

Modelling with neighborhood operators

By:

A. Patrono and A. Saldaña

**Dept. of Earth Resources Surveys and Dept. Land Resource and Urban Sciences,
International Institute for Aerospace Survey and Earth Sciences (ITC),
P.O. Box 6, 7500 AA Enschede, The Netherlands**

Tel: +31 53 4874233, Fax: +31 53 4874336, e-mail: ADYPAT@ITC.NL

Tel: +31 53 4874444, Fax: +31 53 4874399, e-mail: SALDANA@ITC.NL

Summary

This case study proposes the use of the neighborhood operations of ILWIS for two types of analysis: (1) intervisibility-viewshed analysis, and (2) simulation of animal flows.

Getting started

The data for this case study are stored on the ILWIS 2.1 CD-ROM in the directory `d:\appguide\chap16`. If you have already installed the data on your hard-disk, you should start up ILWIS and change to the subdirectory where the data files for this chapter are stored, `c:\ilwis21\data\appguide\chap16`. If you did not install the data for this case study yet, please run the ILWIS installation program (see ILWIS Installation Guide).



- Double-click the ILWIS program icon in the ILWIS program group.
- Change the working drive and the working directory until you are in the directory `c:\ilwis21\data\appguide\chap16`.

Now you are ready to start the exercises for this case study.

16.1 Neighborhood operations

Neighborhood operations are calculations on pixels in which the outcome depends on the neighboring pixels. Neighborhood operations allow to make special spatial analysis using a filter, i.e. a mobile 3 x 3 window. The cell numbers in this imaginative window are coded from 1 to 9. Each cell of the output map is calculated according to specified neighborhood expressions and functions. For more information about neighborhood operations and the syntax used in ILWIS, it is strongly recommended to read the related paragraphs in the *User's Guide* before starting.

16.2 Available data

The following data set is available:

Dtm	Digital elevation model derived from the interpolation of contour lines
Target	a source map that will be used for the intervisibility analysis
Lau_or	Land use map derived from the classification of a LandSat multispectral image, improved with topographic maps.

16.3 Viewshed

The possibility to determine the intervisibility of points in a landscape is important for evaluating visual impacts, planning microwave communication networks, etc. (Burrough, 1986). Determining intervisibility from interpolated contour lines is not an easy task especially in the raster representation because of the large number of profiles that must be extracted and compared.

The site from which the viewshed needs to be calculated is identified on the DEM and hypothetical rays are sent out from this point to all points in the surroundings. Please notice that in DEMs the recorded heights may not take into account some landscape features (such as woods or buildings) and the results may need to be adjusted. The heights of landscape elements may be used to update directly the DEM in order to model their effect on intervisibility.

In Environmental Impact Assessment (EIA) studies, the visual incidence of projects depends essentially on their degree of visibility and perceptibility. The visibility is a spatial function depending on the elevation model of the study area. It is hereby evaluated using a simple viewshed algorithm which determines all pixels which are visible from one or more viewpoint pixels (the "target") situated on a surface (Patrono, 1995). The output is an updated DEM in which pixels not in view are

risen up, in order to “see” the target, and pixels in view are left as originally coded. Comparing the two DEMs (original and updated) it is possible to extract the viewshed.

16.3.1 Input data

The required input is:

- (a) The DEM of the area (corrections should be performed due to the presence of obstacles, e.g. tree height, etc.);
- (b) Distance map (it represents the distances for each pixel from the target pixel(s));
- (c) Mean height of the target (it represents the mean height of the target from the DEM surface);
- (d) Search distance (optional) (limits of intervisibility to be established *a priori*. It can be useful to reduce long computation time);
- (e) Target map (source map representing the target).

16.3.2 The algorithm

The algorithm is based on the comparison of the height of the target and the height of the neighbor interposed between the analyzed pixel and the target itself (Figure 16.1). For each pixel within the search distance, the following formula computes the required height, necessary to be visible from the target. It is calculated to update the DEM as shown in Fig. 16.2:

$$Q = \frac{(a - b) * d}{D} + b \quad [16.1]$$

where,

- Q required elevation to fall in the intervisibility or viewshed. This is the output to be calculated;
- b average elevation of the target plus its height;
- a elevation of the “obstacle” neighbors;
- d distance of the analyzed pixel from the target; the distance map is calculated from the target;
- D distance of the “obstacle” neighbors from the target; the distance map is the same as above.

