

r.survey: a tool for calculating visibility of variable-size objects based on orientation

Txomin Bornaetxea & Ivan Marchesini

To cite this article: Txomin Bornaetxea & Ivan Marchesini (2021): r.survey: a tool for calculating visibility of variable-size objects based on orientation, International Journal of Geographical Information Science, DOI: [10.1080/13658816.2021.1942476](https://doi.org/10.1080/13658816.2021.1942476)

To link to this article: <https://doi.org/10.1080/13658816.2021.1942476>



© 2021 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group.



Published online: 28 Jun 2021.



Submit your article to this journal [↗](#)



Article views: 721



View related articles [↗](#)



View Crossmark data [↗](#)

r.survey: a tool for calculating visibility of variable-size objects based on orientation

Txomin Bornaetxea ^a and Ivan Marchesini ^b

^aDepartment of Geology, University of the Basque Country (UPV/EHU), Leioa, Spain; ^bResearch Institute for Geo-Hydrological Protection (IRPI), Italian National Research Council (CNR), Perugia, Italy

ABSTRACT

Identification of terrain surface features can be done using approaches such as visual observation or remote sensing image processing. Accurate detection of survey targets at the ground level primarily depends on human visual acuity or sensor resolution, and then on acquisition geometry (i.e. the relative position and orientation between the surveyor and the terrain). Further, the delimitation of the observer's viewshed boundary or of the sensor's ground footprint is sometimes insufficient to ensure that all enclosed targets can be correctly detected. Size and orientation can hamper ground target visibility. In this paper we describe a new release of r.survey, an open-source spatial analysis tool for terrain survey assessment. This tool offers the necessary information to assess how terrain morphology is perceived by observers and/or sensors by means of three basic visibility metrics: 3D distance, view angle, and solid angle. It is also fully customizable, allowing single or multiple observation points, ground or aerial point of view, and size setting of the observed target, making it useful for many different purposes.

ARTICLE HISTORY

Received 8 December 2020
Accepted 9 June 2021

KEYWORDS

Terrain survey; viewshed; visibility; geographic information system

1. Introduction

Terrain survey includes detection and measurement of surfaces, and can be conducted over large areas using simple visual observation or remote sensing via instruments such as laser scanners or optical, infrared and radar sensors. It is a key part of many scientific disciplines including geomorphology, biology, biogeography, archeology, landscape studies, and civil engineering (Santangelo *et al.* 2010, Edan *et al.* 2013, Minelli *et al.* 2014, Fábrega-Álvarez and Parcero-Oubiña 2019, Saeidi *et al.* 2019). In this regard, visual observation of terrain, either directly or through photo interpretation, has been one of the most widely used approaches. However, the quality of such observations is difficult to assess due to the implicit subjectivity of the observer. During recent decades, significant progress has been made to objectively quantify what is visible or not visible from certain observation locations. This process is known as viewshed analysis (Higuchi 1983), and many GIS tools have been developed for it (Domingo-Santos and Fernández de Villarán 2017, Sahraoui *et al.* 2018, Fábrega-Álvarez and Parcero-Oubiña 2019).

CONTACT Ivan Marchesini  ivan.marchesini@irpi.cnr.it

© 2021 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group.
This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Surface observation (on Earth or other planets) is not limited to direct eye contact. The use of remote sensors mounted in satellites, aircraft, or unmanned aerial vehicles (UAV), as well as ground-based sensors, is becoming increasingly crucial, providing data that go beyond the capacity of human vision. Hence, the concept of what is visible and not visible can vary depending on the observer's specifications (i.e. sensor characteristics).

Optical cameras, the human eye, and data collection sensors, such as radar, laser scanners, or infrared waves, require a direct line of sight between them and the target terrain to attain full operability. Thus, viewshed areas can provide valuable information for assessing surveyed territories (Chae *et al.* 2017). Visibility analysis has been widely developed and discussed from the human perspective (Higuchi 1983, Fisher 1992, Ogburn 2006, Llobera 2007, Nutsford *et al.* 2015), although other fields such as telecommunication, natural hazards, and remote sensing may also benefit from such spatial information (Proy *et al.* 1989, Dodd 2001, Edan *et al.* 2013, Tajelsir Raouf *et al.* 2020). For example, Bornaetxea *et al.* (2018) proved the relevance of the spatial delimitation of the 'effective surveyed area,' that is, the portion of the territory that should be used to train and validate landslide susceptibility models that are based on field survey landslide inventories. For this work, the authors designed a specific GRASS GIS python module that automatically delineates the 'effective surveyed area,' referred to as `r.survey`. Recently, Knevels *et al.* (2020) transferred the original GRASS GIS-based python tool into R. In this paper we go further and present the new release of the Python version of `r.survey`.

The `r.survey` open-source spatial analysis tool offers basic information to determine how an analyzed surface may be perceived, particularly if an object of a given size can be detected by one or multiple viewpoints. The tool has been optimized for earth surface analysis and, thus, is especially useful for evaluating the visibility of features on terrain slopes, such as landslides, road cuts, minor vegetation, quarry scars, geological outcrops, surface pipelines, parking areas, and burned areas. Visibility can be evaluated in terms of 3D distance, solid angle, and orientation of the terrain with respect to the line of sight (view angle). Therefore, depending on the purpose of the study, one can be interested in searching the closest position to observe a given territory. Conversely, for others, having a frontal view (perpendicular to the plane where the observed object shows its maximum extension) may be more relevant. In some cases, the aim may consist of a combined balance between the closest view and the frontal view to evaluate how much of the human field of view is occupied by a given object.

To summarize and also present an example, `r.survey` provides the user with the necessary information to answer questions such as the following:

- In the field, from how many locations is something visible?
- During fieldwork, which location would provide the best position for observation of a given point or area (according to the distance, angle, or both)?
- Which portion of the territory is visible along a given survey path/flight?
- Are objects of a given size visible from a given survey path/flight?

The remainder of this paper organized as follows. First, in [Section 2](#), we describe the software features that deal specifically with visibility analysis. In [Section 3](#), we present a detailed description of the design of the algorithm and the underlying rationale behind it. In [Section 4](#), we explain the benefits of the parallelized design of the code. In [Section 5](#), we

show the results of different tests carried out in a synthetic digital elevation model (DEM) created specifically for this purpose. We also describe the application of *r.survey* in a specific study area using a real digital elevation model. In [Section 6](#), we discuss the obtained results and the different implications related to the software, as well as the potential applications of *r.survey*. Finally, in [Section 7](#), we summarize the lessons learned and outline the principal conclusions.

2. Related tools

To date, a number of tools have become available for the automatic detection of visible areas either from single or multiple perspectives ([Table 1](#)). These include Viewshed and Viewshed2 in ArcGIS Pro version 2.7 (ESRI 2021), *r.los* or *r.viewshed* in GRASS (Neteler *et al.* 2012, GRASS Development Team 2020) or the Visibility Analysis Plugin for QGIS (Cuckovic 2016). Most approaches consider the viewshed as an ensemble of pixels (cells) in a digital elevation model that presents a direct line of sight with the observer, resulting in a simple binary map showing visible and non-visible pixels. However, as noted by some authors (Domingo-Santos *et al.* 2011, Nutsford *et al.* 2015), qualitative information regarding how something is visible offers much more useful data than simply classifying the terrain as visible and not visible. Indeed, in addition to elevation, aspect and slope play essential roles in how terrain is perceived. Unfortunately, the most popular viewshed algorithms available in most GIS packages do not consider pixel orientation or slope, forcing researchers to develop their own algorithms to obtain more accurate results or to reach the specific objectives related to their field of study. One such example is PixScape, an open-source software program used for viewshed analysis focused on landscape assessment (Sahraoui *et al.* 2018).

The most advanced tool in ArcGIS Pro is Viewshed2, which is able to perform very efficient parallel calculations of visibility from multiple locations to produce several primary outputs, including the number of points from which each cell can be viewed, and the identifier of the viewpoints (maximum 32) from which each cell is visible. The Visibility Analysis plugin for Qgis (Cuckovic 2016) is mainly devoted to producing outputs that are related to the outer edge of the viewshed and to the height that a

Table 1. Main features of visibility tools. GIS: running in a GIS environment; multiple: multiple viewpoints are processed; parallel: parallel processing of multiple viewpoints; number: output map portraying the number of viewpoints each cell is visible from; view angle: output map portraying the view angle (Θ) each cell is visible from; distance: output map portraying the 3D distance (Δ) each cell is visible from; solid angle: solid angle (Ω) of an object in each cell; and identifier: identifiers of the viewpoints a cell is visible from.

| | Viewshed2 | r.viewshed | Visibility Analysis | PixScape | r.survey (this work) |
|--------------------------|------------|------------|---------------------|----------|-------------------------|
| GIS | ArcGIS Pro | GRASS GIS | QGIS | - | GRASS GIS |
| Multiple | yes | no | yes | yes | yes |
| Parallel | yes | no | no | yes | yes |
| Number | yes | no | yes | yes | yes |
| View Angle (Θ) | no | vertical | no | yes | yes |
| Distance (Δ) | no | no | no | no | yes |
| Solid Angle (Ω) | no | no | no | no | yes |
| Identifier | yes | no | no | yes | yes |

cell should reach to become visible from a given viewpoint. It is also able to produce maps of the number of points from which each cell can be viewed. Among GIS tools, Pixscape is probably one of the most complete software programs related to visibility analysis available to date (Sahraoui *et al.* 2018). It offers a user-friendly interface where viewsheds can be calculated interactively for single viewpoint, and also allows many options to be set related to observer height, field of view (horizontal and vertical bounding), or target height. A multiple viewpoints option is also allowed, and there are three main outputs provided in raster or vector format: (i) the number of times each cell can be viewed, (ii) the angular height, and (iii) the angular area. The GRASS GIS tool `r.viewshed` can be used to calculate the viewshed and the vertical angle to the viewpoint.

