7.4 Statistics

7.4.1 Histogram

Functionality

The Histogram operation calculates the histogram of a raster, polygon, segment or point map. Histograms list frequency information on the values, classes, or IDs in your map. Results are presented in a table and optionally in a graph. Summary information of a histogram of a raster map which uses a value or the Image domain, can be viewed in the properties of the histogram: mean, standard deviation, median, predominant, undefineds and percentage intervals.

Histograms of raster maps that are stored using 1 or 2 bytes per pixel are automatically calculated when needed. Thus, there is no need to perform the Histogram operation separately. You can use this operation to calculate histograms of raster maps stored with 4 or 8 bytes per pixel and to calculate histograms of polygon, segment or point maps.

A **raster histogram** lists the number of pixels, the percentages, and the areas per value, class or ID. For value maps (map which use a value, the image, the bit or a bool domain), also the cumulative number of pixels and cumulative percentages are calculated. Further,

- if the input map is an image, the percentages of the total number of pixels excluding pixels with value zero are always calculated;
- if the input map uses a bool or a value domain and if this map contains undefined values, the percentages of the total number of pixels excluding pixels with the undefined value are calculated;
- if the input map uses a class or ID domain, the percentages of the total number of pixels excluding pixels with the undefined value are always calculated.

A **polygon histogram** lists the number of polygons, the perimeter and area of polygons per class, ID, or value. If the input polygon map is a value map, also the cumulative number of polygons, the cumulative perimeter and cumulative area are calculated.

A **segment histogram** lists the number of segments and their length per class, ID or value. If the input segment map is a value map, also the cumulative number of segments and the cumulative length are calculated.

A **point histogram** lists the number of points per class, ID or value. If the input point map is a value map, also the cumulative number of points are calculated.

Histograms are displayed in a table window. To protect the results of the histogram calculation, none of the columns in a histogram are editable (columns are tableowned and read-only), but you can calculate new columns from the existing ones by typing Table Calculation formulas on the command line of the table window.

To show a raster histogram as a graph, open the histogram table, and in the table window, choose Show Graph from the Options menu. In the following dialog box, select the domain (or Value column) for the X-axis and Npix for the Y-axis, choose a 'step' or 'needle' graph.

Input map requirements
No special input requirements; a histogram can be calculated for any kind of raster, polygon, segment or point map.

Domain of output histogram table
Always, a dependent output histogram is created with the same name as the input map. The extension of a histogram is determined by the type of input map: .HIS for raster maps, .HSA for polygon maps, .HSS for segment maps and .HSP for point maps.

- When calculating the histogram of an image, the histogram also uses the Image domain: all values of the Image domain between 0 and 255 appear as records in the histogram table.
- When calculating the histogram of a map which uses the Bit domain, the histogram also uses the Bit domain: all values of the Bit domain (0 and 1) appear as records in the histogram table.
- When calculating the histogram of a map which uses a value domain, the histogram table obtains domain None. Occurring values in the input map appear in a column named Value, this column uses the same domain as the input map.
- When calculating the histogram of a map which uses a Bool domain, the histogram table obtains domain None. Occurring values in the input map (False and True) appear in a column named Value, this column uses the same domain as the input map.
- When calculating the histogram of a map which uses a class or ID domain, the histogram table uses the same domain as the input map.
- When calculating the histogram of a map which uses a Picture domain, then the histogram uses the same domain as the input map: the colors appear as (Red,Green,Blue) values as records in the histogram table.

A raster histogram contains the following columns

domain	- if this is a histogram of a map which uses the Image domain, then all values between 0 and 255 appear as records in the histogram table; - if this is a histogram of a map which uses the Bit domain, then values 0 and 1 appear as records in the histogram table; - if this is a histogram of a map which uses a class or ID domain, then the class names or IDs appear as records in the histogram;

	- if this is a histogram of a map which uses a Picture domain, then the colors as (Red,Green,Blue) values appear as records in the histogram. - if this is a histogram of a map with any other domain, record numbers appear.
`Value`	Column `Value` appears in histograms of maps which use a value domain, bool domain or the color domain: each occurring value in the map is listed.
`Npix`	The number of pixels with a certain value or meaning (pixel frequencies).
`NpixPct`	The relative amount of pixels with a certain value or meaning as a percentage of the total number of pixels in the map.
`PctNotZero`	Column `PctNotZero` appears in histograms of images: the relative amount of pixels with a certain value as a percentage of the total number of pixels in the map that do not have value 0.
`PctNotUnd`	Column `PctNotUnd` appears in histograms of raster maps which use a class, ID, bool or value domain and when the maps actually contain undefined pixels: the relative amount of pixels with a certain value or meaning as a percentage of the total number of pixels in the map that are not undefined.
`NpixCum`	For histograms of images and value maps: the cumulative amount of pixels of pixels with this value or a smaller value.
`NpCumPct`	For histograms of images and value maps: the relative cumulative amount of pixels with this value or a smaller value as a percentage of the total number of pixels in the map (to determine for instance the 1%, 5%, 95%, 99% intervals).
`Area`	The area of pixels with a certain value or meaning as: `npix`* pixel size * pixel size.

A polygon histogram contains the following columns

domain	- if this is a histogram of a domain class or ID map, then the class names or IDs appear as domain entries in the histogram table; - if this is a histogram of another domain value map, then record numbers appear as domain entries in the histogram table.
`Value`	Column `Value` only appears in histograms of maps which use a value domain or a bool domain: each occurring value in the map is listed.
`NrPol`	The number of polygons with a certain value or meaning.
`NPolCum`	For polygon maps which use a value domain: the cumulative number of polygons with this value or a smaller value.
`Perimeter`	The perimeter of polygons with a certain value or meaning.
`PerimeterCum`	For polygon maps which use a value domain: the cumulative perimeter of polygons with this value or a smaller value.
`Area`	The area of polygons with a certain value or meaning.

`AreaCum`	For polygon maps which use a value domain: the cumulative area of polygons with this value or a smaller value.

A segment histogram contains the following columns

domain	- if this is a histogram of a domain class or ID map, then the class names or IDs appear as domain entries in the histogram table; - if this is a histogram of another domain value map, then record numbers appear as domain entries in the histogram table.
`Value`	Column `Value` only appears in histograms of maps which use a value domain or a bool domain: each occurring value in the map is listed.
`NrSeg`	The number of segments with a certain value or meaning (segment frequencies).
`NrSegCum`	For segment maps which use a value domain: the cumulative number of segments with this value or a smaller value.
`Length`	The length of segments with a certain value or meaning.
`LengthCum`	For segment maps which use a value domain: the cumulative length of segments with this value or a smaller value.

A point histogram contains the following columns

domain	- if this is a histogram of a domain class or ID map, then the class names or IDs appear as domain entries in the histogram table; - if this is a histogram of another domain value map, then record numbers appear as domain entries in the histogram table.
`Value`	Column Value only appears in histograms of maps which use a value domain or a bool domain: each occurring value in the map is listed.
`NrPnt`	The number of points with a certain value or meaning (point frequencies).
`NrPntCum`	For point maps which use a value domain: the cumulative number of points with this value or a smaller value.

☞ In the Catalog, the following icons are used:
▨ for raster map histograms,
▨ for polygon map histograms,
▨ for segment map histograms,
▨ for point map histograms.

☞ For more information on how to show histograms in the Catalog, see topic How to customize the Catalog.

☞ In an existing attribute table, you can easily join for instance the Area column of a histogram, for more information refer to How to join areas from a histogram to an attribute table.

☞ To create a graph that shows several histograms, first join the necessary columns into one histogram or table.
☞ To calculate the lengths of segments over a certain area, use the Segment Density operation.
☞ To calculate the directions of segments, use the Segment Direction Histogram operation.
☞ To calculate the number of points over a certain area, use the Point Density operation.

Dialog box

The Histogram operation calculates the histogram of a raster, polygon, segment or point map. Histograms list frequency information on the values, classes, or IDs in your map. Results are presented in a table and optionally in a graph. Summary information of a histogram of a Value raster map can be viewed in the Properties dialog box of the histogram (mean, standard deviation, and percentage intervals).

Dialog box options

Input map:	Select an input raster, polygon, segment or point map. Open the list box and select the desired input map, or drag a raster, polygon, segment or point map directly from the Catalog into this box.
Show:	Select this check box if you want the histogram to be displayed in a table window when the operation has finished. Clear this check box if you do not want to see the histogram: you simply define how the histogram should be created.

The output histogram always obtains the same name as the input map. A dependent histogram is created.

☞ Histograms of raster maps that are stored using 1 or 2 bytes per pixel are automatically calculated when needed. Thus, for these kind of raster maps there is no need to perform the Histogram operation separately. You can use this operation to calculate histograms of raster maps stored with 4 or 8 bytes per pixel and to calculate histograms of polygon, segment or point maps.

Command line

The Histogram operation can be directly executed by typing the following expression on the command line of the Main window:

```
OUTTABLE = TableHistogram(InputMapName)

OUTTABLE = TableHistogramPol(InputPolMapName)

OUTTABLE = TableHistogramSeg(InputSegMapName)

OUTTABLE = TableHistogramPnt(InputPntMapName)
```

where:

OUTTABLE	**is not the user-defined name of a histogram**: the histogram table automatically obtains the same name as the input map.
`TableHistogram`	is the command to start the Histogram operation and calculate the histogram of a raster map.
`TableHistogramPol`	is the command to start the Histogram operation and calculate the histogram of a polygon map.
`TableHistogramSeg`	is the command to start the Histogram operation and calculate the histogram of a segment map.
`TableHistogramPnt`	is the command to start the Histogram operation and calculate the histogram of a point map.
InputMapName	is the name of your input raster map (any domain).
InputPolMapName	is the name of your input polygon map (any domain).
InputSegMapName	is the name of your input segment map (any domain).
InputPntMapName	is the name of your input point map (any domain).

You can only create histograms using the definition symbol =. The output histogram always obtains the same name as the input map and is a dependent object.

Algorithm

The Histogram operation calculates the histogram of a raster, polygon, segment or point map. Histograms list frequency information on the values, classes, or IDs in your map. Results are presented in a table and optionally in a graph. Summary information of a histogram of a Value raster map can be viewed in the Properties of the histogram (mean, standard deviation, and percentage intervals).

A **raster histogram** lists the number of pixels, the percentages and the areas per value, class or ID. If the input raster map uses a Value domain, also cumulative number of pixels and cumulative percentages are calculated.
A **polygon histogram** lists the number of polygons and the perimeter and area of polygons per class, ID, or value. If the input polygon map uses a Value domain, also the cumulative number of polygons, cumulative perimeters and cumulative areas are calculated.
A **segment histogram** lists the number of segments and their length per class, ID or value. If the input segment map uses a Value domain, also the cumulative number of segments and cumulative lengths are calculated.
A **point histogram** lists the number of points per class, ID or value. If the input point map uses a Value domain, also the cumulative number of points are calculated.

Raster map histograms are stored in .HIS and .HI#.
Polygon map histograms are stored in .HSA and .HA#.
Segment map histograms are stored in .HSS and .HS#.
Point map histograms are stored in .HSP and .HP#.

7.4.2 Autocorrelation - Semivariance

Functionality

Autocorrelation calculates the autocorrelation and the semivariance of a raster map. The autocorrelation of a raster map is generated by calculating the correlation between pixel values of a raster map and pixel values of the same raster map for different shifts (lags) in horizontal or vertical directions. The semivariance, a measure for the spatial variability of a raster map, is calculated for the same shifts. The operation is a tool to obtain information on horizontal and vertical patterns in a raster map.

Method
The user has to specify a maximum pixel shift, i.e. the maximum distance for which values of pixels should be compared to each other.