Whenever the height of the considered pixel is smaller than required, the DEM is updated with the computed value. After crossing the original DEM and the updated one, it is possible to extrapolate the viewshed.

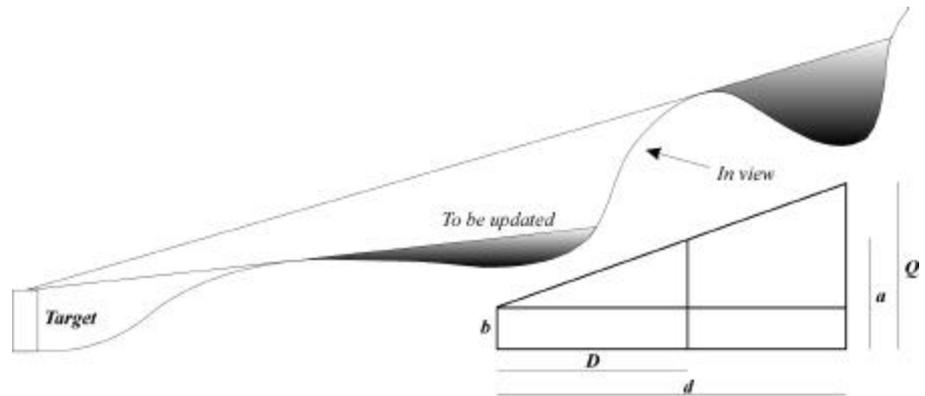


Figure 16.1: Elements involved in intervisibility analyses

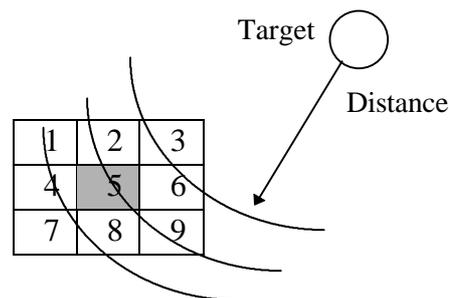


Figure 16.2: Example of neighborhood calculation. Distance is used to establish the obstacles

If the project is a road, for example, the target pixel may have more than one obstacle. This is not considered by the application of the algorithm because it works only with small source areas (e.g. it is impossible to perform the viewshed analysis simultaneously for more than one area upon a time). A possible solution is to cut the project in sub-units (the average height is also more reliable) with homogeneous characteristics. In this way, less obstacles and so less errors are expected.

It is important to mention that when obstacles such as a single tree line are close to the target or, in any case, in critical positions, they can strongly modify the viewshed. This cannot be detected by the presented method if the required information of these specific obstacles is not available (on land cover/land use maps, single tree lines are generally not mapped, which means no possibility to update the DEM *a priori*).

16.3.3 Computing the viewshed from a source landscape unit

The viewshed analysis hereby presented, determines all pixel visible from viewpoint pixels (target).



- Display the raster map `Dtm`.
- Display the map `Target` and write down the coordinates of the viewpoint.
- Open the `PixellInfo` window to check the height values in the surroundings of the target (use the map `Dtm`).
- Compute the distance map `Dist` from the source map `Target` using the `Distance` option (domain `Distance`, precision 1 and range 0-25000).
- Create a map called `Dist_up` with this formula:

```
Dist_up:=iff(Dist=0,1,Dist)↵
```

Accept all the defaults but give precision 1.

The updated map is calculated to avoid further problems with calculations (e.g., division by 0). Using the so calculated distance map, it is now possible to compute the “obstacle neighbor” maps.