All visibility indices produced by the instruments mentioned above do not depend on the size of the object being viewed. Therefore, it may be useful to have an instrument that allows evaluation of whether or not an object of a certain size is visible from one or more perspective. Similarly, it could be useful to have such an instrument capable of identifying the point of view from which a given object, located on a given territory, is most visible because of its position, size, and orientation with respect to multiple points of view. This could be useful for multiple applications, such as evaluation of the effective surveyed area when planning a field trip or assessing the visual impact on the landscape of a planned mining facility, considering its size and morphology. For these reasons, we developed a new version of `r.survey`.

3. Design of `r.survey`

A literature review reveals that automatic visibility analysis tools primarily follow two approaches: sight-line methods (Fisher 1996, Joly *et al.* 2009) and solid angle methods (Germino *et al.* 2001, Domingo-Santos *et al.* 2011). The latter takes advantage of trigonometric calculations to quantify the surface areas of the observer's retina occupied by the target pixel (Llobera 2003, Sahraoui *et al.* 2016a). Meanwhile, the former considers any visible cell in a discretized surface if there is no interruption along the straight line between the observer and the target pixel. The approach employed by `r.survey` combines both strategies. First, the visible pixels (cells) are selected according to the line-of-sight method using the pre-existing GRASS command, `r.viewshed` (Haverkort *et al.* 2009, Petrasova *et al.* 2018, Toma *et al.* n.d.). Then, three principal outputs (visibility metrics) are given, based on trigonometric calculations, to provide quantitative information about how the terrain is perceived from the selected observation points (or viewpoints). This information concerns the three-dimensional distance (Δ), relative orientation (view angle, Θ), and solid angle (Ω) of each target with respect to the viewpoint. A map containing a set of points describing the location of the observer and a digital elevation model (DEM) is the mandatory input for `r.survey`. Maps of buildings and trees that portray height information can be used to alter the DEM. In particular, for trees, the local elevation of the DEM can be altered by adding a random tree height constrained by an average value and accuracy estimation (standard deviation), according to a Gaussian model. Furthermore, the observer's view can be set as spherical or downward. The former

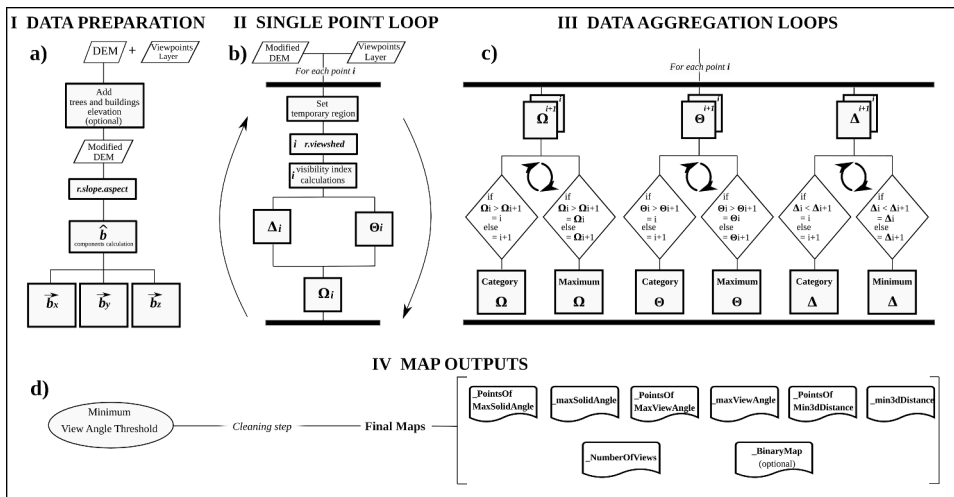


Figure 1. r.survey flowchart. Δ refers to 3D distance raster layers, Θ refers to view angle raster layers, and Ω refers to solid angle raster layers.

implies that the observer can look in any direction, while the latter is devoted to simulating the field of view of a flying object facing the ground.

The average size of the object being viewed is a fundamental input to the model. It must be given as the radius of the circle that has the same area as the object under consideration. Any value is allowed as this value does not depend on the working resolution. If no value is provided, the tool uses a value equal to half the resolution of the DEM. Many other options can be set, for example observer height with respect to the ground or respect to an elevation datum, maximum distance to perform the calculations, a view-angle threshold to exclude, a priori, cells oriented almost perpendicularly to the lines of sight. More details are available via the code availability section.

r.survey is designed as a GRASS GIS python module, which source code was written using the GRASS Scripting library and the pyGRASS library (Zambelli *et al.* 2013). It is organized into four main blocks (Figure 1), data preparation, the single point loop, data aggregation loops, and map outputs.

3.1. Data preparation

The input point map must be provided to the tool as a vector layer. If it represents the positions of a UAV, helicopter, or satellite, it must include an attribute field containing the absolute elevation values with respect to the same datum used by the input DEM. Caution must be taken not to use the points outside the valid (non-null) cells of the DEM. Observer height (with respect to the ground, default 1.75 m) and maximum distance of observation (default 1000 m) are parameters required by the r.viewshed module and by r.survey.

The target object size is provided by the object radius parameter. It is used to represent a circle of the same size as the target object. The circle is located at the center of the cells, and its orientation is defined by the aspect and slope values of the cells.

Because *r.survey* produces different types of outputs, a name to be used for the prefix of the output maps is requested (more details are available in the code availability section).

In the first block of instructions (Figure 1(a)), the DEM was modified according to the presence of possible tree and building maps. The elevations provided by the maps were added to the DEM values. Then, slope and aspect maps were derived for the calculations of the three components, \vec{b}_x , \vec{b}_y , and \vec{b}_z , of the unit vector \hat{b} normal to the terrain surface in each cell.

3.2. Single point loop

In the second block (Figure 1(b)), the coordinates of each viewpoint are used by the *r.viewshed* module to compute the viewshed area (Haverkort *et al.* 2009, Petrasova *et al.* 2018, Toma *et al.* n.d.), selecting only those pixels that have a direct line of sight from the viewpoint. This tool also returns the vertical angle value between the viewpoint and each visible DEM cell, and the curvature of the earth can be considered. In addition, at this step, the field of view can be restricted to the downward vision in cases such as unmanned aerial vehicles (UAVs). Then, the three main outputs are derived by using trigonometric equations: 3D distance, view angle, and solid angle.

3.3. 3D distance

The 3D distance is the three-dimensional linear distance between a viewpoint and a target pixel. Unlike the planar distance, the 3D distance (Δ) measures the length of the line of sight between two points (Figure 2), considering their relative elevation difference and their *x* and *y* coordinate differences (formula 1):

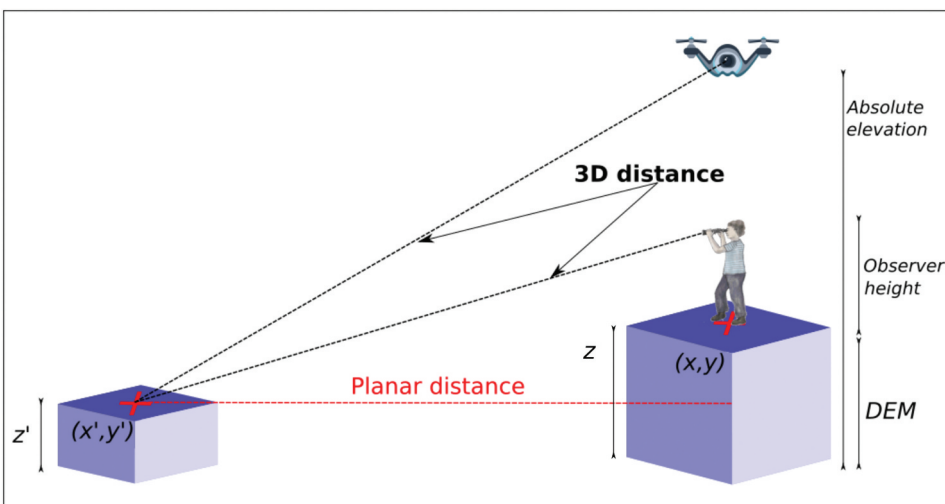


Figure 2. 3D distance between the observer and the object. The observer can be on the ground or in flight.

$$\Delta = \sqrt{|(y - y')|^2 + |(x - x')|^2 + |(z - z')|^2}, \quad (1)$$

where y , x , and z are the coordinates of the viewpoint and y' , x' , and z' correspond to the coordinates of the target pixel. Trigonometry allows for the measurement of an imaginary line connecting any cell and the 3D point defined by z , which is the elevation of the DEM from the viewpoint, plus the observer height (1.75, by default). However, if the absolute elevation option is set (in the case of UAVs, helicopters, aircraft, or satellite surveys), then the attribute table of the point map must contain an attribute column specifying viewpoint elevation. The calculation is then repeated for all available viewpoints. We acknowledge that 3D distance is a straightforward measure to obtain using GIS software. However, 3D distance is also needed for the calculation of the solid angle metric (Section 2.2.3) of the `min3dDistance` map (Section 2.3) and `pointOfViewWithMin3dDistance` map (Section 2.3). Thus, we provided a description of how it is calculated.

3.3.1. View angle

For each viewpoint, `r.survey` calculates the three components \vec{a}_x , \vec{a}_y , and \vec{a}_z of the unit vector \hat{a} describing the direction of the line of sight from that viewpoint toward each visible target. We define the view angle as the angle (Θ) between unit vectors \hat{a} and \hat{b} (Figure 3). As shown in the illustration, the view angle for visible pixels has a well-defined range of values: $90^\circ < \Theta < 180^\circ$. The latter (180°) is the condition of unit vectors having opposite direction. In contrast, $\Theta = 90^\circ$ indicates orthogonal directions, while values below 90° indicate that the unit vectors are oriented in the same direction and, thus, are not visible to each other.

According to formula (2), Θ can be calculated by exploiting the components of \hat{a} and in the east, north, and vertical directions. This calculation was repeated for all available viewpoints.