1. For each pixel, the value of that pixel (at position P) is compared with the value of the pixel one column to the right (position Q: row + 1) and one row below (position R: col + 1). Figure 1 below shows the collection of horizontal pairs of pixel values (PQ) for the first row of a map for shift 1;
   - for all collected values PQ , the horizontal correlation and semivariance are calculated; and
   - for all collected values PR, the vertical correlation and semivariance are calculated;
   - the calculated values are shown in the output table in record number 1 (shift 1).
2. Subsequently, for each pixel, the value of that pixel (at position P) is compared with the values of the pixel two rows to the right (position Q: row + 2) and two columns below (position R: col + 2). Figure 2 below shows the collection of horizontal pairs of pixel values for the first row of the map for shift 2;
   - again for all pairs PQ, the horizontal correlation and semivariance are calculated; and
   - for all pairs PR, the vertical correlation and semivariance are calculated;
   - the calculated values are shown in the output table in record number 2 (shift 2).
3. This continues until the specified maximum pixel shift is reached.

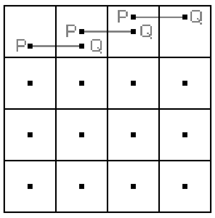


Fig. 1: Collection of horizontal pairs of pixel values PQ for the first row in a map for shift 1.

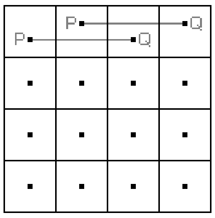
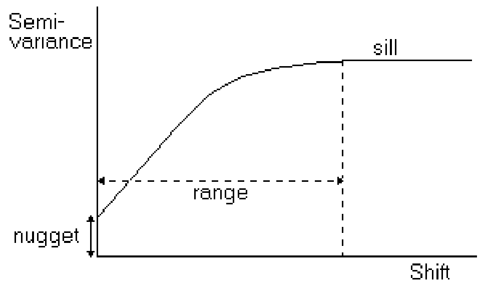


Fig 2.: Collection of horizontal pairs of pixel values PQ for the first row in a map for shift 2.

Autocorrelation and correlogram
The method is based on the assumption that a regional variable becomes random at a great distance. The autocorrelation is a value between -1 and +1. A horizontal correlation value of 1 means a perfect correlation, i.e. for a particular shift, the pixel values at P (column position P) and Q (column position P + shift) are identical. Variation is however rarely as regular as this. In most cases autocorrelation is close to 1 at short shifts, where the values of P and Q are similar. It decreases to 0 at larger shifts, where there is little relation. A plot of the autocorrelation against the shift is known as a correlogram.

Semivariance and semivariogram
The semivariance is a measure of variance. It is a straightforward way of measuring how a value changes between P and Q. In the calculation, it is assumed that the variance of differences only depends on the distance between P and Q. A plot of the semivariance against the shift is known as the semivariogram.

The figure below shows a typical semivariogram:



It can be seen from the figure that the curve rises from a small semivariance value to higher levels at larger shifts, where it levels off. This horizontal part is known as the sill. At these large values of the lag there is no spatial dependence between the data points. The estimates of variances of differences will be invariant with distance. The distance at which the semivariance approaches the sill is the range. The range is the important part of the variogram because it describes how inter-site differences are spatially dependent. A long range indicates high correlation, a small range indicates low correlation. Often the semivariance does not approach zero at the origin; it cuts the y-axis at a positive value of the semivariance. It can be seen as the residual, spatially uncorrelated noise, also called the nugget variance.

Example
The program can be used to detect correlation between pixels at regular horizontal or vertical distance intervals. In this way repetitions, or repeated patterns in a map, can be detected. An autocorrelation graph with regular peaks at certain interval distances indicates a cyclicity in the map. For instance, a map depicting the wave height at sea in a certain area will show an increased autocorrelation at a distance equal to the wave length, when looking in a direction parallel to the propagation direction of the waves.

Input map requirements
The input raster map needs to have the Image domain or a value domain.

Output table
An output table with domain None is created. The record numbers in the table represent the pixel shifts, i.e. the distance between P and Q; record number 1 represents a shift of 1 pixel, record number 2 represents a shift of 2 pixels, etc.

The table contains 4 columns:

- Column `HorCorr` shows correlation values for horizontal shifts, i.e. horizontal autocorrelation;
- Column `VertCorr` shows correlation values for vertical shifts, i.e. vertical autocorrelation;
- Column `HorVar` shows semivariance values for horizontal shifts, i.e. horizontal semi-variance;
- Column `VertVar` shows semivariance values for vertical shifts; i.e. vertical semi-variance.

☞ For better understanding it is useful to visualize autocorrelation and semivariance in a graph.

Dialog box

AutoCorrelation - Semivariance calculates the autocorrelation and the semivariance of a raster map. The autocorrelation of a raster map is generated by calculating the correlation between pixel values of a raster map and pixel values of the same raster map for different shifts (lags) in horizontal or vertical directions. The semivariance, a measure for the spatial variability of a raster map, is calculated for the same shifts.

Dialog box options

Input raster map:	Select an input raster map. Open the list box and select the desired input map, or drag a raster map directly from the Catalog into this box. The raster map need to be a value map.
Maximum pixel shift:	Type a value for the maximum pixel shift (in pixels). The minimum value which you can enter is 1 (shift of 1 pixel); the maximum value that you can enter is the number of rows or columns in your input raster map divided by 2 ('Map/2).
Output table:	Type a name for the output table that will contain the autocorrelation and the semivariance for shifts of 1 pixel, 2 pixels, ..., 'Map/2' pixels.
Show:	Select this check box if you the output table to be displayed when the operation has finished. Clear this check box if you do not want to see this table immediately: you simply define how the output table should be created.

Description:	Optionally, type a description for the output table. The description appears in the title bar when the output table is displayed.

A dependent output table is created.

Command line

The Autocorrelation - Semivariance operation can be directly executed by typing the following expression on the command line of the Main Window:

OUTTABLE = `TableAutoCorrSemiVar`(InputMapName, *max shift*)

where:

OUTTABLE	is the name of the output table.
`TableAutoCorrSemiVar`	is the command to start the Autocorrelation - Semivariance operation.
InputMapName	is the name of your input raster map (value map).
max shift	is the parameter to indicate the maximum pixel shift, measured in the number of pixels.

When the definition symbol = is used, a dependent output table is created; when the assignment symbol := is used, the dependency link is immediately broken after the output table has been calculated.

Algorithm

The AutoCorrelation - Semivariance operation calculates the autocorrelation and the semivariance of a raster map. The autocorrelation of a raster map is generated by calculating the correlation between pixel values of a raster map and pixel values of the same raster map for different shifts (lags) in horizontal or vertical directions. The semivariance, a measure for the spatial variability of a raster map, is calculated for the same shifts.

Autocorrelation is defined as:

$$AutoCorrelation = \frac{n\sum xy - \sum x \sum y}{\sqrt{\left(n\sum x^2 - \left(\sum x\right)^2\right) * \left(n\sum y^2 - \left(\sum y\right)^2\right)}}$$

where:

n	number of combinations
x	value of pixel in raster map
y	value of pixel in shifted raster map
Σx	sum of all x values
Σy	sum of all y values
Σxy	sum of all x * y values
Σx^2	sum of all x * x values
Σy^2	sum of all y * y values

Semivariance is defined as:

$$SemiVariance = \frac{\sum x^2 + \sum y^2 - 2\sum xy}{2m}$$

where:

Σxy	sum of all x * y values
Σx^2	sum of all x * x values
Σy^2	sum of all y * y values
m	total number of pixels in direction of shift (= n + shift)

Shift

The default value for the maximum pixel shift is 'Map/4', where 'Map' refers to the number of pixels in the direction of the shift. 'Map' is the number of columns of the input map for shifts in horizontal direction, and the number of rows for shifts in vertical direction. This means that the autocorrelation and semivariance will be calculated for shifts of 1 pixel, 2 pixels etc. up to a shift of 'Map/4' pixels. A smaller value than 'Map/4' will increase calculation speed.

The maximum pixel shift allowed is 'Map/2'. So, for example, for a map of 200 rows, the maximum shift in vertical direction is 200/2 = 100. If the shift would be larger than 'Map/2', i.e., more than half the map, the calculation cannot be performed for all pixels for the entire shift as for the pixels in row 'Map/2' + 1 there is no pixel to correlate this pixel with; that pixel would need to be in row 'Map/2' + 1 + 'Map/2' = 'Map' + 1; i.e. outside the map.

In this example of a map with 200 rows and columns, no other pixel can be found anymore for pixels in rows 191, 192, etc. for a pixel shift of 10 or more. For these pixels, pairs of pixel values cannot be collected; this means that the pixels in the last columns and the last lines of a map will not be taken into account.

7.4.3 Principal Component Analysis

Functionality

The Principal component analysis operation is mathematical method to uncover relationships among many variables and to reduce the amount of data needed to define the relationships. With principal component analysis each variable (input map) is transformed into a linear combination of orthogonal common components (output maps) with decreasing variation. The linear transformation assumes the components will explain all of the variance in each variable. Hence each component carries different information which is uncorrelated with other components.

Principal component analysis results in a linear transformation of a set of (satellite) raster maps into a set of output raster maps each explaining a common component in the input raster maps. The number of output raster maps is taken identical to the number of input raster maps to enable the user to determine the actual amount of reduction.

The output raster maps are listed in decreasing order of variance. This enables the reduction of maps because the last number of transformed maps have little or no variation left (may be virtually constant maps), do not add significance to the common components and may hence be discarded.

The input consists of a map list of raster maps which will be used to determine the common components. The operation uses the covariance matrix as a basis to calculate the output raster maps. Undefined values in the input raster maps are ignored.

The output of the principal component analysis operation is a map list containing transformed raster maps (components) and an output matrix denoting the transformation coefficients. The output raster maps are orthogonal. The number of output raster maps is taken identical to the number of input raster maps. The output raster maps are listed in decreasing order of variation (variance). The output matrix contains an extra line showing the variance of each component map (transformed raster map).

☞ Just as Principal Component Analysis uses the covariance matrix to calculate the map transformations, Factor analysis uses the correlation matrix (standardized variance-covariance matrix). The difference in approach is that Principal Component Analysis is more of a mathematical manipulation whilst Factor analysis is regarded as a statistical technique (Davis, 1986).

Principal components analysis can be used for several purposes, e.g.,:

- Data compression. For example you have 7 TM bands. Using principal components most of the variance in a seven band TM data set can be explained in two or three components.
- Pre-processing procedure prior to classification of the data.
- To find targets of interest, e.g., the component with the lowest variance may contain some interesting information (Mather, 1987).

Input requirements

The operation requires a map list with minimum of 2 raster maps (bands) and maximum 80 raster maps. All raster maps in the map list must use the Image domain or the same value domain and have the same georeference. Undefined values in the input raster maps are ignored.

Output objects of this operation

1. At first instance, the operation results in a matrix which contains the eigenvectors of the calculated covariance matrix; the total variance per band is also calculated.
2. When you open the matrix is (e.g. by double-clicking it in the Catalog), the matrix is calculated.
   Subsequently, all output raster maps are defined as linear combinations of the eigenvectors and the input maps. When the eigenvectors to construct the first component (OutMap1) are for instance: 0.43, 0.65, 0.63 (as the first row in the matrix); the first output map is calculated as 0.43*InMap1 + 0.65*InMap2 + 0.63*InMap3.

The number of output maps is the same as the number of input maps. The output maps obtain the same name as the matrix but the map names are followed by 1, 2, etc., where OutMap1 represents the first axis, i.e. explains the largest amount of variation of all input maps, OutMap2 the second orthogonal axis, etc.

3. Furthermore, the output maps are combined into an output map list; the output map list obtains the same name as the matrix.

Domain and georeference of output raster maps
The output raster maps use the system Value domain and have the same georeference as the input raster maps.

Dialog box

The Principal Component Analysis operation is a linear transformation of a set of (satellite) raster maps to uncover relationships among many variables and to reduce the amount of data needed to define such relationships. The input raster maps are transformed, i.e. recalculated based on the covariance matrix, into output raster maps called components.

