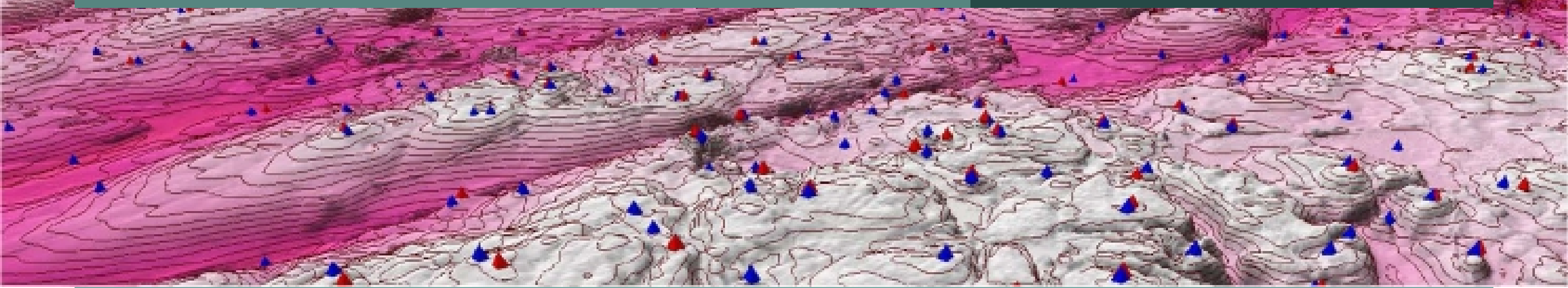
TECHNICAL REPORT NO. 13



Automatic generation of contour lines based on DK-DEM

Brigitte Rosenkranz, Thomas Knudsen, Hanne E. Mortensen, Peter Bøving Michealsen

Danish Ministry of the Environment
National Survey and Cadastre

Indhold

Brigitte Rosenkranz, Thomas Knudsen, Hanne E. Mortensen, Peter Bøving Michealsen
Automatic generation of contour lines based on DK-DEM

National Survey and Cadastre—Denmark, technical report series number 13
ISBN 978-78-92107-39-8
Technical Report
Published 2012-02
This report is available from http://www.kms.dk

Summary and Introduction

Poul Frederiksen, Thomas Knudsen

This report comprises a collection of 4 papers by 4 authors, giving 4 different views of the technical and historical background for the production of a new height contour set for use in Danish topographic maps.

The contour set is based on DK-DEM, the new high resolution LiDAR measured national elevation model, introduced in 2009. The new contour set replaced an older set originating mostly from 19th century field surveys, but continuously refined and improved by cartographers, for more than a century. Peter Michaelsen describes this aspect in the paper *Højdegengivelse på topografiske kort – en historisk redegørelse* (Elevations on Topographical Maps—a Historical Account)

The old field surveys were necessarily limited by the precision of the technology of their times, and hence typically suffered from long wavelength biases. The main goal of the new contour set is to reduce these biases and hence get the more accurate heights of the new height model reflected in the contours shown on the maps.

The old contours were, however, highly acclaimed by map users, since the century long cartographic tender loving care had resulted in a data set with very high cartographical expressivity. In other words, the old contours communicated an easily interpreted impression of the actual landscape, to the map user.

On the other hand, automatically generated contours based on high resolution models, are typically very hard to interpret, since the large amount of short wavelength details disturb the overall impression of the landscape.

Worse yet, such contours are not easily generalized: While the amount of detail is reduced, the generalized contours are still hard to interpret. Generalization is a geo-communicative task, and turning classical cartographic skills into algorithms is not an easy task. Even tried-and-tested algorithms sometimes result in absurdities such as contours crossing each other (hence assigning two distinct heights to the same point in the field).

So in the work with the new set of contours, we have strived to infuse a generous dose of classical cartographic skills into the highly automated process. We have, however, not attempted to construct a "mechanical cartographer".

Rather, we have followed two complementary strategies: The first is to recycle existing ancillary data sets, the second to employ what may be described as *expert in the loop* filtering, where a human cartographic expert supervises a parameterized filtering algorithm, making sure that the filtering parameters are set to an optimum defined by a hard-to-quantify goal of cartographic expressivity.

Brigitte Rosenkranz demonstrates the first strategy in the paper *Update of historical contours by automatically generated contours, based on the laser scanned Digital Terrain Model DK-DEM*: Here, it is described how the existing file of spot heights has been adjusted to fit with the new height model. Spot heights are typically hilltops which, on the map, are labelled with the actual height of their apex. The spot height data were selected, refined and revised through many editions of map series by expert cartographers in order to deliver improved support for the interpretation of the height contours. Fitting the old data set to the new height model comprises two steps: First, to tie the old and less-accurate spot height position to the actual horizontal position of a hilltop in the new and much more accurate height model; Second, to replace the old height label with a new one corresponding to what is actually found in the height model.

The second strategy, *expert in the loop* filtering, is described by Thomas Knudsen in the paper *An Algorithm for Production of Cartographically Optimized Height Contours*. The paper describes a process where a filtering method (combining gaussian smoothing and non-isotropic shape-preserving filters) was developed in an iterative process involving evaluation sessions and interviews with cartographers.

But even the best contour sets need a final brush up to make them publication ready. This was the case for the old contour sets, and this is the case for the new contour set. The brush up procedure (which includes a modest amount of generalization; removal of very small elements, and the adjustment of top level contours to make room for spot height labels), is described by Hanne Elisabeth Mortensen in the paper *Generalizing contour lines to various map scales*

As can be derived from the summary above, producing a new generation of height contours is a somewhat daunting task. And rightfully so, since the mission is to replace a classic and highly acclaimed cartographical data set. The initial impression is, however, that the mission is accomplished: The first products based on the new contours have been released, and the general user feedback is very positive.

Update of historical contours by automatic generated contours, based on the laser scanned Digital Terrain Model DK-DEM

Brigitte Rosenkranz

Abstract

This paper describes two programs and the process developed by the National Survey of Denmark to update and relocate height points and generate cartographically nice and at the same time mathematical correct contour lines. Both, the production procedures and the challenges of generating the nation wide contour lines are described in detail.