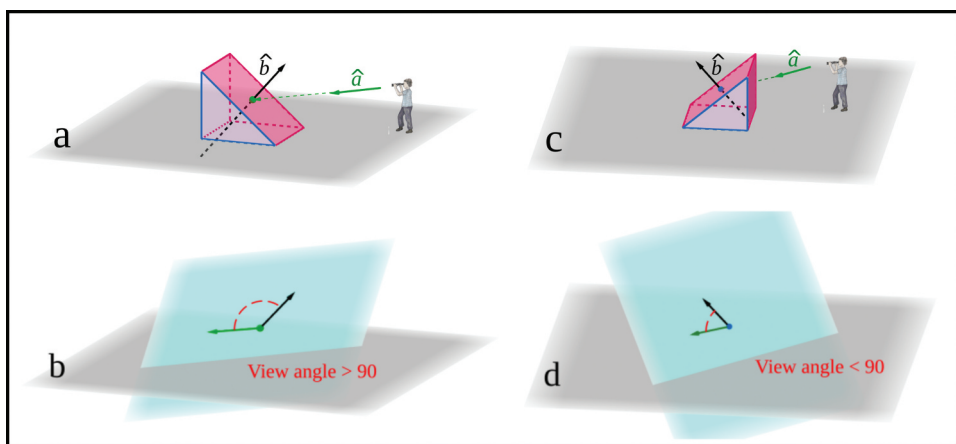


Figure 3. View Angle scheme for a single viewpoint (green arrow) and single cell (black arrow). Illustrations a and b represent the situation of a slope visible to the observer. Illustrations c and d show the view angle for a slope that is not visible to the observer.

$$\Theta = \arccos \left[\frac{\vec{a}_x \cdot \vec{b}_x + \vec{a}_y \cdot \vec{b}_y + \vec{a}_z \cdot \vec{b}_z}{\sqrt{\vec{a}_x^2 + \vec{a}_y^2 + \vec{a}_z^2} \cdot \sqrt{\vec{b}_x^2 + \vec{b}_y^2 + \vec{b}_z^2}} \right] \quad (2)$$

3.3.2. Solid angle

The solid angle is one of the best and most objective indicators for quantifying visibility (Domingo-Santos *et al.* 2011). Unlike all other visibility indices, the value of the solid angle depends on the size and orientation of the object being viewed. The solid angle provides a measure of the portion of the field of view occupied by the observed object. For example, to an observer, a circle will appear smaller the further away it is located. This is because the area occupied by this circle is reduced with respect to the total field of view, and, hence, the solid angle value decreases. The idea is that any observed object would be progressively less perceived as far away and the more tilted it is with regard to the viewpoint.

For the calculation of the solid angle ($\Omega(s)$) of an object of size s , we use two approximations. First, we use a circle to describe the equivalent size of the object; use of a circle provides an important advantage because, regardless of how tilted the circle is with respect to the observer, there is always a direction in which its diameter remains unaltered in the ellipse. Such an ellipse results from the projection of the circle in the plane perpendicular to the line of sight and is located at a distance d from the observer (see Figure 4). Therefore, based on this property, r.survey calculates the area of the

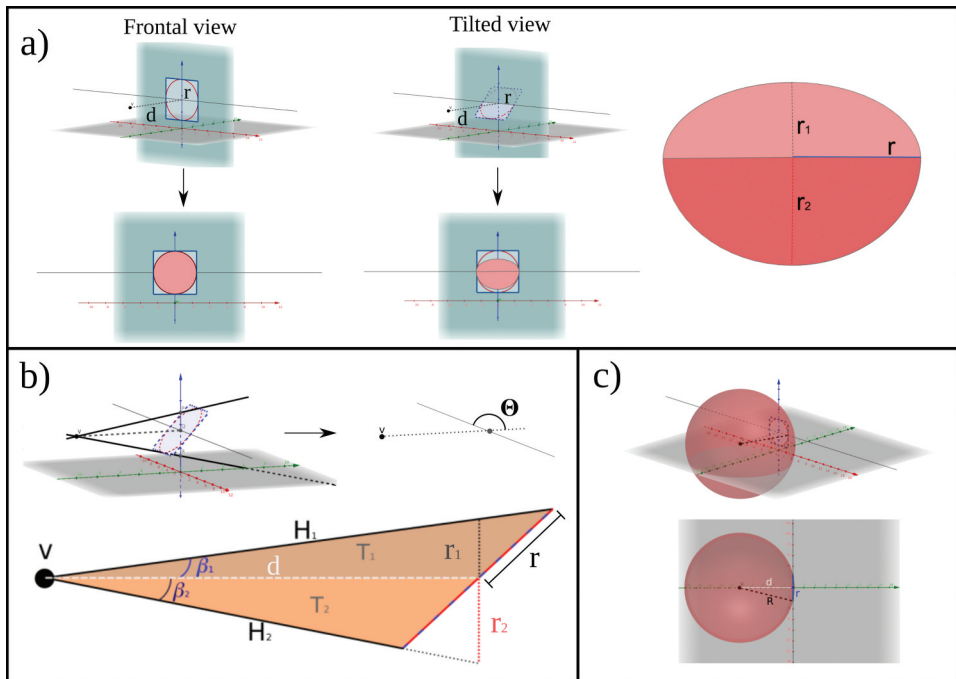


Figure 4. Solid Angle scheme. (a) Projected ellipse of a tilted circle. (b) Trigonometric measure scheme where it is noted that r_1 and r_2 are perpendicular to d . (c) Spherical view.

projected ellipse (A_{ellipse}). The circle radius (r) is set by default to half of the DEM resolution to correspond to the equivalent circle inscribed within each cell. Nevertheless, the user can specify this value if the target size is known (more details are available in the code documentation, see the code availability section).

The projected ellipse can be split into two different semi-ellipses whose major axes are equal to the diameter of the original circle. In [Figure 4\(b\)](#), we show how the minor axes of the two semi-ellipses depend on (i) the target pixel 3D distance (d) from the viewpoint (v) and (ii) the view angle (Θ) value. Focusing on the plane that passes through the center of the cell, the viewpoint, and the vertical axis, two triangles can be defined (T_1 and T_2 in [Figure 4\(b\)](#)) and used to calculate the semi-axes r_1 and r_2 according to Equations 3(a)–5(b).

$$r_1 = \tan(\beta_1) \cdot d \quad (3a)$$

$$r_2 = \tan(\beta_2) \cdot d \quad (3b)$$

where

$$\beta_1 = \text{acos} \left[\frac{r^2 - H_1^2 - d^2}{-2 \cdot H_1 \cdot d} \right] \quad (4a)$$

$$\beta_2 = \text{acos} \left[\frac{r^2 - H_2^2 - d^2}{-2 \cdot H_2 \cdot d} \right] \quad (4b)$$

and

$$H_1 = \sqrt{r^2 + d^2 - 2 \cdot r \cdot d \cdot \cos(270 - \Theta)} \quad (5a)$$

$$H_2 = \sqrt{r^2 + d^2 - 2 \cdot r \cdot d \cdot \cos(-90 + \Theta)} \quad (5b)$$

The solid angle of a surface is by definition equal to the projected spherical surface in the evolving sphere (A_{spheric}) divided by the square of the radius of the sphere [$R = (r^2 + d^2)^{0.5}$] ([Figure 4\(c\)](#)). In the second approximation made by *r.survey*, we use the ellipse surface area (A_{ellipse}) instead of the projected spherical surface in the evolving sphere. This assumption becomes less biased when $d \gg r$ when $A_{\text{spheric}} \simeq A_{\text{ellipse}}$. Consequently, we assume that the solid angle can be approximated using Equation (6).

$$\Omega(s) \simeq \frac{A_{\text{ellipse}}}{(r^2 + d^2)}, \quad (6)$$

where the solid angle value is in steradians. This calculation is repeated for all viewpoints.

An estimate of the percentage error due to the approximation of the projected spherical surface with the ellipse surface is given in [Figure 5](#) as a relative percentage of the difference between the exact solid angle and the approximated value (Equation (7)).

$$\text{Error} = \frac{(A_{\text{sphere}} - A_{\text{ellipse}})}{A_{\text{sphere}}} \cdot 100 \quad (7)$$

We compared the value of the solid angle and of the approximate solid angle of two circles with radius 2.5 and 25 m located at a distance ranging from 1 to 1000 m from the observer. If we observe both the solid and dotted curves, the error decreases rapidly and

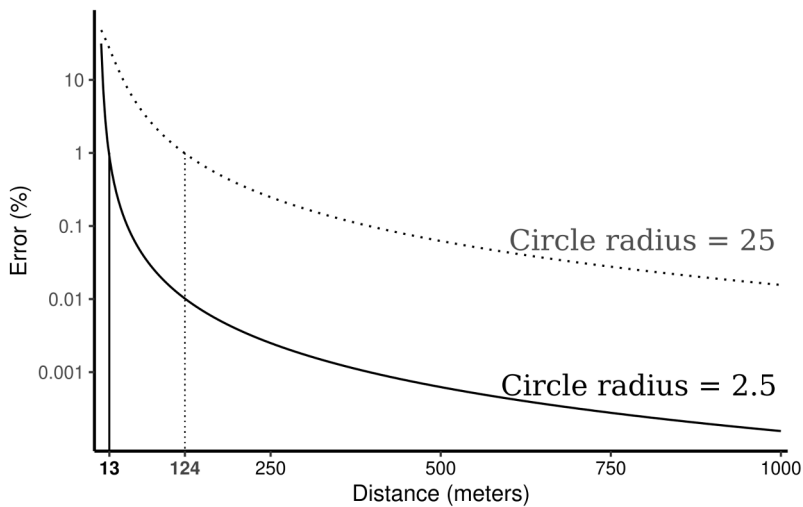


Figure 5. Error plot. Black and dotted lines represent the percentage error between the real solid angle and the approximate solid angle with respect to the distance for two circles with radii 2.5 and 25 m, respectively.

becomes negligible (less than 1%) at a distance from the observer larger than 13 and 124 m, respectively, which, in relative terms, implies fewer than 3 pixels when the radius is half the DEM resolution. The error becomes even smaller (0.1%) after 40 and 394 m (fewer than 8 pixels).

3.4. Data aggregation loops and output maps

When a vector map including more than one viewpoint is provided as input to `r.survey` Δ , Θ , and $\Omega(s)$ visibility metrics are calculated for each location, producing a map for each metric for each viewpoint. Visibility metric maps are then aggregated (third block of the code; see Figure 1) and elaborated to generate the final cartographic products.