The raster format does not allow to calculate exactly the neighbor pixel representing the obstacle due to the differences between the 4-connected and the diagonal neighbors. A good estimation can then be represented by a “mix” of two pixels, most likely a 4-connected and a diagonal one. Using the above distance map `Dist`, the pixels occupying the positions closest to the target are selected. They are supposed to be right in between the analyzed pixel and the target itself (see Fig. 16.2). The lower limit of 61 that you will find in the formulas (bigger than the distance between two diagonally connected pixels, smaller than two times the pixel size - here 30) is established to avoid division by 0 and because by default it is assumed that pixels in contact with the target “can see it”. An upper limit (optional) is included (time analysis increases according to increasing distance ranges).



- Compute the 4-connected obstacle map:

```
Min_4=nbminp4(Dist_up#)↵
```

- Compute the diagonal obstacle map:

```
Min_d=nbminp(Dist_up#)↵
```

- Compute the average distance (domain `Distance`) of the obstacle (`D` in equation 16.1):

```
Dis_neb=(Dist_up#[Min_4]+Dist_up#[Min_d])/2↵
```

Select domain Value with the same specifications as Dist_up.



- Using the map Dtm as the starting map for the Iteration programme, update it to Dtm_up. Perform 10 iterations and select the Propagation option:

```
iff(inrange(Dis_neb,61,5000),max(Dtm,
(((Dtm#[Min_4]+Dtm#[Min_d])/2-44)/
Dis_neb)*Dist_up+44)),Dtm)↓
```

Accept all defaults but give precision 0.1.

(Please, notice that this computation takes a long time).

In words: the procedure performs iterations with propagation until Dtm is not updated any more. If the pixel is included in the analyzed range the theoretical height value is computed according to Equation 16.1 (see also Fig. 16.1). If the original value is higher then no change occurs, otherwise Dtm is updated.

Note: - Instead of `inrange(Dis_neb,61,1500)` and/or 10 iterations, it is possible to increase or decrease the analysis with different ranges and/or different number of iterations.

- 44 is the total elevation of the target (the elevation is ~14 m and its height is ~30 m).



- Compute a new map Vsbst to display the viewshed Dtm_up, using the Pseudo representation as follows, modifying the color for the value 1 (into red):

```
Vsbst=iff((Dtm_up<>Dtm,1,Dtm)↓
```

Notice the “diagonal” effects of the obstacle-neighbor selection due to the raster representation. You may proceed as follows:

- Extract from Vsbst the areas with value 1.
- Apply the Majority filter and overlap it again on Dtm. Display the result using the same representation as above.



Change the height and/or the positions of the target and repeat the exercise.

The final viewshed is provided in the file Viewshd.mpr.

16.5 Ecological features analysis for mapping animal flows

Living things adapt to the surrounding environment by achieving an equilibrium with its different components. One of the problems generally encountered when analyzing the effects of landscape structure (independently if natural or anthropogenic) on animal communities is the fact that these communities or assemblages form a very intricate system of processes and relations which operate at different spatial and temporal scales. Another aspect of landscape structure relevant to animal dynamics is the spatial arrangement of different landscape elements, both man-made and natural. Not only distance is important for movement estimations, but also the extent to which certain landscape elements form a dispersal barrier (Forman, 1995; Knaapen *et al.*, 1992; Wiens, 1992).

In landscape ecology, in general, and in management and planning applications, in particular, the need for connectivity of habitat is evident. In landscape planning and EIA, there is a need to consider the effects of isolation of landscape elements. Attention should be given to preservation of such elements of connectivity and to constructing corridors and networks to overcome isolation effects caused by human intervention.

As for any other land-planning activity, all these analyses require as a major input spatial-georeferenced information which has to be modeled and mapped according to behavior, movement habits, frequency, habitat suitability, etc., of the analyzed species.

Reclassification, distance function and neighbor modelling represent a feasible tool/solution to estimate and outline areas of most probable organism flows.

Reclassification can be used to derive first, e.g. from a land use-cover map, landscape classes that affect the diffusion rate because of their related viscosity. The heterogeneity of the physical environment, e.g. edge areas, corridors, etc., also influence animal fluxes; the neighbor analysis functions allow to extract these and similar landscape features. Distance functions can be used then to simulate diffusion rates; edge maps, slope maps, reclassified land-use map, etc., can be combined in order to estimate appropriate resistance factors (e.g. edges have a positive influence while motorways represent obstacles). Once the distance map is performed from the pre-selected habitats using the mentioned resistance values, fluxes between them can be simulated and mapped using neighbor operators (Patrono, 1995).