Dialog box options

Input map list:	Select an input map list. Open the list box and select the desired map list, or drag a map list from the Catalog into the box. You can also click the create button to create a new map list.
Output matrix:	Type a name for the output matrix which will contain the transformation coefficients. The output map list will obtain the same name as the matrix; the output raster maps will obtain the name of the matrix followed by 1, 2, 3, etc.
Show:	Select this check box if you want the output raster maps to be calculated and the output matrix displayed when the operation has finished. Clear the check box if you do not want to see the matrix immediately: you simply define how matrix and thus the output raster maps should be calculated.
Description:	Optionally type a description for the matrix. The description appears in the title bar when the matrix is displayed.

A matrix is created. When the matrix is calculated, the output raster maps are created, and the output map list is created.

Command line

The Principal Component Analysis operation can be directly executed by typing the following expression on the command line of the Main Window:

MATRIX = `MatrixPrincComp`(InputMaplist)

where:

MATRIX	is the name of the output matrix; the output map list will obtain the same name; the output raster maps also use this name followed by 1, 2, 3, etc.
`MatrixPrincComp`	is the command to start the Principle components operation.
InputMaplist	is the name of the input map list; all raster maps in the input map list must use the Image domain or the same value domain and have the same georeference.

A matrix is created containing the Principal Components Analysis coefficients. In addition, a map list is created containing the resulting transformed raster maps (principal components).

Algorithm

The Principal component analysis operation results in a linear transformation of a set of (satellite) raster maps. The map values are transformed into raster map values, called components.

The following steps apply:

1. The covariance matrix of the input raster maps is computed.
2. The orientation of the components is computed based on the eigenvectors and eigenvalues of the covariance matrix.
3. The vectors in each column are normalized. The normalized values are the principal component analysis coefficients.

   With two input raster maps the principal components analysis coefficients matrix looks like:

   $$\begin{bmatrix} a_1, a_2 \\ b_1, b_2 \end{bmatrix}$$

   where:

a_1 and a_2	=	coefficients of the first component
b_1 and b_2	=	coefficients of the second component.

   With M bands an MxM matrix of principal component analysis coefficients is calculated.

4. The pixel values are transformed into the new raster maps (principal component):

   $$x = a_1 X + a_2 Y$$
   $$y = b_1 X + b_2 Y$$

   where:

x and y	=	spectral values in the first and second component
X and Y	=	spectral values in the two input raster maps

a_1 and a_2	=	coefficients of the first component
b_1 and b_2	=	coefficients of the second component.

A map list is created which stores these formulas to calculate the components.

Reference

- Davis, J.C. 1986. Statistics and data analysis in geology. John Wiley & Sons, Inc., New York.

7.4.4 Factor analysis

Functionality

The Factor analysis operation is used to uncover relationships among many variables and to reduce the amount of data needed to define the relationships. With factor analysis each variable is transformed into a linear combination of orthogonal common factors with decreasing variation. The linear transformation assumes the factors will explain all of the correlation in each variable. Hence each factor carries different information which is uncorrelated with other factors.

Factor analysis results in a linear transformation of a set of (satellite) raster maps into a set of output raster maps each explaining a common factor in the input raster maps. The number of output raster maps is taken identical to the number of input raster maps to enable the user to determine the actual amount of reduction. The output raster maps are listed in decreasing order of variance. This enables the reduction of maps because the last number of transformed maps have little or no variation left (may be virtually constant maps), do not add significance to the common factors and may hence be discarded.

The input consists of a map list of raster maps which will be used to determine the common factors. The operation uses the correlation matrix as a basis to calculate the output raster maps. Undefined values in the input raster maps are ignored.

The output of the Factor analysis operation is a map list containing transformed raster maps (factors) and an output matrix denoting the transformation coefficients. The output raster maps are orthogonal. The number of output raster maps is taken identical to the number of input raster maps. The output raster maps are listed in decreasing order of variation (variance). The output matrix contains an extra line showing the variance of each factor map (transformed raster map).

☞ Just as Factor analysis uses the correlation matrix (standardized variance-covariance matrix) to calculate transformed raster maps, Principal components analysis uses the variance-covariance matrix. The difference in approach is that Factor analysis is regarded as a statistical technique, whereas the Principal Component operation is more a mathematical manipulation (Davis, 1986).

Factor analysis can be used for several purposes, e.g.,:

- for separating derivatives within a single population. A typical example in geology is the discrimination between rock types based on chemical analysis of samples representing a magmatic differentiation sequence (Davis, 1973).
- as a pre-processing procedure prior to classification of data.

Input requirements
The operation requires a map list with a minimum 2 raster maps (bands) and maximum of 80 raster maps. All raster maps in the map list must use the Image domain or the same value domain and have the same georeference. Undefined values in the input raster maps are ignored.

Output objects of this operation

1. The operation results in a matrix which contains the eigenvectors of the calculated correlation matrix; the total variance per band is also calculated.
2. When you open the matrix is (e.g. by double-clicking it in the Catalog), the matrix is calculated.
   Subsequently, all output maps are defined as linear combinations of the factor loadings and the input maps. When the factor loadings to construct the first output map (OutMap1) are for instance: 0.43, 0.65, 0.63 (as the first row in the matrix); the first output map is calculated as 0.43*InMap1 + 0.65*InMap2 + 0.63*InMap3.
   The number of output maps is the same as the number of input maps. The output maps obtain the same name as the matrix but the map names are followed by 1, 2, etc., where OutMap1 represents the first factor, OutMap2 the second uncorrelated factor, etc.
3. Furthermore, the output maps are combined into an output map list; this map list obtains the same name as the matrix.

Domain and georeference of output raster maps
The output raster maps use the system Value domain and have the same georeference as the input raster maps.

Dialog box

The Factor analysis operation is a linear transformation of a set of (satellite) raster maps to uncover relationships among many variables and to reduce the amount of data needed to define such relationships. The pixel values of the input raster maps are transformed, i.e. recalculated based on the correlation matrix, into output raster maps called factors.

Dialog box options

Input map list:	Select an input map list. Open the list box and select the desired map list, or drag a map list from the Catalog into the box. You can also click the create button to create a new map list.
Output matrix:	Type a name for the output matrix which will contain the transformation coefficients.

	The output map list will obtain the same name as the matrix; the output raster maps will obtain the name of the matrix followed by 1, 2, 3, etc.
Show:	Select the check box if you want the output raster maps to be calculated and the output matrix displayed when the operation has finished. Clear the check box if you do not want to see the matrix immediately: you simply define how the matrix and thus the output maps should be calculated.
Description:	Optionally type a description for the matrix. The description appears in the title bar when the matrix is displayed.

A matrix is created. When the matrix is calculated, the output raster maps are created, and the output map list is created.

Command line

The Factor analysis operation can be directly executed by typing the following expression on the command line of the Main Window:

MATRIX = `MatrixFactorAnal`(InputMaplist)

where:

MATRIX	is the name of the output matrix; the output map list will obtain the same name; the output raster maps also use this name followed by 1, 2, 3, etc.
`MatrixFactorAnal`	is the command to start the Factor analysis operation.
InputMaplist	is the name of the input map list; all raster maps in the input map list must use domain Image or the same value domain and have the same georeference.

A matrix is created containing the Factor analysis coefficients. In addition, a map list is created containing the resulting transformed raster maps (factors).

Algorithm

The Factor analysis operation results in a linear transformation of a set of (satellite) raster maps. The map values are transformed into raster map values, called factors.

The following steps apply:

1. The correlation matrix of the input raster maps is computed.
2. The orientation of the factors is computed based on the eigenvectors and eigenvalues of the correlation matrix.
3. The vectors in each column are normalized. The normalized values are the factor analysis coefficients.

   Looking at two input raster maps the factor analysis coefficients matrix looks like:

   $$\begin{bmatrix} a_1, a_2 \\ b_1, b_2 \end{bmatrix}$$

where:

a_1 and a_2	=	coefficients of the first factor (factor loadings)
b_1 and b_2	=	coefficients of the second factor (factor loadings)

If M bands are used, an MxM matrix of factor analysis coefficients is computed.

4. The pixel values are transformed into the new raster maps (factors):

$$x = a_1 X + a_2 Y$$
$$y = b_1 X + b_2 Y$$

where:

x and y	=	spectral values in the first and second factor (output maps)
X and Y	=	spectral values in the two input raster maps
a_1 and a_2	=	coefficients of the first factor
b_1 and b_2	=	coefficients of the second factor.

A map list is created which stores these formulas to calculate the factors.

Reference

- Davis, J.C. 1986. Statistics and data analysis in geology. John Wiley & Sons, Inc., New York.

7.4.5 Variance-Covariance matrix

Functionality

The variance-covariance operation calculates variances and covariances of a number of raster maps in a map list. The variance is a means to express the variation of pixel values within a single raster map, i.e. a measure of the variation to the mean of the DN (Digital Number) values in a raster map. The covariance is a measure to express the variation of pixel values in two raster maps. It denotes the joint variation to the common mean of pixel values of the maps.

The resulting covariance matrix contains computed variances and covariances. Furthermore, the mean and standard deviation of each individual raster map is calculated and shown.

General information

The following shows how the covariance matrix is composed for three raster maps (band1, band2, band3):

	band1	**band2**	**band3**
band1	Var1	CoVar1-2	CoVar1-3
band2	CoVar2-1	Var2	CoVar2-3
band3	CoVar3-1	CoVar3-2	Var3

The diagonal elements (Var) contain the respective variances of the bands. Since covariance calculations are symmetric, the covariance matrix is also, e.g. CoVar1-2 is identical to Covar2-1.

Input requirements
The operation requires a map list in which all raster maps use the Image domain or the same value domain, and the same georeference. Pixels in the input maps that have the undefined value are ignored in the calculation.

Output matrix
After performing the operation, the covariance matrix, and the mean and standard deviation of the individual raster maps are shown on the screen. To show the matrix once more after closing this window, you have to perform the operation again. The values of the covariance matrix are stored by the object definition file of the map list (.MPL). When the map list contains a single raster map the resulting matrix will only contain the variance of the map.

Furthermore, after calculating a covariance matrix or a correlation matrix, the Properties dialog box of the input map list shows you the Optimum Index Factors, i.e. the combinations of three input maps with largest sum of standard deviations and smallest correlation. This can give you an indication which raster maps to use to create a color composite. For more information, see How to calculate Optimum Index Factor.

☞ As the output covariance matrix is not stored as a separate object, this operation cannot be executed from the command line.

Dialog box

The Variance-Covariance operation calculates variances and covariances of a number of raster maps in a map list. The variance is a means to express the variation of pixel values within a single raster map, i.e. a measure of the variation to the mean of the DN (Digital Number) values in a raster map. The covariance is a measure to express the variation of pixel values in two raster maps. It denotes the joint variation to the common mean of pixel values of the maps.

Dialog box options

Map list:	Select an input map list (open the list box and select the desired map list, or drag a map list from the Catalog into this list box) or create a new map list by clicking the create button. All raster maps in the map list must use the Image domain or the same value domain and must have the same georeference.

The covariance matrix is calculated and shown. The matrix is not stored as a separate object, but the matrix values are stored by the object definition file of the map list (.MPL). To show the matrix once more after you closed the matrix window, you have to perform the operation again.

Algorithm

Variances and covariances are computed for M bands in the map list. The result is presented as a M x M covariance matrix.

The values of the covariance matrix are calculated according to the following formulae:

$$VAR_B = \frac{\sum x_B{}^2}{N} - \frac{\left(\sum x_B\right)^2}{N}$$

where:

x_B	= pixel values in map B.
N	= the number of pixels which are not undefined.

and

$$COVAR_{B1,B2} = \frac{\sum x_{B1} * x_{B2}}{N} - \frac{\sum x_{B1}}{N} * \frac{\sum x_{B2}}{N}$$

where:

x_{B1}	= pixel values in map B1.
x_{B2}	= pixel values in map B2.
N	= the number of pixels which are not undefined.

7.4.6 Correlation matrix

Functionality

The operation calculates correlation coefficients of raster maps in a map list. Correlation coefficients characterize the distribution of pixel values in two raster maps. Furthermore, the mean and standard deviation of each individual raster map is calculated.