Seven output raster maps are created, plus an additional eighth map if the optional binary map flag (-b) is specified. The `min3dDistance` map portrays the value of the minimum three-dimensional distance between each pixel and the closest viewpoint in meters. The `maxViewAngle` map shows the value of the maximum view angle between each pixel and the viewpoints, in degrees. Given that 180° implies a face-to-face view and 90° signifies an orthogonal view between the observer and the terrain, `maxViewAngle` is a measure of the frontal view from which each single cell is visible. Another output is the `maxSolidAngle` map, which shows the value of the maximum solid angle among those measured from different viewpoints. In this map, the pixels closest to the observation points have to be interpreted with special attention, considering that the error, with respect to the real solid angle, can reach 10% in the immediate neighboring pixels (see Section 2.2).

The other three maps (`pointOfViewWithMin3dDistance`, `pointOfViewWithMaxAngle`, and `pointOfViewWithMaxSolidAngle`) are used to register, in each cell, the identifier of the viewpoint from where an observer can obtain, respectively: (i) the minimum values of

3D distance, (ii) the maximum values of view angle, and (iii) the maximum value of the solid angle. Another relevant output is the `numberOfViews` map, which portrays the number of viewpoints from where each pixel is visible. In contrast, `r.survey` can provide a binary map of visibility. A value of 1 implies that the pixel is visible from at least one observation point, and null values represent the pixels that are not visible from any of the observation points. All the cited maps are filtered based on the choice of the user by the view angle threshold (fourth block in [Figure 1](#)). Setting a value slightly larger than 90° for this threshold has the effect of removing, from the output maps, all those cells oriented nearly perpendicular to the lines of sight.

4. Multi-core processing

One of the main improvements with respect to the first version of `r.survey` (Bornaetxea *et al.* 2018) is the possibility of running blocks II and III ([Figure 1](#)) in parallel, splitting the points representing the observer location into p sets, where p is the number of parallel processes required by the user (with $p = <$ to the number of available CPU cores). When the aim is to derive the values of the visibility metrics along a given path (e.g. a road or a UAV track), viewpoints can be very dense in terms of number per unit of distance, and the higher the number of viewpoints, the longer the computational time. Parallel processing makes such tasks more tractable in reasonable times.

Parallel computation was implemented using the Python multiprocessing library and the ability of GRASS GIS to set a temporary spatial region centered on the considered point without affecting the parallel computation of other points.

We tested the influence of the multi-core processing on the computational time of the survey using the entire Gipuzkoa Province (Basque Country) as our study area (Bornaetxea *et al.* 2018), which has an area of 1980 km^2 . We used a set of 1000 viewpoints randomly located along a road network in the study area. We used three different DEMs at resolutions of 100×100 , 50×50 , and 20×20 m, in combination with two different maximum distance parameter values (1000 m and 10,000 m).

Here, we describe the tests performed using a 64 core (AMD Opteron 4 \times Sixteen-Core a 2.5 GHz and cache 2048 KB) server with 320 GB of RAM and running Ubuntu 18.04 GNU/Linux OS with a 4.15.0-101-generic kernel image. Proportionally similar results were obtained when working on standard laptop or desktop machines with 8 or 12 cores and lower RAM.

The results show that the computational time, using the 100 m DEM resolution and with the maximum distance equal to 1000 m, was 14,479 and 526 s for 1 and 48 cores, respectively. The values became 49,313 and 1795 s, respectively, using the 20 m DEM resolution and maximum distance equal to 10,000 m. Hence, we observe a computational time reduction equal to a factor larger than 27.

We exploited the speedup (S_p) and efficiency (E_p) indexes (Mergili *et al.* 2014) to evaluate the gain in computing time due to parallel processing. S_p is defined as T_0/T_p , where T_0 and T_p represent the execution times with a single core or p cores, respectively. $S_p = p$ indicates an ideal speedup, with no relevant irreducible sequential part of the code. E_p is equal to $T_0/(p \cdot T_p)$ and in case of ideal performance is equal to 1.0.

The processing performance has weak dependence on DEM resolution and even weaker dependence on the maximum distance parameter ([Figure 6](#)). Speedup and

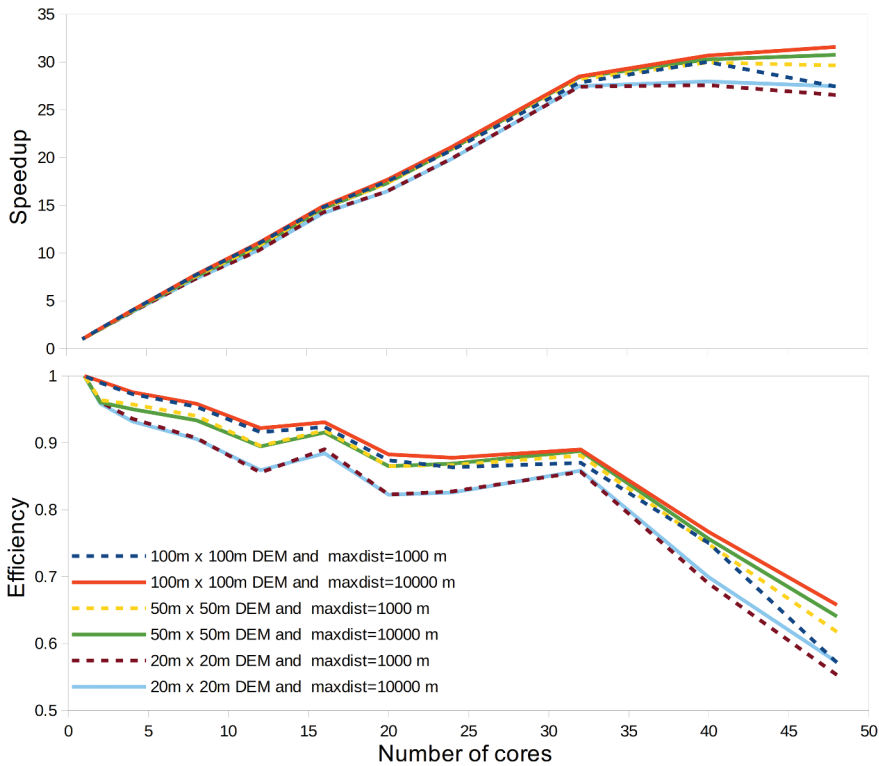


Figure 6. Speedup and efficiency curves of *r.survey* multi-core processing. Each line represents a DEM with a given resolution and different maximum distance.

efficiency remained at relatively high values until 32 cores, and then an abrupt reduction in performance was observed. With 32 cores, speedup reached a computational time reduction equal to a factor close to 27 or 29 (depending on the experimental settings), and the efficiency was close to 0.9. These numbers describe the efficient computational performance of the parallel code. We argue that the rapid reduction in the performance for a number of cores larger than 32 is because the CPU architecture is in effect, including hyper-threading, and that the real number of cores is only 32. The additional CPUs are detected as logical CPUs by the GNU/Linux kernel, and are not able to efficiently manage the concurrent access to the same execution unit (Qun *et al.* 2016).

5. Performance assessment

We carried out several experimental tests in both controlled and real-world environments to illustrate and validate the products of *r.survey*.

5.1. Performance assessment using synthetic data

To further assess performance, we designed a synthetic straight valley with a slight inclination at the bottom, simulating a small, regular river valley (Figure 7). The initial set of experiments was based on viewpoints located at ground level. First, a single

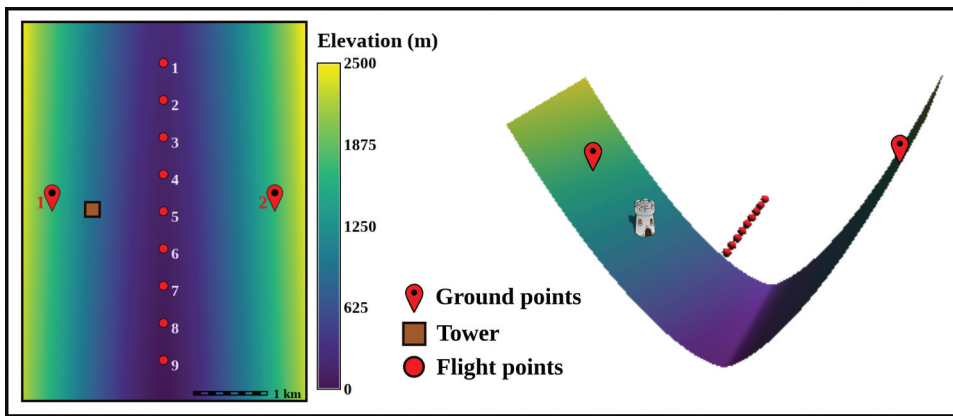


Figure 7. Details of the synthetic valley and the observer viewpoints used in Section 4.

viewpoint located in the upper part of one side of the valley was used. Further, we set a large tower (50 m tall, square footprint of 30×30 m) in the middle of the slope. Subsequently, we added a symmetric observation point to the opposite slope (Figure 7) and repeated the calculations. During these tests, we set the maximum visible distance to 3500 m but left the remaining options as default (including the observer elevation which was 1.75 m above the ground level).

The second set of experiments included multiple viewpoints located at relevant distances from the ground. The red dots in Figure 7 simulate the viewpoints of a helicopter flying at a constant elevation of 1000 m above sea level. The tower was then removed and the process was again run twice. In one case, we used the spherical field of view, while we reduced it to a downward view in the other case.

In all experiments, the resolution of the DEM was 5×5 m, and the object radius used for calculating the solid angle was 2.5 m, although this value can be modified according to the user's requirements.

