

Landscape relevance analysis with GRASS

Elisabetta MIORELLI, Paolo ZATELLI

Dipartimento di Ingegneria Civile e Ambientale,
Università degli Studi di Trento,
via Mesiano 77, 38100 Trento,
tel. 0461.882618, Fax 0461.882672,
e-mail: paolo.zatelli@ing.unitn.it

Abstract

One of the most important variables for territorial planning in mountainous areas is the impact of new works on the landscape. This paper describes a study on the evaluation of the impact of new works by mapping the land visibility from the main roads. This is done by creating a set of points along the road lines and evaluating the visibility from each point. By summing the resulting maps, a visibility index, given by the number of points along the roads from which the area is visible, is obtained. Moreover, it is possible to link to each point along the roads the number of visible areas, providing in this way a “panoramic” index for roads tracts. All these procedures have been made automatic by scripts for the GRASS GIS.

Riassunto

Uno dei fattori che la pianificazione del territorio deve tenere in particolare considerazione, soprattutto nelle zone montuose, è l'impatto che la realizzazione di nuove opere può avere sul paesaggio. Questo articolo descrive uno studio per la valutazione dell'impatto sul paesaggio di nuove opere attraverso la costruzione di mappe della visibilità del territorio, ottenute mediante la disposizione di una serie di punti lungo gli assi viari a distanza costante e valutando per ogni punto le aree visibili. Sommando le mappe ricavate per ogni punto si ottiene un indice della visibilità di ciascuna area, dato dal numero di punti lungo la viabilità da cui l'area è visibile. E' altresì possibile associare a ciascun punto lungo la viabilità il numero di aree visibili, valutando in questo modo la "panoramicità" dei tratti di strada. Tutte le procedure sono state automatizzate realizzando script per il GIS GRASS.

1. Introduction

New infrastructures and buildings have great impact on mountainous regions, both for the ecological balance and landscape conservation, because of the particular morphology of the terrain. Here, even small buildings or roads can bear huge impacts on the landscape for wide areas.

The prevision of the impact of new infrastructures on the landscape can be difficult using traditional cartography or CAD systems, especially when the impact is on medium or large areas. The right tool to perform such task is a Geographic Information System, since it combines the ability to create visibility maps to the management of all the geographic data. It is therefore possible to integrate landscape analysis with other land management analysis, for example for regional planning.

2. Study area

The study area is a secondary valley of the Trentino Region (Italy): Valle del Fersina. The Fersina valley, which altitude varies from 600 to 2.071 meters a.s.l., is typically “V” shaped, deeply

marked by the Fersina stream that flows through in southwest directions. This is a narrow valley, with a particular history that makes it unique with respect to population and landscape. In fact, unlike its surroundings, this valley had been settled during 1200 by coal miners originating from Germany, which developed a different land settlement from the usual model spread among Trentino Region. In the same way, due to the difficulties in communications the valley had been isolated during the past centuries from its neighbourings, keeping its linguistic and cultural tradition unchanged.

Nowadays the valley maintains a nearly undamaged natural landscape and a strong identity that makes it particularly vulnerable to the impact of new works (roads and infrastructure in general), which nonetheless represent basic steps to the development of the area that otherwise could be excluded from progress.



Figure 1 – 3D view of the study area.

3. Materials and methods

This study aims to the evaluation of two intertwined aspects of landscape:

1. the impact that a modification in one site has on the landscape;
2. the “panoramic grade” of road tracts.

This is done by evaluating the part of the study area which is visible from a set of points along the paved and unpaved roads. These data are used:

1. digital terrain model as raster with resolution of 10 meters (nominal scale of 1:10000), this is used to describe the morphology of the land;
2. roads network as vector of 1:10000 nominal scale, it is used to generate points along the roads;
3. a set of point along the roads network, whence visibility is evaluated.

The first two set of data are provided by the local government, Provincia Autonoma di Trento, the third one can be automatically generated by a GIS or CAD system, for example by the *v.to.points* module in GRASS. The resolution of the DTM, 10 meters, has been used for all the raster maps.

