ADDRESSS training course, 19-28 August 2010,
Balaton Limnological Research Institute, Hungary

SCOP++
Brief Instruction

DERIVATION OF SURFACE MODELS FROM POINT CLOUDS

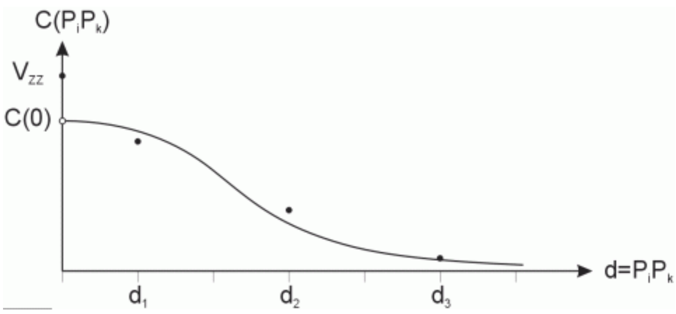
Christian Doppler Laboratory
"Spatial Data from Laser Scanning
and Remote Sensing"

1. Theoretical Background

This section will give an overview on the computation of terrain heights in SCOP++ and about the underlying data structure. It will present the theoretical aspects in compact form. More detailed information is available in the SCOP++ manual and references to it will be given in the text.

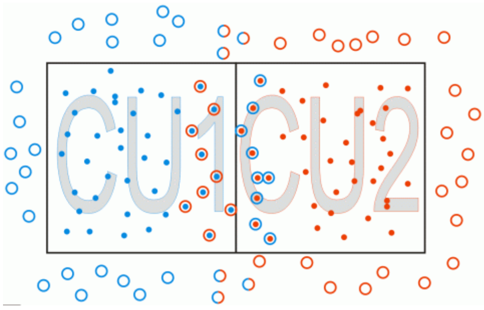
Linear Prediction

The method to compute terrain heights from irregularly distributed points in SCOP++ is the linear prediction. It is very similar to the well-known method of Kriging and shares its theoretical foundations: The terrain heights are considered as a stochastic process, the realization of a spatial random variable. Observations (i.e. measurements of the terrain heights) and the stochastic properties (correlations, ...) are used to estimate the value (i.e. the terrain height) of this random variable at different locations (i.e. ground plan positions, xy). Of course, this can be considered as an interpolation process as well. It can be shown that if the covariance function has been determined correctly, linear prediction provides the most accurate estimator for the terrain height.
First, the trend i.e. the average terrain height over large areas, is determined as an adjusting polynomial of low degree, and is subtracted from the observed terrain heights. Essential for the estimation of heights (z-values) is the covariance function, which is a function of the horizontal point distance. At a distance of zero it is the variance of the terrain heights i.e. $\mathbf{E}(z^2)$, where E denotes the expectancy. For distances larger than zero it is the expectancy $\mathbf{E}(z_1*z_2)$. This covariance function can be estimated from the data as indicated in the figure.



It can be seen that the correlation of heights decreases with the distance of points. In the figure two values are shown for the distance zero: C(0) and V_{zz}. The first denotes the variance of the terrain heights whereas the second is augmented by the measurement variance. Their difference is the so-called filter value, and it is the square of the observation accuracy. Formulas for the computation of the heights from this are given in the reference manual (e.g. p. 261). There, each point can have an individual accuracy i.e. an individual V_{zz}. This accuracy controls the filtering i.e. the removal of random measurement errors in the data points. The residual (the filter value) will be in the range of the accuracy.

In principle, this computation method is global. This means that for the computation of one (or more) terrain heights all measured points have an influence. This is also true for very large areas. To avoid this, and also in order to keep the matrices for the computation small, the computation is performed patch-wise in so-called computing units (CU). These CUs are rectangular, most often quadratic areas which may include e.g. 500 originally measured points. In order to obtain a continuous surface, an overlap is necessary between adjacent computing units.