In computing an animal flow, the propagation of an element can be estimated using neighbor analysis. One way to stress efficiency is to attribute a pixel to a flux only if it represent the fastest connection. For example, if in the 3x3 window, 3 of the 4-connected pixels are already part of a path it becomes unnecessary to include the central one.

The basis of this exercise is a land use/land cover map. The animal flow will be analyzed from the habitats represented by woods.



- Display the raster map `Lau_or`. Notice that the values correspond to:

<u>Use</u>	<u>Class</u>
Roads	1
Settlements	2
Industries	3
Railroads	4
Lakes	5
Rivers	6
Rivers (secondary)	7
Anthropogenic areas	8
Fields (Agric.)	9
Pastures and grasslands	10
Bushes and grasslands	11
Woods	12
Mixed	13

- Create a value map `Lau_orv` where: `Class 1 = 1, ..., Class 13 = 13`.

The second input required for the analysis is a distance map highlighting the different resistances of the landscape elements.

The first step is to prepare a source map considering in increasing order of resistance:

- (a) Edges of fields, woods, bushes, pastures and grasslands (corresponding to the classes 9, 12, 11 and 10 of `Lau_or`).



- Extract from map `Lau_orv` the edges (domain Value) of fields, woods etc., using a neighborhood operator. Call the result `Edg`:
`Edg=iff(inrange(Lau_orv,9,12), iff(nbcnt4
(Lau_orv#=Lau_orv)=4,0,Lau_orv),0))`

Note: 4-connectivity involves only the upper, left, right and lower neighbors of each pixel.

- (b) Riversides (all rivers), roadsides and lake shores (corresponding to the classes 6-7, 1 and 5 of `Lau_or`).



- Lake shores can be extracted as follows:
- Create a shore map (domain `Bit`) using the formula:

$$\text{Temp} = \text{iff}(\text{Lau_orv} = 5, 1, ?) \downarrow$$
- Dilate the shore map using the binary filter `Dilate4` to transform the map `Temp` into `Tempd`.
- Calculate the lake shores as:

$$\text{Lakes} := \text{Tempd} - \text{Temp} \downarrow$$
- Repeat the computation for river and road sides (you may overwrite the maps `Temp` and `Tempd` or create new maps). Save the results as `Roads` and `Rivers`. Calculate all of them and delete all the temporary maps.

- (c) Bush, pasture and grassland areas (corresponding to the classes 10 and 11 of `Lau_or`).



- Extract the bush, pasture and grassland areas from map `Lau_orv` (leaving the rest equal to 0) and call the result `Bpg` (domain `Value`).

- (d) The anthropogenic areas, fields (agriculture), mixed areas, rivers (secondary) (corresponding to the classes 8, 9, 13 and 7 of `Lau_or`).



- Extract them from map `Lau_orv` leaving the rest equal to 0, and call the result `Afmr` (domain `Value`).

Also barrier elements must be taken into consideration:

- (e) Roads, settlements, industries, railroads, lakes, rivers (not secondary) (corresponding to the classes 1, 2, 3, 4, 5 and 6 of `Lau_or`).



- Extract them from map `Lau_orv` leaving the rest equal to 0. Call the result `Barr` (domain `Value`).

The last calculation, involves the combination of all the previous maps (be sure all the landscape elements have been evaluated!!!!).



- Compute a weight map `W_map` (domain `Value`) combining the maps calculated in points (a) to (e) assigning the following resistance values (respect the priority):

<u>Map</u>	<u>Value</u>
Edg	1
Roads, Rivers, Lakes	2
Bpg	5
Afmr	10
Barr	-1

```
W_map=iff(Edg,1,iff((Roads)or(Rivers)or(Lakes),2,...
```

```
↵
```



- Create the source map `Woods` (Class 12) with domain `Class`, from `Lau_or`.

(Hint: the rest should remain undefined for the following distance calculation).

- Using the map `Woods` as a source and `W_map` as weight map, compute a distance map `Rdist` (domain `Distance` and precision 0.1).
- Display it using the `Pseudo` representation.

