Appl Geomat (2011) 3:35–48
DOI 10.1007/s12518-011-0045-1

ORIGINAL PAPER

SKYPLOT_DEM: a tool for GNSS planning and simulations

Stefano Gandolfi · Luca La Via

Received: 15 February 2010 / Accepted: 30 January 2011 / Published online: 19 February 2011

Abstract Satellite survey techniques have become commonplace for several applications such as surveying, environmental positioning, and navigation. The possibility of using this technique is strongly dependent on the possibility of receiving data from at least four satellites, as the accuracy is dependent on both the number of visible satellites and on their relative positions in the sky. From a theoretical standpoint, the Global Positioning System (GPS) constellation of satellites was designed to guarantee the continuous visibility of a minimum of four satellites worldwide, but this was based on the major approximation of considering the earth as an ellipsoid. The presence of natural (such as mountains) and anthropic (such as buildings) obstacles reduce sky visibility, in some cases significantly, thereby reducing the position accuracy or the possibility of determining a position (in the case of less than four satellites). The evolution of satellite positioning technologies will result in the possibility that, in the near future, many constellations can be used to determine the position of a receiver. Additionally, the Glonass constellation will be improved, and the new Galileo constellation will be deployed. Starting from digital elevation models (DEM), orbits (broadcast or precise), and the Keplerian parameters of new constellations, an open-source software has been realized with the aim of performing more accurate planning and simulations of the skyplot (the map of the visible satellites) in a point or region for a defined window of time. The software, developed in the C++ language, permits the simulation of the current constellation and the simulation of future scenarios by considering the interoperability between NAVSTAR-GPS, GLONASS, Galileo, COMPASS, and other systems. The simulation, conducted using DEM, permits the definition of a more correct obstacle map for each point. Of course, the DEM is more detailed and accurate, resulting in a more precise obstacle map and thus a more precise simulation (DEMs derived from Lidar data can even be used to account for the blockage caused by buildings). This paper begins with an introduction concerning the approach used to determine the skyplot and the obstacle map derived from a DEM, continues by explaining the logical architecture of the software and then presents simulation results in different conditions (with or without the DEM, using different constellations, and considering either one point or a regional analysis).

Keywords GNSS · DEM · Simulation · Planning

Abbreviations

DEM	Digital elevation model
DOP	Dilution of precision
GDOP	Global dilution of precision
GMT	Generic mapping tools
GNSS	Global navigation satellite systems
GPS	Global positioning system
HDOP	Horizontal dilution of precision
MJD	Modified Julian date
PDOP	Precise dilution of precision
RINEX	Receiver independent exchange format
UTM	Universal transverse mercatore
VDOP	Vertical dilution of precision
YDOYHMS	Year, day-of-year, hour, minute, seconds
YMDHMS	Year, month, day, hour, minute, seconds
YUMA	Format Almanac constellation (.alm)

S. Gandolfi (✉) · L. La Via
Engineering Faculty, DICAM-ARCES, University of Bologna,
Bologna, Italy
e-mail: stefano.gandolfi@unibo.it

Introduction

The final position accuracy of Global Navigation Satellite System (GNSS) systems is influenced by several parameters, including the number of the visible satellites and the relative location of the GNSS receiver. Many commercial software packages for planning and/or data processing permit simulation and planning using tools that show the number and location of visible satellites along with some correlated parameters over time from one position on the earth or in space.

The most common mode for evaluating the goodness of the available satellite constellation for positioning are the dilution of precision parameters (Hofmann-Wellenhof et al. 2001), which are representative of the relative geometric quality of the satellite constellation.

Other software packages are also available that permit the planning or simulation of any new constellations, such as the software *Visual* from Delft University (Verhagen 2002), which represents a complete tool for planning and simulation of the GNSS constellation but considering the earth as an ellipsoid and without morphology.

This study considers the possibility of integrating the classic skyplot (considering the earth's surface as a pure ellipsoid) with other information such as a Digital Elevation Model (DEM) and other constellations like Glonass and Galileo (or more if ephemerides or Keplerian orbital parameters are available).

The software has been implemented in C++ and is now available for scientific purposes by sending a request to the authors.

Principal objectives of this software

The development of this code permits the realization of a map of all the visible satellites through the definition of a "mask map" as derived by a DEM and the simulation of new constellations using the orbital parameters or ephemerides (Fig. 1).

Obstacle map realization using a DEM as a reference

Usually, a skyplot is generated automatically by considering the earth's surface as a pure ellipsoid. A cut-off angle can be defined to eliminate all the satellites that are geometrically visible but have too low of an elevation for the considered simulation. In other cases, it is also possible to create a mask (by a numerical editing or graphical tool) that simulates the presence of some obstacles. It is clear that ignoring the presence of natural or anthropic obstacles yields an unrealistic simulation, and, in some cases, such as

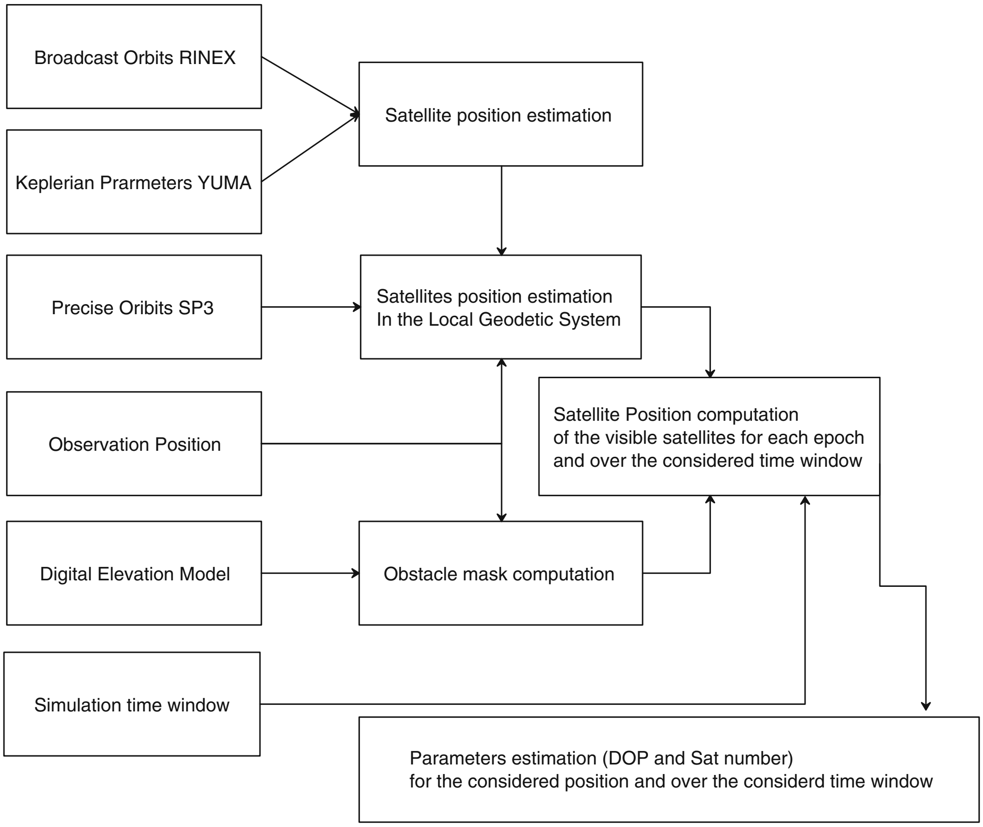
Broadcast Orbits RINEX
Keplerian Prarmeters YUMA
Satellite position estimation
Precise Oribits SP3
Satellites position estimation
In the Local Geodetic System
Observation Position
Satellite Position computation
of the visible satellites for each epoch
and over the considered time window
Digital Elevation Model
Obstacle mask computation
Simulation time window
Parameters estimation (DOP and Sat number)
for the considered position and over the considerd time window

Fig. 1 Flowchart of the software

in high-density urban areas, this can have a significant effect.

The evolution of the survey techniques permits the use of many instruments that are able to realize three-dimensional models of the territory morphology and towns using remote-sensing or the Lidar technique. These models can be used to evaluate the obstacle mask at each point within the area covered by the model.

Simulation of new constellations using theoretical orbital parameters

This study also considers the possibility of simulating new constellations (such as Galileo or the complete Glonass segment) or the constellations that are already present in the sky (such as the NAVSTAR-GPS or Glonass constellations). This work assumes that all present and future constellations will be fully interoperable.

