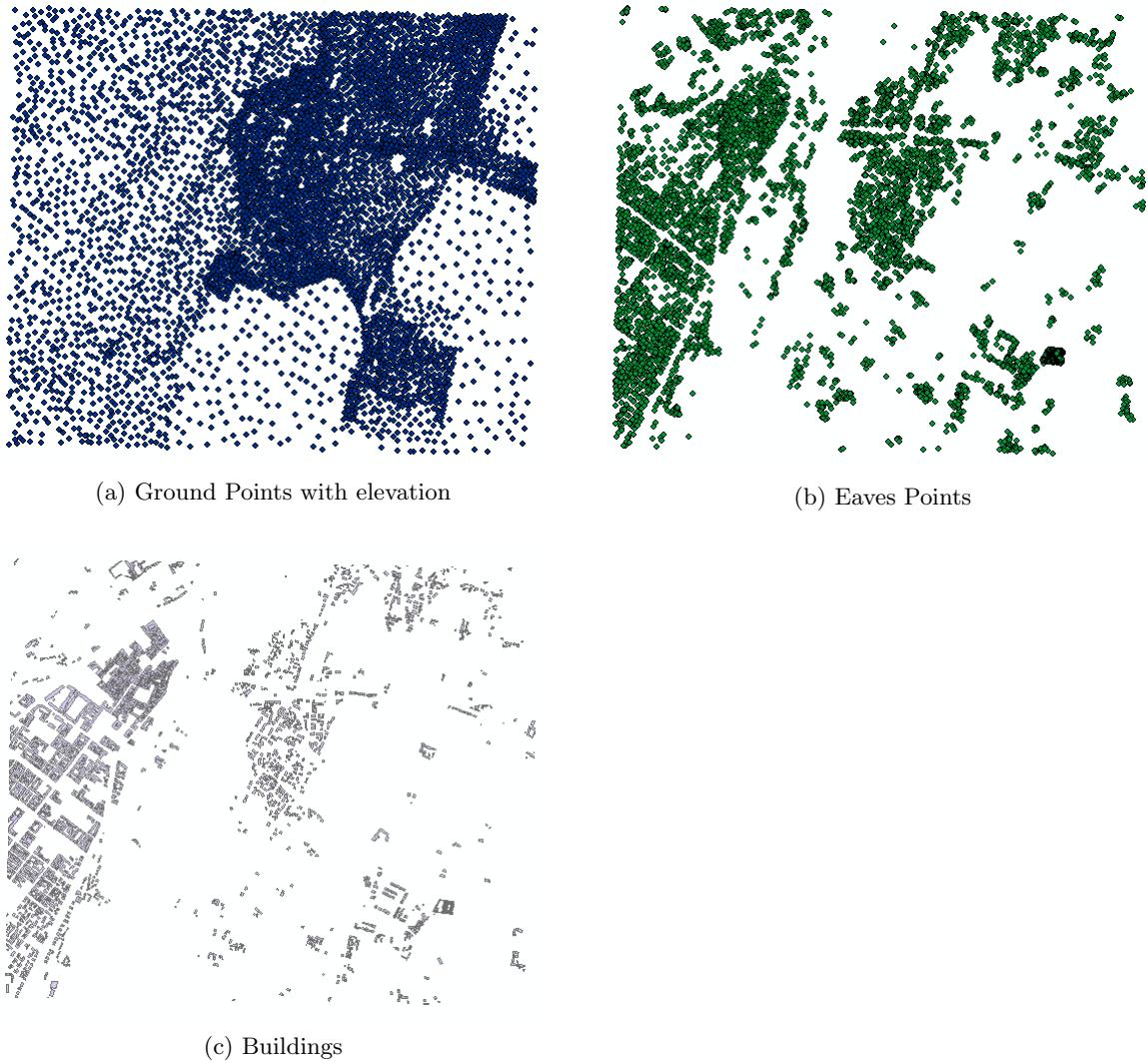


Figure 17: Extracted points from the database

Then, by using the extracted "ground points", a DTM model has been obtained by making a *Raster Interpolation* selecting the tools in 3D Analyst of ArcToolbox. The model has been obtained with two different algorithms: the *Natural Neighbor* and *IDW*¹³. In both the cases, the value to consider to make the interpolation was the elevation. Moreover, by selecting *Minus* in the tool *Raster Math* it has been possible to get a model in which the difference between the two obtained models was reported. They are shown in Figure 18. In some zones of the model there are points marked in white, meaning that

¹³Inverse Distance Weighted

the difference is high, around 27 meters, and points marked in black, meaning that the difference is low, around -15 meters.