The visibility map and all the subsequent elaborations have been carried out using the FOSS GIS GRASS¹, which provides two modules for landscape analysis:

1. *r.cva*, which generates a cumulative viewshed map giving the user the possibility of choosing the viewpoints (all points in the DTM, systematic grid, random samples or user given sites). This module is not available for GRASS 6.0 at time of writing, therefore has not been used.
2. *r.los*, which creates a raster map in which the visible cells from a user-specified observer location are assigned an integer value representing the vertical angle (in degrees) under which those cells are seen. Non visible cells are assigned the NULL value. If the map is reclassified as a binary map with value 1 for visible cells and NULL otherwise, the output map is equal to that of the *r.cva* module with the last viewpoints input option.

The latter GRASS module has been used to generate a visibility map for each point, with a syntax of this kind:

```
r.los input=dtm output=visibility_map coordinate=east,north  
obs_elev=1.75 max_dist=dist
```

Where *input* is the DTM map name, *output* the name of the output visibility map, *coordinate* the pair of east and north point coordinate, *obs_elev* the height above the terrain of the observer and *max_dist* the maximum distance for which the visibility is evaluated.

The output visibility map contains in each cell the vertical angle (in degrees) under which the cells is seen or the NULL value if the cell is not visible. This map has been reclassified, assigning value 1 to visible cells and NULL to non visible ones.