Geodetic aspects of the problem

To realize a skyplot able to show the visible satellites, it is necessary to locate the satellite positions epoch by epoch, with respect to the local geodetic system placed at the point of interest. If we use precise orbits, this requires a transformation of the system from a geocentric to a local geodetic system. Otherwise, using orbital parameters would require a computation of the satellite position in the geocentric reference system (epoch by epoch), and then the aforementioned system transformation would be used to report the satellite position in the local geodetic reference frame.

The starting point for the obstacle map was a DEM, which was realized by regular grids on a cartographic projection (e.g., UTM). The software is able to compute the position of the observation point in all the required reference systems (geodetic and cartographic), starting from a position expressed in one of the other systems. Thus, the obstacle map is computed in a cartographic system (UTM) and the satellite parameters are reported in a local geodetic system.

It is well known that there are some differences in convergence angle (γ) between the cartographic anomaly and the bearing. In the UTM system, this difference can be significant when the point is located very far from the central meridian of the region and/or when the latitude is very high or low (close to polar regions). Considering a half-amplitude of three degrees and a point located at the middle latitudes, the error due to this approximation is about two degrees (3.49E-02 rad). This difference is considered negligible for the resulting simulations, especially when considering the approximation of the Digital Elevation Models with respect to the true context.

Creation of an obstacle map using DEM

The use of a digital elevation model to create the obstacle map is one of the main tasks of this software. In this first release, the software uses DEM in a standard ASCII grid format, and it only considers the UTM projection. The ASCII grid files (.grd) contain five header lines that provide information about the size and limits of the grid, followed by a list of *Z* values. The fields within the ASCII grid files must be space-delimited. The software has been designed to allow for the possibility of extending the number of formats and cartographic projections.

The algorithm used to construct the obstacle map is based on the algorithm described by Zatelli and D'incà (2004). The algorithm divides the horizons into *n* equal parts (a priori defined by the user), and, for each one, it computes the maximum elevation angle from the observation point and the grid nodes. In this case, the *n* parts are useful to parallelize the process.

To optimize the search area, which can decrease the software performance in the case of very large or dense DEMs, we have also considered the default *cut-off angle* a priori defined by the user. In other words, if we do not want to consider all of the satellites that are below a cut-off angle (because under a certain cut-off angle the observation quality decreases and the atmospheric effect can be significant), this cut-off angle can also be used to reduce the grid search area. Considering a DEM characterized by *n* rows and *m* columns, and considering, for example, that the nodes are equally spaced by *x* meters in each direction, and supposing that the maximum height ($h_{\max}$) of the DEM is known, that the height of the observation point (h_p) can be derived or obtained from the DEM, if the observation point is on the surface or defined a priori by the user, and that a cut-off angle (α) is a priori imposed by the user, it is possible to compute the maximum search distance ($d_{\max}$) from the observing point where the search can be realized.

$$d_{\max} = \frac{\left(h_{\max} - h_p\right)}{\tan\alpha} \tag{1}$$

This formula makes it possible to define the maximum search distance from the observing point and to speed up the computation.

Figure 2 provides an example indicating that to define the elevation mask in a direction, it is not necessary to perform a computation on all the grids; rather, computation until the maximum radius is sufficient. As a numerical example, consider three mountains (a, b, and c in Fig. 2) d_a=4, d_b=8, and d_c=15 km from an observation point

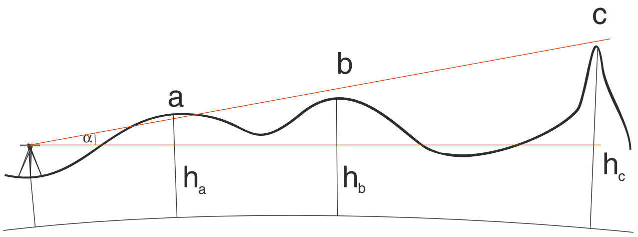

Fig. 2 Example of the maximum distance computation that was performed for the search of obstacles

along a general direction with heights of h_a=1,200, h_b=1,400, and h_c=2,200 m, respectively, where the height of the observation point is 200 m. Defining an a priori a cut-off angle of 10° formula (1) results in a d_{max} of 11.3 km.

Therefore, the search area is restricted to approximately 11 km and not to the complete DEM. In this example, the algorithm does not consider mountain C because it is too far from the observation point, and the obstacle induced by this mountain is lower (in terms of elevation angle) with respect to the a priori imposed cut-off.

Figure 3 shows an example where the research area is restricted to the area inside of a circle with radius d_{max}.

Finally, the position of the observation point will not necessarily be coincident with a node of the DEM. In these cases, we assume that the observation point is coincident with the nearest node of the DEM.

Computation of dilution of precision parameters

The mathematics used to define the satellite positions starting from orbital parameters (ephemeris) is well known and well explained in the literature (Hofmann-Wellenhof et al. 2001). The second part of the problem is to start from the satellite position in a topocentric reference system and compute the Dilution of Precision parameters (DOP), which indicate the satellite geometry quality. First, the satellite position must be transformed from a topocentric reference system to a local geodetic reference system with the origin as the observation point.

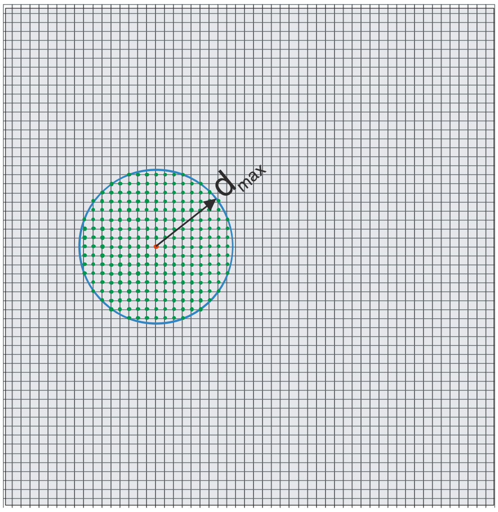

Fig. 3 Example of the DEM nodes that are investigated to define the obstacle mask

As is well known, the DOP parameters are position-dependent, and the formulas that permit this transformation are generally composed of a six-parameter transformation (three translation parameters and three rotation parameters).

We assume that $\underline{\rho}_{i0} = (X_{io}, Y_{io}, Z_{io})$ is the observation position and $\underline{\rho}_t = (X(t), Y(t), Z(t))$ is the position of a satellite with respect to a topocentric reference system. To obtain the coordinates of the satellite in the local geodetic system, we have to apply the transformation (Eq. 2) using R as expressed in (Eq. 3).

$$\begin{bmatrix} X^{neu}(t) \\ Y^{neu}(t) \\ Z^{neu}(t) \end{bmatrix} = R \cdot \begin{bmatrix} X(t) - X_{io} \\ Y(t) - Y_{io} \\ Z(t) - Z_{io} \end{bmatrix} \tag{2}$$

$$R = \begin{bmatrix} -\sin\varphi_{i0}\cos\lambda_{i0} & -\sin\varphi_{i0}\sin\lambda_{i0} & \cos\varphi_{i0} \\ -\sin\lambda_{i0} & \cos\lambda_{i0} & 0 \\ \cos\varphi_{i0}\cos\lambda_{i0} & \cos\varphi_{i0}\sin\lambda_{i0} & \sin\varphi_{i0} \end{bmatrix}, \tag{3}$$

where φ_{i0} and λ_{i0} are the latitude and longitude of the observer, respectively.

In this case, considering that the required information is the direction (versor) of the satellite with respect to the observation point in the local geodetic reference frame and disposing of the coordinates of the two objects (observation point and satellite) in the topocentric reference system, it is possible to directly calculate the versor in the same reference system using (Eq. 4).

$$\underline{\rho} = \frac{\underline{\rho}_{(t)} - \underline{\rho}_{i0}}{\left\| \underline{\rho}_{(t)} - \underline{\rho}_{i0} \right\|} \tag{4}$$

In this case, the direction from the observation point and the satellites as referenced to the local geodetic system can be obtained by applying a three-dimensional rotation.