5.1.1. Ground-level viewpoints

In the first experiment, a single viewpoint was used (Figure 8). All three output maps (solid angle, view angle, and 3D distance) highlight that the largest portion of the territory is included as part of the considered maximum visible distance, except for the two portions in the NE and SE corners of the maps and the few aligned cells hidden behind the tower. The 3D distance map (Figure 8(d)) shows the minimum values for pixels that are close to the viewpoint; these values then grow radially to reach a maximum value larger than 3500 m. This is expected because the maximum distance value was used through r . viewshed to define a bi-dimensional buffer around the viewpoint location. The view angle map (Figure 8(b)) shows a clear differentiation between one slope of the valley and the opposite slope, with values that are always larger than 90° , but never reach 180° . The highest view angle values are on the opposite slope as that is where the terrain is oriented toward the observer. We also observe that, given the synthetic DEM (a valley) and the viewpoint location (one side of the valley), it is not possible to obtain a visual angle equal to 180° (perfect frontal view). Indeed, the combination of distance and visual

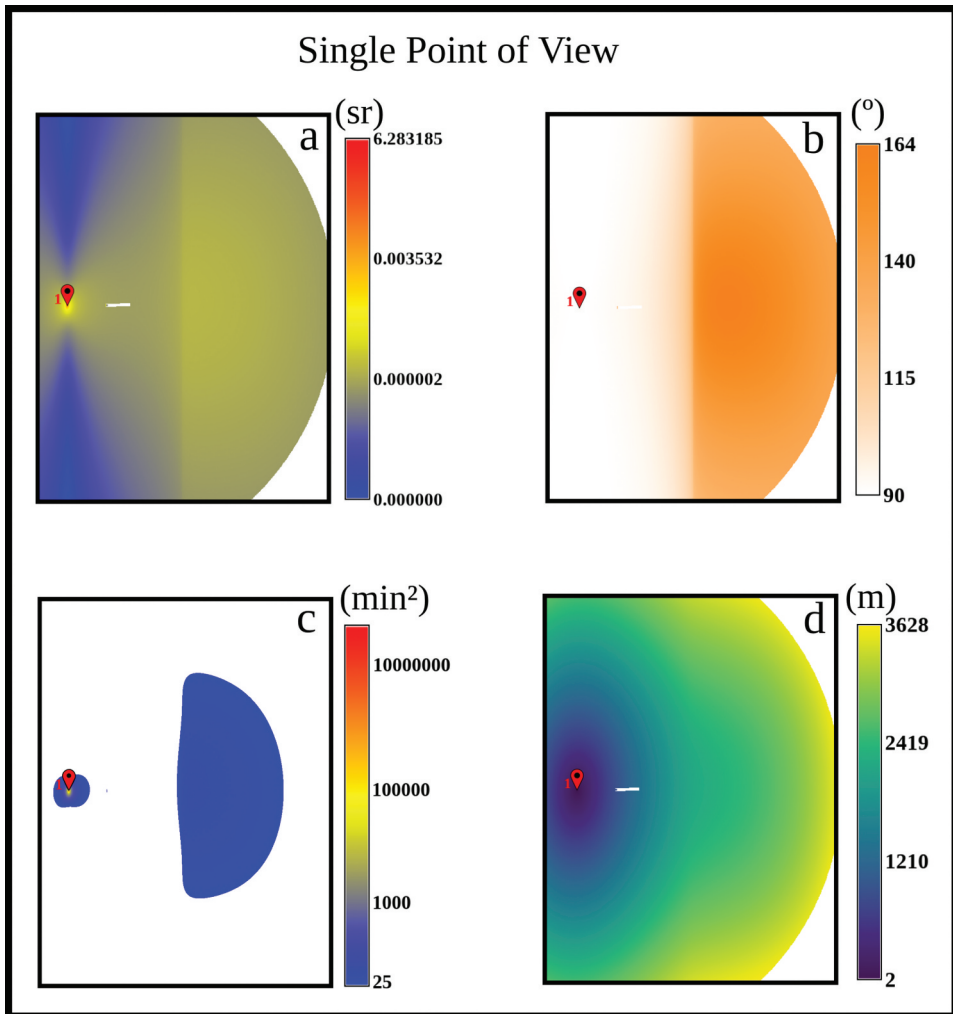


Figure 8. Results of the experiment with one single observer. (a) Solid angle in steradians (sr) (maxSolidAngle map). (b) View angle in degrees ($^{\circ}$) (maxViewAngle map). (c) Solid angle values transformed into min^2 and filtered by a threshold of 25. (d) 3D distance in meters (m) (min3dDistance map).

orientation is represented by the solid angle map (Figure 8(a)), which shows that, in addition to the values in the pixels surrounding the viewpoint, the best view from this point to the ground corresponds to the opposite slope. The lowest values are located in the north–south direction, where, as the view angle is close to 90° , and as the 3D distance increases, the solid angle becomes very small, reaching values smaller than 2×10^{-6} steradians.

As a reference, it is well known that 4π steradians (sr) subtend the entire visible sphere and that 1 sr is equal to approximately 3283 square degrees (deg^2) and to $1.18 \cdot 10^7$ square minutes (min^2). It means that $2 \cdot 10^{-6} \text{ sr} \simeq 23.6 \text{ min}^2$. For comparison, letter acuity (the capacity to resolve a letter) is approximately 25 min^2 (Healey and Sawant 2012) for a

human with perfect vision and in controlled conditions, that is, high contrast between the letters and the background. For simplicity purposes, [Figure 8\(c\)](#) shows only those pixels with a solid angle greater than 25 min^2 . In this case, there is a significant reduction in the portion considered visible. Moreover, under natural conditions (mist, vegetation, low contrast, etc.), the capability to detect and map features on the slope likely requires higher solid angle values, and, hence, the visible area should be even smaller. Therefore, the solid angle threshold (which depends on the observing sensor), as well as the average size of the object, can be customized by users in real-world applications.

On the other hand, it is interesting to note how the solid angle, view angle, and 3D distance maps change when an additional viewpoint is located on the opposite side of the valley ([Figure 9\(a–c\)](#)). Considering these two viewpoints, it is useful to verify from which viewpoint the single cells are more visible. In [Figure 9\(d–f\)](#), we show the `pointsOfView*` maps that portray the ID of the viewpoints from where each single cell is visible. Additionally, the `numberOfView` map ([Figure 9\(g\)](#)) indicates the number of points from which each pixel is visible. Indeed, the latter (green color) shows the portion of the terrain visible from both points of view. In addition, this map shows how the cells hidden by the tower in the experiment using a single viewpoint are now visible from the second viewpoint. [Figure 9\(d,e\)](#) allow us to clarify that the eastern corners are visible from the eastern viewpoint (ID = 2), while the opposite occurs for the western corners.

The 3D distance map ([Figure 9\(c\)](#)) shows the highest values at the bottom of the valley. The corresponding `pointOfViewWithMin3dDistance` map ([Figure 9\(f\)](#)) allows us to verify that, as expected, each side of the valley has a minimum distance from the viewpoints lying on the same side. However, the part hidden by the tower for point 1 is instead visible from point 2.

A similar behavior can be found in view angle maps ([Figure 9\(b,e\)](#)). In this case, the values in each slope correspond to the viewpoint located on the opposite side of the valley, with the exception of the four corners of the map.

The maximum solid angle results deserve special attention ([Figure 9\(a,d\)](#)). In particular, looking at the `pointOfViewWithMaxSolidAngle` map ([Figure 9\(d\)](#)), a buffer area of several meters can be observed surrounding each viewpoint location. This implies that, inside these buffers, the best viewpoint is the closest one. In contrast, similar to the previous experiment, to observe something outside the boundaries delimited by the buffer area, the viewpoint located on the opposite side of the valley should be preferred.

5.1.2. Flight observation points

In this experiment, nine points were used to evaluate the performance of `r.survey`. Points were located at a constant elevation of 1000 m above sea level along the valley ([Figure 10](#)).

As expected, the resulting maps cover different extensions when the downward-looking direction option is enabled or disabled. If not used, the field of view of the sensor and/or observer encompasses the total spherical field of view. In contrast, when forcing the downward looking direction, only the portion of the valley under the viewpoint's elevation is considered. The values corresponding to the visible pixels were identical in both cases.

Because it is a constant elevation flight, the relative distance between the points and the ground changes as a result of the inclination of the synthetic valley. This is highlighted in the minimum 3D distance map ([Figure 10\(i,j\)](#)), where the values of the cells located close to viewpoint 1 are smaller than those of the cells located in proximity of viewpoint 9.

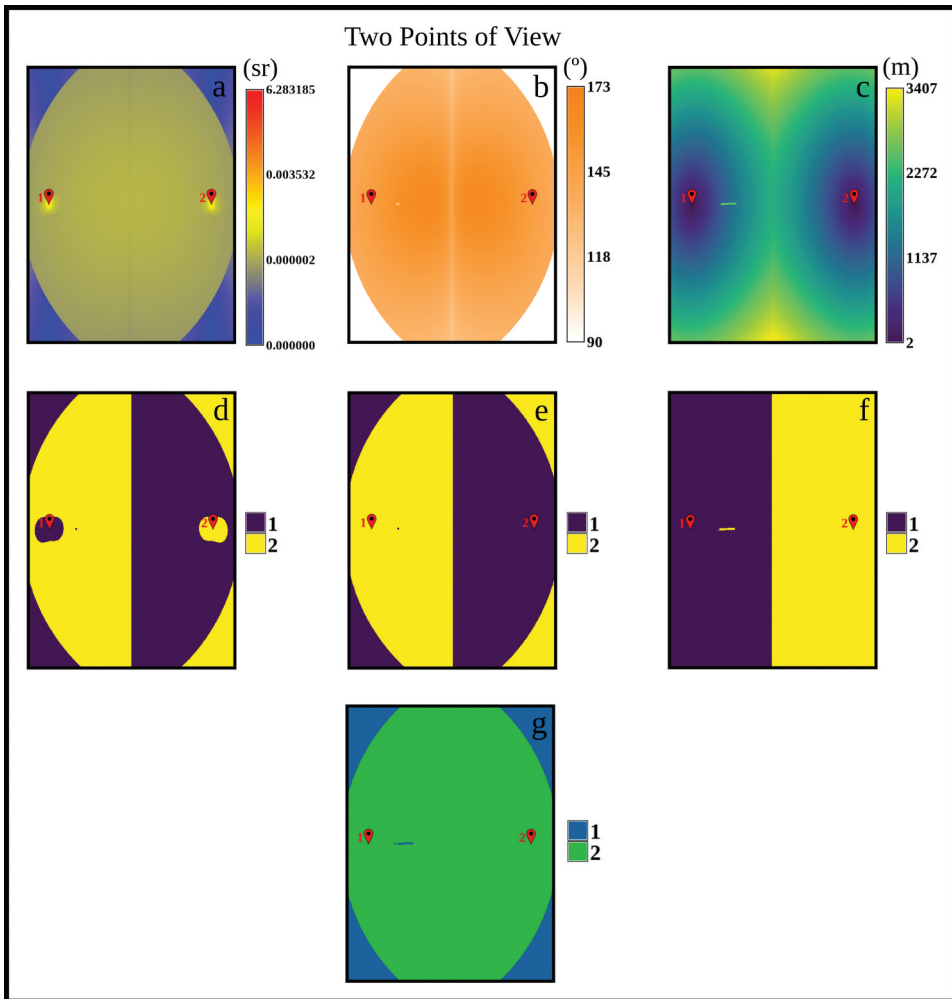


Figure 9. Results of the experiment with two observers. (a) Solid angle in steradians (sr) (maxSolidAngle map). (b) View angle in degrees ($^{\circ}$) (maxViewAngle map). (c) 3D distance in meters (m) (min3dDistance map). (d) Points of view of maximum solid angle (pointOfViewWithMaxSolidAngle map). (e) Points of view of maximum view angle (pointOfViewWithMaxAngle map). (f) Points of view of minimum 3D distance (pointOfViewWithMin3dDistance map). (g) Number of views (numberOfViews map).