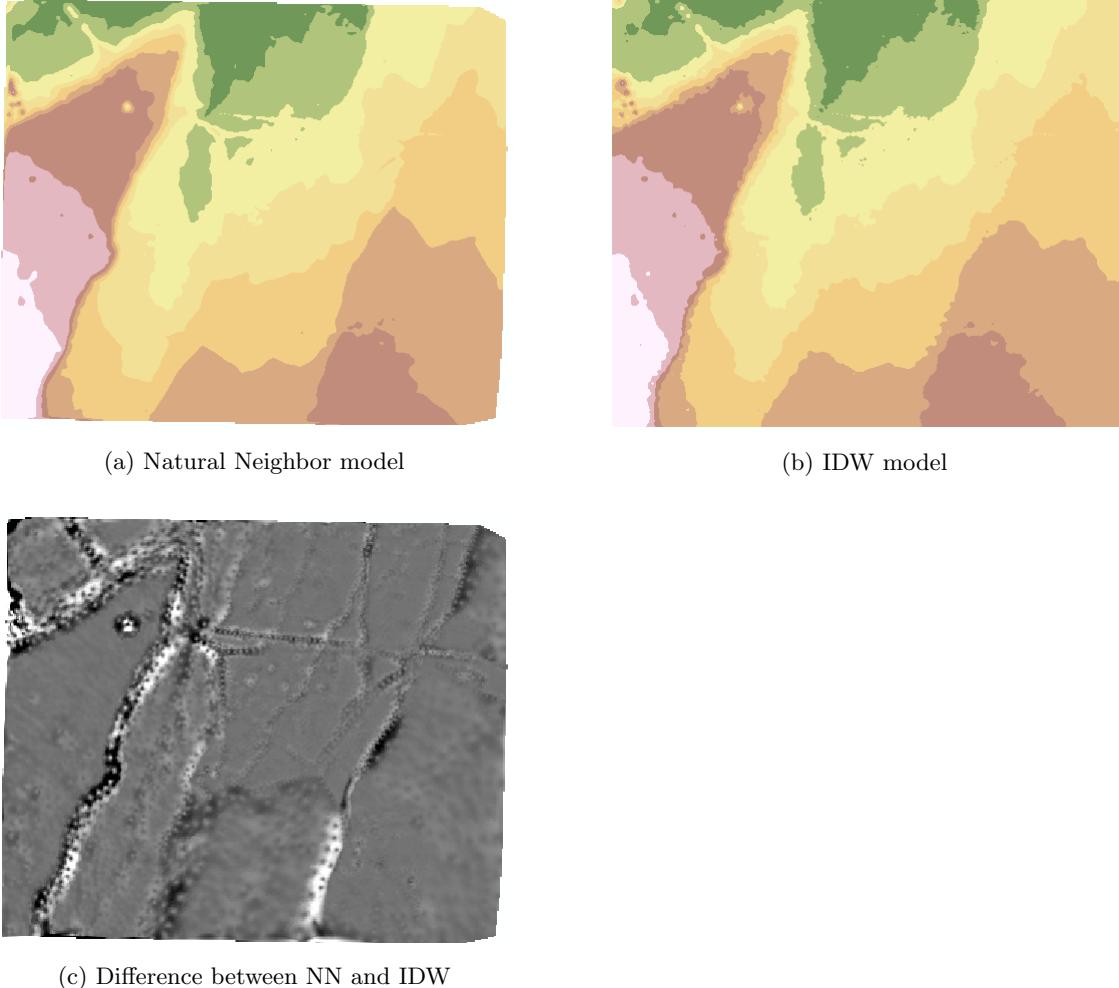


Figure 18: Obtained models

The next step was to merge the information about the eaves points with the buildings, by applying a *spatial correlation* using not the attributes but the *position*. In this way, a new database was created, composed by the information contained in the "building database" and the information (the attributes) contained in the "eaves database". With this new database, a new *field* about the height of the buildings has been created, doing the difference between eaves height (elevation) and ground height (elevation). Moreover, two new field were created called "area" and "perimeter" by using the tool *calculator tool*. In Figure 19 is reported a screenshot of the 3 new fields Height Buildings, Area and Perimeter.