After obtaining the coordinates of all the visible satellites with a positive elevation angle, we can write a matrix A composed of k rows and four columns (Eq. 5). The k parameters are the number of visible satellites for the considered epoch.

$$A = \begin{bmatrix} -\frac{X^1(t)-X_{i0}}{\rho^1_{i0}(t)} & -\frac{Y^1(t)-Y_{i0}}{\rho^1_{i0}(t)} & -\frac{Z^1(t)-Z_{i0}}{\rho^1_{i0}(t)} & -c \\ -\frac{X^2(t)-X_{i0}}{\rho^2_{i0}(t)} & -\frac{Y^2(t)-Y_{i0}}{\rho^2_{i0}(t)} & -\frac{Z^2(t)-Z_{i0}}{\rho^2_{i0}(t)} & -c \\ -\frac{X^3(t)-X_{i0}}{\rho^3_{i0}(t)} & -\frac{Y^3(t)-Y_{i0}}{\rho^3_{i0}(t)} & -\frac{Z^3(t)-Z_{i0}}{\rho^3_{i0}(t)} & -c \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ -\frac{X^k(t)-X_{i0}}{\rho^k_{i0}(t)} & -\frac{Y^k(t)-Y_{i0}}{\rho^k_{i0}(t)} & -\frac{Z^k(t)-Z_{i0}}{\rho^k_{i0}(t)} & -c \end{bmatrix}, \tag{5}$$

Table 1 Orbital parameters considered for the Galileo constellation (part A)

ID	Right ascension at week(deg)	Mean anomaly (deg)	ID	Right ascension at week (deg)	Mean anomaly (deg)
51	0	0	66	4	253
52	0	40	67	4	293
53	0	80	68	4	333
54	0	120	69	18	27
55	0	160	70	18	67
56	0	200	71	18	107
57	0	240	72	18	147
58	0	280	73	18	187
59	0	320	74	18	227
60	4	13	75	18	267
61	4	53	76	18	307
62	4	93	77	18	347
63	4	133	78	0	20
64	4	173	79	4	20
65	4	213	80	18	20

where $\rho_{i0}^{k}(t)$ represents the geometric distance between the observation point and the *k*-satellite at the epoch *t*. The cofactor matrix Q_{XX} can be written as in (Eq. 6).

$$Q_{XX} = \left(A \cdot A^{T}\right)^{-1} = \begin{pmatrix} q_{XX} & q_{XY} & q_{XZ} & q_{t} \\ q_{YX} & q_{YY} & q_{YZ} & q_{t} \\ q_{ZX} & q_{ZY} & q_{ZZ} & q_{t} \\ q_{tX} & q_{tY} & q_{tZ} & q_{tt} \end{pmatrix} \tag{6}$$

This matrix contains all terms needed to determine the DOP parameters (Hofmann-Wellenhof et al. 2001).

In particular, if we are interested in the determination of some **global** parameters such as GDOP, PDOP, or TDOP, they can be calculated as in (Eq. 7) below.

$$\begin{aligned} GDOP &= \sqrt{q_{XX} + q_{YY} + q_{ZZ} + q_{tt}} \\ PDOP &= \sqrt{q_{XX} + q_{YY} + q_{ZZ}} \\ TDOP &= \sqrt{q_{tt}} \end{aligned} \tag{7}$$

Otherwise, if we are interested in other DOP parameters such as HDOP or VDOP (horizontal and vertical components with respect to the local geodetic system), we need to transform the cofactor matrix from the topocentric system to the local geodetic system using (Eq. 8). The HDOP and VDOP can then be determined as in (Eq. 9).

$$Q_{xx} = RQ_{XX}R^{T} = \begin{pmatrix} q_{xx} & q_{xy} & q_{xz} \\ q_{yx} & q_{yy} & q_{yz} \\ q_{zx} & q_{zy} & q_{zz} \end{pmatrix} \tag{8}$$

$$\begin{aligned} HDOP &= \sqrt{q_{xx} + q_{yy}} \\ VDOP &= \sqrt{q_{zz}} \end{aligned} \tag{9}$$

Using these formulas, it is possible to obtain the DOP parameters and the positions of all the satellites from the observer's point of view.

Operational aspects and use of code

As shown in Fig. 1, some input parameters are required to make the code work, which are summarized as follows:

1. Position of observation site or area of study
2. Orbital parameters
3. Time window to be considered
4. Digital Elevation Model
5. Cut-off angle

For each chosen configuration, the user specifies the types of results that the code produces by choosing an option between those that are available, based on the aspect

Table 2 Orbital parameters considered for the Galileo constellation (part B)

Health	0
Eccentricity	0
Major semiaxis (km)	29600
Orbital inclination (deg)	56
Rate of right ascension (r/s)	0
Argument of Perigee (deg)	0
Af0(s)	0
Af1(s/s)	0
Week	328
Time of applicability (s)	405504

of the problem he/she is interested in. It is possible to choose between two available modes:

1. Single-point
2. area of study

which will be examined in the following paragraphs.

Single-point mode

The single-point mode is used to determine how effective the site is over the time window selected and how the introduction of new constellations could improve the positioning at the chosen site.

The code for this mode analyzes how the distribution of satellites evolves over the time window considered. In this mode, the following results are considered both numerically and graphically (a graphical interface has not yet been developed; the graphical display was instead conducted using a standard platform such as GMT (Generic Mapping Tools) or Gnuplot, both of which work in both Linux and Windows environments). In particular, three products will be generated:

1. Skyplot (Graphic of satellite paths in the sky studied from the observation point)
2. Graphic of the number of visible satellites versus time
3. Graphic of the GDOP versus time

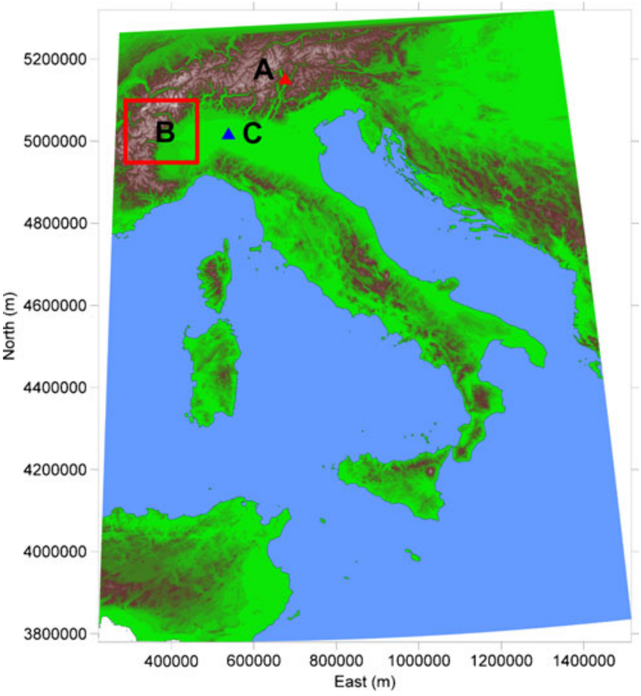

Fig. 4 Map of the points and areas considered for the simulation. *A* is related to the example reported in section "Example 1: single-point mode" for a single-point simulation, while *B* and *C* are the areas considered in sections "Example 2: simulation in the presence of high mountains (north-western Italy)" and "Example 4: simulation in a city using a DEM realized by an airborne Lidar survey"

Fig. 5 Location used for the simulation test (by Google Earth)

If other DOP parameters are necessary, a simple upgrade of the software can be realized.

Area of study mode

In the area of study mode, the code is supposed to research all of the DEM points included within a rectangular range bounded by two opposite vertices of the rectangle ($N_{\min}$, $E_{\min}$) and ($N_{\max}$, $E_{\max}$). This option is recommended to assess whether there are particular areas on average in which the System GNSS operation is critical and whether the contribution of additional constellations significantly improves the results.