The maximum view angle values are concentrated at the very bottom of the valley exactly under each viewpoint (Figure 10(e,f)), where the frontal view is total (180°), and at both sides of the valley, where the unit vectors of the line of sight and of the normal to the terrain have opposite direction.

We observe that the solid angle values are quite homogeneous in the maps (Figure 10(a,b)), at least when the downward option is used. This again confirms that r.survey is working as expected. Compared to what we obtain from the ground positions, from an aerial position, we expect to have a nearly equivalent capacity across the different cells to detect objects on the terrain.

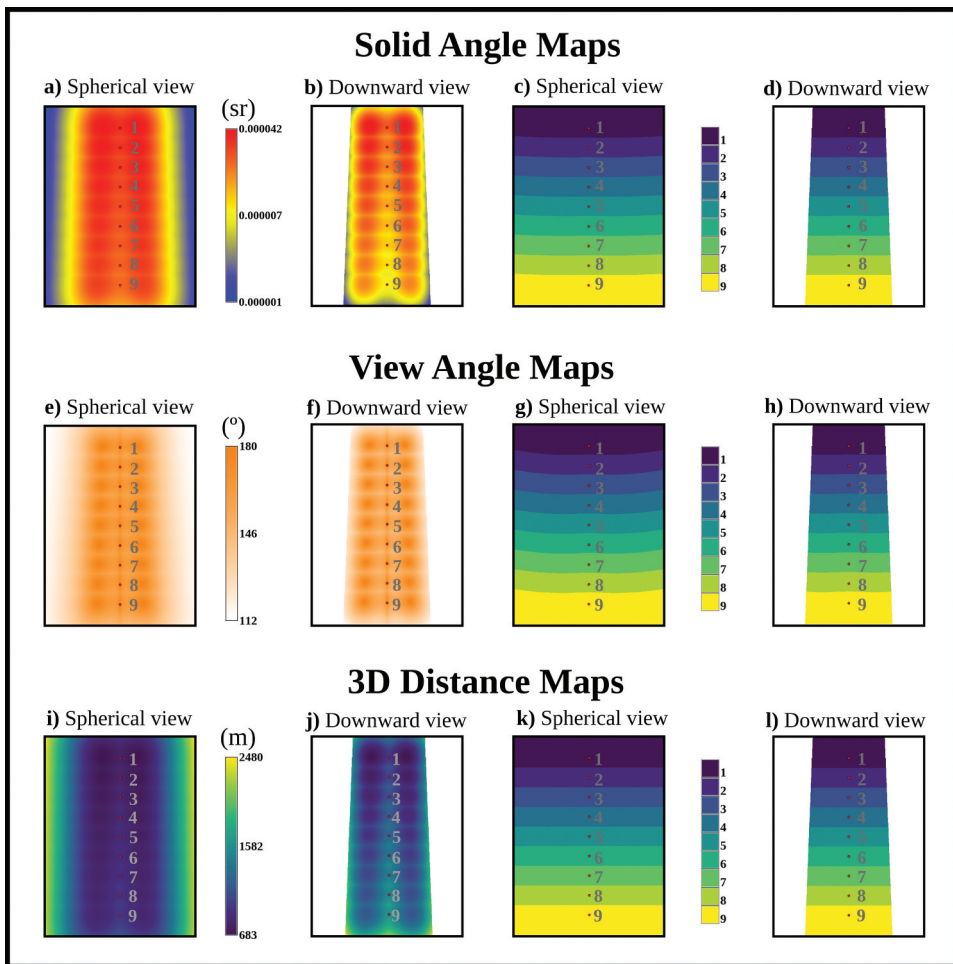


Figure 10. Results of different experimental settings simulating helicopter flight. Downward and spherical view maps share the same scale bar. Maps (a) and (b) show the solid angle in steradians (sr) (maxSolidAngle maps). Maps (e) and (f) view angle in degrees ($^{\circ}$) (maxViewAngle maps). Maps (i) and (j) show 3D distance in meters (m) (min3dDistance maps). Maps (c), (d), (g), (h), (k), and (l) show the corresponding point-of-view values (pointOfView* maps).

Finally, considering that our synthetic valley was designed not to have irregular roughness, there are no hidden areas, and all terrain is visible from every point of view (see Section 5.2 for examples in a real DEM). However, because of the regular horizontal distance of 500 m between each viewpoint, the pointOfView* maps (Figure 10(c,d,g,h,k,l)) show a quite regular distribution as well.

5.2. Performance assessment using a real-world DEM

To exemplify the potential of the software in a real case study, we ran r.survey again for Gipuzkoa Province (Bornaetxea *et al.* 2018). The viewpoints, of which there were 18,103, were regularly distributed along the road network of the studied area, and we used the

available 5×5 m resolution DEM. In addition, we chose a large maximum visible distance (6000 m) and used the default values for the remaining options (which implies a radius of 2.5 m for the target object used for calculating the solid angle).

As expected, even if we set a large maximum visible distance value, there are few hidden areas because of the presence of different aspects of land forms (Figure 11(a)). Overall, the viewshed covered 85% of the territory. In Figure 11(b), we excluded from the viewshed area those cells where the solid angle was smaller than 100 min^2 . We consider this value to be a reasonable threshold for visual acuity, under natural conditions, compared to the controlled conditions in a laboratory, where the minimum threshold is 25 min^2 (Healey and Sawant 2012). Therefore, Figure 11(b) shows the area in which an object with an average size of $\pi \cdot 2.5^2 \sim 20 \text{ m}^2$ can be detected. Note that viewshed area is significantly reduced and covers 62% of the territory. Figure 11(c,d) show a 3D representation of a small portion of the studied area. The red lines represent the roads from which visibility metrics were calculated. We also show two different locations (i.e. A and B), where a landslide could hypothetically occur. The colors of the DEM indicate the values of the solid angle calculated for objects with a circle radius of 2.5 m (Figure 11(c)) and 7 m (Figure 11(d)), which are equivalent to circles having an area of $\sim 20 \text{ m}^2$ and $\sim 150 \text{ m}^2$, respectively. The values of the solid angle changed significantly in the two cases. In particular, it was observed that location B was placed in an area where a landslide of $\sim 20 \text{ m}^2$ would not be visible (solid angle is close to 0.0 in Figure 11(c)). In contrast, a greater landslide (area $\sim 150 \text{ m}^2$) in the same location would have been visible because the solid angle values shown in Figure 11(d) were larger than 100 min^2 . On the other hand, in location A, both a small and a large landslide would have been visible because the solid angle values were close to or above 1000 min^2 (Figure 11(c,d)).

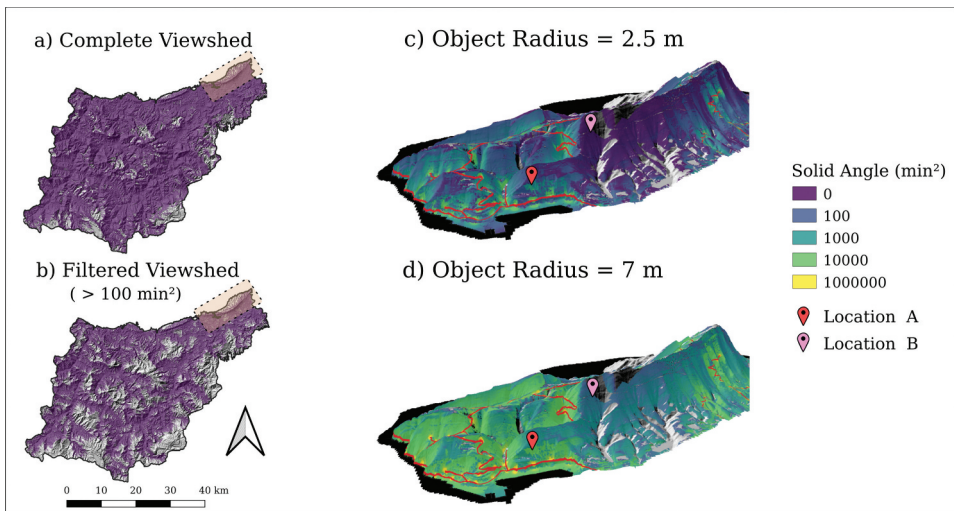


Figure 11. Results of the application of *r.survey* in a real-world DEM. (a) Complete viewshed. (b) Viewshed filtered based on solid angle values ($>100 \text{ min}^2$). (c) Solid angle map for a circle having a radius of 2.5 m in a zoomed-in portion of the terrain. (d) Solid angle map for a circle having a radius of 7.0 m, in a zoomed-in portion of the terrain.