Analyzing satellite data the images often show a degree of correlation. This means that when spectral values in one band are high the values in another band are expected to be high as well. Plotting values from highly correlated bands in a feature space will result in an ellipsoid denoting that the two bands contain dependent information. From a set of highly correlated bands only one adds real value whilst the other ones may be derived or estimated. Calculating a correlation matrix helps to detect the redundancy and identifies possible reductions in the number of bands, e.g., to be used in a color composite.

Correlation coefficients are normalized covariance values. A correlation coefficient ranges from -1 to +1. Diagonal elements are always 1. A correlation close to +1 indicates a direct relationship between two bands. This suggests that if the reflectance (DN) of a pixel in one band is known, the reflectance of that pixel in the other band can be derived or estimated. A correlation close to -1 indicates an inverse relationship between the reflectance values of one band and the reflectance values in the other one.

Example
The following example shows a correlation matrix for 4 input bands. Bands 1 and 2 are highly correlated with a correlation coefficient of 0.97. Bands 1 and 4 seem independent with a correlation coefficient of -0.10.

	band 1	**band 2**	**band 3**	**band 4**
band 1	1.00	0.97	0.50	-0.10
band 2	0.97	1.00	0.43	-0.21
band 3	0.50	0.43	1.00	0.83
band 4	-0.10	-0.21	0.83	1.00

The high correlation between band 1 and band 2 suggests the removal of one of these bands; in a color composite use band 1 or band 2, and band 3 and band 4.

Input map requirements
The operation requires a map list in which all raster maps use the Image domain or the same value domain and the same georeference. Pixels in the input maps that have the undefined value are ignored in the calculation.

Output matrix
After performing the operation, the correlation matrix, and the mean and standard deviation of each individual raster map are shown on the screen. To show the matrix once more after closing this window, you have to perform the operation again. The values of the correlation matrix are stored by the object definition file of the map list (.MPL).

Furthermore, after calculating a correlation matrix or a covariance matrix, the Properties dialog box of the input map list shows you the Optimum Index Factors, i.e. the combinations of three input maps with largest sum of standard deviations and smallest correlation. This can give you an indication which raster maps to use to create a color composite. For more information, see How to calculate Optimum Index Factor.

☞ As the output correlation matrix is not stored as a separate object, this operation cannot be executed from the command line.

Dialog box

The Correlation matrix operation calculates correlation coefficients of the raster maps in a map list. Correlation coefficients characterize the distribution of pixel values in two raster maps. Furthermore, the mean and standard deviation of each individual raster map is calculated.

Dialog box options

Map list:	Select an input map list (open the list box and select the desired map list, or drag a map list from the Catalog into this list box) or create a new map list by clicking the create button. All raster maps in the map list must use the Image domain or the same value domain and must have the same georeference.

The correlation matrix is calculated and shown. The matrix is not stored as a separate object, but the matrix values are stored by the object definition file of the map list (.MPL). To show the matrix once more after you have the matrix window, you have to perform the operation again.

Algorithm

The Correlation matrix operation calculates correlation coefficients of input raster maps of a map list. Undefined values in the maps are ignored.

The correlation matrix is calculated in two steps. In the first step, the covariance matrix of the raster maps in the map list is calculated. In the second step, the elements of the covariance matrix are normalized according to the following formula:

$$CORR_{B1,B2} = \frac{COVAR_{B1,B2}}{\sqrt{VAR_{B1} * VAR_{B2}}}$$

where:

$COVAR_{B1,B2}$	= the covariance computed of two input raster maps (B1, B2).
VAR_{B1}	= the variance in the first input raster map (B1).
VAR_{B2}	= the variance in the second input raster map (B2).

7.4.7 Neighbour polygons

Functionality

The Neighbour polygons operation finds adjacent (or neighbouring) polygons in a polygon map and calculates the length of the boundaries of adjacent polygons.

Input map requirements
The input map needs to be a polygon map with a class, ID, or group domain.

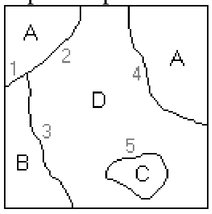
Output table
An output table with domain None is created.

The output table has 3 columns:

- The first two columns list the class names, IDs, or group names of neighbouring, i.e. adjacent, polygons. In the table, the combinations of neighbouring polygons is shown twice. It means that, for example, all boundaries between polygons A and B are shown not only as PolA-PolB but also as PolB-PolA;
- Column Length lists the lengths of the summed boundaries between these polygons.

For example, an area with certain crops needs to be protected against rabbits. A fence is needed at the boundary of this area. It is therefore interesting to calculate the boundary of the area. The output of the Neighbour polygon operation is an output table which shows the length of the boundaries between polygons.

Example
Input map:



Output table:

Polname 1	Polname 2	Length	(segments)
A	B	100	1
A	D	1150	2 + 4
B	D	650	3
C	D	750	5

In this example, you can see that for class A, there are 2 polygons. Note that there are 2 polygon boundaries between the polygons of class A and class D. In the output table, the lengths of these two segment boundaries are summed.

Dialog box

The Neighbour polygons operation finds adjacent (or neighbouring) polygons in a polygon map and calculates the length of the boundaries of adjacent polygons.

Dialog box options

Input polygon map:	Select an input polygon map. Open the list box and select the desired input map, or drag a polygon map directly from the Catalog into this box. The polygon map needs to use a class, ID or group domain.
Output table:	Type a name for the output table that will contain the lengths of the boundaries of neighbouring polygons.
Show:	Select this check box if you want the output table to be displayed when the operation has finished. Clear this box if you do not want to see this table immediately: you simply define how the output table should be created.
Description:	Optionally, type a description for the output table. The description appears in the title bar when the output table is displayed.

A dependent output table is created.

Command line

The Neighbour polygons operation can be directly executed by typing the following expression on the command line of the Main window:

```
OUTTABLE =  TableNeighbourPol(InputPolMapName)
```

where:

OUTTABLE	is the name of the output table.
`TableNeighbourPol`	is the command to start the Neighbour polygons operation.
InputPolMapName	is the name of your input polygon map.

When the definition symbol = is used, a dependent output table is created; when the assignment symbol := is used, the dependency link is immediately broken after the output table has been calculated.

Algorithm

The Neighbour polygons operation finds adjacent (or neighbouring) polygons in a polygon map and calculates the length of the boundaries of adjacent polygons.

Steps

- The program scans the polygon map polygon by polygon.
- For each polygon the boundary is checked for neighbouring polygons. The lengths of the boundaries of a polygon with its neighbouring polygons are calculated.
- All polygon boundaries, for example all boundaries between polygons A and B, are summed.
- These values are presented in an output table. The polygon names, in the first two columns of the table, are arranged in alphabetic order.

7.4.8 Segment direction histogram

Functionality

The Segment direction histogram operation calculates directions and lengths within segments, i.e. between all stored coordinate pairs of the segments. The output is a table with directions from 0 to 179° and the length and number of the segment parts in that direction.

This operation can be used to analyze the pattern of lines (segments) in a segment map. To get information on, for instance, the structure and lithology of rocks, the direction, length, and frequencies of features such as faulting and jointing can be calculated.

Example
When segments only consist of begin and end nodes without any intermediate points (see Fig. 1), directions and lengths will be calculated for the complete segments.

Usually however segments consist of a begin and end node and multiple intermediate points (see Fig. 2). Directions and lengths will then be calculated between all stored coordinate pairs in the segments, i.e. for all segments from the begin node to the first intermediate point, from the first intermediate point to the second, etc., until the end node.

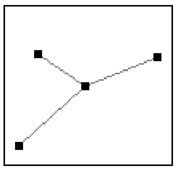


Fig. 1: Segments without intermediate points.

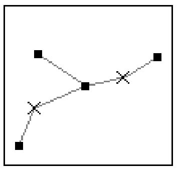


Fig. 2: Segments with intermediate points.

Input map requirements
No special requirements for the input segment map.

Output table
An output table with domain None is created.

The output table has three columns:

- Column Direction shows the compass direction in degrees from 0 (north-south trending) to 179°,
- Column Length shows the total length of all segment parts in that direction; and
- Column NrSeg (frequency) shows the total number of the segment parts in that direction.

☞ For a better understanding of the pattern of segments, you can display graphically the output data as a rose diagram.

☞ To reduce the number of intermediate points within segments of a segment map, you can use the Tunnel segments operation.

☞ The operation calculates direction, length and frequency for all segments in the map. Therefore, the input segment map has to contain only those segments of which directions are of interest; the map should not contain segments which show, for example, a frame or boundary of an area. To select the proper segments, you can use the Mask segments operation prior to the Segment direction histogram operation.

☞ The operation is only useful when your segments are rather straight lines. As this operation calculates directions and lengths between all stored coordinate pairs of segments, it is generally not very practical to use this operation on segment maps which store for instance roads or rivers; these features often contain too many curves.

Dialog box

The Segment direction histogram operation calculates directions and lengths within segments, i.e. between all stored coordinate pairs of the segments. The output is a table with directions from 0 to 179° and the length and number of the segment parts in that direction.

Dialog box options

Input segment map:	Select an input segment map. Open the list box and select the desired input map, or drag a segment map directly from the Catalog into this box.
Output table:	Type a name for the output table that will contain the direction, length and frequency of the segments in the input map.
Show:	Select the check box if you want the output table to be displayed when the operation has finished. Clear this box if you do not want to see this table immediately: you simply define how the output table should be created.
Description:	Optionally, type a description for the output table. The description appears in the title bar when the output table is displayed.

A dependent output table is created.

Command line

The Segment direction histogram operation can be directly executed by typing the following expression on the command line of the Main window:

OUTTABLE = `TableSegDir`(InputSegMapName)

where:

OUTTABLE	is the name of the output table.
`TableSegDir`	is the command to start the Segment direction histogram operation.
InputSegMapName	is the name of your input segment map.

When the definition symbol = is used, a dependent output table is created; when the assignment symbol := is used, the dependency link is immediately broken after the output table has been calculated.

Algorithm

The Segment direction histogram operation calculates directions and lengths within segments, i.e. between all stored coordinate pairs of the segments. The output is a table with directions from 0 to 179° and the length and number of the segment parts in that direction.

Steps

1. The program scans the map segment by segment.
2. Within each segment, i.e. between all stored coordinate pairs in a segment, directions are calculated. This means that not the total direction of a complete segment is calculated but the direction from the begin node to the first intermediate point, from the first intermediate point to the second, etc., until the end node. Figure 1 and Figure 2 below show segments with and without intermediate points.

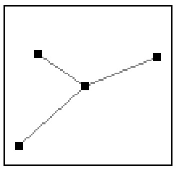


Fig. 1: Segments without intermediate points.

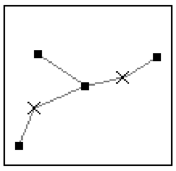


Fig. 2: Segments with intermediate points.

3. The compass directions are sorted from 0 to 179°.
4. Then, for each direction:
   - when the segments in the map have no intermediate points at all, the total lengths of segments in that direction and the number of segments in that direction are calculated;
   - when the segments do have intermediate points, the total length of the segment parts in that direction and the frequency of the segment parts in that direction are calculated.

7.4.9 Point statistics

Point statistics may help to get an impression of the nature of your point data prior to for instance a point interpolation.

For point statistics a point map is required in which:

- the points themselves are values (domain Value point map), for instance concentration values, or
- the points have an identifier (domain Identifier point map) and values are stored in a column of an attribute table.

Output of point statistics are tables. Graphs can be created from the tables.

Spatial correlation: calculates autocorrelation (as Moran's I) and semivariance (as Geary's c) for point values in a point map. The user is encouraged to compare his or her data set with a data set consisting of the same point locations, with a set of attribute values, approximately in the same range as the measured variable, but created at random (using one of the RND functions in Table Calculation). If the graphs are very much the same for the measured data and the random data, no autocorrelation exists between the data points. Hence, point interpolation is not useful.

7.4.10 Spatial correlation

Functionality

Point statistics may help to get an impression of the nature of your point data, for instance prior to a point interpolation.