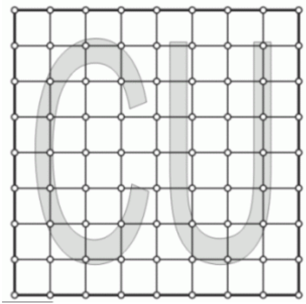


The points used for the computation of the terrain heights in the left CU (no.1) are drawn in blue. The full points are those inside the area of the computing unit, whereas those in the overlap zone are shown as rings. For CU2 (right) the points are drawn in red.

Although linear prediction is the preferred interpolation method in SCOP++, other simpler techniques are offered as well. Especially for deriving Digital Surface Models (DSM) the Moving Planes interpolation or Triangulation respectively are used.

SCOP DTM and Data Structure

In SCOP++ the original points are not stored, as this is the case in a TIN (triangular irregular network). Instead, heights are computed in a grid structure and stored in an array format (grid heights).



Additionally, structure lines can be considered in the computation e.g. where there is a sharp edge in the surface. These lines are captured during data gathering (photogrammetry) or in by processing point clouds (ALS) and stored in the DTM, too. Details can be found in the manual (p. 35). There, also the extension to a hybrid grid, containing inter-meshed break lines is described.
The distance between two grid points in the ground plan, the extension of the computing units, the width of the overlap, the accuracy of the measurements and other parameters have to be specified in order to compute a DTM. Concerning the user interface and further details on these parameters the interested reader is, again, referred to the manual (p. 125ff).

2. Data formats

The following data formats are relevant for the course “Topographic Information Systems” (input data, intermediate data):

xyz-file

.xyz (.pts respectively)

Line by line easting, northing, and height, (additional columns like correlation coefficient, … optional)

Reasonable way to save bulk data.

Example:

```
1483.299  5340216.688  64.758  0.895
1485.545  5340218.601  65.753  0.955
1487.091  5340219.944  67.773  0.935
1483.876  5340216.822  65.776  1.000
```

wnp-file

*.wnp

Header
Line by line wnp-code, easting, northing, and height
Final line

The wnp-code consists of a two-digit code and a following four- or six-digit point number (e.g.: `500001` resp. `50000001`).
Reasonable way to save break lines, form lines, and bulk data.

Example:

```
999991   0.000       0.000        0.000
0        0.000       0.000        0.000
999998   0.000       0.000        0.000
300001   67404.160   358962.480   620.500
300001   67401.560   358970.950   619.980
500001   67400.550   358978.430   619.850
500001   67400.030   358966.860   619.950
999999   0.000       0.000        0.000
```

Important wnp-codes: Bulk data 30, form lines 40, break lines 50.

Xyz- and wnp-files may be visualised spatially in the program GVE (Graphics Viewer Editor). The ASCII files can be viewed very fast with the program Windows Commander and edited with powerful text editors. Access both programs from the “quick launch” in the task bar.

3. Instructions for DSM/DTM derivation and visualisation with SCOP++

Start SCOP++ with the icon "quick launch" on the task bar.

New project

Project ► ***New…*** ► ***Name*** enter project name, ***Location*** enter location for saving ► ***ok***

At the location for saving, a folder with the name of the project appears. It contains the files "project name.spr", "project name.TOP", and "project name.cmf".

Attention: The project name may exist only once on a computer!
HINT: Leave the default directory in ***Location*** to ease the overview on the already used names.

Open an existing project

Project ► ***Open…*** ► ***Browse*** project name.spr ► ***ok***

Add a Model overlay (This dialog opens automatically after creating a new project)

Overlays ► ***Add model overlay*** ► ***Name*** enter overlay name ► ● ***Both*** ► ***ok***

Model-only: Enables the visualisation (shading, height coding, isolines) of a DSM (*.dtm).
Data-only: Enables to edit and visualise the data set (*.xyz, *.wnp, etc.).
Both: Enables DSM derivation, visualisation of the data set and of the DSM, editing of the data set.

A button with the overlay name appears on the left side of the window (from now on called ***ModelOverlay***)

In the project directory, a folder with the overlay name appears. The results of the DSM derivation are written into this folder.