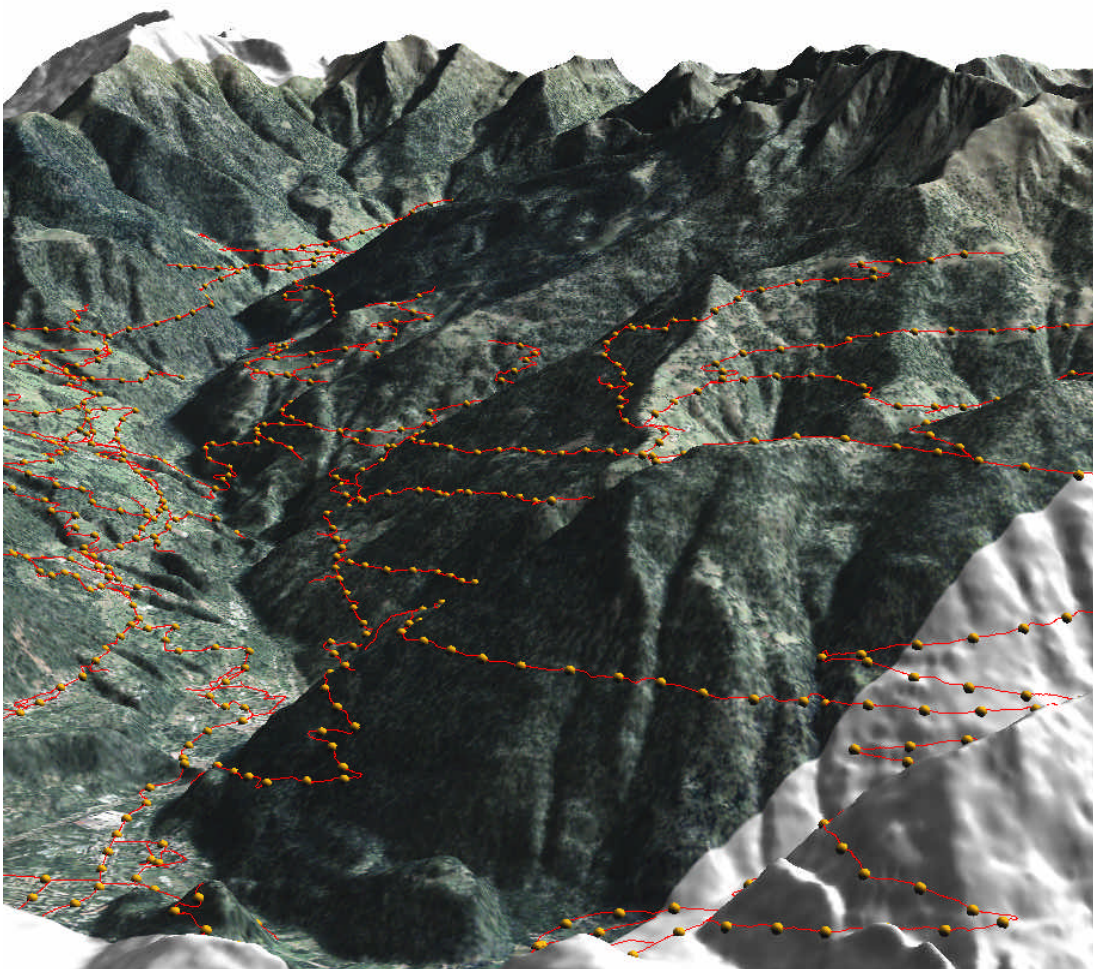


Figure 2 – Roads network and viewpoints set.

¹<http://grass.itc.it>

This creates a binary map (1=visible, 0=not visible) for each point, the global visibility map is obtained by simply summing pixel wise each visibility map: in this way the final map contains in each cell the number of points by which it can be seen. This can be easily done in GRASS using map algebra with the **r.mapcalc** module or by patching the maps with **r.patch**.

This approach is possible only when a small number of viewpoints is involved, since the **r.los** module must be run for each point and the resulting maps must be summed.

The number of points used in this application ranges between 135, for points placed at 200 meters of mutual distance, to 747 points when the distance is 50 meters, therefore the manual application of the procedure above is not feasible.

4. Scripts

Some scripts have been developed to automatically generate a visibility map for each point along the road and to sum all the resulting maps and to manipulate the database associated to the viewpoints. The first script:

1. reads the east, north coordinate pair from a vector map;
2. applies the *r.los* module to these points;
3. converts the resulting map into a binary map (1=visible, 0=not visible) for each point.

A second script sums all the resulting maps into a global visibility map, optionally deleting the maps which have been added, to save disk space.

A third script determines the total number of visible DTM cells for each point and adds this value to the point's record in the database table associated to the viewpoints map. For this purpose a new column is added to the table.

These scripts are reported in appendix A.

5. Results

Several visibility maps have been created by changing the mutual distance between points along the roads (50, 100 and 200 m) and the maximum distance for which the visibility is evaluated (1000, 2000 and 3000 m). Seven further points, which have been indicated by local authorities as panoramic points, have been added to the viewpoints' set in all the analysis. Some of the results are reported in the figures following, all the images have been created in GRASS except fig. X, created by QGIS.

6. Conclusions

GRASS offers powerful tools for visibility analysis by combining its modules. The use of scripts allows the automation of repetitive tasks and the combination of different modules, providing the possibility of creating complex applications in a simple and fast way.

Moreover, GRASS visualization tools allow an effective rendering of the results.

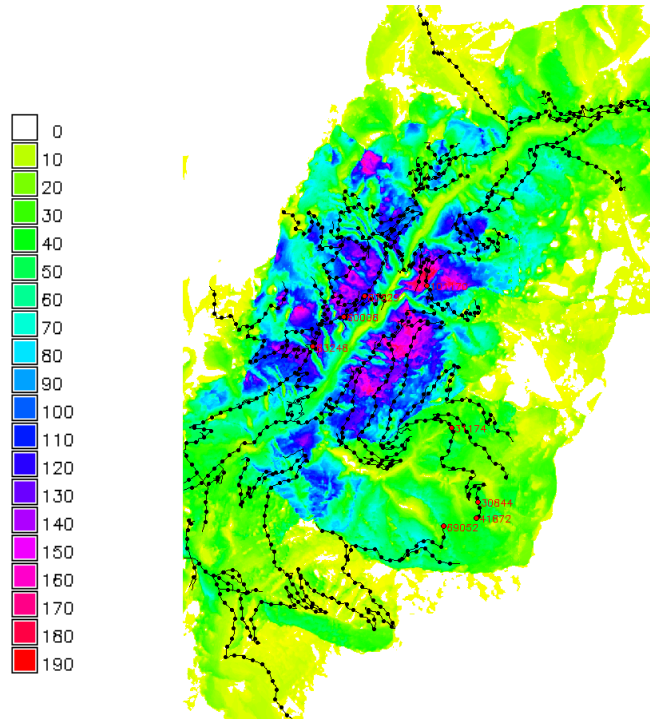


Figure 3 – Global visibility map, the seven red points are the panoramic points, labeled with the number of visible DTM cells. Numbers in the legend refer to the number of viewpoints from which the cell is visible.