1 Introduction

In 2007 the National Survey and Cadastre (KMS) invested in a new laser scanned digital elevation model DK-DEM back in 2007. Until then, the contour lines used in KMS were to some extent still based on data from 19th-century military surveys. Owning the new laser scanned elevation model naturally obligates KMS to use this new model for the generation of updated nation wide contour lines.

The new terrain (DK-DTM) and surface model (DK-DSM) are interpolated by triangulation from a geo-referenced point cloud. KMS owns user rights to the DK-DSM and DK-DTM, but not for the point cloud. Therefore the contour lines were generated from the terrain model (DK-DTM), stored as an esri-ascii grid. The key values of the model are:

Point cloud density	$0.45\,\text{points}/m^2$
Cell size	$1.6m \times 1.6m$
RMS error	$5.9\,cm$
Standard deviation	$3.44\,cm$

All details about the quality assessment of DK-DEM can be found in (Rosenkranz, 2011).

The DK-DEM provided the starting conditions for generating contour lines. The main application for contour lines still is the visualization of the landscape. This means that is very important to have a good cartography. For that reason one of our main success criteria was to satisfy our cartographers but also to generate mathematically and geographically correct contour lines. Well designed contour lines are always supported by marked cartographically important top points. In a high resolution digital elevation model like DK-DEM a huge amount of important, but also unimportant local maximum points are included. Over a long period of the time cartographers decided which points are necessary to mark in maps. As no mathematical filtering processes can take this decisions, we were using these well funded historical decisions as input to our contour generation process.

Every filtering processes reduces the amount of information included. However, a regular filtering process would result in minimizing everything, also cartographically important extremes. This is of course not desired and would furthermore result in contour lines that do not fit to the cartographically important top points. For avoiding the discrepancy between contour lines and extreme points the contour line generation process which is described in this paper was developed. `Kaku.m` is a filtering program where the level of filtering depends on the distance of the grid point to the closest top point. The exact location of the top points are computed by the program `FindHighLowElevPoints.py`. At this exact top point location the grid is not filtered which results in the absolute correct value for these top points and the associated top contour lines. More details about the mathematics behind the contour line generation is reported in the companion paper (Knudsen, 2011). In addition to the automatic filtering a further generalization was necessary to fulfill the success criteria of cartographers. The generalization processes carried out as the final step is described in (Mortensen, 2011). Information about the history of KMS' contour lines can be read in (Michaelsen, 2011).

2 Data Preparation

The original data were delivered in blocks of $1\,km \times 1\,km$ with a cell size of $1.6m$. For minimizing the task of snapping the contour lines across block borders the $1\,km \times 1\,km$ grid were merged into 900 times bigger grids ($30\,km \times 30\,km$). See figure 1. Contour lines calculated from a grid with a cell size of $1.6m$ would be disturbed by high frequency noise and result in huge file sizes. For minimizing both the noise and huge file sizes, we were working with a generalized grid with a cell size of $10m$.The elevation value was interpolated from the nearest neighbor point to the $1.6m$ grid.

During the past 100 years KMS was working with many different sets of historical top and hollow points. These points were digitized during the past 15 years. The first digitalization was done manually by entering height values from existing maps and later on by photogrammetric measurements. Many of the top points are located in forest areas and could not be measured directly by photogrammetry. During the digitizing project these points were allowed to be moved horizontally up to $200m$. For locating the correct horizontal position of these points, we carried out a archive search which resulted in one file with the positions of all cartographic important top- and hollow points in the UTM32N_ETRS89 reference system. These points are used as input positions for the determination of the actual height points for the new contour lines calculated based on DK-DEM.

3 Method

For calculating these cartographic and mathematically correct contour lines two programs were developed. The first program

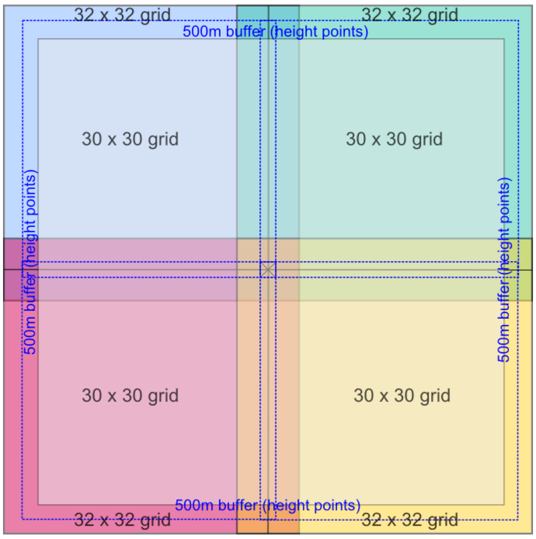


Figure 1: Interrelating buffers of the grid and the input height data to the contour lines workflow. For guaranting snapping of the contour lines at block borders, it was worked with different buffers. The original working tile was 30 *km* x 30 *km* where a buffer of 1 *km* was added to fit contour lines. Due to the differentiated filtering around important heigth points a diffentent buffer (500 *m*) was added to the height imput file. After calculation the contour lines were cut into $30\,km \times 30\,km$.

(FindHighLowElevPoints.py) updates the cartographically important top points by relocating the historical points and inserting the new measured values. The second program (`Kaku.m`) filters the grid and uses the distance to the actual top points for its differentiated filtering process.

3.1 FindHighLowElevPoints.py

FindHighLowElevPoints.py is a Python program, working with the grid as a 2-dimensional matrix. The program reads the historical top points from the prepared input text file (see figure 3). It builds a small search matrix from the grid with the size of $n \times n$ cells around the historical extreme point. Within the search matrix a maximum and a minimum filter with a footprint of $m \times m$ is applied to the data. This means the search matrix is divided into $(n/m)^2$ smaller matrices where the specific extreme within the footprint is found. If there are more footprints with the same extreme value the shortest distance to the historical point decides which extreme point is chosen (see figure 4). Finally, the geographic coordinate from this height point is calculated from the header information of the used ESRI-grid. Experiments with different maximum filter sizes and footprints resulted the following parameters.

- Search radius $n = 10$ cells (100 *m*)
- Footprint for maximum/minimum filter m=5 cells (50 *m*).