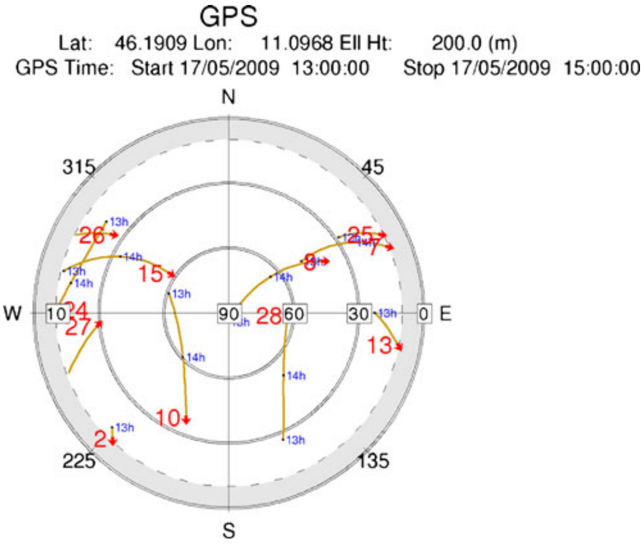

Fig. 6 Skyplot obtained by simulation A

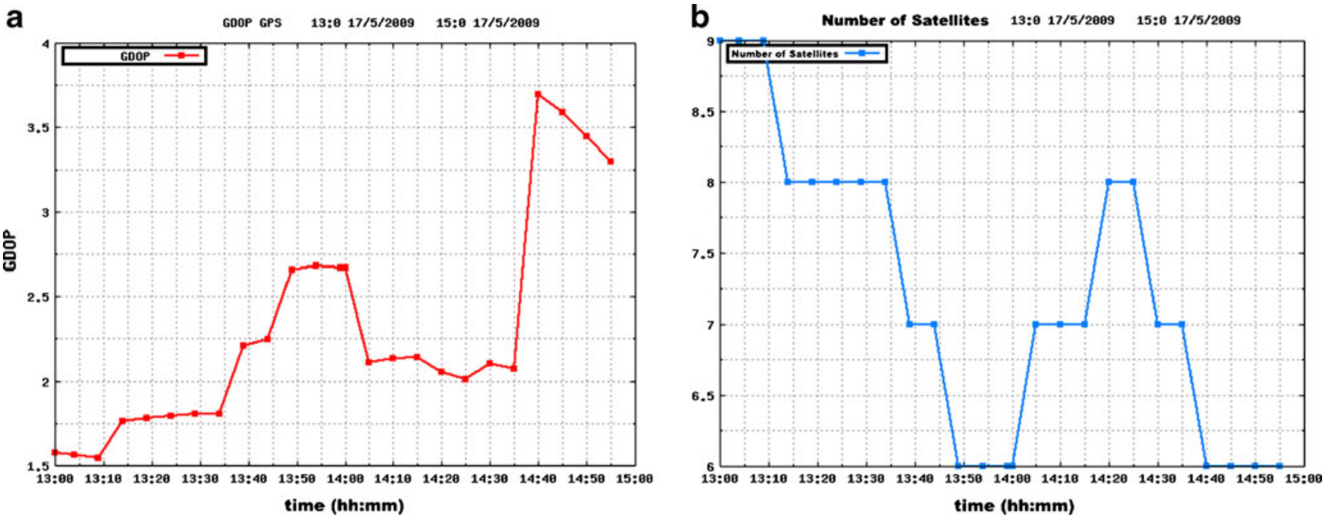

Fig. 7 Skyplot GDOP (*left*) and number of GPS satellites (*right*) as a function of time as obtained by simulation A

In this mode, the interest is directed mainly to the average parameters one can get at various points of the DEM within the time window considered. Thus, by choosing this option, it is possible to generate two maps (in grid format) representing:

1. the average number of visible satellites
2. the average GDOP

Implementation details

The SkyplotDem program has been written in the C++ language (Alexandrescu 1996; Josuttis 1999; David et al. 2001; Martin 2003; Gamma et al. 2005) to gain all the advantages of object-oriented programming. This program was developed on the Ubuntu 8.04 distribution of the Linux operating system.

This program can be best explained using a behavioral representation called a "sequence diagram" in Unified Modeling Language (Lee and Tepfenhart 2001).

Skyplot represents the heart of the program, as this is the class that instantiates all other classes and call the appropriate methods on these objects instantiated by performing calculations capable of achieving the objectives set by the program Skyplot_Dem.

One possible operation mode of the program is to run a satellite skyplot without considering any obstacle, considering the visibility of satellites in a fixed point as an argument and an elevation angle below which no satellite track will be considered. In the case of simulations in a scenario with obstacles, the program uses a file that contains the mapping of the obstacles.

The first thing that is performed by the algorithm is to calculate the position in different reference systems useful for further calculations. Specifically, methods *Llh2Xyz*, *Xyz2Llh*, *Ll2Utm*, and *Utm2Ll* called on Point object are used for converting the location coordinates obtained at the end of the observation point in different systems of geographic, geocentric, and cartographic UMT coordinates. Skyplot also creates an object *Broadcast* and exploits the ephemeris file (to be broadcast or almanac) for reading all relevant information to determine the location of the satellites. This object also uses the *DateTime* object to synchronize the period of revolution of the satellites with the earth day. In fact, this object deals with formatting dates according to various formats such as GPS weeks, seconds measurement of GPS weeks, Julian Day, and other formats (Marshall 2002). In this class, transformations by a given date to the other are handled depending on the type and

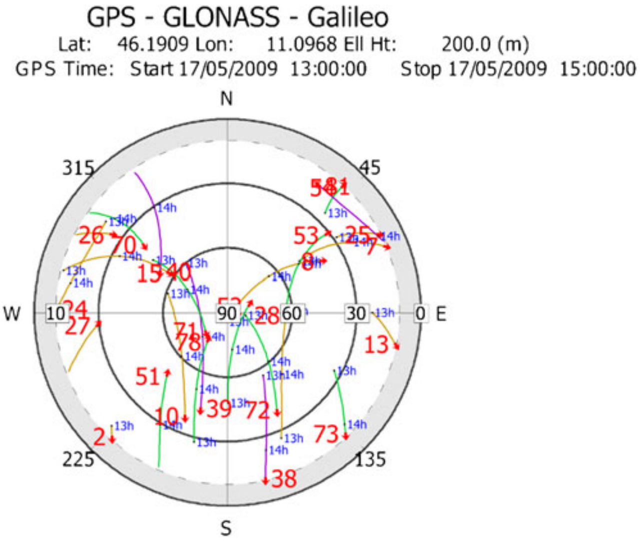

Fig. 8 Skyplot obtained by simulation B

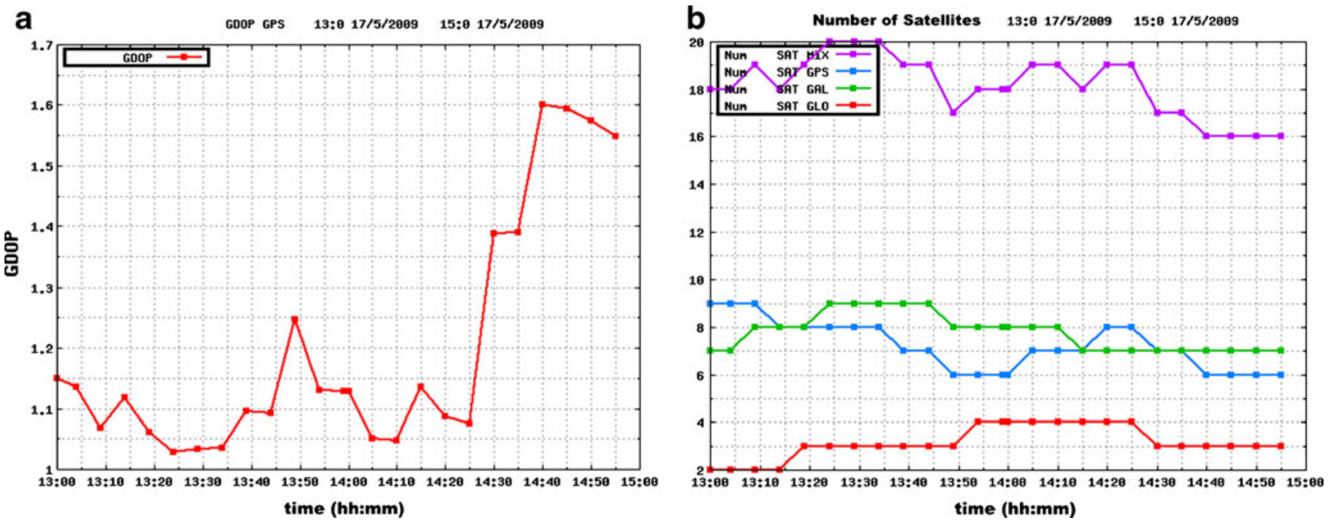

Fig. 9 Skyplot GDOP (*left*) and number of GPS, GLONASS, or Galileo satellites (*right*), varying with time as obtained by simulation B

date format added. Following, the different date formats are showed:

1. GPSWeek, GPS Week, and seconds of Week, seconds of week GPS.
2. Modified Julian date (MJD)
3. Year, day-of-year, hour, minute, seconds (YDOYHMS)
4. Year, Month, day, hour, minute, seconds (YMDHMS)

The *Broadcast* class allows reading of the file ephemeris broadcast which gives the position of the satellites in geocentric system. In fact, the function *BcRead* allows just the reading of the ephemeris file looking for each block within the file broadcast all information for the satellite.