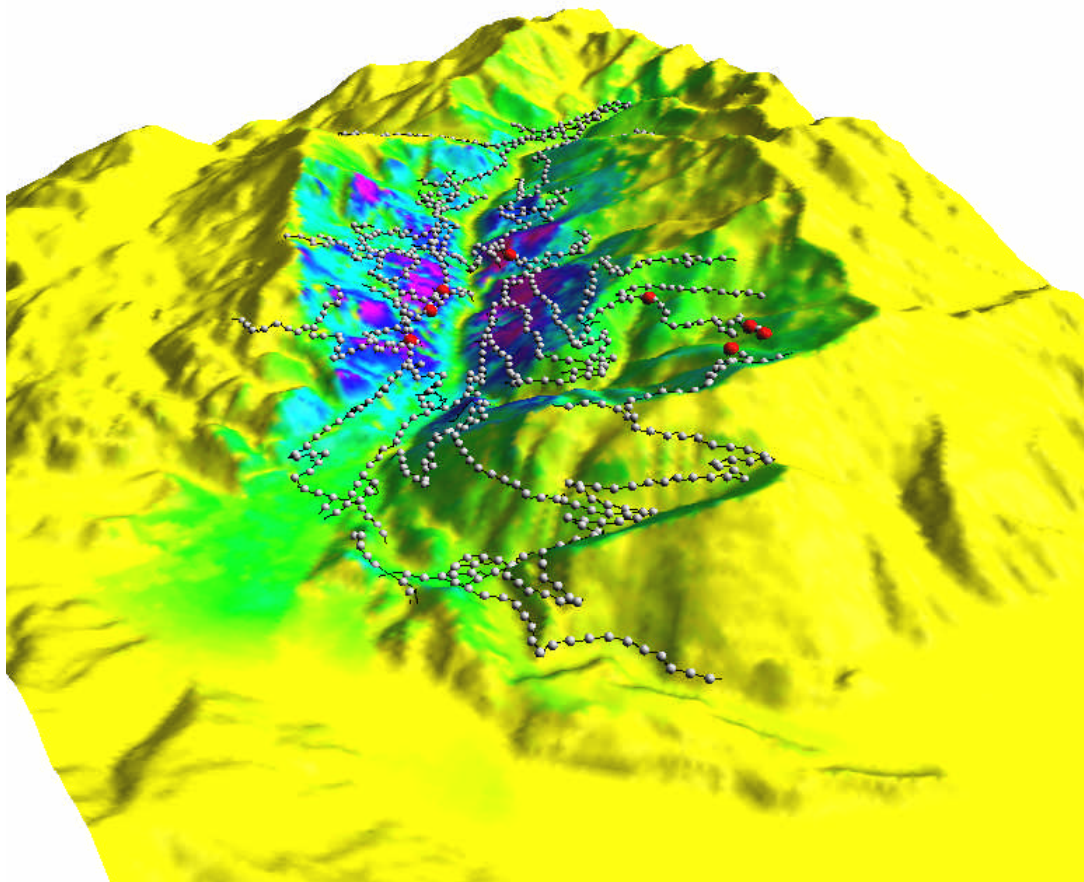


Figure 4 – 3D global visibility map, the seven red points are the panoramic points.