Spatial correlation measures dependence among nearby values in a spatial distribution. Variables may be correlated because they are affected by similar processes, or phenomena, that extend over a larger region. Odland (1988, p.7) mentions that spatial autocorrelation 'exists whenever a variable exhibits a regular pattern over space in which values at a certain set of locations depend on values of the same variable at other locations'.

For example, if the concentration of a certain pollutant is very high at a certain location, it will most likely also be high in the direct surroundings. In other words, the concentration is autocorrelated at small distances. At larger distances, it is less likely that the concentration will be equally high. The correlation will probably be lower, and the variance higher.

Distance intervals between points are calculated, and autocorrelation (as Moran's I) and semivariance (as Geary's c) are calculated for points that are more or less at the same distances towards each other.

The user is encouraged to compare his or her data set with a data set consisting of the same point locations, with a set of attribute values, approximately in the same range as the measured variable, but created at random (using one of the RND functions in Table Calculation). If the graphs are very much the same for the measured data and the random data, no autocorrelation exists between the data points. Hence, point interpolation is not useful.

Input map requirements
The input point map should be a value map.

Output table
An output table with domain `None` is created.

The output table contains 4 columns:

- Column `Distance` lists distance intervals;
- Column `NrPairs` lists the number or point pairs found within each distance interval;
- Column `Correlation` lists the correlation for these point pairs within a distance interval;
- Column `Variance` lists the semivariance for these point pairs within a distance interval.

Dialog box

Spatial correlation calculates autocorrelation (as Moran's I) and (semi) variance (as Geary's c) for point values in a point map. Point statistics may help to get an impression of the nature of your point data, for instance prior to a point interpolation.

Dialog box options

Input point map:	Select an input point map. Open the list box and select the desired input map, or drag a point map directly from the Catalog into this box. You can select a point map with a value domain, or a point map with a class or ID domain which has a linked attribute table with values.
Column:	In case you selected an input point map with a class or ID domain, select an attribute column (value domain) from the attribute table.
Output table:	Type a name for the output table that will contain the autocorrelation and semi- variance.
Show:	Select this check box if you want the output table to be displayed in a table window when the operation has finished. Clear this check box if you do not want to see this table immediately: you simply define how the output table should be created.
Description:	Optionally, type a description for the output table. The description appears in the title bar when the table is displayed.

A dependent output table is created.

Command line

The Spatial correlation operation be invoked by typing the following expression on the command line of the Main Window:

```
OUTTABLE = TableSpatCorr(InputPointMap)
```

where:

OUTTABLE	is the name of your output spatial correlation table.
`TableSpatCorr`	is the command to start the Spatial correlation operation.
InputPointMap	is the name of your input point map (value map).

When the definition symbol = is used, a dependent output table is created; when the assignment symbol := is used, the dependency link is immediately broken after the output table has been calculated.

Algorithm

First, distances between all points are calculated. Distance groups are created for point pairs that are more or less at the same distance to each other. Then, within every distance group, the (auto) correlation and (semi) variance of points is calculated.

In ILWIS, spatial autocorrelation between points is calculated as Moran's I (Odland):

$$I = \frac{n}{\Sigma\Sigma w_{ij}} \frac{\Sigma\Sigma w_{ij}(x_i - \bar{x})(x_j - \bar{x})}{\Sigma(x_i - \bar{x})^2}$$

In ILWIS, semivariance is calculated as Geary's c (Odland):

$$c = \frac{n-1}{2\Sigma\Sigma w_{ij}} \frac{\Sigma\Sigma w_{ij}(x_i - x_j)^2}{\Sigma(x_i - \bar{x})^2}$$

Reference

– Odland, J. 1988. Spatial autocorrelation. In: G.I. Thrall (Ed.), Sage University Scientific Geography Series no. 9. Sage Publications, Beverly Hills.

7.4.11 Pattern analysis

Functionality

Data in a point map represent spatial occurrence of a particular phenomenon. To acquire knowledge about the occurrence of the phenomenon, the spatial distribution of the points in the map can be examined. Point pattern analysis is a technique that is used to obtain information about the arrangement of point data in space, to be able to make a statement about the occurrence of certain patterns.

Three types of patterns can be distinguished (see Figure 1):

- In a situation of *complete spatial randomness* (CSR), no correlation exists between locations of points.
- In a *clustered pattern*, subgroups of points tend to be significantly closer to each other than to other subgroups of points.
- In a *regular pattern*, distances between adjacent points tend to be further apart than for CSR. The pattern under consideration is compared to the situation of CSR. If point items tend to attract each other, and there is environmental heterogeneity, one can call the pattern 'grouped'. On the other hand, if individual point items tend to repel each other, and points are more spread out than in CSR, one may call the pattern 'regular'.

There are two basically two different techniques; the location of points can be studied:

- with reference to the area, i.e. *measures of dispersion*, or
- with respect to each other, i.e. *measures of arrangement*.

Measures of arrangement
Measures of arrangement are techniques that examine characteristics of the locations of points relative to other points in the pattern. In this technique measured frequencies of occurrences of *reflexive nearest neighbours* (RNN) are compared with expected frequencies of occurrences in a situation of CSR. The CSR is simulated for the same area and the same number of points. Two points are considered first order RNN if they are each other's nearest neighbour. This definition can be extended to higher orders; second order RNNs are points that are each others second-nearest neighbours etc. The frequencies are calculated for RNNs of first to sixth order.

Boots & Getis (1988, p. 69) remark that 'most researchers suggest that more higher order values in excess of CSR expectations indicate measure of regularity in arrangement of points, whereas lower empirical values imply elements of grouping in the pattern'. If more (first order) RNNs occur than is expected in CSR, it can be concluded that isolated and relatively uniformly arranged couples exist.

Measures of dispersion
Measures of dispersion are techniques that examine characteristics of the locations of points with respect to the area e.g. the pattern is analyzed by calculating distances between individual points, and comparing these with the distances that would be found in a CSR. In this technique the mean distance between RNNs are tested against the expected distances in CSR. If the individual points are closer than they would be for CSR, this indicates a clustered pattern. If, on the other hand, individual points are further apart than they would be in CSR, a more regular pattern is assumed.

Input map requirements
An input point map is required with at least two valid points.

Output table
An output table with domain None is created.

The output table contains the following information:

Measures of dispersion
In the output table, in which the mean distance between RNNs are tested against the expected distances in a (simulated) CRS, eight columns are stored;

- Column `Distance` lists distances. These distances should be considered as the distance from any point in the input point map, i.e. a kind of search radius;
- Column `Prob1Pnt` lists the probability that within a certain distance (column `Distance`) of any point, one other point will be found;
- Column `Prob2Pnt` lists the probability that, within that distance, from any point, two other points will be found;
- The same goes for columns `Prob3Pnt`, `Prob4Pnt`, `Prob5Pnt` and `Prob6Pnt`;
- For a dataset of *n* points, `ProbAllPnt` is the summation of `Prob1Pnt + Prob2Pnt + .. + Probn-1Pnt`, divided by (*n* - 1).

Figure 1 shows the different behaviour of `Prob1Pnt` and `ProbAllPnt` for three fundamental types of patterns. Significant features that can be read from the graphs include the following (all values relate only to this example):

- For the `Prob1Pnt` curves, the distance for which `Prob1Pnt` becomes 1 is the maximum distance between first order RNN (random: ± 600, clustered: ± 200, regular: ± 300). For the regular pattern this means for instance that each point has (at least 1) first order RNN within a distance of ± 300. The fact that the curve is vertical at ± 300 means that this is valid for all points, i.e. all points have a first order RNN at the same distance of 300.
- For the `ProbAll` curves, the curve becomes flat at a distance equal to the cluster cross section (for clustered pattern: ± 600). Note the steps in the curve for the regular pattern.
- Due to boundary effects, flattening occurs for a distance equal to the area side (for all patterns: ± 2000). Therefore it is useful to include in your research also points outside the area to overcome these boundary effects.

☞ To study the differences between the three datasets (the original one and the two special ones), graphs can be drawn: choose Show Graph from the Options menu in the table window; display `Distance` against `Probn`, like in Figure 1.

Measures of arrangement
The Properties dialog box of the output table contains additional information (available in the on pattern analysis is displayed in two tables, both with columns Order, Observed value and Assumed with CSR.

The first table lists the order of the reflexive nearest neighbour (nearest, second nearest, etc.), the observed number of times such a pair is found in the dataset, and the number of times this would happen in the situation of CSR. If the number of pairs found is much larger than in CSR, this indicates that the points are not randomly distributed. Instead they cluster. If the number is smaller, the points are distributed regularly.

The second table lists the mean distance to the reflexive nearest neighbour for every order, and the mean distance in a CSR. If the mean distance for the data set is much smaller than the one for CSR, the points tend to cluster. If the distance is larger, the distribution is more regular.

Advantages/disadvantages of dispersion vs. arrangement
The advantages of measurements of arrangement are:

1. The technique is density free, i.e. no estimates have to be made, for example the expected number of points per quadrant;
2. The technique is free of edge effects, because one studies the arrangement of points with respect to each other is studied rather than with respect to the area.

The advantages of the dispersion technique are:

1. It is more rigorous than the arrangement technique, because it is more sensitive to certain differences in some pattern characteristics. For arrangement, identical values may sometimes be expected for patterns that are different in some way;
2. The statistical theory for dispersion is better developed, so that this method is less subjective.

Note that higher order RNN analysis (refined analysis) might give more information than first order analysis alone; take the example of couples on a dance floor; the distance to the first order RNN will be significantly less than in CSR, while higher order RNN distances will be higher.

☞ In the output table, the columns ProbnPnt cannot be interpreted easily without more information. The user is encouraged to create other datasets (with an equal amount of points, in about the same area) in which the points are randomly distributed, e.g. by using for instance the RND() function in TabCalc, and regularly distributed.

Dialog box

Point pattern analysis is a technique that is used to obtain information about the arrangement of point data in space, to be able to make a statement about the occurrence of certain patterns.

Dialog box options

Input point map:	Select an input point map. Open the list box and select the desired input map, or drag a point map directly from the Catalog into this box. The point map needs to have at least two valid points.
Output Table:	Type a name for the output table that will contain columns with probabilities of finding more points for different distance intervals.
Show:	Select this check box if you want the table to be displayed when the operation has finished. Clear this check box if you do not want to see this table immediately: you simply define how the table should be created.
Description:	Optionally, type a description for the table. The description appears in the title bar when the table is displayed.

A dependent output table is created.

Command line

The Pattern analysis operation can be directly executed by typing the following expression on the command line of the Main window:

```
OUTTABLE = TablePattAnal(InputPointMapName)
```

where:

OUTTABLE	is the name of your output table.
`TablePattAnal`	is the command to start the Pattern analysis operation.
InputPointMapName	is the name of your input point map.

When the definition symbol = is used, a dependent output table is created; when the assignment symbol := is used, the dependency link is immediately broken after the output table has been calculated.

Algorithm

Data in a point map represent spatial occurrence of a particular phenomenon. To acquire knowledge about the occurrence of the phenomenon, the spatial distribution of the points in the map can be examined. Point pattern analysis is a technique that is used to obtain information about the arrangement of point data in space to be able to make a statement about the occurrence of certain patterns.

Measures of dispersion
The mean distance between RNNs are tested against the expected distances in a CRS.

First a number of different distance figures are defined. Then from distance 0 to the upper limit of each distance figure, the probability of finding one or more other points is calculated for output columns `Prob1Pnt, Prob2Pnt, ..., Prob6Pnt` where:

- Column `Distance` lists distances. These distances should be considered as the distance from any point in the input point map, i.e. a kind of search radius;
- Column `Prob1Pnt` lists the probability that within a certain distance (column Distance) of any point, one other point will be found;
- Column `Prob2Pnt` lists the probability that from any point, two other points will be found;
- The same goes for columns `Prob3Pnt, Prob4Pnt, Prob5Pnt` and `Prob6Pnt`;
- For a dataset of *n* points, ProbAllPnt is the summation of `Prob1Pnt + Prob2Pnt + ... + Probn-1Pnt`, divided by (*n* - 1).