In this example, we calculated solid angle maps for objects with a larger size with respect to the DEM cell resolution. We highlight that the calculated solid angle depends on the size of the object. Therefore, the object must be thought to be oriented according to the slope and aspect of the cells in which it is located. Consequently, we stress that the solid angle values calculated by *r.survey* in a given cell must not be summed with those of the surrounding pixels because they are only consistent within the pixel itself.

6. Discussion

How to assess whether or not a portion of terrain is visible is an ongoing subject of research (Klouček *et al.* 2015, Nutsford *et al.* 2015, Sahraoui *et al.* 2018, Abdulhafedh 2019, Fábrega-Álvarez and Parceró-Oubiña 2019, Palmer 2019, Saeidi *et al.* 2019, Mancini 2020). In fact, for survey activities, visibility (line of sight) is not sufficient if the objects of the survey are too small, too distant, or oriented in such a way that they cannot be perceived and/or mapped. This paper presents a multipurpose tool that returns relevant information to determine how features and/or objects of the analyzed terrain are actually visible. The tool relies on three visibility metrics which measure: (i) how far the target is (*min3dDistance*), (ii) how it is oriented with respect to the viewpoint (*maxViewAngle*), and (iii) how much it can be perceived (*maxSolidAngle*). Whether the observer is a human being, an optical camera mounted in an unnamed aerial vehicle (UAV), or a radar sensor mounted on a satellite, the user can take advantage of the information provided by those visibility metrics and delimit the 'effective survey area' according to specific aims. In addition, *r.survey* can alter the DEM to consider the presence of trees and buildings. We acknowledge that visibility and perception also depend on other environmental conditions such as color contrast with the background and atmospheric extinction (Domingo-Santos and Fernández de Villarán 2017). However, these conditions may vary with time and are not considered here.

Compared to existing tools available for the automatic detection of visible areas, *r.survey* presents similarities, differences, and some improvements (Table 1). The main similarities are the ability to process multiple viewpoints and to calculate a map describing the number of viewpoints from which each cell is visible. Improvements include the introduction of view angle, 3D distance, and solid angle output maps, which are not present in other tools, especially those integrated in GIS packages. A major difference is the identification of the points from which each individual cell is visible. In other types of software, this identification produces a list of points. In *r.survey*, three different output maps show, for each cell, the identification of the viewpoint (i) located most frontally (maximum view angle), (ii) located at the minimum distance (minimum 3D Distance), and (iii) from which the cell is most visible (maximum solid angle). Only Pixscape provides outputs comparable to *r.survey*; however, for both angular height and angular area, it follows the same approach as Domingo-Santos *et al.* (2011), so it only offers the values corresponding to the cell size (Sahraoui *et al.* 2016a, 2016b). This approach limits the inference of the real visibility of objects of a certain size compared with the *maxSolidAngle* of *r.survey*. Additionally, as a result of the 3D distance and view angle layers, which are not provided by PixScape, *r.survey* can be exploited in a wider field of potential applications. The possibility of setting multiple elevation values to the observation points (such as in flight mode) and the option of altering the size of the target object

whose solid angle will be calculated provides high versatility. Furthermore, if we consider that our software is fully integrated in a powerful GIS environment (GRASS GIS), it facilitates the development of any other output in post-processing. However, the authors acknowledge that Pixscape is optimized for its primary application in computing landscape visibility. Thus, it provides specific outputs for this purpose, such as a wide variety of visibility metrics or the tangential view. Meanwhile, *r.survey* has been optimized for earth surface analysis and, thus, evaluation of the visibility of features on terrain slopes (or included in the DEM).

The illustrated examples in [Section 5](#) describe the informative content of the products of *r.survey*. The NumberOfViews map ([Figure 9\(g\)](#)) provides information on the number of viewpoints a target is visible from; this is an issue relevant to fields such as visual impact assessment studies (Shang and Bishop 2000, Joly *et al.* 2009, Palmer 2019, Saeidi *et al.* 2019). PointOfView* maps allow identification of the position with the best visibility (maximum solid angle), the most frontal view (maximum view angle), or the nearest location (minimum 3D distance) with respect to a given target ([Figures 9 and 10](#)). Such information could be exploited to define the best position of telecommunication antennas or tourist viewpoint locations (Dodd 2001, Edan *et al.* 2013, Tajelsir Raof *et al.* 2020). Meanwhile, the maxViewAngle layer could be beneficial for remote-sensing applications. Knowing the elevation above sea level and the orbital path followed by a satellite, the orientation of the terrain with respect to the sensor could provide useful information for assessing layovering, shadowing, and foreshortening (Herrera *et al.* 2013, Lu *et al.* 2019, Mondini *et al.* 2019, Letortu *et al.* 2020).

We believe that the maxSolidAngle map is a valuable product of *r.survey*. Similar to Domingo-Santos *et al.* (2011), our tool can calculate the solid angle from multiple viewpoints. However, unlike the tool proposed by those authors, not all the potentially thousands of solid angle maps are made available to the user. Instead, for each pixel, the maximum solid angle value is registered (maxSolidAngle map), together with the information about the identifier of the viewpoint from where each pixel is best visible (pointOfViewWithMmaxSolidAngle map). We did not choose, as suggested by Domingo-Santos *et al.* (2011), to sum up values of the solid angles calculated for the different viewpoints because theoretically this can lead to cumulative solid angle values that are unrealistically larger than 4π sr.

Another advantage of *r.survey* is the possibility of calculating the solid angle map for target objects with different sizes from the DEM resolution. Again, this is different from the work of Domingo-Santos *et al.* (2011), where the solid angle refers to the pixel size. We maintain that choosing the target object size is quite relevant as it allows us to delineate target-size specific 'effective survey areas' (Bornaetxea *et al.* 2018, Knevels *et al.* 2020); that is, the portion of the territory where an observer is actually able to detect target objects. The 'effective survey area' can be defined by thresholding the solid angle map based on a value that depends, for example, on human visual acuity. This value can be further modified to consider non-optimal environmental conditions during the survey. The illustrative examples in [Section 5.2](#) and [Figure 11](#) show one potential application of this concept. We demonstrated how the effective surveyed area could be significantly reduced even using a conservative threshold of 100 min^2 . Furthermore, [Figure 11\(c,d\)](#) show that two landslides located in the same position can be considered

visible depending on their size, and the tool provides the necessary information to evaluate it.

The parallel design of the code allows calculations to be sufficiently accurate in terms of spatial resolution in an affordable processing time. This is of major importance for its usage in real cases, allowing the assessment in a local to regional context with the only needs being a digital elevation model and a set of viewpoints. Therefore, as a tool designed for relatively large spatial analysis, the approach followed for the solid angle calculation (even being an approximation) offers a correct metric to define whether or not something is visible for areas from a medium distance and beyond (after approximately 140 m according to Higuchi (1983)). The only exceptions are the immediate pixel areas around the observer, where more specific approaches should be applied (Domingo-Santos *et al.* 2011).

7. Conclusions

In this paper, we presented an effective algorithm and a useful software tool for earth surface survey assessment. Using a synthetic study area, the results endorse the idea that *r.survey* provides relevant spatial information for assessing the perception of the terrain from remote observers located either above the ground or mounted in any aerial device (such as UAVs, helicopters, and satellites).

This software provides outputs not available in other similar packages, such as 3D distance or view angle; it also offers some relevant setting options that make *r.survey* highly customizable for multiple purposes. The main advances with respect to the rest of the viewshed analysis algorithms are (i) the possibility that the observer is on the ground or in a flying device and (ii) the capability to estimate the solid angle according to the size of the object to be surveyed, rather than forcing it to the pixel size. This last option, specifically, is devoted to consider the relevance of the size of the object that one is attempting to detect in a survey.

In summary, we highlight that the parallelized design of the new release of *r.survey* increases its efficiency and allows process performance in high spatial resolution digital elevation models for large areas with an affordable processing time.

We offer the *r.survey* code in the form of an open-source license to facilitate and encourage future improvements by additional researchers. In addition, we intended to overcome the gaps in the binary representation of a complex GIS product such as the viewshed, providing users with very basic information to let them decide what is considered visible and what is not.

Acknowledgments

The authors kindly appreciate the useful comments provided by the three anonymous reviewers and the editor that contributed to improve the manuscript. In addition, the authors want to thank Frida Clerissi for the English language revision and Michele Santangelo and Alessandro Mondini for the useful suggestions and information provided.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work was supported by the postdoctoral fellowship program of the Basque Government obtained by one of the authors [grant numbers POS_2019_1_0020] in collaboration with the Geological Survey of Canada; the research group IT1029-16 of the University of the Basque Country (UPV/EHU); and the geomorphology group of CNR IRPI.

Notes on contributors

Txomin Bornaetxea is post-doctoral researcher at the University of the Basque Country (EHU/UPV) thanks to a research scholarship sponsored by the regional Basque Government since 2020. He reached the PhD degree in Physical Geography on 2018 and his research is focused on the field of landslide susceptibility modelling in regional scale. Since 2017 until 2020 he worked as lecturer in the degree of Geography at the same university, giving lessons about Climatology, Geomorphology, Photo Interpretation and Thematic Cartography among others.

Ivan Marchesini is permanent researcher at the Italian Consiglio Nazionale delle Ricerche (CNR) since 2012. He is a GIS/geospatial specialist with more than 15 year of expertise on Open-Source GIS. He works in different fields of applied research such as fluvial geomorphology, slope stability, erosion processes, environmental impact assessment, estimation of CO² consumption by atmospheric weathering and mines reclamation.

ORCID

Txomin Bornaetxea  <http://orcid.org/0000-0002-1540-3991>

Ivan Marchesini  <http://orcid.org/0000-0002-8342-3134>

Data and codes availability statement

The data and code, as well as the user's guide, that support the findings of this study are available with the identifier at the public links: <https://doi.org/10.6084/m9.figshare.13246595>
<https://doi.org/10.5281/zenodo.3993140>

The digital elevation model of Gipuzkoa Province with 5 × 5 meters of resolution, as well as the road network, was downloaded from the official geospatial data repository www.geo.euskadi.eus.