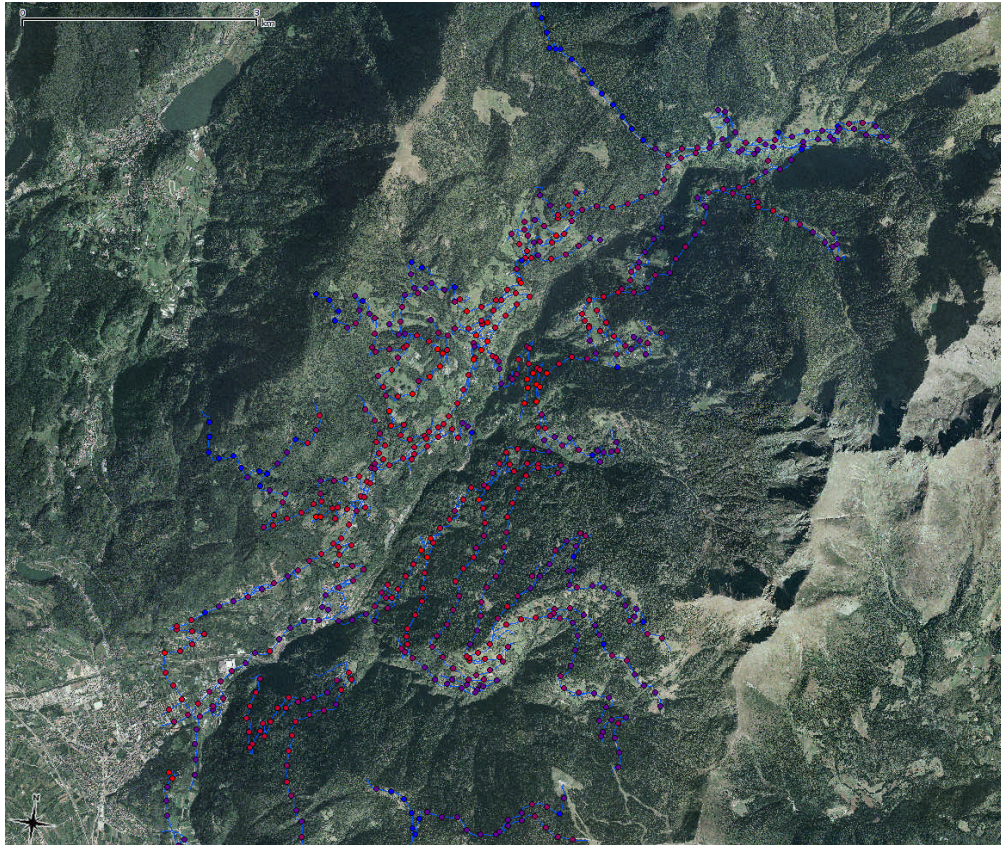


Figure 5 – Global visibility map, red viewpoints have high visibility score, indigo medium visibility and blue points low visibility.