The formula to calculate probabilities within certain distances reads:

$$ProbAllPnt = \frac{\sum_{i=1}^{n-1} P_i}{n-1}$$

Measures of arrangement
For every order the expected number of reflexive neighbours is calculated. Multiplying the probability that a point in a CSR pattern is a reflexive neighbour by the number of points gives you the expected number. Note that a reflexive point must be a member of a pair, so the observed value will always be an even number.

To calculate the mean distance to nearest neighbour, the nearest neighbour distance (d) for each of the points is summed and divided by the number of points. The formula reads:

$$d = \frac{\Sigma d_i}{n}$$

where:

d_i	is the nearest neighbour distance for a point in the map
n	is number of points.

Note that the mean distance to nearest neighbour is not corrected for boundary effects.

Reference

– Boots, B.N. & A. Getis, 1988, Point Pattern Analysis. Sage University Scientific Geography Series no. 8. Sage Publications, Bevery Hills.

7.5 Interpolation

7.5.1 Densify

Functionality

The Densify operation allows you to reduce the pixel size of a raster map, i.e. the number of rows and columns is increased in the output map. When the input map is a value map, new values can be assigned to the pixels in the output map by means of nearest neighbour, or a bilinear or bicubic interpolation. When the input map is a domain class or ID map, only the nearest neighbour method is available.

Densify should be used after you performed a point interpolation. Furthermore, you can use Densify before printing raster maps with the Image domain or a value domain, to improve the quality of the printed maps.

Example
When an input raster map has 400 rows by 400 columns with a pixel size of 25m, and Densify is applied with an enlargement factor 2.5,
the resulting output raster map has 1000 rows and 1000 columns and a pixel size of 10m.

Mind that a byte map of 1000 rows and 1000 columns requires 1 MB disk space, while a map of 400 rows and 400 columns merely requires 160 kB disk space. The effect of using an enlargement factor of 2.5 is that the original map size increases by the square of the enlargement factor.

Nearest Neighbour
For each output pixel, the class name, ID or value of the nearest pixel in the input map is assigned.

Bilinear versus bicubic interpolation
A bilinear interpolation takes much less time than a bicubic interpolation.
A bilinear interpolation results in discontinuity when taking the first derivative of a densify output map. A bicubic interpolation remains continuous up to the second derivative.

This means that when you obtained a value raster map from a point interpolation, the point interpolation results can be improved with the Densify operation:

- Densify with a bilinear interpolation gives a quick results;
- Densify with a bicubic interpolation is advised if after Densify, you intend to apply first derivative filters like DFDX, DFDY, or SHADOW on the densify output map.

Input map requirements
For nearest neighbour, the input raster map can have any domain. For a bilinear or bicubic interpolation, the input raster map should be a value map. The input raster map must have a georeference which is not georeference None.

Domain and georeference of output map
The output map uses the same value domain as the input map. The value range and precision can be adjusted for the output map.

The operation always creates a new georeference for the output map. The output georeference obtains the same name as the output map and is calculated from the input georeference and the specified enlargement factor.

Dialog box

The Densify operation allows you to reduce the pixel size of a map. The user is asked to enter an enlargement factor which determines the number of rows and columns in the output map. For example, when using an enlargement factor of 2.5 on an input map consisting of 400 lines and 400 columns (e.g. pixel size of 25m); the output map will contain 1000 lines by 1000 columns (resulting in a pixel size of 10m).

Thus the number of rows and columns in the map is increased and, as the position of each pixel in the output map is known, values are assigned to these pixels by means of nearest neighbour, or a bilinear or bicubic interpolation using values from the input map.

Dialog box options

Input raster map:	Select an input raster map to be densified. Open the list box and select an input map, or directly drag a raster map from the Catalog into this box.
Densify factor:	Enter a value by which the number of rows and columns of the input map should be multiplied (real value >= 1).
Interpolation method:	Select an interpolation method which is used to determine the pixel values in the output map.
◉ Nearest Neighbour:	To assign to each output pixel the value of the nearest pixel in the input map; this option is available for input maps with any domain.
◉ Bilinear:	To calculate for each output pixel a value from 4 surrounding pixels in the input map; this option is available for value maps.
◉ Bicubic:	To calculate for each output pixel a value from 16 surrounding pixels in the input map; this option is available for value maps.
Output raster map:	Type a name for the output raster map.
Value range:	Accept the default value range, or specify your own range of possible values in the output map.

Precision:	Accept the default precision of output values, or specify your own precision.
Show:	Select this check box if you want the output map to be displayed in a map window when the operation has finished. Clear this check box if you do not want to see this map immediately: you simply define how the output map should be created.
Description:	Optionally, type a description for the output map. The description appears in the title bar when the output map is displayed.

A dependent output map is created. Further, a new georeference is created for the output map; this georeference obtains the same name as the output map.

Restriction
For a Bilinear or a Bicubic interpolation, the input raster map should be a value map.

Command line

The Densify operation can be directly executed by typing the following expression on the command line of the Main Window:

OUTMAP = `MapDensify`(InputMapName, *factor*, `NearestNeighbour|Bilinear| Bicubic`)

where:

OUTMAP	is the name of your output raster map.
`MapDensify`	is the command to start the Densify operation.
InputMapName	is the name of your input raster map.
factor	is the enlargement factor (real value > 1).
`NearestNeighbour|Bilinear|Bicubic`	is the parameter which indicates to assign output values from the nearest pixels in the input map or to calculate new values by a bilinear interpolation or a bicubic interpolation.

When the definition symbol = is used, a dependent output map is created; when the assignment symbol := is used, the dependency link is immediately broken after the output map has been calculated. Further, a georeference factor is created with the same name as the output map.

Algorithm

The Densify process consists of several steps:

- from the specified enlargement factor and the georeference of the input map, a new georeference factor is created; this georeference determines the number of rows and columns in the output map; thus the XY-coordinate is known for each output pixel;

- then these positions are looked up in the input map and, according to the selected interpolation method, the nearest pixel, or 4 (bilinear), 16 (bicubic) input pixels around this position are used to calculate a value for the output map.

For a detailed description of the nearest neighbour method, bilinear and bicubic interpolation, refer to Resample : algorithm.

With a nearest neighbour interpolation, the pixels of the output map are assigned the same class names, IDs or values as the pixels of the input map.
A bilinear interpolation takes much less time than a bicubic interpolation.
A bilinear interpolation results in discontinuity of the first derivative. This means that a bilinear interpolation gives artefacts when you for example want to use a DFDX, DFDY, or a SHADOW filter on the map later on.

A bicubic interpolation remains continuous up to the second derivative.

7.5.2 Contour interpolation

Functionality

Contour interpolation is an operation which first rasterizes contour lines of a segment map with a value domain, and then calculates values for pixels that are not covered by segments by means of a linear interpolation. Furthermore, on the command line, this operation can also use an input raster map.

When using Contour interpolation on a segment map containing height (contour) information, the resulting raster map is a Digital Elevation Model (DEM). To visualize a DEM you can use Display3D.

Input map requirements
The input segment map should be a value map. The Contour interpolation operation creates an intermediate raster map with rasterized segments. For all pixel values with undefined values in this intermediate map, an output value is calculated.

When using Contour interpolation from the command line, also a raster map (with rasterized segments) can be used as input.

Domain and georeference of output map
The output map uses the same value domain as the input map. The value range and precision can be adjusted for the output map.
The georeference for the output map has to be selected or created; you can usually select an existing georeference corners.

☞ At the borders of the area in which interpolation takes place, artefacts may occur. Therefore make sure that the area of interest is smaller than the area in which interpolation takes place. After the contour interpolation, you can perform a map calculation with another raster map which contains your study area by using a mask.

- ☞ The contours should touch the boundary of the map, otherwise discontinuities will appear.
- ☞ Several megabytes of temporary disk space are required to perform the interpolation.
- ☞ The quality of the obtained Digital Elevation Model will not only depend on the density of digitized contour lines or on the precision you choose for the domain of the output map but also on the selected pixel size of the georeference for the output raster map. The smaller the precision and the smaller the pixel size, the better your DEM will be.
  However choosing a small precision will result in a raster map that is stored using 4 or 8 bytes per pixel, while choosing a small pixel size will result in a raster map with more pixels. Choosing a small precision or a small pixel size thus result in a larger output raster map on disk.
- ☞ An example of how output values are interpolated from the contour lines is presented in Contour interpolation : algorithm.

Dialog box

Contour interpolation is an operation which first rasterizes contour lines of a segment map with a value domain, and then calculates values for pixels that are not covered by segments by means of a linear interpolation. Furthermore, on the command line, this operation can also use an input raster map.

When using Contour interpolation on a segment map containing height (contour) information, the resulting raster map is a Digital Elevation Model (DEM).

Dialog box options

Contour map:	Select an input segment map with a value domain.
Output raster map:	Type a name for the output raster map.
Georeference:	Select a georeference for the output raster map or create a new georeference by clicking the little create button ☼.
Value range:	Accept the default value range, or specify your own range of possible values in the output map.
Precision:	Accept the default precision for the output values, or specify your own precision.
Show:	Select this check box if you want the output map to be displayed in a map window when the operation has finished. Clear this check box if you do not want to see this map immediately: you simply define how the output map should be created.
Description:	Optionally, type a description for the output map. The description appears in the title bar when the output map is displayed.

A dependent output map is created.

Command line

Contour interpolation can be directly executed by typing one of the following expressions on the command line of the Main window:

OUTMAP = `MapInterpolContour`(InputSegmentMap, Georeference)

OUTMAP = `MapInterpolContour`(InputRasterMap)

where:

OUTMAP	is the name of your output raster map.
`MapInterpolContour`	is the command to start the Contour interpolation operation.
InputSegmentMap	is the name of your input segment map (value domain).
InputRasterMap	is the name of an input raster map (value domain).
Georeference	is the name of an existing georeference that should be used for the output raster map.

When the definition symbol = is used, a dependent output map is created; when the assignment symbol := is used, the dependency link is immediately broken after the output map has been calculated.

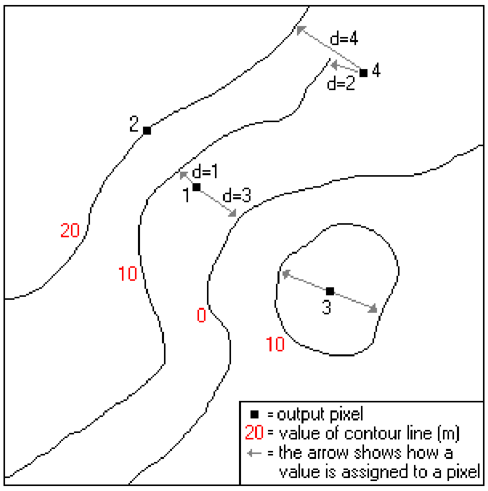
Algorithm

Contour interpolation first rasterizes contour lines in the segment map (see also Segments to Raster). This results in values for all pixels that are located on the segments.

Then values have to be calculated for pixels that fall in between the segments. For each undefined pixel, the distance is calculated towards the two nearest contour lines (see also Distance calculation). The distances are calculated forwards and backwards, until no more changes occur. Then a linear interpolation is performed using the two distance values. This returns the value for the undefined pixel.

Example
The pixel values in this example are calculated as follows (see figure below):

1. The value for output pixel no. 1 is determined by the contour lines with value 0m and value 10m. The distance to these contour lines is respectively 1 and 3 and therefore the pixel value will be ((3*10)+(1*0))/4 = 7.5m. This interpolation matches reality.
2. Pixel no. 2 is situated on a contour line with value 20, and therefore obtains this value. This interpolation also matches reality.
3. The value for output pixel no. 3 is determined by the contour line with value 10m, and therefore the pixel value will also be 10m. When this area indeed represents a flat area, the interpolation matches reality. However, when the area represents a peak or a valley, this interpolation is wrong. You can then obtain a correct interpretation by adding a small segment representing the highest point of the top or the lowest point of the valley to the segment map.


d=4
4
d=2
2
d=1
1
d=3
20
10
0
3
10
■ = output pixel
20 = value of contour line (m)
← = the arrow shows how a value is assigned to a pixel

4. The value for output pixel no. 4 is determined by the contour lines with value 10m and value 20m. The distance to these contour lines is respectively 2 and 4 and therefore the pixel value will be ((2*20)+(4*10))/6 = 13.33m. This is incorrect, because the output pixel value is most likely to lay within the values 0m and 10m. To avoid these kind of artefacts, extend the contour line (segment) with value 10 to the border of the map. The output pixel value will now be determined by the contour lines with value 0m and 10m, and the interpolation will be better.