Distance	Hei Buildi	Area	Perimeter
0	3.81	85,434	37,514
0	3.24	76,637	36,03
0	3.75	63,359	31,884
0	5.15	215,79	70,937
0	7.71	227,089	62,601
0	9.38	196,267	56,196
0	4.86	541,602	111,592
0	4.17	320,9	81,327
0	4.05	372,099	91,458
0	2.71	44,587	27,381
0	7.71	255,468	67,6
0	2.49	29,059	21,703
0	8.36	176,324	53,242
0	5.63	148,472	48,845
0	5.49	204,501	58,61
0	4.79	366,266	83,045
0	6.41	222,325	64,96
0	4.63	235,584	68,061
0	5.63	129,005	45,489
0	7.94	202,508	57,687
0	1.9	26,757	21,027
0	4.62	150,629	49,136
0	6.38	467,947	103,574
0	4.74	395,835	81,646
0	4	36,202	24,068
0	4	188,827	56,999
0	1.87	50,464	28,415

Figure 19: New fields Height Buildings, Area and Perimeter

Finally, by using a new tool called ArcSCENE and by exploiting the new field obtained above, a 3D model of the buildings has been created, choosing an *Extrusion* of the real height of buildings multiplied by 10, to make them more visible on the model. It's reported in Figure 20.

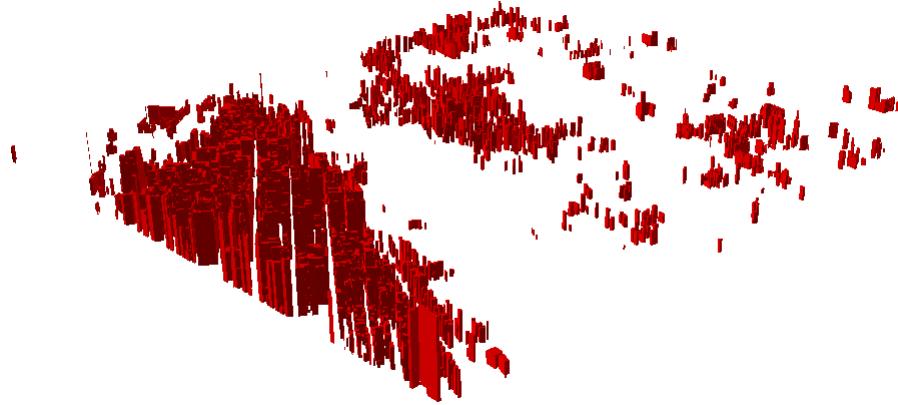


Figure 20: 3D model of buildings

Moreover, the model has been modified by classifying the buildings in 10 classes according to their heights and by plotting them with different colours, each of them indicating a range of values of heights. It can be seen in Figure 21

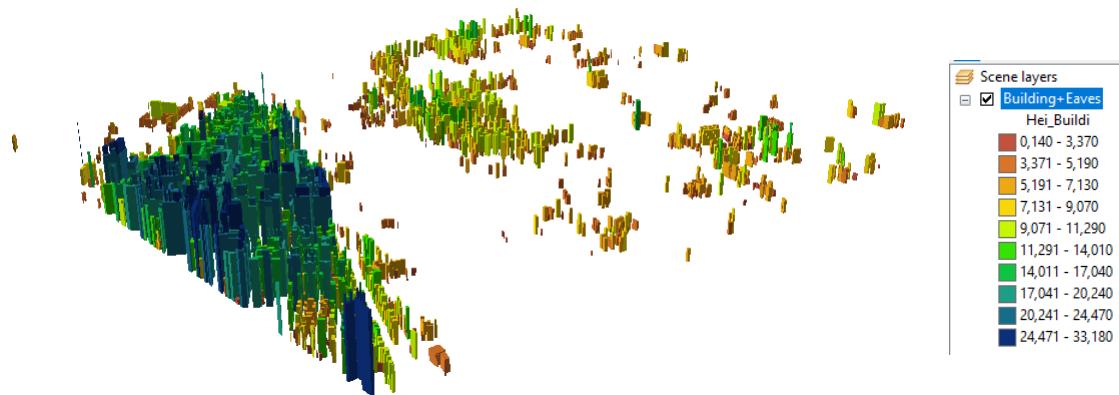


Figure 21: 3D model of buildings with different colours

8 GIS Part 2

8.1 Exercise 1

The goal of this exercise was to create new entities and new attributes by using the tool ArcMap and by starting from an *orthophoto*¹⁴.