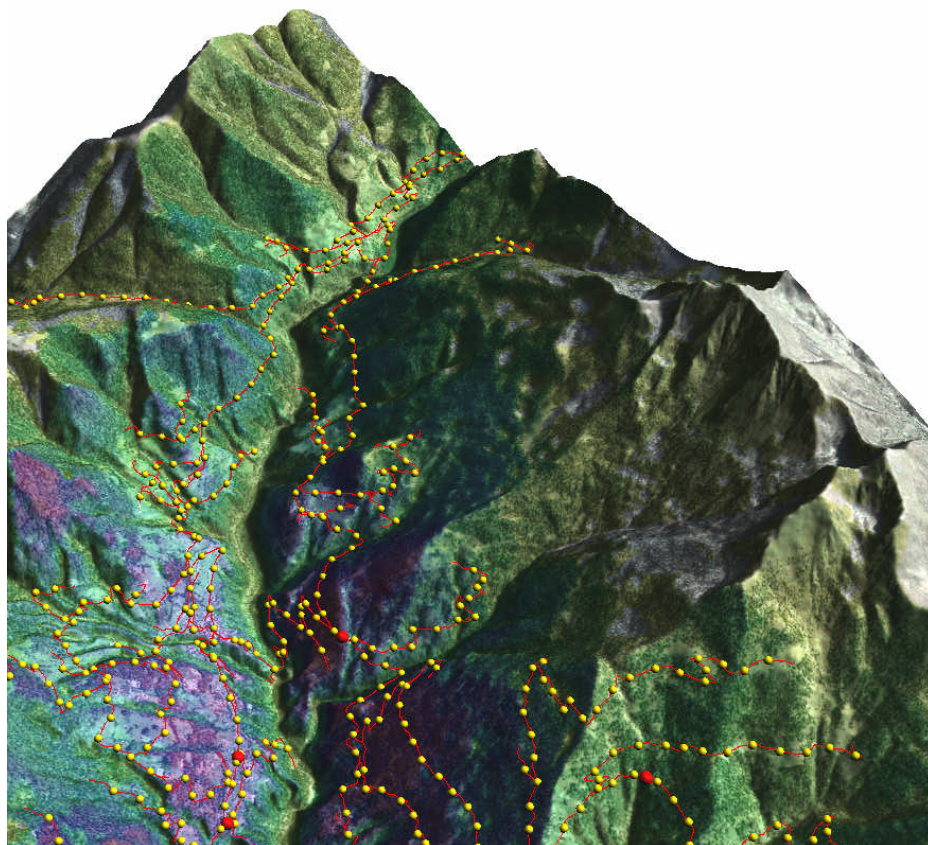


Figure 6 – Global visibility map, blended with orthophoto, the legend is the same of figure 3.

8. Bibliography

Wheatley D., (1995), *Cumulative viewshed analysis: a GIS-based method for investigating intervisibility, and its archaeological application*, in Lock G. R. and Stancic Z. (eds.) *Archaeology and Geographic Information Systems: A European Perspective*, London: Taylor and Francis, 171-186

Appendix A

Script 1: runs the *r.los* module for each point in a vector map. The input CSV file is create from the GRASS vector point map with the *v.to.db* module:

```
v.to.db map=puntipanoramici type=point layer=1 qlayer=1
option=coord column=x,y -p > coordinate_puntipanoramici.csv
```

The *dist* parameter is the maximum distance for which the visibility is evaluate, *mappa_visibilita_num*, with *num* equal to the progressive number of the point, are the names of the output map, *mappa_visibilita_num_bin* the names of the output binary maps.

```
#!/bin/sh
# warning: the csv file must contain the coordinates in decimal notation,
# not scientific
```

```
GISDBASE=/home/paolo/misc/elisabetta
LOCATION_NAME=valdeimocheni
MAPSET=frassilongo
```

```
dist=3000
File=coordinate_puntipanoramici.csv
map=puntipanoramici
mappa_visibilita=visibilita_puntipanoramici
```

```
#####
# nothing to change below this line
#####
```

```
input=""
```

```
START_DATE=`date`
echo "Start: "$START_DATE
```

```
#number of records in $map database
num=`v.to.db map=$map type=point layer=1 qlayer=1 option=coord\
column=x,y -p|wc -l`
let "num=$num-1"
echo $num
```

```
g.region rast=valdeimocheni.dtm
```

```
for ((a=1; a <= $num ; a++))
do
read line
#num=${line%%\|*0}
coll=`expr index "$line" \|`
#num1=`expr match "$line" '\(\|1*\)`
est=${line:coll:7}
col2=coll+8;
nord=${line:col2:7}
echo $est $nord
g.remove rast=$mappa_visibilita_"$a
r.los input=valdeimocheni.dtm output=$mappa_visibilita_"$a\
coordinate=$est,$nord obs_elev=1.75 max_dist=$dist
r.null map=$mappa_visibilita_"$a null=0
r.mapcalc $mappa_visibilita_"$a_"bin=\
if("visibilita_"puntipanoramici_"$a">"0")"
```

```

g.remove rast=$mappa_visibilita_"_$a

STOP_DATE=`date`
echo "Start: "$START_DATE
echo "Stop: "$STOP_DATE

```

Script 2 : sums all the resulting maps of the previous script. The single point visibility maps are deleted at the end of the process.