The output of FindHighLowElevPoints.py are text files (rcxyz-files) including the location and value of the top points (see figure 5). The location and value are given by:

- Row and the column (DK-DTM 32 *km*x32 *km* grid)

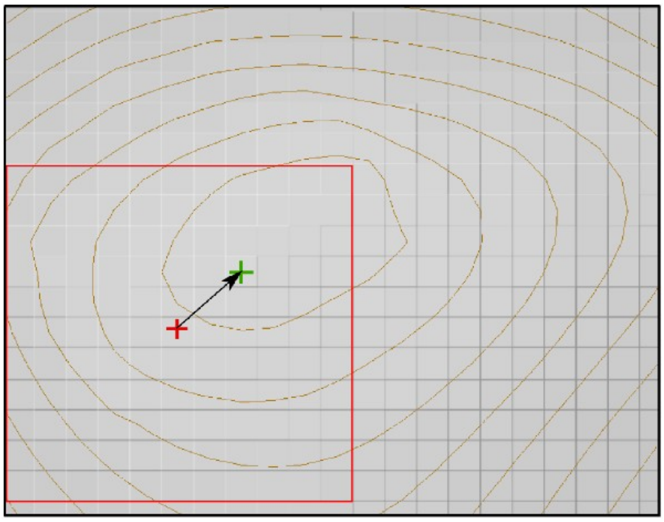


Figure 2: New placement of the (red) historical to the updated (green) height point, shown in the 10 *m* grid. The search matrix (in this case 105 *m* x 105 *m*) is shown as the red box.

```
"navn","y","x","z","type"
6400_580,6401192.35,597704.14,-999.00,8514
6400_580,6399648.56,591529.45,-999.00,8514
6400_580,6400602.12,592234.98,-999.00,8514
6400_580,6400607.21,592615.02,-999.00,8514
```

Figure 3: Input file to FindHighLowElevPoints.py. Note that a header is expected. 'navn' is the name of the 30x30 block, 'x' is easting, 'y' is northing, 'z' is elevation, 'type' is type maximum (8512) or minimum (8514).

- Easting, northing (UTM32N_ETRS89)
- z value (DK-DTM 32 *km*x32 *km* grid).

The rows and columns are used as input in `Kaku.m` for implementing the differentiated filtering.

```
617 2569 543695.00 6185825.00 69.09 8512
625 2397 541975.00 6185745.00 59.30 8512
660 2111 539115.00 6185395.00 71.67 8512
618 2052 538525.00 6185815.00 67.12 8512
```

Figure 5: Output file of FindHighLowElevPoints. The first column describes the row, the second the column of the 30x30 grid. The third and fourth are x- and y-position in UTM32N_ETRS89 and the fifth is the z-value.

3.2 Kaku.m

`Kaku.m` is an Octave program that filters the grid and calculates a more smoothed grid. It works with the grid as a 2-dimensional matrix, which is filtered iterative and in several steps with both an anisotropic (Perona/Malik) filter and an isotropic (Gaussian) filter. During the iteration process it gradually keeps a more and more strict value for the local extrema. The location of the cartographically important top points (calculated with FindHighLowElevPoints.py) are used for the differentiated filtering processes. The distance to the top point location determines the weight of the unfiltered (original) to the

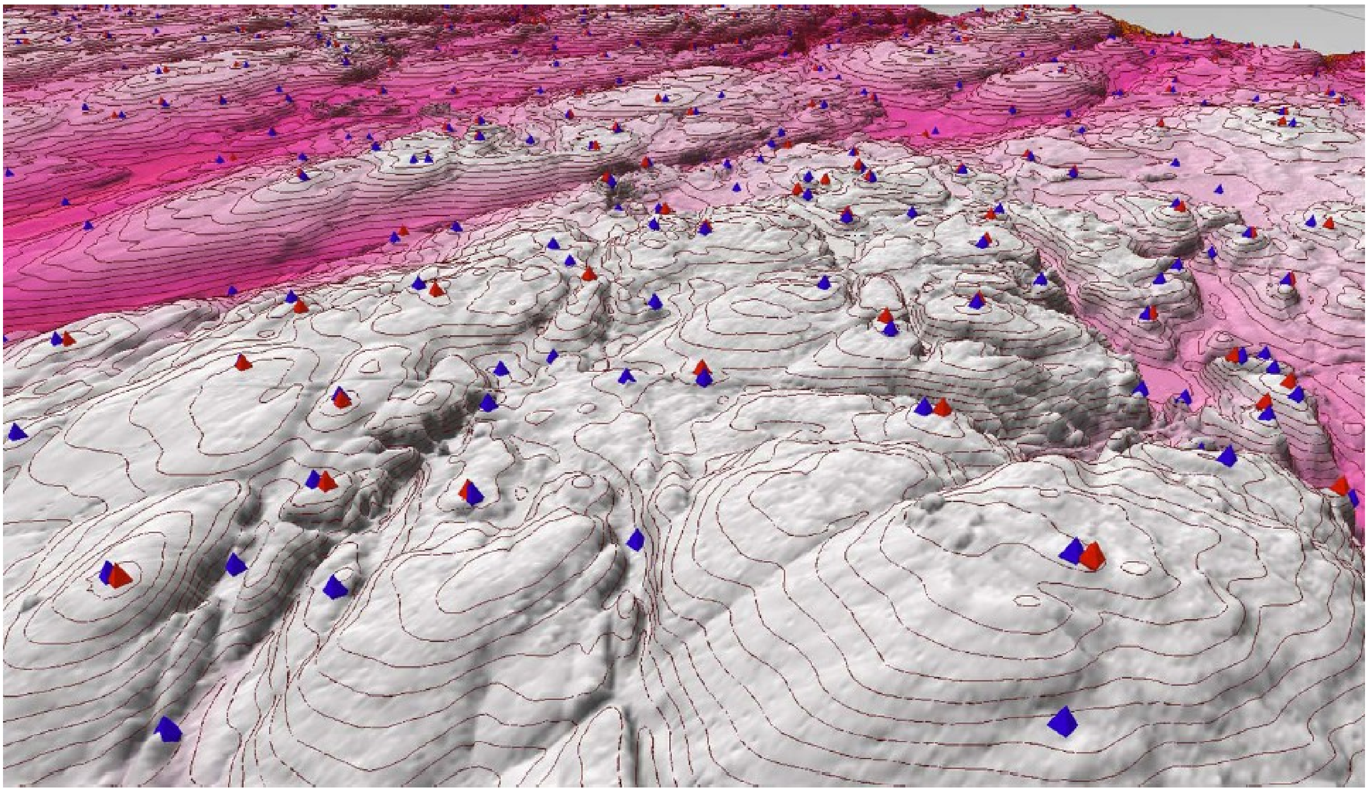


Figure 4: FindHighLowElevPoints.py moves the historical height points (blue) to their updated locations (red) in the DHM grid

filtered grid. The output is a grid that is both, smoothed (easier to interpret) and preserves the height of cartographically important top points. `Kaku.m` can be run with the following options