Data import (This dialog opens automatically after creating a new overlay)

ModelOverlay ► ***Import/Export*** ► ***Data*** ► ***Import…*** ► ***Browse*** (set ***Filename*** and ***Data format***) ► ***ok***

Limits

Either the whole area or a smaller part can be considered for DSM/DTM derivation or visualisation. You can digitise the area of interest in ***Limits...***, set coordinates with ***Limits...*** ► ***Edit*** or take the whole area with ***Limits...*** ► ***Set to max.*** The limits are valid for the checked options in ***Limits for*** (Screen, Model, Views, Data).

Attention: Limits are always set for all overlays!

Moving planes

ModelOverlay ► ***Model*** ► ***Grid width*** set grid width, should fit roughly the point density ► ● ***Moving planes*** ► ***Details moving planes...***

Window ***DTM*** overlay name***: Moving planes parameters***:

Structure
Size of the CU = grid width • (gridlines per CU – 1)

Method specification:
● **Tilted planes:** each grid point is computed from a tilted plane based on the surrounding control points
● **Nearest neighbour:** grid height = height of nearest neighbour of grid point

Triangulation

ModelOverlay ► Model ► Grid width set grid width, should fit roughly the point density ► ● ***Triangulation ► Details triangulation...***

Window ***DTM*** overlay name***: Triangulation parameters***:

Structure
Size of the CU = grid width • (gridlines per CU – 1)

Linear prediction

ModelOverlay ► Model ► Grid width set grid width, should fit roughly the point density ► ● ***Classic prediction ► Details classic prediction...***

Window ***DTM*** overlay name***: Classic prediction parameters***:

Size of the net CUs
● **Derive:** automatic determination of the size of the Computing Units, in every CU the number of points should be about the value set in **Points per CU to aim**, or
● **Define** Size of the CU = grid width • (gridlines per CU to the east – 1) • (Gridlines per CU to the east – 1),

Overlap specification
Overlapping ‚gross' computing units large,

Specification selection Filter,

Smoothing/Filtering specifications
Mean filter values Set values for bulk data, spot heights, form lines und break lines (ALS data normally contain only bulk data)

Result: overlay name.dtm

Attention: To get a calculation protocol check in ***Options ► Protocol*** (you can find the protocol in ***ModelOverlay ► Model ► Protocol*** after DSM/DTM computation)
Attention: To get a calculation protocol check in ***Options ► Protocol*** (you can find the protocol in ***ModelOverlay ► Model ► Protocol*** after sDTM computation)

Shading

Requirement: In the respective ModelOverlay a DSM exists already.

Mode ► final
ModelOverlay ► Shade… ► Properties… ► set resolution and illumination ► ***ok***
ModelOverlay ► Shade… ► display

Height coding

Requirement: In the respective ModelOverlay a DSM exists already.

Mode* ► *final
ModelOverlay* ► *Zcode…* ► *Properties… ► set resolution and height coding, in ***Predefined palettes*** you can find predefined palettes for different tasks, but you can also define palette files, level files or look up tables yourself within SCOP++ or external (text editor). Optionally, a certain height band can be considered ► ***ok***
ModelOverlay* ► *Zcode…* ► *display

Export of Visualisations

Height codings and Shadings may be exported as follows:
ModelOverlay* ► *Import/Export* ► *Views* ► *Export…* ► *Browse (select ***Filename***) ► ***●Isolines ●Shading ●Z-coding ●Profiles ►ok***

4. Handling of error messages and program crashes

Error messages are explained in the SCOP++ manual (section 12.3).

Crash of SCOP++:
Kill the processes scop++.exe and RPC_TdmServer.exe in the task manager, start SCOP++ again (possibly twice).
Delete all *.TOP and *.TTT files in the directories d:\ScopProjects\topdb\cfg, cvt, dpm, grf, mkt, sys.
Deregister the project (***Project* ► *Deregister*** ► mark the respective project ►***ok***) and possibly delete the project.