All the information for a complete description of the orbit and that needed to calculate the position of satellites are contained in the following files. The formulas for these calculations have been implemented in the method *BcOrb* which, in turn, to make this calculation uses a private method of the same class that is called *BcCalc* (Marshall 2002).

Skyplot_Dem finally gets the position of satellites carrying the call of these methods (*BcRead, YuRead*, and *BcOrb*) on the class broadcast.

After obtaining the position of the observer in the two coordinate system (XYZ and LLH) and the information about the satellite from the broadcast file, the program generates a map of the obstacles from DEM files.

The *Map* object, with the call *MakeMap*, creates a map of the obstacles to that point. For this calculation, the *Map* object uses the object cell which divides the calculation of the map into four quadrants that have the vertex at the point of the observer and, with the method calls *Quadro90, Quadro180, Quadro270*, and *Quadro360*, create a partial map in four sections. The basis of computation is the calculation of the distance and the height point: after getting these two figures, it immediately calculates the angle α that identifies the height of obstacles:

$$\alpha = \mathrm{atan}(\mathrm{difference\_height}, \mathrm{distance}) \tag{10}$$

The algorithm works as if for each quadrant two rays were created that originate at the point indicated as observation point. These two lines identify the angle used to assess the presence of obstacles that may impede the visibility of the satellites. This procedure is performed many times as the number of steps used to cover the entire quadrant.

Considering an interval of 10°, then this action is repeated in each quadrant in nine directions (0°, 10°, 20°....). If in one direction there are more obstacles, the angle considered is the bigger one of obstruction in that

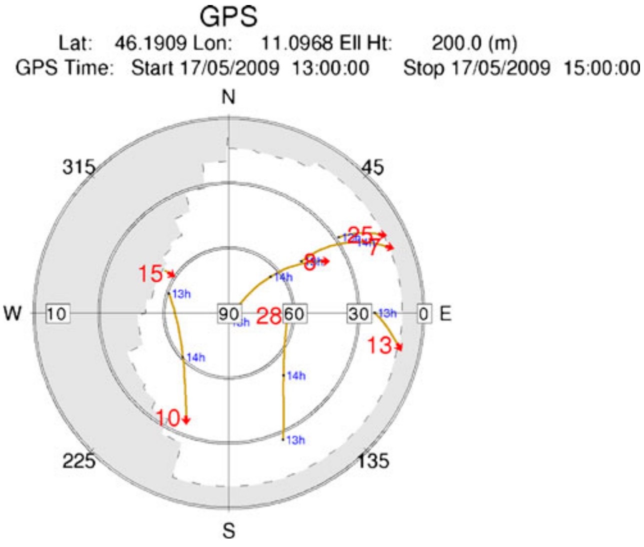

Fig. 10 Skyplot obtained by simulation C

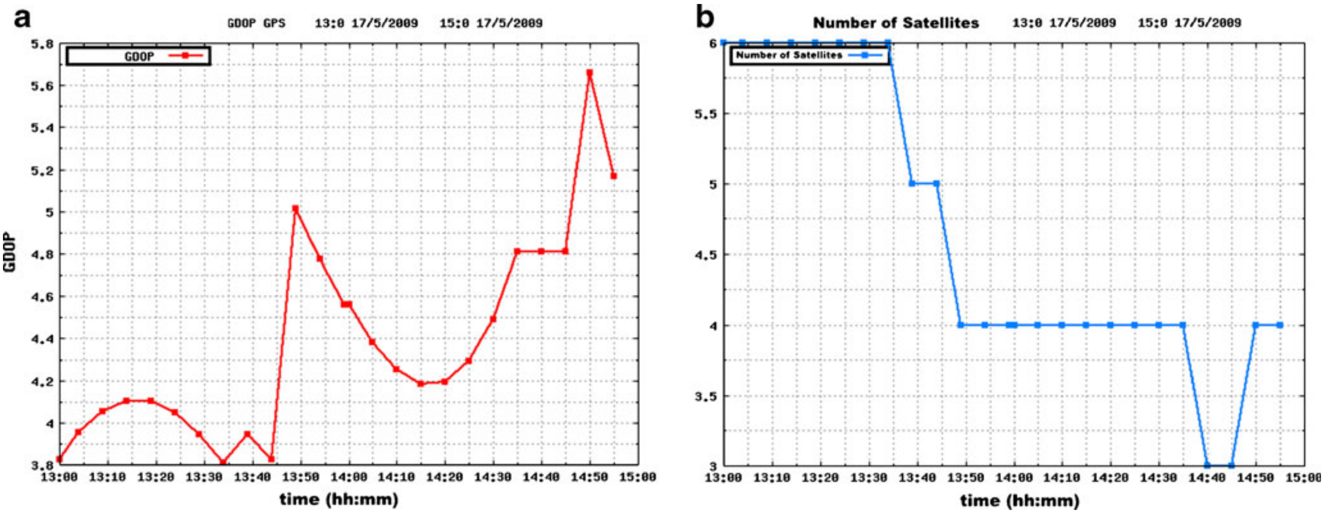

Fig. 11 Skyplot GDOP (*left*) and number of GPS satellites (*right*) over time as obtained by simulation C

direction. Not only that, when you identify the point within the DEM, you can evaluate the upper limit of the distance between the observer and a possible obstacle that must be taken into account in applying the formula.

Indeed, applying the formula (Eq. 10) and considering the greatest obstacle and a minimum cut-off of α, then the maximum distance to consider for each point must be less than the value $d_{\max}$. This reduces the computational burden of the algorithm without calculating the angle of obstruction for each point of the DEM.

The program Skyplot_Dem analyzes the two dates that define the time period within which to run the simulation to calculate the positions of the satellites in the required period. In fact, after reading, the values for *StartTime* and *EndTime* begin a loop to calculate the position of all satellites epoch by epoch. When running this cycle, the program starts from the initial date and then increase that time to 5 min (set as the default option) for each cycle and calculates, through the call *BcOrb*, the positions of the satellites visible at that time.

In the same cycle, after having obtained the coordinates of the geocentric positions of the satellites, a map of the obstacles, and having considered the corners of the obstacles, the software creates a local geodetic system to determine the location of the satellites relative to the plane of the observer's horizon.

Given the satellites in the local geodetic system, the program also considers the angle of obstruction that the signal may encounter: in fact, the file obtained from the object *Map* consists of two columns where the values of the first column contain the azimuth angle while the second column the elevation angle of the obstacles. This allows the determination the location of the visible satellites.

The last stage is therefore that of the graphic representation used to make visible all the parameters calculated by the program. For this application, GMT (Wessel and Smith 1995) has been used for easy graphical representation.

How the software works

The software works with a command line mode. By typing Skyplot_Dem, a header shows the different possibilities that the user can consider. In particular, there are three options:

1. "- -"the program runs without **any** DEM
2. **"-D** *filename***"** the program considers a DEM file in ASCII grid format for the definition of the obstacle map of the considered point