- p.ITER - amount of iterations
- p.dzmax - maximum modification value
- p.D - size of weighting distance array
- p.overlap - block overlap in cells
- p.k1 - filter size and type.

KMS has up to now produced contour lines for 5 different scales with 4 different equidistances. Different options have been used for the different scale generation. The parameters that were used are described in chapters 3.3 and 3.4. The contour lines were calculated from the filtered grid with the GDAL contour generating tool gdal_contour. The contours were exported to ESRI-shape format and cut with a buffer of 50 m (p.overlap) to each block.

More information about `Kaku.m's` mathematics and filtering method is described in detail in the companion paper (Knudsen, 2011).

3.3 Scale 1:10 000, 1:25 000, 1:50 000

The first contour lines KMS generated from DK-DEM data were 5 m contour lines for scale 1:25 000. The best result could be reached with the following `Kaku.m` parameters.

p.dzmax	1.25 m	maximum modification
p.NITER	20	amount of iterations
p.D	4 cells	search radius
p.overlap	5 cells	overlap between blocks
p.k1	gaussian, footprint 15 cells, variance 4 cells	filtering type

Manual snapping

For interrelating lines over block borders, a buffer of 1 km was added to the original grid. See figure 1. This resulted in a 32 $km \times 32\,km$ working grid. Nevertheless, it showed out that this buffer was not enough to avoid contour lines that could not be snapped over the block borders. Due to different filtering in the blocks with compared to the blocks without a top point the snapping was not successful in 3 000 locations. These locations resulted from 296 top points that were closer to the block border than 50 m. These dangling contour lines were handled of manually.

3.4 Scale 1:100 000

As contour lines for 1 : 100 000 maps have to be much more generalized than more detailed products, this calculation was carried out with a different setup. The historical input file was less detailed and the filtering was more aggressive for this map scale. `Kaku.m` was run with the following parameters for the 1:100,000 map scale:

```
# footprint is the size of the matrix where the maximum/minimum
      filter is appended
F=np.ones((footprint,footprint))

# use maximum filter (find maximum inside the footprint)
MAX=scipy.ndimage.maximum_filter(B,footprint=F)

# use mimimum filter (find minimum inside the footprint)
MIN=scipy.ndimage.minimum_filter(B,footprint=F)

T=MAX>MIN
if extrType[[k]]==8512: # Top points
    # find rows(m)/cells(n) with a maximum value
    [m,n]=np.where(np.logical_and(T,B==MAX))

elif extrType[[k]]==8514: # Hallow points
    # find rows(m)/cells(n) with a minimum value
    [m,n]=np.where(np.logical_and(T,B==MIN))

if m.shape[0]!=0:
    # build a vektor with rows/cols of all lokal maxima/minima
          around point k (one input point)
    [Bnm]=np.dstack((m,n))
    Bxy=GridHandling.grid2xy(Bnm,hB)
    Bxyz=np.array([[]])

    # add the z value to the vektor
    Bxyz=np.column_stack((Bxy,B[(m),(n)]))
    if extrType[[k]]==8512:

    # find maximum of all lokal maxima
        [p]=np.where(Bxyz[:,2]==np.max(MAX))
    elif extrType[[k]]==8514:

    # find minima of all lokal minima
        [p]=np.where(Bxyz[:,2]==np.min(MIN))
```

Listing 1: Using maximum/minimum filter for finding extreme points.

p.dzmax	25 m	practically overruled
p.NITER	1	amount of iterations
p.D	4 cells	search radius
p.overlap	5 cells	overlap between blocks
p.k1	gaussian, footprint 14 cells, variance 5 cells	filtering type

Multiple top points input

When generating 1:100 000 scale lines a buffer of 500 m was added to the historical top points input file (used in FindHighLowElevPoints.py). This means that points with a distance shorter than 500 m to the neighbor block were included in more than one contour line filtrations. Hence the manual snapping that had been carried out when generating the $1:25000$, $1:50000$ and $1:10000$ contour lines was eliminated. Elevation points less than 500 m from the 30 $km \times 30\,km$-border can now be included up to 4 times (once in each block corner).

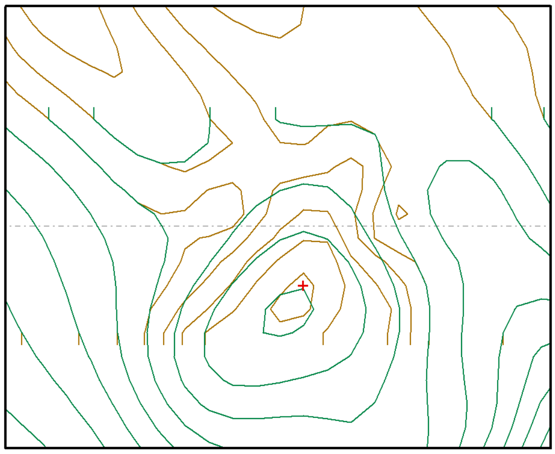


Figure 6: Assembling the contours at the border showed out to be more challenging than expected. The green contours show the grid filtered without a height point. The brown contours show the differentiated filtered grid weighted proportional to the distance to the height point in red. These contour lines were snapped manually for the 1:10 000, 1:25 000 and 1:50 000 scalled maps.