Note that the *r.mapcalc* module allows up to 256 input maps, therefore if the number of maps to sum is greater the sum is broken into parts.

```

#!/bin/sh
GISDBASE=/home/paolo/misc/elisabetta
LOCATION_NAME=valdeimocheni
MAPSET=frassilongo

dist=3000
File=coordinate_puntipanoramici.csv
input2=""
map=puntipanoramici
mappa_visibilita=visibilita_puntipanoramici

#####
# nothing to change below this line
#####

START_DATE=`date`
echo "Start: "$START_DATE

#number of records in $map database
num=`v.to.db map=$map type=point layer=1 qlayer=1 option=coord column=x,y\
-p|wc -l`
let "num=$num-1"
echo $num" maps to sum"

let "cent=$num / 100"

for ((a=1; a <= $cent ; a++)); do
    input=""

    echo "Processing "$(($a-1)*100+1)" to "$($a*100)

    for ((b=1; b <= 100 ; b++)); do
        let "c = ($a -1) * 100 + $b"
        if [ $b -eq 1 ]; then
            input=$mappa_visibilita_"_$c_"bin
        else
            input=$input+$mappa_visibilita_"_$c_"bin
        fi
    done

    r.mapcalc visibilita_"_$map_"tmp_"_$a=$input

    if [ $a -eq 1 ]; then
        input2=$mappa_visibilita_"tmp_"_$a
    else
        input2=$input2+$mappa_visibilita_"tmp_"_$a
    fi
done

let "num2=$num-$cent*100"
input=""

echo "Processing "$(($cent*100+1))" to "$num

```



```

for ((b=1; b <= $num2 ; b++)); do
    let "c = ($a -1) * 100 + $b"
    #echo "Processing "$c
        if [ $b -eq 1 ]; then
            input=$mappa_visibilita_"$c_"_bin
        else
            input=$input+$mappa_visibilita_"$c_"_bin
        fi
done
    let "a =$a +1"

r.mapcalc $mappa_visibilita_"_tmp_"_"$a=$input
input2=$input2+$mappa_visibilita_"_tmp_"_"$a

echo "Summing "$(($cent+1))" temporary maps."

r.mapcalc $mappa_visibilita_"_tot_"_"$dist=$input2

echo $mappa_visibilita_"_tot_"_"$dist "created."
echo " "
echo "Cleaning up temporary maps"

for ((b=1; b <= $a ; b++))
do
    g.remove rast=$mappa_visibilita_"_tmp_"_"$b
done

for ((a=1; a <= $num ; a++))
do
    g.remove rast=$mappa_visibilita_"_"$a"_bin"
    g.remove rast=$mappa_visibilita_"_"$a
done

echo ""$num" maps removed"

STOP_DATE=`date`
echo "Start: "$START_DATE
echo "Stop: "$STOP_DATE

```

Script 3 : determines the total number of visible DTM cells for each point and adds this value to the point's record in the database table associated to the viewpoints map. For this purpose a new column is added to the table. At the end of the process the table is printed un the GRASS terminal and the vector map displayed on the monitor with the new attribute.

```

#!/bin/sh
GISDBASE=/home/paolo/misc/elisabetta
LOCATION_NAME=valdeimocheni
MAPSET=frassilongo
dist=3000

map=puntipanoramici
mappa_visibilita=visibilita_puntipanoramici

#####
# nothing to change below this line
#####

START_DATE=`date`
echo "Start: "$START_DATE

# run once
echo "ALTER TABLE $map ADD COLUMN visib_$dist int" | db.execute

#number of records in $map database
num=`v.to.db map=$map type=point layer=1 qlayer=1 option=coor column=x,y\`

```

```

        -p|wc -l`
let "num=$num-1"
echo $num

for ((a=1; a <= num ; a++)); do
    echo "Processing "$a
    # number of non zero cells = number of visible cells
    line=`r.sum rast=$mappa_visibilita_"$a"_bin`
    col1=`expr index "$line" =`
    col2=`expr index "$line" .`
    let "col1=$col1+1"
    let "lungh=$col2-$col1-1"
    punti=${line:col1:lungh}
    echo "UPDATE $map SET visib_$dist=$punti WHERE cat=$a" | db.execute
done

STOP_DATE=`date`
echo "DB updated."
echo "select * from $map"|db.select|less

d.vect map=$map type=point,line,boundary,centroid,area,face display=shape,attr
attrcol=visib_$dist icon=basic/x size=6 layer=1 color=black fcolor=gray
llyer=1 lcolor=red bgcolor=none bcolor=none lsize=6font=romans xref=left
yref=center

echo "Start: "$START_DATE
echo "Stop: "$STOP_DATE

```