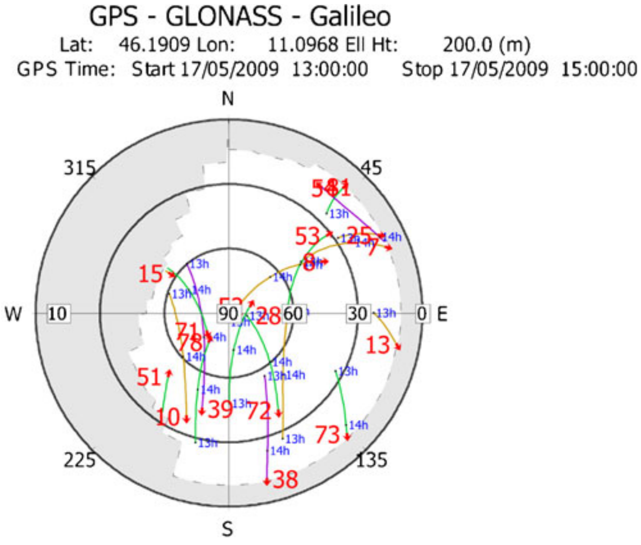

Fig. 12 Skyplot obtained by simulation D

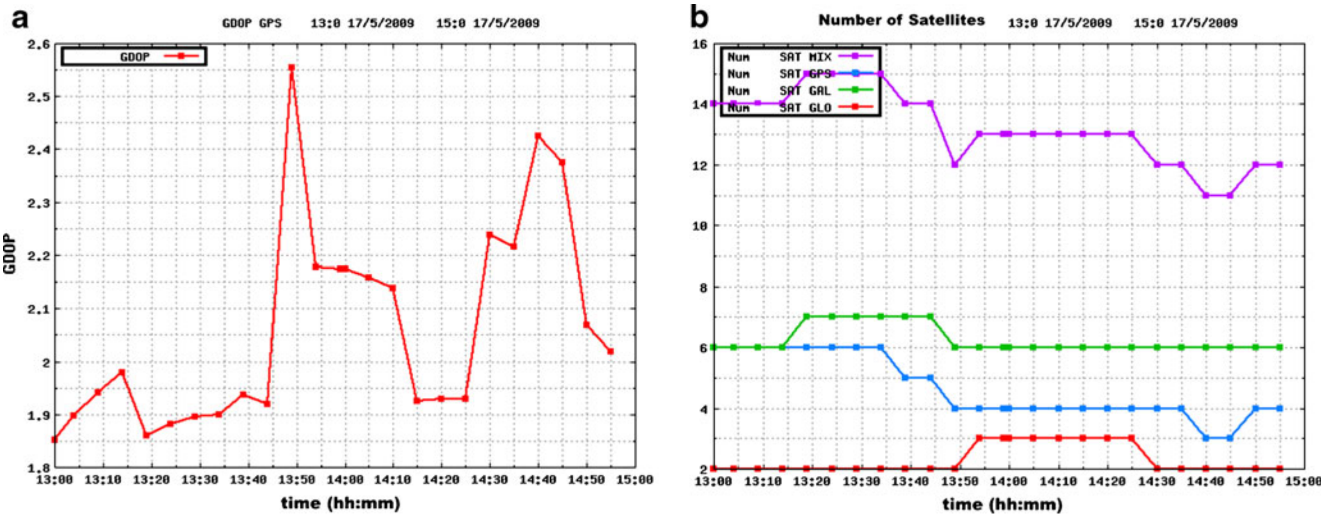

Fig. 13 Skyplot GDOP (*left*) and number of GPS, GLONASS, and Galileo satellites (*right*) over time as obtained by simulation D

3. **"-M** *filename***"** the program considers the DEM file in ASCII grid format for the definition of the obstacle map and completes the simulation for an area

All the other data necessary for the simulation are reported in an auxiliary file *Skyplot.inf* (the window time, the cut-off angle, etc. are reported in an external ASCII file *skyplot.inp*.

An example of this file (Skyplot.inp) is provided below.

```
START_DATA
2009 05 17 12 00 00
2009 05 17 15 00 00
END_DATA
START_ORB
verg0830.09n
YumaGAL.txt
END_ORB
START_COOR
LLh 44.05107108 11.24702217 0.00
END_COOR
START_CUT
10
END_CUT
```

In particular:

Between the START_DATA and END_DATA fields, the considered window time has to be inserted using the YYYY MM DD hh mm ss format.

Between the START_ORB and END_ORB fields, the list of the files containing orbits or ephemeris (broadcast, precise or YUMA) have to be reported.

Between the START_COOR and END_COOR fields, the position of the simulation or the xmin, ymin, xmax, and ymax of the simulation over a region in one of the reference systems considered (UTM, LLh, XYZ) has to be introduced by specifying at the beginning which type of coordinate systems are considered (e.g., LLh 44.05107108 11.24702217 0.00).

Between the START_CUT and END_CUT fields, the cut-off angle has to be reported.

Examples

This section will provide some examples and results obtained from the execution of the code described above. We show results obtained in both the "single-point" and "study area" modes, showing results in each case when considering only the GPS constellation, as well as when adding both the GLONASS and Galileo constellations. For the simulation of the Galileo orbits, the Keplerian parameters used in that project are available on the ESA Website and are shown in Tables 1 and 2.

The examples are divided into two categories: single-point simulation and areal parameter estimation. Figure 4 shows the region and point involved in the simulation.

Example 1: single-point mode

In the first example, we report the results of a simulation of a site near the location Mezzolombardo (Italy). The

Table 3 Summary of threshold parameters obtained in the four simulations in single-point mode

Simulation	*N* sat min	*N* sat max	GDOP min	GDOP max
a	6	11	1.25	3.75
b	13	27	0.9	1.6
c	3	9	3	100
d	12	19	1.3	3.2

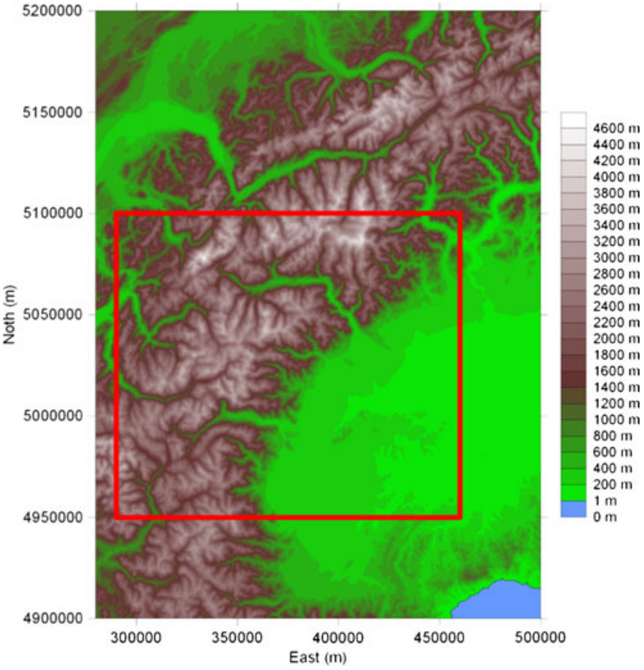

Fig. 14 Test area for example 3 (north-western part of Italy)

simulation was conducted over an interval of 24 h for the first day, 17 May 2009.

Figure 5 shows the location and morphology of the site. Details of the four simulations conducted at this point are described in detail:

1. A simulation using only the GPS constellation (from the ephemeris file: verg0830.09n) with a cut-off of 10° and without the use of a digital model of the ground (results reported in Figs. 6 and 7).
2. A simulation using the GPS constellation (from the ephemeris file: verg0830.09n) as well as the GLONASS and Galileo constellations from the Yuma file (and thus nominal constellations) with a cut-off of 10°, without the use of a digital model of the ground (results reported in Figs. 8 and 9).
3. A simulation using only the GPS constellation (from the ephemeris file: verg0830.09n) with a cut-off of 10° and with the use of digital model GTOPO30 (results reported in Figs. 10 and 11).
4. A simulation using the GPS constellation (from the ephemeris file: verg0830.09n) as well as the GLONASS and Galileo constellations from the Yuma file