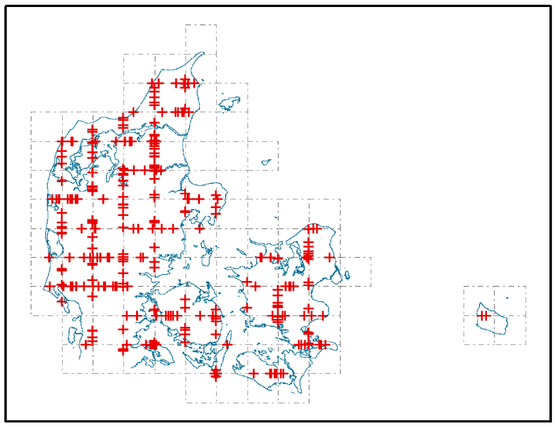


Figure 7: Height points with a distance less than 50 m to the border. These points had to be controlled and most of them corrected manually for the 1:10 000, 1:25 000 and 1:50 000 scalled map.

4 Further Generalization

These automatically generated contour lines were derived from data collected between 2005 and 2007. These data are supposed to be used in KMS' map production together with different vector data having different actuality and accuracy. So for consistency, the following generalization procedures were carried out:

- small contours ($< 1000m^2$) were removed.
- small contours including a top point were enlarged.
- contours running through a vectorized lake and contours outside the vectorized coast-zone were removed.
- crosses were added within hallows.

The whole generalization process is described in detail in the companion paper (Mortensen, 2011).

5 Conclusion

Developing the process of this new contour line generation was a very challenging and instructive progress. It can best be described as expert-in-the-loop development where a very close corporation between software developers and cartographers took place.

Both historical data and cartographers expert knowledge were incorporated in the filtering software. The historical top points as starting points for the search of actual top points and the cartographers expert knowledge via developing-review cycles.

Working this way enabled us to improve the filtering methods and correct unexpected unsuitabilities directly. The contour lines that have been generated with the setup described are already implemented in the 4 KMS mapping products (M718, M6113, Kort25, Kort10). These products are judged constantly by different experts where most gave very positive feedback.

References

Thomas Knudsen. An algorithm for production of cartographically optimized height contours. Technical Report 13, National Survey and Cadastre KMS, 2011.

Peter Michaelsen. Højdegengivelse på topografiske kort – en historisk redegørelse. Technical Report 13, National Survey and Cadastre KMS, 2011.

Hanne E. Mortensen. Generalizing contour lines to various map scales. Technical Report 13, National Survey and Cadastre KMS, 2011.

Brigitte Rosenkranz. Quality assessment of the danish elevation model (DK-DEM). Technical Report 12 (ISBN 978-78-92107-38-1), National Survey and Cadastre KMS, 2011.

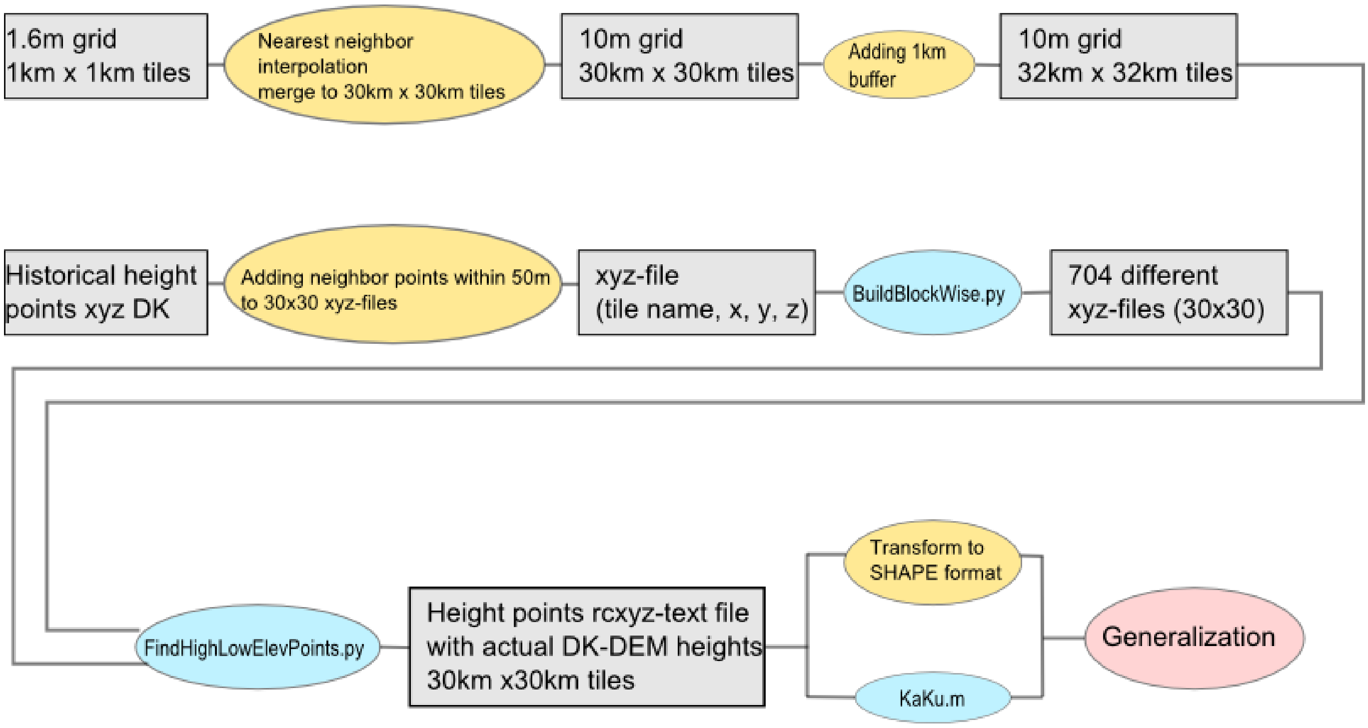


Figure 8: Workflow of the contour and height points generating process. Starting with esri-asci grid data from DK-DEM with a cell size of $1.6\,m$ and a tile size of $30\,km \times 30\,km$. A new $10\,m$ grid was generated by nearest neighbor interpolation and $1\,km$ buffer was added. This resulted in $32\,km \times 32\,km$ tiles. The historical height points were transformed to a ETRS89 text file, where the historical points from neighboring block were added within a distance of $50\,m$. Thus overlap points were presented more often. The height point text file was divided into $30\,km \times 30\,km$ blocks. These text files were together with the $32\,km \times 32\,km$ grid used as input for FindHighLowElevPoints.py. The output of FindHighLowElevPoints.py (rcxyz-text file) was both used for the filtering process within `Kaku.m` and generation of a shape file with the actual top points with FME. `Kaku.m` results in a filtered grid and contour lines which were generalized for mapping purposes.