Now you can compute the possible flux paths as follows:



- Using `Rdist`, transform barrier areas into an impossible distance of 30000. Save results as `Rdistc`:

```
Rdistc=ifundef(Rdist,30000,Rdist)↵
```

Select a proper domain and range.

- Create a map `Woodsb` as follows:

```
Woodsb=ifundef(Woods,1,2)↵
```

- Calculate a map called `Path` using the `Iteration` programme. `Woodsb` is the starting map; select the **Until no changes** and the **Propagation** options:

```
iff(Woodsb=2,2,iff((nbcnt(Woodsb#=1)=9)or(Rdistc>1500)or(nbcnt4(Woodsb#=2)>2),1,iff(nbminp(Rdistc#,Woodsb#=1)=5,2,1)))↵
```

(Please, notice that this computation takes a long time).

In words: the algorithm performs iterations with propagation until `Woodsb` does not change any more. If `Woodsb` is woods, then leave 2. Otherwise check if all the neighboring elements of `Woodsb` equal 1 or the distance is > 1500 (notice that higher limits imply longer computing time) or the present pixel has more than two 4-connected “woods” pixels (it is not necessary for the animal to occupy the central pixel if at least 3 of the 4-connected are already occupied). If true write 1 else check if the central `Rdistc` pixel corresponds to the minimum distance of the “still not occupied” neighbors. If true write 2, else 1.



- Compute a `Final` path map assigning 1 to possible paths, 2 to original wood areas and 3 to the rest.

It can occur that meaningless path islands have been created without “real connections” in the 1500 m pre-selected range. At this point, we need to distinguish connected from isolated patches.



- Apply the `AreaNumbering` function (8-connected), creating a new domain, to the map `Woods` and call the output `An1`.
- Transform it in a `Value` map called `An1_v` where 1 is the undefined area of `An1`, 2 corresponds to `Class 1` of `An1`, 3 is `Class 2` of `An1`, etc.



- Calculate a new map `Prop` using the `Iteration` programme. Take the map `An1_v` as the starting map. Select the **Until no changes** and the **Propagation** options and apply the formula:

```
iff((Final=1)and(An1_v=1)and(nbcnt(An1_v#>1)>0),nbmax(An1_v#),An1_v)↵
```

Each flow pixel is attributed to its original wood patch.

(Please, notice that this computation takes a long time).

- Using MapCalc calculate a new map Conn where the “bridge pixels” (two different flows get in contact) are coded as 4. Apply the formula:

```
iff((Final=1)and(nbmax(Prop#)<>nbmin(Prop#,Prop#>1)),4,Final)↓
```

- Calculate a new map Propc using the program Iteration with propagation; select the Until no changes option. Use the map Conn as the starting map. Apply the formula:

```
iff((Conn=1)and(nbcnt(Conn#=4)>0),4,Conn)↓
```

-
- ! The connecting flows are highlighted propagating the “bridge pixels” (value 4 in Conn). Notice that within the same patch there can be connecting flows that are not coded as 4 being the patch the same. When calculating An1 with the Area Numbering if the 4-connected option is chosen the result is a higher number of single patches and so more real connections can be delineated
-

- ! Repeat the exercise using different ranges e.g. 1000 m and/or 2000 m instead of 1500 m.
-

References

- Burrough, P.A. (1986). *Principles of Geographical Information Systems for land resources assessment*. Clarendon Press. Oxford.
- Forman, R.T.T. (1995). *Land mosaics. The ecology of landscapes and regions*. Cambridge University Press. Cambridge.
- Knaapen, J.P, Scheffer, M. and Harms, B. (1992). Estimating habitat isolation in landscape planning. *Landscape and Urban Planning* 23, 1-16
- Patrono, A. (1995). *Theory and practice in landscape analysis for Environmental Impact Assessment (EIA)*. ESM-7 Lecture Notes, ITC. Enschede.
- Wiens, J.A. (1992). Ecological flows across landscape boundaries: a conceptual overview. In: A.J. Hansen and F. di Castri, (eds.), *Landscape Boundaries*, pp. 217-235. Springer-Verlag. New York.