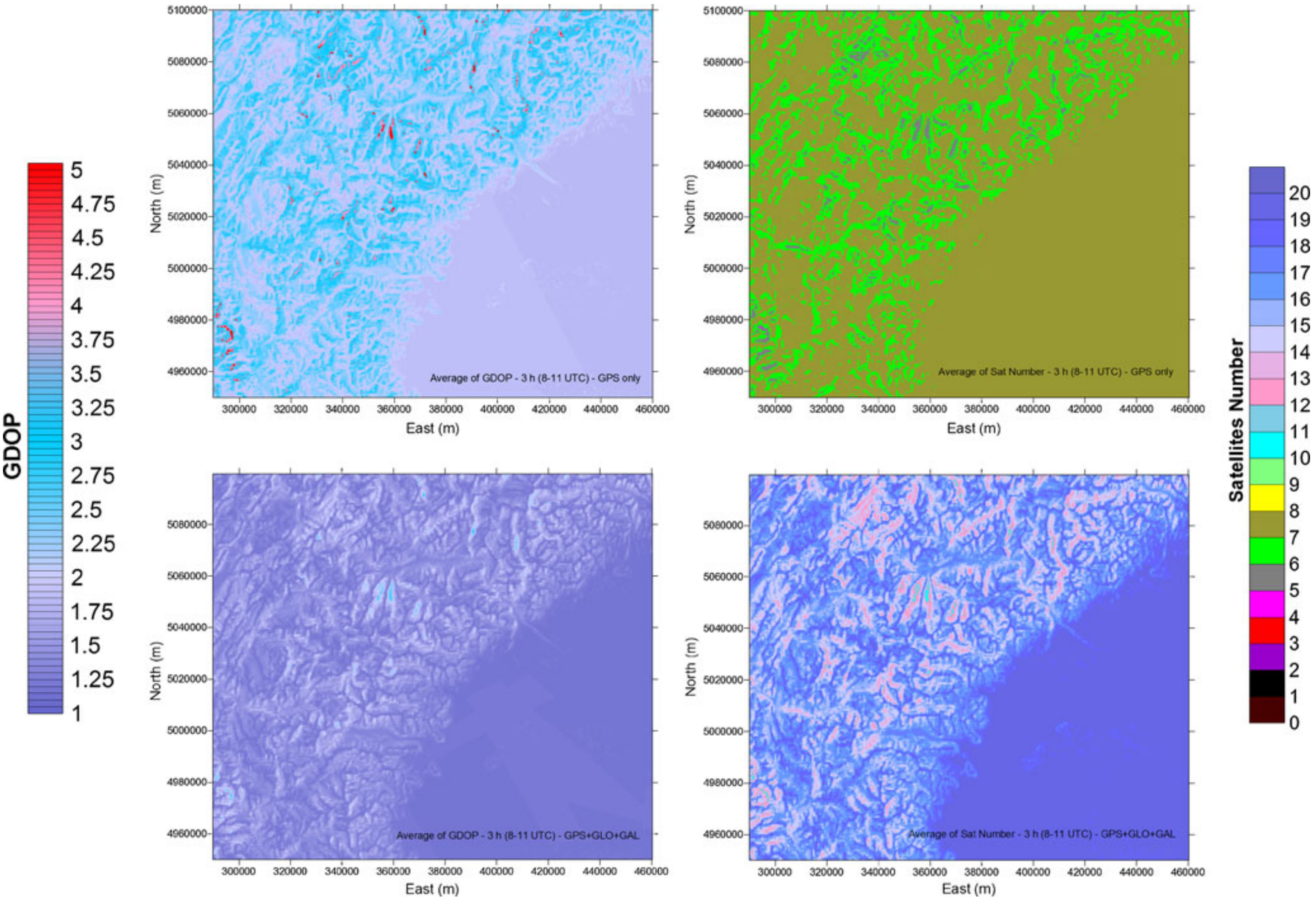

Fig. 15 Average GDOP (*left*) and the average number of satellites (*right*) obtained in the test area for the 3 h of observation (8–11 UTC, March 15, 2009), considering only the GPS constellation (*upper figures*) and the GPS, Glonass, and Galileo constellations (*bottom figures*)

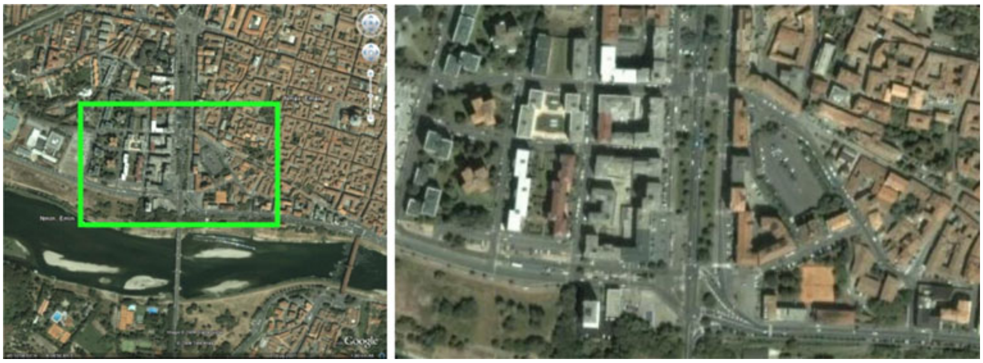

Fig. 16 Test area in Pavia city. The *left* picture represents a wide area in Pavia city, and the *right* picture represents the investigated area (pictures are taken by Google Earth)

(and thus nominal constellations) with a cut-off of 10° and with the use of digital model GTOPO30 (results reported in Figs. 12 and 13).

As shown by the four simulations performed here, the considered scenarios change drastically depending on which constellations are considered and also on which digital model of the area is used.

Table 3 shows the global statistical parameters obtained for the four simulations.

Using the Digital Elevation Model to define the obstacle mask results in significant changes in the predicted availability of the GPS constellation, and for some windows of time, only three or four satellites are available. In these conditions, navigation and surveying are quite impossible or the accuracy will be very low.

Example 2: simulation in the presence of high mountains (north-western Italy)

Another test area is the north-western part of Italy, characterized by the presence of flat areas and high mountains (such as the Alps), as in Fig. 14.

For this area, only a 3-h window was considered, and only the GPS and GPS + Glonass + Galileo conditions were simulated. These results are reported in Fig. 15.

In this case, the presence of high mountains can play an important role, and the derived results showed that both the number of satellites and the associated DOP parameters had an inverse relationship with topography. The simulation provides evidence that the sky visibility is strongly reduced in some valleys, and the average number of satellites is between five and six. Obviously, the GDOP map also shows that in some valleys, the DOP value rises to five, which represents the worst condition for a GPS survey.

Example 4: simulation in a city using a DEM realized by an airborne Lidar survey

The last example is a simulation in an urban area (Fig. 16), the city of Pavia (Italy). In this case, a DEM derived from an airborne Lidar survey was considered (Fig. 17).

This simulation considered only four scenarios. Detailed simulations were performed using only the GPS constellation and using the three constellations (GPS, Glonass, and Galileo), using a time window of only 1 or 3 h.

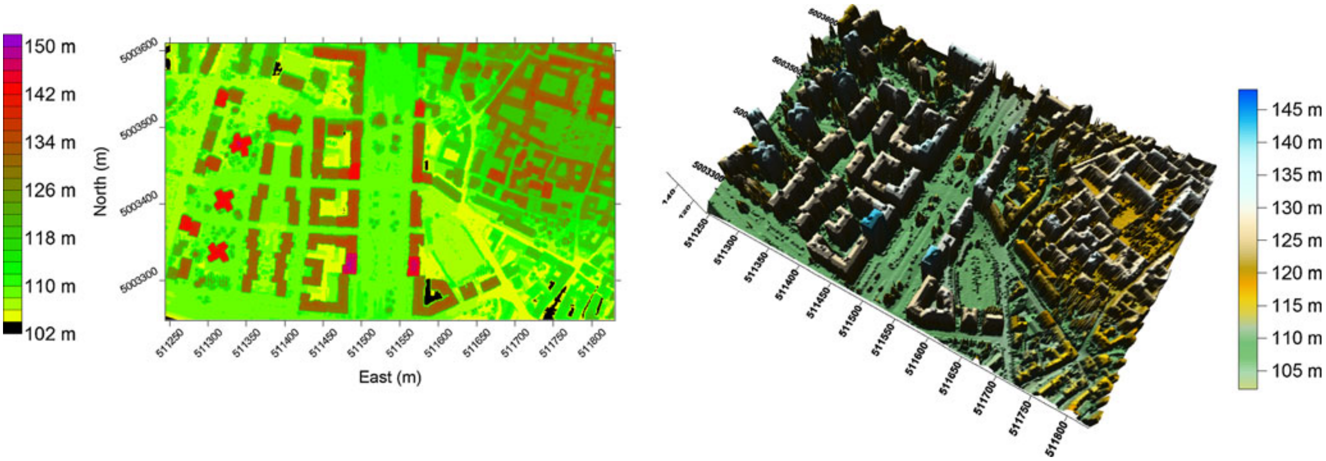

Fig. 17 Digital elevation model of a part of Pavia city derived by an airborne Lidar survey used for the simulation