An Algorithm for Production of Cartographically Optimized Height Contours

Thomas Knudsen

Abstract

We introduce a filtering algorithm for digital terrain models, intended to improve the geo-communicative aspect of of contours based on the filtered model. The algorithm is based on a combination of gaussian (isotropic) filtering and non-isotropic filtering. It was developed empirically, gradually adjusting the approach through input and opinion from expert cartographers.
Keywords: Digital surface model, digital elevation model, raster, compressive sampling, horizontal accuracy, LiDAR

```
F = G;
for i = 1:N
    % Perona/Malik filtering.
    F = imsmooth (F, "P&M", 2, 0.25, "method2");
    D = find (isnan (F));
    F(D) = G(D);

    % Gaussian filtering.
    F = filter2 (fspecial ("gaussian", 5,2), F);
    D = find (isnan (F));
    F(D) = G(D);
endfor
```

Listing 1: Filtering process See section 4.1 for details.

1 Introduction

Historically topographic contour curves were based on comparatively sparse data from field surveys. The curves were hand drawn by cartographers, which were free to include their (ample) field experience and geo-communicative skills into the interpretation of the (sparse) surveying data.

Such historical topographical contours tend to be visually beautiful and communicatively rich: they give the user a very good impression of the landscape portrayed *even though they may be slightly offset in comparison to the actual height of the landscape* (e.g. the 20 m contour may trace through parts of the landscape with heights between, say 19 m and 21 m).

With modern laser scanned or photogrammetrically derived height models the situation is different: Here we have ample amounts of data, but not much field experience. In such cases topographical contours are generated by interpolation in the height model or by triangulation in a point cloud. Numerically, such automatically generated contours are superior to their old style counterparts. Unfortunately they also tend to be jagged (i.e. dominated by high frequency noise) and very hard to interpret for a human map user.

Since the early 1970s a large number of contour simplification algorithms have been published. Such algorithms are often successful in reducing the data volume (i.e. number of points) in the contour set – but they are seldom successful in convincing experienced cartographers, with high geo-communicative standards.

1.1 What this paper is

In this paper we present a filtering algorithm developed in close collaboration with cartographers, and intending to improve the geo-communicative aspect of the contours, while not necessarily come anywhere near the classical ideal.

1.2 What this paper is not

This paper is *not* scientific in the strict sense of the word: our succes metric is buried deep in the brains of the cartographers commenting on the results during an iterative development process. We finally arrived at the algorithm presented in section 4 below, which was deemed *good enough*.

Explaining *why* and *in which sense* the algorithm is good enough is outside of the scope of this paper. Also, for clarity all corner cases, boundary matching etc. etc. aspects of the actually implemented algorithm is left out of the discussion (although a few comments on their nature is given). These aspects can be studied in the source code (Matlab) of the actually implemented algorithm, which is available from the author.

2 Historical background

For more than a century, height contours on Danish maps were based on material from 19th-century military surveys. While not having high absolute precision, these contours were known for conveying correct *impressions* of the landscape, as decades of revision by skilled cartographers perfected the graphical communication aspect.

But with modern requirements for high precision heights (hydrology, orthophotos. . .), the need for a new DEM became increasingly evident. Hence, a new national coverage, high resolution DEM was released in 2009. With a new DEM, we also need new contours. But raw contours from direct interpolation in the grids are way too detailed for cartographic communication. And simplification by Douglas-Peucker or similar algorithms (Douglas and Peucker, 1973) typically results in hard-to-interpret, topologically problematic contours. This is far from the highly regarded cartographical expressiveness of the old contours.

3 Approach

To regain some cartographic expressiveness, we developed a contour generalization based on filtering of the raw grid, followed by a small dose of Douglas-Peucker generalization of the resulting contours. The real challenge was to get the cartographic expression right since the design criterion of the filter is not given by a skill metric, but rather buried in the skulls of

Thomas Knudsen thokn@kms.dk, Kort & Matrikelstyrelsen, Rentemestervej 8, 2400 Copenhagen NV, Denmark

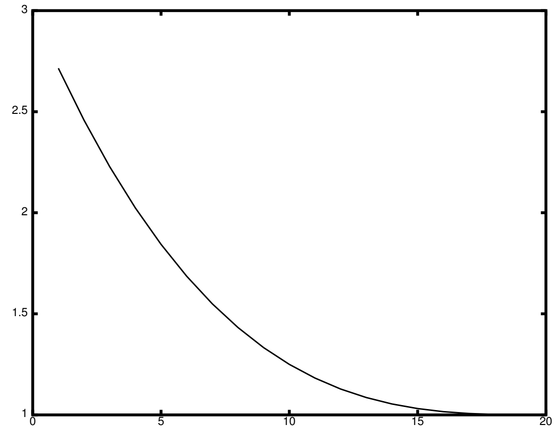


Figure 1: Maximum deviation exaggeration factor for the recovery process

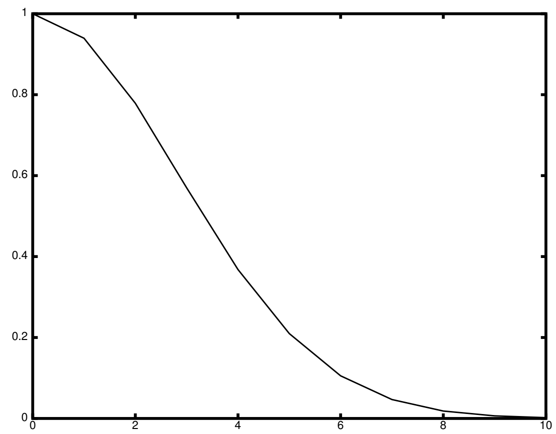


Figure 2: Distance weight function for the recovery process

```
for i = 1:N
    % At the first iteration we only reduce very large deviations,
        then gradually extend the domain by reducing limits
        towards "max acceptable"
    maxdev = dzmax ∗ (1+2 ∗ ((N−i)/N)^3)

    % Mask: W==1 where deviation from raw DEM exceeds
        maxdev
    W = (G − F);
    W(abs (W) < maxdev) = 0;
    W(W > 0) = 1;
    W(W < 0) = 1;
    W(isnan (W)) = 0;

    % ignore very small deviating areas
    W = bwmorph (W, "open");

    % Force hilltops into the mask
    W(HT) = 1;

    % Compute distance weight for recovery process
    D = bwdist(W);
    W = exp (−(D/4).^2);
    W(W<0.1) = 0;

    % Recover by adding back deviation
    F += (W .∗ (G − F));
endfor
```