References

- Abdulhafedh, A., 2019. An innovative GIS method for evaluating the visibility of the road using the ArcMap-Tools. *Open Access Library Journal*, 6, 1–23. doi:10.4236/oalib.1105586
- Bornaetxea, T., et al., 2018. Effective surveyed area and its role in statistical landslide susceptibility assessments. *Natural Hazards and Earth System Sciences*, 18 (9), 2455–2469. doi:10.5194/nhess-18-2455-2018.
- Chae, B.G., et al., 2017. Landslide prediction, monitoring and early warning: a concise review of state-of-the-art. *Geosciences Journal*, 21, 1033–1070. doi:10.1007/s12303-017-0034-4
- Cuckovic, Z., 2016. Advanced viewshed analysis: a Quantum GIS plug-in for the analysis of visual landscapes. *The Journal of Open Source Software*, 1 (4), 32. doi:10.21105/joss.00032.

- Dodd, H.M., 2001. *The Validity of Using a Geographic Information System's Viewshed Function as a Predictor for the Reception of Line-of-Sight Radio Waves*. Thesis (PhD). Virginia Tech.
- Domingo-Santos, J.M., et al., 2011. The visual exposure in forest and rural landscapes: an algorithm and a GIS tool. *Landscape and Urban Planning*, 101 (1), 52–58. doi:10.1016/j.landurbplan.2010.11.018.
- Domingo-Santos, J.M. and Fernández de Villarán, R., 2017. Visibility analysis. In: S.D. Brunn, ed. *International encyclopedia of geography*. American Cancer Society. Hoboken, NJ: John Wiley & Sons, 1–14. doi:10.1002/9781118786352.wbieg0906.
- Edan, J.D., Idowu, T.O., and Zango, I.S., 2013. Viewshed analysis of MTN telecommunication network in Jimeta metropolis, Adamawa State, Nigeria. *International Journal of Environmental and Ecology*, 3, 99–106.
- ESRI, 2021. *Release notes in ArcGIS Pro 20.7* [online]. Available from: <https://pro.arcgis.com/es/pro-app/latest/get-started/release-notes.htm> [Accessed 18 February 21].
- Fábrega-Álvarez, P. and Parcerou-Oubiña, C., 2019. Now you see me. An assessment of the visual recognition and control of individuals in archaeological landscapes. *Journal of Archaeological Science*, 104, 56–74. doi:10.1016/j.jas.2019.02.002
- Fisher, P.F., 1992. First experiments in viewshed uncertainty: simulating fuzzy viewsheds. *Photogrammetric Engineering & Remote Sensing*, 58, 345.
- Fisher, P.F., 1996. Extending the applicability of viewsheds in landscape planning. *Photogrammetric Engineering & Remote Sensing*, 62 (11), 1297–1302.
- Germino, M.J., et al., 2001. Estimating visual properties of Rocky Mountain landscapes using GIS. *Landscape and Urban Planning*, 53 (1–4), 71–83. doi:10.1016/S0169-2046(00)00141-9.
- GRASS Development Team, 2020. Geographic Resources Analysis Support System (GRASS GIS) Software, Version 7.0. Open Source Geospatial Foundation.
- Haverkort, H., Toma, L., and Zhuang, Y., 2009. Computing visibility on terrains in external memory. *ACM Journal of Experimental Algorithmics*, 13, 5–23. doi:10.1145/1412228.1412233
- Healey, C.G. and Sawant, A.P., 2012. On the limits of resolution and visual angle in visualization. *ACM Transactions on Applied Perception*, 9 (4), 1–21. doi:10.1145/2355598.2355603.
- Herrera, G., et al., 2013. Multi-sensor advanced DInSAR monitoring of very slow landslides: the Tena Valley case study (Central Spanish Pyrenees). *Remote Sensing of Environment*, 128, 31–43. doi:10.1016/j.rse.2012.09.020
- Higuchi, T., 1983. *Visual and spatial structure of landscapes*. Cambridge: The MIT Press.
- Joly, D., et al., 2009. A quantitative approach to the visual evaluation of landscape. *Annals of the Association of American Geographers*, 99 (2), 292–308. doi:10.1080/00045600802708473.
- Klouček, T., Lagner, O., and Šimová, P., 2015. How does data accuracy influence the reliability of digital viewshed models? A case study with wind turbines. *Applied Geography*, 64, 46–54. doi:10.1016/j.apgeog.2015.09.005
- Knevels, R., et al., 2020. Event-based landslide modeling in the Styrian Basin, Austria: accounting for time-varying rainfall and land cover. *Geosciences*, 10 (6), 217. doi:10.3390/geosciences10060217.
- Letortu, P., et al., 2020. The potential of Pléiades images with high angle of incidence for reconstructing the coastal cliff face in Normandy (France). *International Journal of Applied Earth Observation and Geoinformation*, 84, 101976. doi:10.1016/j.jag.2019.101976
- Llobera, M., 2003. Extending GIS-based visual analysis: the concept of visualsapes. *International journal of geographical information science*, 17(1),25–48. doi:10.1080/713811741
- Llobera, M., 2007. Reconstructing visual landscapes. *World Archaeology*, 39 (1), 51–69. doi:10.1080/00438240601136496.
- Lu, P., et al., 2019. Landslide mapping from multi-sensor data through improved change detection-based Markov random field. *Remote Sensing of Environment*, 231, 111235. doi:10.1016/j.rse.2019.111235
- Mancini, M.F., 2020. An algorithmic approach to viewsheds analysis for cultural landscape: Manziana and the Bracciano Lake Area. In: M. Di Nicolantonio, E. Rossi, and T. Alexander, eds. *Advances in additive manufacturing, modeling systems and 3D prototyping, advances in intelligent systems and computing*. 24-28 July 2019 Cham: Springer International Publishing, 169–179. doi:10.1007/978-3-030-20216-3_16.

- Mergili, M., et al., 2014. A strategy for GIS-based 3-D slope stability modelling over large areas. *Geoscientific Model Development*, 7 (6), 2969–2982. doi:10.5194/gmd-7-2969-2014.
- Minelli, A., et al., 2014. An open source GIS tool to quantify the visual impact of wind turbines and photovoltaic panels. *Environmental Impact Assessment Review*, 49, 70–78. doi:10.1016/j.eiar.2014.07.002
- Mondini, A.C., et al., 2019. Sentinel-1 SAR Amplitude Imagery for Rapid Landslide Detection. *Remote Sensing*, 11 (7), 760. doi:10.3390/rs11070760.
- Neteler, M., et al., 2012. GRASS GIS: a multi-purpose open source GIS. *Environmental Modelling & Software*, 31, 124–130. doi:10.1016/j.envsoft.2011.11.014
- Nutsford, D., et al., 2015. Personalising the viewshed: visibility analysis from the human perspective. *Applied Geography*, 62, 1–7. doi:10.1016/j.apgeog.2015.04.004
- Ogburn, D.E., 2006. Assessing the level of visibility of cultural objects in past landscapes. *Journal of Archaeological Science*, 33 (3), 405–413. doi:10.1016/j.jas.2005.08.005.
- Palmer, J.F., 2019. The contribution of a GIS-based landscape assessment model to a scientifically rigorous approach to visual impact assessment. *Landscape and Urban Planning*, 89, 80–90. doi:10.1016/j.landurbplan.2019.03.005
- Petrasova, A., et al., 2018. Viewshed Analysis. In: A. Petrasova, et al., eds. *Tangible modeling with open source GIS*. Cham: Springer International Publishing, 127–132. doi:10.1007/978-3-319-89303-7_9.
- Proy, C., Tanré, D., and Deschamps, P.Y., 1989. Evaluation of topographic effects in remotely sensed data. *Remote Sensing of Environment*, 30 (1), 21–32. doi:10.1016/0034-4257(89)90044-8.
- Qun, N.H., et al., 2016. Hyper-threading technology: not a good choice for speeding up CPU-bound code. In: Y.K. Chen, ed. *3rd International Conference on Electronic Design (ICED)*, 11–12 August 2016 Phuket: IEEE, 578–581. doi:10.1109/ICED.2016.7804711.
- Saeidi, S., et al., 2019. Assessing the visual impacts of new urban features: coupling visibility analysis with 3D city modelling. *Geocarto International*, 34 (12), 1315–1331. doi:10.1080/10106049.2018.1478891.
- Sahraoui, Y., et al., 2018. Integrated GIS software for computing landscape visibility metrics. *Transactions in GIS*, 22, 1310–1323. doi:10.1111/tgis.12457
- Sahraoui, Y., Clauzel, C., and Foltête, J.C., 2016a. Spatial modelling of landscape aesthetic potential in urban-rural fringes. *Journal of Environmental Management*, 181, 623–636. doi:10.1016/j.jenvman.2016.06.031
- Sahraoui, Y., Youssoufi, S., and Foltête, J.C., 2016b. A comparison of in situ and GIS landscape metrics for residential satisfaction modeling. *Applied Geography*, 74, 199–210. doi:10.1016/j.apgeog.2016.07.011
- Santangelo, M., et al., 2010. Remote landslide mapping using a laser rangefinder binocular and GPS. *Natural Hazards and Earth System Sciences*, 10, 2539–2546. doi:10.5194/nhess-10-2539-2010
- Shang, H. and Bishop, I.D., 2000. Visual thresholds for detection, recognition and visual impact in landscape settings. *Journal of Environmental Psychology*, 20, 125–140. doi:10.1006/jevps.1999.0153
- Tajelsir Raouf, M.A., Fritsch, D., and Abdalla, R., 2020. Signal coverage of low-land areas using geographic information systems, case study: Kassingar Area, Sudan. In: A. Froehlich, ed. *Space fostering African societies: developing the African continent through space, Part 1, Southern Space Studies*. Cham: Springer International Publishing, 219–230. doi:10.1007/978-3-030-32930-3_15.
- Toma, L., et al., n.d. *r.viewshed - GRASS GIS manual* [online]. Available from: <https://grass.osgeo.org/grass78/manuals/r.viewshed.html> [Accessed 26 May 2020].
- Zambelli, P., Gebbert, S., and Ciolli, M., 2013. Pygrass: An Object Oriented Python Application Programming Interface (API) for Geographic Resources Analysis Support System (GRASS) Geographic Information System (GIS). *ISPRS International Journal of Geo-Information*, 2, 201–219. doi:10.3390/ijgi2010201