With the tool *Editing* it has been possible to create on the image new entities like buildings and parking lines, and it has been possible to create new attributes like "Floor numbers", "Height" of the building and the name of the street where the parking lines were. In Figure 22 are reported the selected buildings and parking line on the image.

¹⁴Photo representing satellites imagery

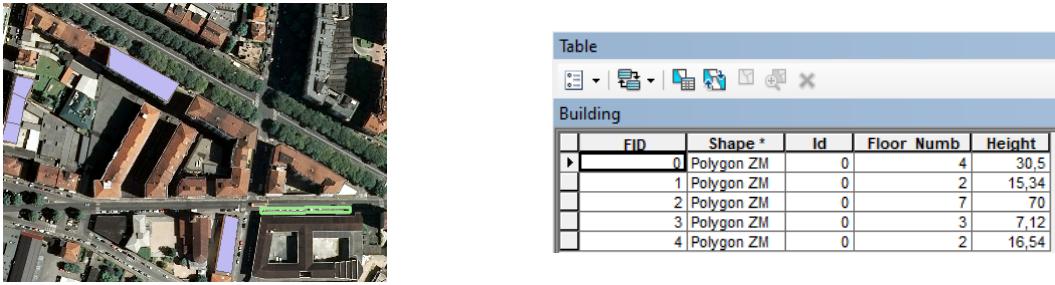


Figure 22: New entities and attributes of "Building" selected on the image

8.2 Exercise 2

In this exercise, by using the tool *Conversion Tools* of ArcMap and selecting *From GPS* and then *GPX to Features*, a file .GPX containing information about a point collected by the GPS was imported and transformed into a database, shown in Figure 23.

GPS_point									
	FID	Shape*	Id	Name	Descript	Type	Comment	Symbol	DateTimes
0	0	Point ZM	0 001	WPT		Flag, Blue	2019-03-18T14:01:56Z		265,545197
1	1	Point ZM	0 002	WPT		Flag, Blue	2019-03-18T14:12:15Z		248,120148
2	2	Point ZM	0 003	WPT		Flag, Blue	2019-03-18T14:16:11Z		246,679398
3	3	Point ZM	0 004	WPT		Flag, Blue	2019-03-18T14:16:44Z		247,19809
4	4	Point ZM	0 005	WPT		Flag, Blue	2019-03-18T14:17:07Z		247,554504
5	5	Point ZM	0 006	WPT		Flag, Blue	2019-03-18T14:17:12Z		247,735168
6	6	Point ZM	0 007	WPT		Flag, Blue	2019-03-18T14:17:16Z		247,738403
7	7	Point ZM	0 008	WPT		Flag, Blue	2019-03-18T14:18:49Z		247,55127
8	8	Point ZM	0 009	WPT		Flag, Blue	2019-03-18T14:18:57Z		247,542099
9	9	Point ZM	0 010	WPT		Flag, Blue	2019-03-18T14:18:58Z		247,541138
10	10	Point ZM	0 011	WPT		Flag, Blue	2019-03-18T14:19:22Z		247,332886
11	11	Point ZM	0 012	WPT		Flag, Blue	2019-03-18T14:21:01Z		247,154831
12	12	Point ZM	0 013	WPT		Flag, Blue	2019-03-18T14:21:04Z		247,158295
13	13	Point ZM	0 014	WPT		Flag, Blue	2019-03-18T14:21:08Z		247,165817
14	14	Point ZM	0 015	WPT		Flag, Blue	2019-03-18T14:21:09Z		247,167725
15	15	Point ZM	0 016	WPT		Flag, Blue	2019-03-18T14:21:11Z		247,169632
16	16	Point ZM	0 017	WPT		Flag, Blue	2019-03-18T14:24:32Z		247,577789
17	17	Point ZM	0 018	WPT		Flag, Blue	2019-03-18T14:26:28Z		247,216354
18	18	Point ZM	0 019	WPT		Flag, Blue	2019-03-18T14:26:35Z		247,073029
19	19	Point ZM	0 020	WPT		Flag, Blue	2019-03-18T14:26:37Z		247,076431

Figure 23: List of attributes of imported point

Moreover, it has been created a new feature starting from an .xml file containing data. The result on the image is reported in Figure 24.