Listing 2: Recovery process. See section 4.2 for details.

skilled cartographers. In order to leverage these skills, we used an iterative filter design, gradually modifying the filtering process, while doing production runs and evaluations with a group of very experienced cartographers.

The final filtering design included iterative application of both isotropic Gaussian filter kernels, and the highly anisotropic Perona/Malik filter. But several devils lurk in the details: additional complications come from the wish to avoid growth of data-void areas (due to the spreading effect of the filter kernel convolution), and to avoid modifications to local extrema (by their nature low pass filters tend to erode hilltops and fill up valleys). But with these complications accounted for, we finally have a new set of contours that approximate the cartographic expressiveness of the original contour set, while having much better absolute heights and consistency with the gridded DEM.

3.1 Hilltops

Historically contours are not labelled on Danish topographic maps. So at first glance, only relative heights can be read from the map. Absolute heights are derived by comparison with point heights printed at selected hilltops. From the existing cartographic data, we know which hilltops are the most important clues for landscape interpretation. It is an important secondary goal for our algorithm to preserve the height of these hilltops.

4 Algorithm

The algorithm finally derived at consists of two steps, where each step consists of an iterative process. The first step is a filtering process, while the second is a recovery process, where grid nodes that have drifted too far away from their original value are gradually forced back into an acceptable deviation interval.

The Octave implementation of the Matlab programming language was used in the iterative development of the filtering procedure. The same language is used as pseudo-code vehicle for describing the algorithm in listings 1 and 2.

4.1 The filtering process

The filtering process is described in the simplified Octave code of listing 1. At entry, the raw grid is in the matrix G, which is preserved during the computation. The filtering process is repeated N times (typically N is in the range 1–20). At exit the filtered grid is in the matrix F.

The imsmooth function call in line 4 performs two passes of an anisotropic Perona/Malik filtering (Perona and Malik, 1987, 1990), with a diffusion parameter of 0.25.

The filter2 function call in line 9 performs a plain gaussian filtering of the grid, with a 5×5 kernel with a standard deviation of 2 in units of the grid node length.

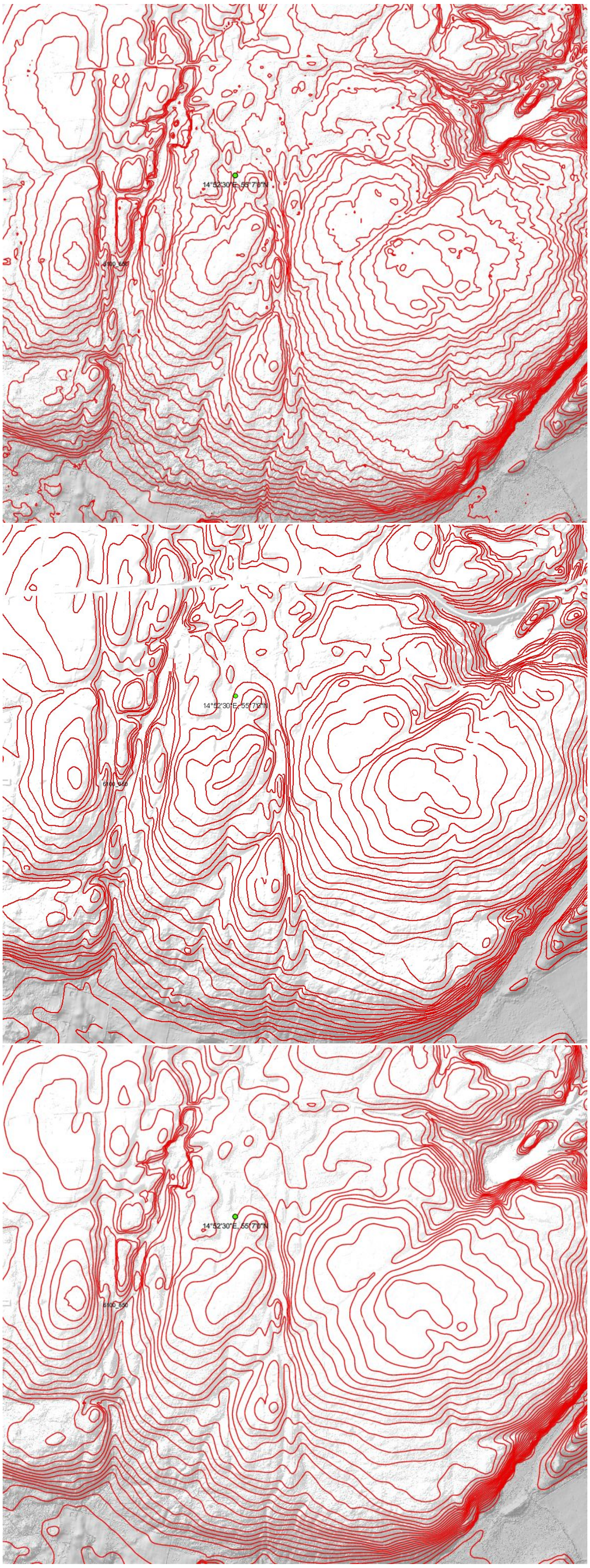


Figure 3: Upper panel: Contours generated from the unfiltered terrain model. Centre panel: Historical contours, based on ground surveys, Lower panel: Final contours generated from the filtered terrain model. Note that the historical (hand drawn) contours are discontinued where they cross roads. For the new contours, we aim at filtering out the bending effect induced by the road profile.