Reference

– Gorte, B.G.H. and Koolhoven W., 1990. Interpolation between isolines based on the Borgefors distance transform. ITC Journal 1990-3, pp. 245-247. ITC, Enschede.

7.5.3 Point interpolation

A point interpolation performs an interpolation on randomly distributed point values and returns regularly distributed point values. This is also known as gridding. In ILWIS, the output values are raster values.

In an ILWIS point interpolation, the input map is a point map where:

- the points themselves are values (domain Value point map), for instance concentration values, or
- the points have an identifier (domain Identifier point map) and values are stored in a column of an attribute table.

The output of a point interpolation is a raster map. For each pixel in the output map, a value is calculated by an interpolation on input point values.

Point interpolation methods

- Nearest point: assigns to pixels the value, identifier or class name of the nearest point. This method is also called Nearest Neighbour or Thiessen; this method does not require a domain Value point map.
- Moving average: assigns to pixels weighted averaged point values. For each pixel in the output map, weight factors for all points are determined by a weight function. Weights may for instance approximately equal the inverse distance. The weight function is implemented in such a way that points outside a user-defined 'limiting' distance obtain weight zero. This speeds up the subsequent weighted averaging.
- Trend surface: calculates pixel values by fitting a surface through all point values in the map. The surface may be of the first order up to the sixth order. A trend surface may give a general impression of the data. Surface fitting is performed by a least squares fit. It might be a good idea to subtract the outcome of a trend surface from the original data.
- Moving surface: calculates pixel values by fitting a surface through weighted point values. Weights for all points are calculated by a weight function. Weights may for instance approximately equal the inverse distance. Points outside a user-specified limiting distance obtain weight zero by the weight function. This speeds up the subsequent surface fitting. Surface fitting is performed by a least squares fit.

☞ When interpolating point values, it is for time efficiency reasons, strongly advised to choose a rather large pixel size for the output map. Further interpolation on the raster map values can be performed using the Densify operation.

☞ When using a moving average or a moving surface point interpolation, it is important to choose a limiting distance that contains enough points. Rule of thumb: when you perform the calculation again with a limiting distance increased by a factor 2, and you find profound differences in outcomes, you chose the limiting distance too small in the first calculation.

☞ You can use the Spatial correlation operation to get some insight in your point data (before interpolating).

7.5.4 Nearest point

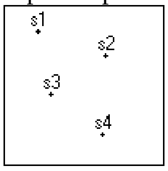
Functionality

The nearest point operation requires a point map as input and returns a raster map as output. Each pixel in the output map is assigned the class name, identifier, or value of the nearest point.

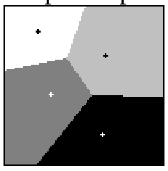
For example

Schools, hospitals, water wells, etc. can be represented by points. The output of a nearest point operation on such a point map gives the 'service area' of the schools, hospitals or water wells, based on the shortest distance (as the crow flies) of a point and pixels.

Input map:



Output map:



This operation is also known as Nearest neighbour or Thiessen Map; if every pixel in the output map is accessible, this operation offers a quick way to obtain a Thiessen map from point data.

Input map requirements
No special input map requirements. Input maps may be of domain type class, ID, or value or a class/ID map with an attribute table.

Domain and georeference of output map
The output raster map uses the same domain as the input point map or uses the domain of the attribute column.
The georeference for the output map has to be selected or created; you can usually select an existing georeference corners.

Dialog box

The nearest point operation requires a point map as input and returns a raster map as output. Each pixel in the output map is assigned the class name, identifier or value of the nearest point. This operation is also known as Nearest neighbour or Thiessen Map. It offers a quick way to obtain a Thiessen map from point data.

Input point map:	Select an input point map. Open the list box and select the desired input map, or drag a point map directly from the Catalog into this box.
Attribute:	Select this check box if you want to select an attribute column from an attribute table which is linked to the input point map (class or ID map). Clear this check box to use the input point map.
Output raster map:	Type a name for the output raster map that will contain, for each pixel, the class name, identifier, or value of the closest point.
Georeference:	Select the name of an existing georeference or create a new georeference (click the create button).
Value range:	In case the output map uses a value domain, accept the default value range, or specify your own range of possible values in the output map.
Precision:	In case the output map uses a value domain, accept the default precision of output values, or specify your own precision.

Show:	Select this check box if you want the output map to be displayed in a map window when the operation has finished. Clear this check box if you do not want to see this map immediately: you simply define how the output map should be created.
Description:	Optionally, type a description for the output map. The description appears in the title bar when the output map is displayed.

A dependent output map is created.

Command line

The Nearest Point operation can be directly executed by typing the following expression on the command line of the Main window:

```
OUTMAP = MapNearestPoint(InputPointMapName,Georeference)
OUTMAP = MapNearestPoint(PointMap.Column,Georeference)
```

where:

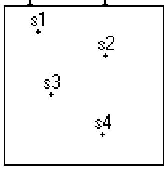
OUTMAP	is the name of the output raster map.
`MapNearestPoint`	is the command to start the Nearest Point operation.
InputPointMapName	is the name of the input point map.
PointMap.Column	is the name of the input point map (class or ID) and the attribute column in the attribute table linked to the map.
Georeference	is the name of an existing georeference that should be used for the output raster map.

When the definition symbol = is used, a dependent output map is created; when the assignment symbol := is used, the dependency link is immediately broken after the output map has been calculated.

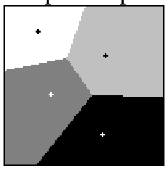
Algorithm

To each pixel in the output map, the class name, identifier or value of the closest input point is assigned.

Input map:



Output map:



7.5.5 Moving average

Functionality

The moving average operation is a point interpolation which performs a weighted averaging on point values and returns a raster map as output. In short, the output value for a pixel is calculated as the sum of the products of weights and point values, divided by sum of weights.
Weight values are calculated in such a way that points close to an output pixel obtain large weights and points further away obtain small weights. Thus, the values of points close to an output pixel are of greater importance to the output pixel value than the values of points that are further away.
The user has to specify the method to calculate weight values. There are two options: inverse distance and linear. For more information on weight calculation methods, see Moving average : algorithm.
Further, the user has to specify a limiting distance. Points that are further away from an output pixel than the limiting distance, obtain weight zero by the weight calculation and thus have no influence on the output value for that pixel. This speeds up the calculation of the output pixel value.

Input map requirements
The input point map should be a value map, or a point map with a class or ID domain which has an attribute table with a value column.

Domain and georeference of output map
The output raster map uses the same value domain as the input point map. The value range and precision can be adjusted for the value domain of the output map.
The georeference for the output map has to be selected or created; you can usually select an existing georeference corners.

☞ For time efficiency reasons, it is advised to choose a rather large pixel size for the output raster map. Further interpolation on the raster values can be performed with the Densify operation.
☞ Make sure that there are enough points within the limiting distance; in other words, choose the limiting distance value large enough. Rule of thumb: when you perform the calculation again with a limiting distance increased by a factor 2, and you find profound differences in outcomes, you chose the limiting distance too small in the first calculation.
☞ You can use the Spatial Correlation operation to get some insight in your point data (before interpolating).

Dialog box

Moving average is a point interpolation which performs a weighted averaging on point values and returns a raster map as output. In short, the output value for a pixel is calculated as the sum of the products of weights and point values, divided by sum of weights.

Dialog box options

Input point map:	Select an input point map. Open the list box and select the desired input map, or drag a point map directly from the Catalog into this box. You can select a point map with a value domain, or a point map with a class or ID domain which has a linked attribute table with values.
Attribute:	In case you selected an input point map with a class or ID domain, select an attribute column (value domain) from the attribute table.
Weight function:	Select the weight function to calculate weights for the points
Inverse distance:	$(1 / d^n) - 1$
Linear decrease:	$1 - d^n$
	d = distance of point towards output pixel divided by the limiting distance; n = weight exponent
Weight exponent:	Type a value for *n* (usually 1 or 2) as used in the inverse distance or linear decrease weight function.
Limiting distance:	Type a value for the limiting distance. Points that are further away from an output pixel than the limiting distance are assigned weight zero and thus have no influence on the output value for that pixel.
Output raster map:	Type a name for the output raster map that will contain the weighted averaged point values.
Georeference:	Select the name of an existing georeference or create a new georeference.
Value range:	Accept the default value range, or specify your own range of possible values in the output map.
Precision:	Accept the default precision of output values, or specify your own precision.
Show:	Select this check box if you want the output map to be directly displayed.
Description:	Type a description for the output map. The description appears in the title bar when the output map is displayed.

Command line

The Moving Average operation can be directly executed by typing one of the following expressions on the command line of the Main window:

OUTMAP = `MapMovingAverage`(InputPointMapName, Georeference, *WeightFunction*)

OUTMAP = `MapMovingAverage`(PointMap.Column, Georeference, *WeightFunction*)

where:

OUTMAP	is the name of the output raster map.
`MapMovingAverage`	is the command to start the Moving Average operation.
InputPointMapName	is the name of the input point map with a value domain.
PointMap.Column	is the name of an input point map with a class or ID domain which has a linked attribute table, and the name of a value column in this attribute table.
Georeference	is the name of an existing georeference for the output raster map.
WeightFunction	is an expressing to define the type of weight function to use (see below).

The *WeightFunction* can be defined as:

`InvDist`(*Exp*,*LimDist*)

or

`Linear`(*Exp*,*LimDist*)

where:

`InvDist`	is the option for the inverse distance method.
`Linear`	is the option for the linear method.
Exp	is a value for the weight exponent (usually 1 or 2).
LimDist	is a value for the limiting distance: points that are further away from an output pixel than the limiting distance obtain weight zero.

When the definition symbol = is used, a dependent output map is created; when the assignment symbol := is used, the dependency link is immediately broken after the output map has been calculated.

A full expression thus might look like:

```
OUTMAP= MapMovingAverage(RegionX,RegionX,InvDist(1,500))
```

Algorithm

Moving average performs a weighted averaging on point values and returns a raster map as output.

1. First for each output pixel, the distances of all points towards this output pixel are calculated to determine weight factors:
   - If the distance of a point towards the output pixel is larger than the user-specified limiting distance, weight zero is assigned. This speeds up the weighted averaging of step 2.
   - If the distance of a point towards an output pixel is smaller than the limiting distance, weights are calculated by a weight function. Two weight functions are available: inverse distance and linear.

   Inverse distance: weight = $(1/d^n) - 1$
   Linear: weight = $1-d^n$

   d = distance of a point towards output pixel divided by the limiting distance
   n = weight exponent

   These weight functions ensure that points close to an output pixel obtain large weights and points further away obtain small weights. Graphs of the weight functions are presented in a next topic. These graphs show the manner in which weight values decrease when distance increases, for different values of n.

2. Then, for each output pixel, the sum of the products of weights and point values, divided by sum of weights returns the desired output pixel value.

   output value = $\Sigma (w_i * val_i) / \Sigma w_i$

   w_i = weight for point i
   val_i = point value of point i

☞ For time efficiency reasons, it is strongly advised to choose a rather large pixel size for the output raster map. Further interpolation on the raster values can be performed with the Densify operation.

☞ The larger you choose the limiting distance value, the longer the operation will take because more point values are used to calculate an output pixel value.

☞ Be sure that you take the limiting distance large enough. Rule of thumb: when you perform the calculation again with a limiting distance increased by a factor 2, and you find profound differences in outcomes, you chose the limiting distance too small in the first calculation.