Figure 24: New feature

8.3 Exercise 4

The goal of the fourth exercise was to generate a *contour line* with 10 meters of resolution and to create the *slope map* and the *aspect map*, by using an imported DTM.

In order to create the Slope Map, the tool *Raster Surface* of 3D Analyst on ArcMap has been used and the *Slope* option has been selected. The Output measurement was selected on DEGREE. The obtained map is reported in Figure 25 and it describes a new raster data where the colors describe the value of the slope expressed in degrees.

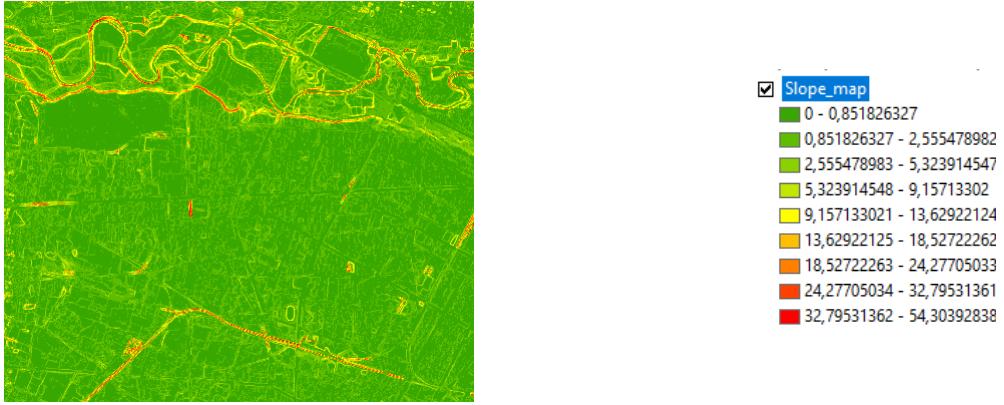


Figure 25: Slope Map and its legend

In the same way it has been possible to create the Contour Line, i.e. lines which describe the same point with the same elevation. It's shown in Figure 26. The interesting part of creating a Contour Line is that it allows to pass from Raster Data to Vector data. Indeed, it can be also seen the Attribute Tables containing all the information about the elevations that has been created and it can be used to select only the wanted elevation one wants to plot.

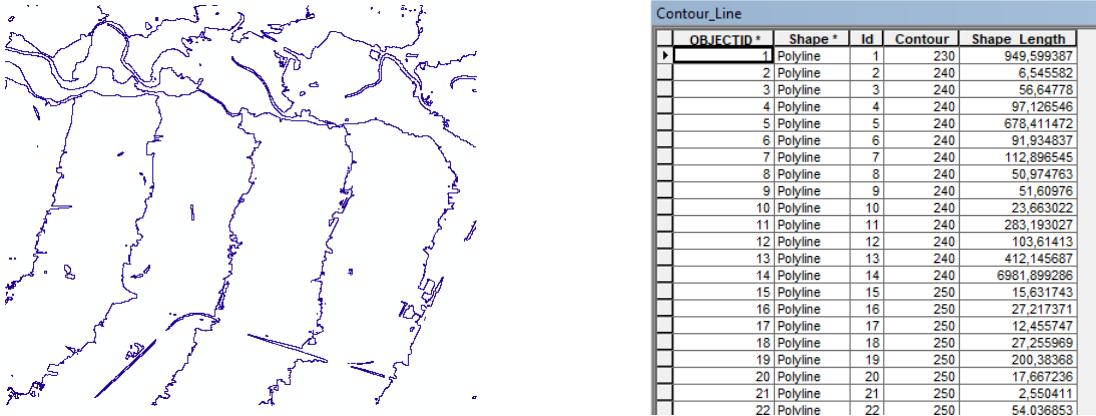


Figure 26: Contour Lines and its database

8.4 Exercise 5

In this exercise, the goal was to try using the layout function useful to obtain a printable map, exporting it in PDF or JPEG file. For example, in Figure 27 is reported an example of a rough Layout, where it has been added the *Scale Map* and *North Arrow*.