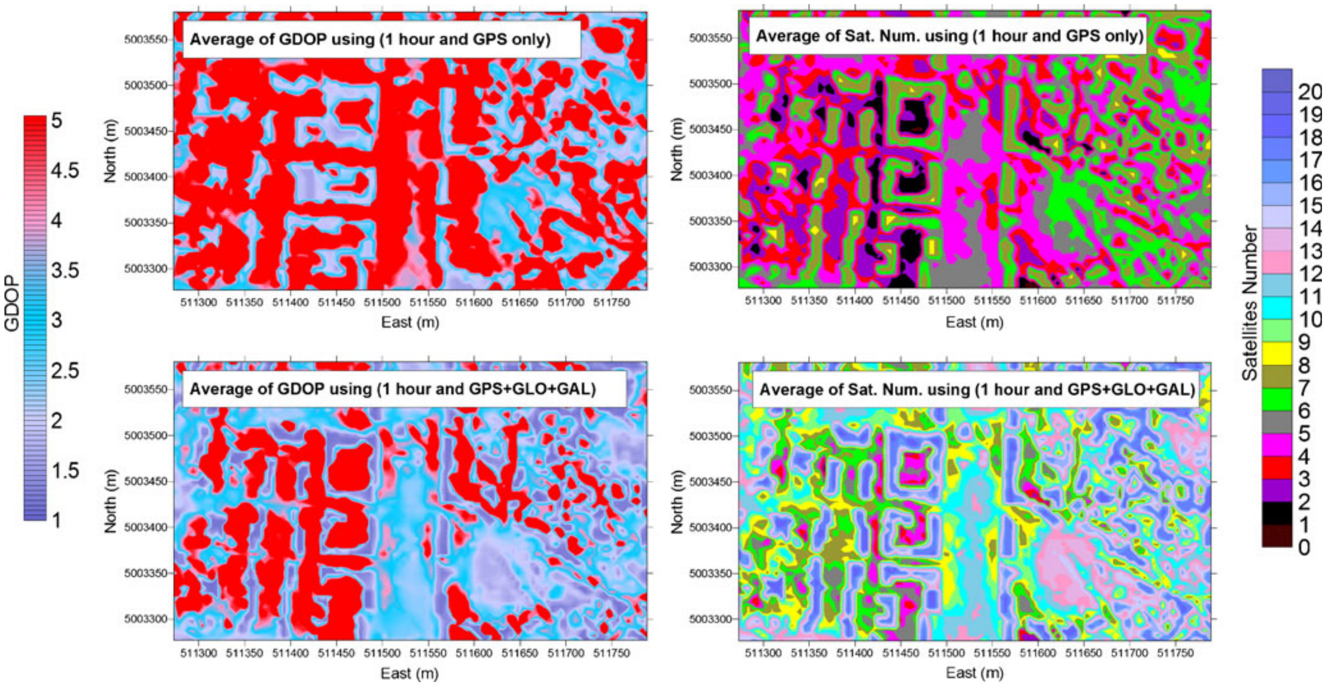

Fig. 18 Results of the simulation using 1 h of observations. The *graphs* on the *left* report the averages of the GDOP considering just the GPS constellation (*upper figures*) and also considering the three constellations (GPS, Glonass, Galileo) *bottom figures*. The *graphs* on the *right* report the average number of satellites

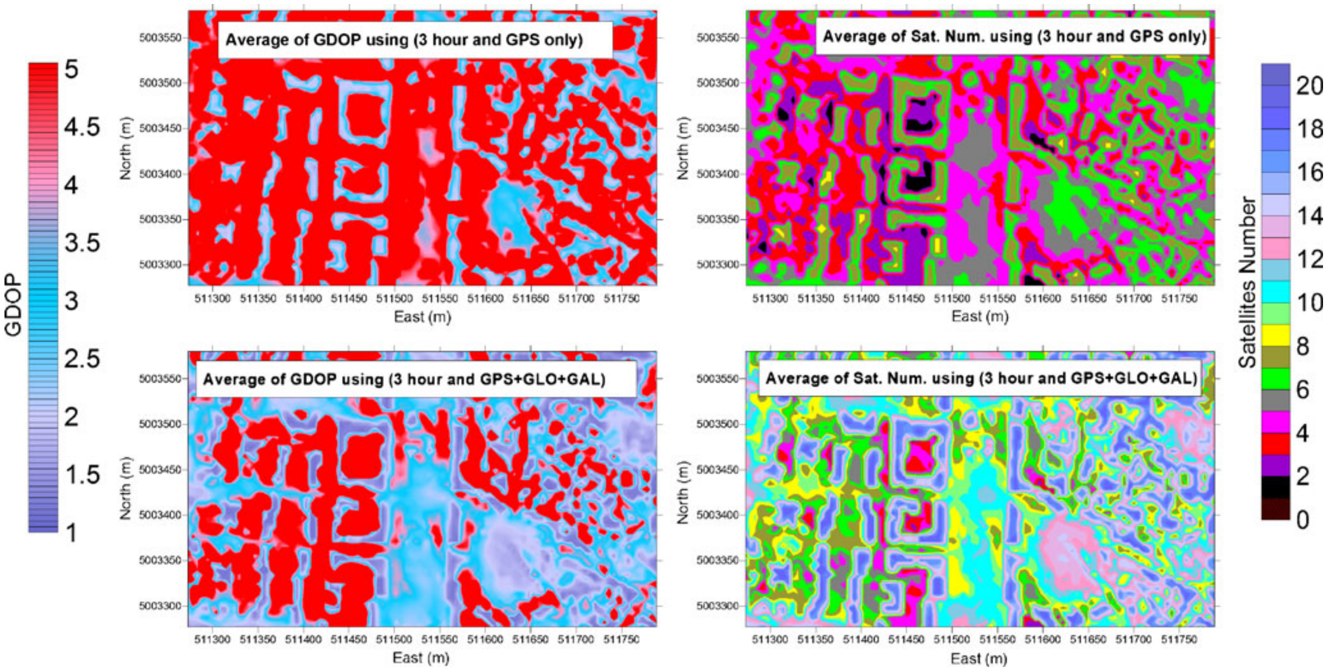

Fig. 19 Results of the simulation using 3 h of observations. The *graphs* on the *left* report the average of the GDOP considering the GPS constellation (*upper figures*) or the three constellations (GPS, Glonass, Galileo) *bottom figures*. The *graphs* on the *right* report the average number of satellites

Figures 18 and 19 present the results derived using 1 and 3 h of observation, respectively.

This last example shows that these tools can be useful for GNSS planning. In an urban area, and particularly in medieval historical city centers, the availability of GPS satellites is drastically reduced, and only certain areas can guarantee the acquisition of a congruent number of satellites. This example also highlights the need for an increase in the number of visible satellites. In particular, considering the three constellations, the area with a congruent number of satellites is drastically enlarged.

Conclusions

The software designed here is useful both for planning GNSS campaigns and for simulation of future scenarios, especially if interoperability is to be achieved between constellations. The possibility to use Digital Elevation Models for areas of interest constitutes a real added value permitting simulations with a more realistic obstacle map. Considering the results obtained from some of these simulations, the DEM can significantly change the theoretical accuracy especially in the urban context (where a DEM realized by airborne Lidar can give very detailed results) or in the case of the presence of very high mountains such as the Alps (one of the areas used for these simulations). Other tests conducted on the Apennine mountains give results not particularly different from the simulation without any superimposed obstacles. The possibility to simulate additional satellite constellations could be a very interesting tool, especially in urban areas. In the performed examples, the improvement both in terms of the number of visible satellites and the Dilution of Precision parameters is evident and changes dramatically in urban areas, where, in many cases, the GPS constellation alone is not enough to permit navigation and surveying.

Acknowledgments Research was carried out in the framework of the PRIN2006, a National Program financially supported by MIUR (Italian Ministry of the University and Research).

References

Alexandrescu A (1996) Modern C++ design: generic programming and design patterns applied. Addison-Wesley, Canada

Musser DR, Derge GJ (2001) STL tutorial and reference guide, Addison-Wesley Professional

Gamma R, Helm R, Johnson R, Vlissides J (2005) Design Patterns Elements of Reusable Object-oriented Software, Addison Wesley

Hofmann-Wellenhof A, Lichtenegger HE, Collins J (2001) GPS: theory and practice (Fifth Revised Edition). Springer-Verlag Wien, New York

Lee RC, Tepfenhart WM (2001) UML and C++: a practical guide to object-oriented development, 2nd edn. Prentice-Hall, New Jersey

Marshall J (2002) Creating and viewing skyplots. GPS Solut 6(1–2):118–120

Fowler M (2003) UML distilled: a brief guide to the standard object modeling language, Addison-Wesley Professional

Josuttis NM (1999) The C++ standard library: a tutorial and reference, Addison Wesley

Verhagen S (2002) Studying the performance of global navigation satellite systems: a new software tool. GPS World 13 (6):60–65

Wessel P, Smith WHF (1995) New version of the generic mapping tools released, *EOS*. Trans Am Geophys Uni 76:329

Zatelli P, D'incà D (2004) New modules for satellite surveying planning in GRASS. Geomatics workbooks 3:1–14

Web sites

http://rmmcweb.cr.usgs.gov/elevation/dpi_dem.html

http://www.esa.int/esaNA/SEMF5OZCU8E_galileo_2.html

http://www.geog.ubc.ca/courses/klink/gis.notes/ncgia/u38.html/OUT38.2, june 2007