Weight functions
In the operations *Moving average* and *Moving surface*,you have to select a weight function which calculates weights for the points. Two weight functions are available: inverse distance and linear.

The actual functions are determined by two parameters:

- *Limiting distance* (D_0)
  This parameter determines the maximum search radius for the interpolation. For every pixel in the output map the program will search for existing input points within the search radius. Points which are outside the search radius will have no influence on the computed value of the current output pixel. Selection of a relatively small limiting distance will result in a number of available points which is approaching the mathematical minimum necessary to apply the operation. This will result in strange irregular surfaces, and may result in discontinuities. Especially with large maps, selection of a very large limiting distance will cause long computing time with hardly any significant influence on the resulting surface.
- *Weight exponent* (n)
  This parameter determines the influence of the distance on the weight.

Selection of *weight function* should be determined upon:

- *Point measurement accuracy*
  If measurement accuracy is very high and the local variations, over a pixel, is small, an Inverse distance function may be selected. In selecting the Inverse distance function, you are ensured that the computed surface will coincide with the point measurements.
- *Local variation magnitude*
  In maps with measurement errors or very close measured points with different results, the Linear distance function should be selected. This might decrease the overall error, by correcting erroneous measurements with other close points. The consequence is that the computed surface will not necessarily coincide with the measured points.

Figures 1 and 2 below show in which way weight values decrease with increasing distance for different values of n.

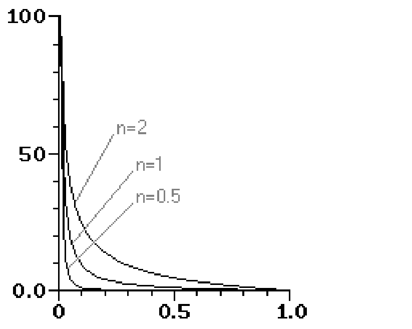


Fig. 1: Inverse distance weights = $(1/d^n) - 1$

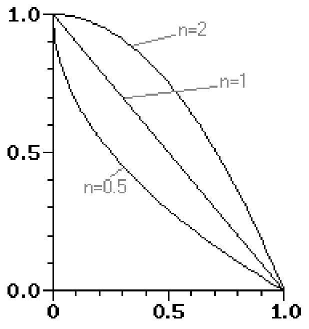


Fig. 2: Linear weights = $1 - d^n$

The X-axes in Figures 1 and 2 represent **d** : the distance of a point towards an output pixel divided by the limiting distance. Value 1 indicates the limiting distance. After the limiting distance, weight values are zero.

The Y-axes in Figures 1 and 2 represent the calculated **weight values** for different values of **n**.

7.5.6 Trend surface

Functionality

The Trend Surface operation is a point interpolation which requires a point map as input and returns a raster map as output. One polynomial surface is calculated by a least square method approaching all point values in the map. The calculated surface values are assigned to the output pixels.
The trend surfaces in this operation range from a simple plane to a comprehensive polynomial surfaces. There is no rule which can be used to determine which surface should be used in the calculation. In general, a low order surface will be most appropriate, because they are less sensitive to extreme values.

Requirements for the input point map
The input point map should have a value domain or, if the input map has a class or ID domain then the column of the attribute table must have a domain Value.

Domain and georeference of output raster map
The output raster map uses the same value domain as the input point map. The value range and precision can be adjusted for the value domain of the output map. The georeference for the output map has to be selected or created; you can usually select an existing georeference corners.

☞ The Additional Info button in the properties form of the output raster map can be pressed to display the formula of the surface.
☞ The calculated trend surface can be subtracted from the original map. If the residuals of this subtraction do not have a strong spatial correlation, the calculated trend surface can be seen as good description of the distribution of the points.
☞ Further interpolation on raster map values can be performed using the Densify operation.
☞ You can use the Spatial Correlation operation to get some insight in your point data (before interpolating).

Dialog box

The Trend Surface operation is a point interpolation which requires a point map as input and returns a raster map as output. One polynomial surface is calculated by a least square fit approaching all point values in the map. The calculated surface values are assigned to the output pixels.

Dialog box options

Input point map:	Select the name of the input point map. Open the list box and select the desired input map, or drag a point map directly from the Catalog into this box. No special input map requirements.
Trend Surface	Select one of the 8 trend surfaces.
Output Raster Map:	Type the name of the output raster map.

Georeference:	Type the name of an existing georeference or create a new georeference.
Show:	Select this check box if you want the output map to be displayed in a map window when the operation has finished. Clear this check box if you do not want to see this map immediately: you simply define how the output map should be created.
Value range:	Accept the default value range, or specify your own range of possible values in the output map (a liberal estimate will do).
Precision:	Accept the default precision of output values, or specify your own precision.
Description:	Optionally, type a description for the output map. The description appears in the title bar when the output map is displayed.

A dependent output map is created.

Command line

The Trend Surface operation can be directly executed by typing the following expression on the command line of the Main window:

OUTMAP = `MapTrendSurface`(InputPointMap, GeoReference, *SurfaceType*)

where:

OUTMAP	is the name of your output raster map.
`MapTrendSurface`	is the command to start the Trend Surface operation.
InputPointMap	is the name of the input point map.
GeoReference	is the name of an existing georeference that should be used for the output raster map.
SurfaceType	is the name of the surface: `Plane` | `Linear2` | `Parabolic2` | `3` | `4` | `5` | `6`

When the definition symbol = is used, a dependent output map is created; when the assignment symbol := is used, a so-called source map is created.

Algorithm

Trend Surface performs a trend surface interpolation on point values and returns a raster map. The operation determines the best fitting surface by minimizing the least square error:

$$\text{total error} = \sum_{i=1}^{n} \left(f(x_i) - m(x_i) \right)^2 \qquad x_i = \binom{x}{y}$$

in which $f(x_i)$ is the calculated value of the surface on a certain location and $m(x_i)$ is the measured value on that same location.

The value of the trend surface determines the value of the pixel in the output raster map.

☞ For time efficiency reasons, it is strongly advised to choose a rather large pixel size for the output raster map. Further interpolation on the raster values can be performed with the Densify operation.

7.5.7 Moving surface

Functionality

The Moving Surface operation is a point interpolation which requires a point map as input and returns a raster map as output. For each output pixel a polynomial surface is calculated by a least square method approaching all point values. As points closer to an output pixel are more important than points further away, a weight function has to be specified. Points that are further away from an output pixel than the user-specified limiting distance, are assigned weight zero.

The surfaces in this operation range from a simple plane to a comprehensive polynomial surfaces. There is no rule which can be used to determine which surface should be used in the calculation. In general, a low order surface will be most appropriate, because they are less sensitive to extreme values.

The user has to specify the method to calculate weight values. There are two options: inverse distance and linear decrease. For more information on weight calculation methods, see Point Interpolation - Moving Surface : algorithm.

Requirements for the input point map
The input point map should have a value domain or, if the input map has a class or ID domain then the column of the attribute table must have a domain Value.

Domain and georeference of output raster map
The output raster map uses the same value domain as the input point map. The value range and precision can be adjusted for the value domain of the output map. The georeference for the output map has to be selected or created; you can usually select an existing georeference corners.

☞ When interpolating point values, it is for time efficiency reasons strongly advised to choose a rather large pixel size for the output map.
☞ Further interpolation on raster map values can be performed using the Densify operation.
☞ Make sure that there are enough points within the limiting distance; in other words, choose the limiting distance value large enough. Rule of thumb: when you perform the calculation again with a limiting distance increased by a factor 2, and you find profound differences in outcomes, you chose the limiting distance too small in the first calculation.
☞ You can use the Spatial Correlation operation to get some insight in your point data (before interpolating).

☞ To find only the trend, use trend surface. This is relatively fast operation using the residuals as input for a Moving Surface operation, can be a good way to split regional and local phenomena.

Dialog box

The Moving Surface operation is a point interpolation which requires a point map as input and returns a raster map as output. For each output pixel a polynomial surface is calculated by a least square method approaching all point values. As points closer to an output pixel are considers more important than points further away, a weight function has to be specified. Points that are further away from an output pixel than the user-specified limiting distance, are assigned weight zero.

Dialog box options

Input point map:	Select the name of the input point map. Open the list box and select the desired input map, or drag a point map directly from the Catalog into this box.
Weight function:	Select the weight function to calculate weights for the points:
Inverse distance:	weight = $\left(\frac{1}{d^n}\right)-1$
Linear decrease:	weight = 1 - d^n
	d = distance of point towards output pixel divided by limiting distance n = weight exponent
Weight exponent:	Type a value for *n* as used in the inverse distance or linear weight calculation.
Limiting distance:	Type a value for the limiting distance. Points that are further away from an output pixel than the limiting distance, are assigned weight zero and thus have no influence on the output value for that pixel.
Surface:	Select one of the surfaces.
Output Raster Map:	Type the name of the output raster map.
Georeference:	Type the name of an existing georeference or create a new georeference.
Show:	Select this check box if you want the output map to be displayed in a map window when the operation has finished. Clear this check box if you do not want to see this map immediately: you simply define how the output map should be created.
Value range:	Accept the default value range, or specify your own range of possible values in the output map (a liberal estimate will do).
Precision:	Accept the default precision of output values, or specify your own precision.

Description:	Optionally, type a description for the output map. The description appears in the title bar when the output map is displayed.

A dependent output map is created.

Command line

The Moving Surface operation can be directly executed by typing the following expression on the command line of the Main window:

OUTMAP = `MapMovingSurface`(InputPointMap,GeoReference, *SurfaceType*,*WeightFunction*)

where:

OUTMAP	is the name of your output raster map.
`MapMovingSurface`	is the command to start the Moving Surface operation.
InputPointMap	is the name of the input point map.
GeoReference	is the name of an existing georeference that should be used for the output raster map.
SurfaceType	is the name of the surface: `Plane` | `Linear2` | `Parabolic2` | `3` | `4` | `5` | `6`
WeightFunction	is an expression for the weight function (see below).

The *WeightFunction* can be defined as:

`InvDist`(*Exp*,*LimDist*)

or

`Linear`(*Exp*,*LimDist*)

where:

`InvDist`	is the option for the inverse distance method
`Linear`	is the option for the linear method
Exp	is a value for the weight exponent
LimDist	is a value for the limiting distance: points that are further away from an output pixel than the limiting distance obtain weight zero.

When the definition symbol = is used, a dependent output map is created; when the assignment symbol := is used, a so-called source map is created.

A full expression might look like:

```
OUTMAP =  MapMovingSurface(MapX,GeoRefX,Plane,
          InvDist(1,500))
```

Algorithm

Moving Surface performs a moving surface interpolation on point values and returns a raster map.

For each pixel:

1. The distances of all points towards this output pixel is calculated to determine weight factors:

	weight = 0	$(d \geq 0)$
Inverse distance:	weight = $\left(\frac{1}{d^n}\right) - 1$	$(0 \leq d \leq 1)$
Linear decrease:	weight = $1 - d^n$	$(0 \leq d \leq 1)$

d = distance of a point towards output pixel divided by the limiting distance
n = weight exponent

These weight functions ensure that points close to an output pixel obtain large weights and points further away obtain small weights. Also see the Graphs of the weight functions topic for information on the way in which weight values decrease when distance increases, for different values of *n*.

2. The value on the specified surface is calculated. This is done with the least square method. This methods minimizes the total error:

total error = $\sum_{i=1}^{n} \left(f(x_i) - m(x_i) \right)^2$ $\qquad x_i = \binom{x}{y}$

in which $f(x_i)$ is the calculated value of the surface on a certain location and $m(x_i)$ is the measured value on that same location.

3. The value of the surface determines the value of the pixel in the output raster map.

☞ For time efficiency reasons, it is strongly advised to choose a rather large pixel size for the output raster map. Further interpolation on the raster values can be performed with the Densify operation.

☞ The larger you choose the limiting distance value, the longer the operation will take because more point values are used to calculate an output pixel value.

☞ Be sure that you take the limiting distance large enough. Rule of thumb: when you perform the calculation again with a limiting distance increased by a factor 2, and you find profound differences in outcomes, you chose the limiting distance too small in the first calculation.