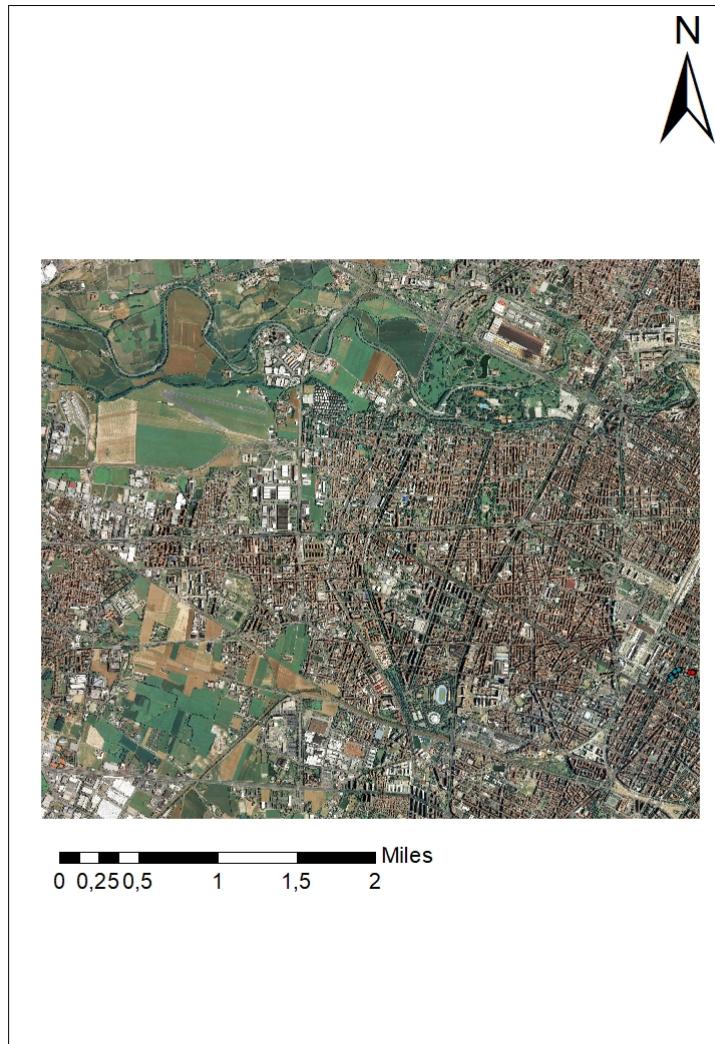


Figure 27: Example of Layout

8.5 Exercise 6

The goal of this exercise was to associate an elevation to a Raster Data: for each pixel of an orthophoto, an elevation obtained with a DSM was assigned. The obtained result is reported in Figure 28.

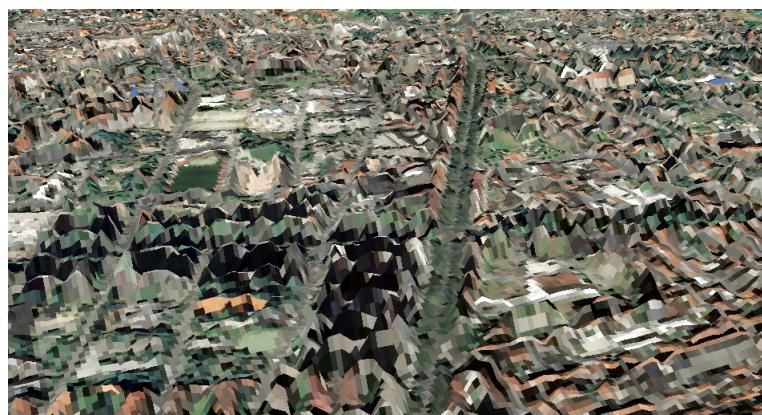


Figure 28: Orthophoto with elevation