The find... incantations in lines 5–6 and 9–10 ensures that data void areas are not allowed to grow by diffusion through the filter kernels: after each iteration, any extensions of the data void areas are reset to their original value. Incidentally this also means that no filtering is carried out on the boundary of data void areas.

4.2 The recovery process

The recovery process works by iterative reduction of outliers using a smooth spatial transition. It is described in the simplified Octave code of listing 2.

At entry, the raw grid is in the matrix G, which is preserved during the computation. The filtered grid is in the matrix F (which is modified during the computation), and the hilltops to preserve are indexed in the mask matrix HT.

The filtering process is repeated N times (typically N is in the range 1–20), and the maximum tolerable deviation between the raw grid and the filtered grid is determined by the parameter dzmax (typically dzmax is in the range 0.5–1.0). At exit the filtered grid is in the matrix F.

The maxdev computation in line 3 determines which exaggeration of dzmax to use in this iteration. At the final iteration maxdev = dzmax. See figure 1, where maxdev is shown for dzmax = 1.

The W computations in lines 6–10 creates a mask which is zero everywhere G an F differ by less than maxdev.

The bwmorph function call in line 13 carries the mask through a morphological *opening* (i.e. an erosion followed by a dilation), removing salt-and-pepper noise (singular outliers) in the mask (Haralick et al., 1987).

In line 16 we force the hilltops into the mask. Since these data are single ones in a sea of zeros, they can first be added after the morphological filtering in the previous step.

The bwdist function call in line 19 computes the *distance transform* (Fabbri et al., 2008) of the mask W. It returns a matrix D which in all elements holds the distance (in units of grid node length) to the nearest non-zero element of W.

The exp function call in line 20 computes a weight matrix for the recovery process using the weight function shown in figure 2: the closer to a gross deviation a grid node is situated, the harder it is dragged back towards the raw grid value.

Finally in line 24 the recovery is carried out by adding back the weighed deviations.

5 Post-processing

The filtering is carried out in 30 km $\times$ 30 km boxes. After filtering of the grids, initial contours are generated using the gdal_contour module from the geographic object orientated data abstraction layer (GDAL) package Warmerdam (2010).

The initial contours are fed into the established production work flow of the KMS: curves are fitted at grid boundaries, the amount of data is reduced by a gentle Douglas-Peucker reduction, etc. Finally the contours are stored in the production data base, ready for delivery to users.

6 Example

Figure 3 shows a small example from the "Rytterknægten" hill top on Bornholm. It is evident from the figure that in terms of graphical communication skill factor (defined as "matching the historical contours"), the final filtered contours are a dramatical improvement upon the raw contours.

7 Final remarks

As already mentioned, the algorithm presented here has as its success criterion *to meet the standards of a panel of cartographers*. Mathematically speaking, this is a very ill defined criterion, so it has evidently not been met through mathematical reasoning – rather by iterations of tests and discussions. When the panel was satisfied, the development process stopped, and the production started.

This means that the final algorithm carries traces from the steps taken during the development iterations: equivalent results can most probably be reached in a more elegant manner.

As it stands, however, the algorithm is fast enough to produce contours at 2.5 m equidistance from the entire Denmark (approximately 43 000 km^2) in about 12 hours, running on a plain office PC.

In other words improvements in terms of speed and elegance is interesting from an academic viewpoint, but from a production viewpoint the algorithm is *good enough* to leave as is.

Acknowledgments: A large number of cartographers gave valuable input to the process leading to the algorithm presented above. I would like to thank them all. Special thanks go to Tom Jensen, Tom Andersen, Peter B. Michaelsen, Hanne Mortensen, Jens Bo Rykov, Anna Brolund Jensen, Mogens Skov, Flemming Hjorth, Michael Eriksen, and Anders Nielsen, which have been particularly active in different ways, and at different times during the project.

References

David H. Douglas and Thomas K. Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *The Canadian Cartographer*, 10(2):112–122, 1973.

R. Fabbri, L. da F. Costa, J. C. Torelli, and O. M. Bruno. 2D euclidean distance transform algorithms: A comparative survey. *ACM Computing Surveys*, 40(1):2:1–2:44, 2008.

R. M. Haralick, S. R. Sternberg, and X. Zhuang. Image analysis using mathematical morphology. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 9(4):532–550, 1987.

Pietro Perona and Jitendra Malik. Scale-space and edge detection using anisotropic diffusion. In *Proceedings of IEEE Computer Society Workshop on Computer Vision*, pages 16––22, November 1987.

Pietro Perona and Jitendra Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629—639, July 1990. doi: 10.1109/34.56205.

Frank Warmerdam. GDAL—geospatial data abstraction library, 2010. URL http://gdal.org/.

Generalizing contour lines to various map scales

Hanne E. Mortensen

Abstract

This paper describes the generalization process carried out as the last step, in generating cartographic contour lines for different map scales. To implement the new initial contours and calculated height points into an established production workflow, some generalization algorithms has been developed and existing algorithms has been adjusted. The generalization process was carried out in an iterative trial and error process.

1 Introduction

In 2009 The Danish National Survey and Cadastre (Kort- og Matrikelstyrelsen KMS) released a new Digital Elevation Model (DEM), based on laser scanning technology. With this new DEM, it was obvious to create new contours with a higher accuracy, than the existing contours. The existing contours were, to some extend, based on data from the 19th-century military field surveys, which is described in article "*Højdegengivelse på topografiske kort – en historisk redegørelse*".

To create smooth cartographic contours, which are consistent with the absolute height from the DEM, a filtering algorithm to generalize contours and a computation program to generate updated height points, has been developed.

The contour generalization is based on filtering of the raw grid, followed by a gentle Douglas-Peucker generalization. The algorithm is described in article *"An Algorithm for Production of Cartographically Optimized Height Contours"*.

Parallel to this, the historical heights points were updated by relocating the points and inserting the accurate heights and position, measured by the laser scanning. The production procedures are described in article *"Updating historical cartographic contours and top points by automatic generated contours, based on the new laser-scanned Digital Terrain Model DEM-DK"*. These historical height points are hilltops, which are necessary to preserve for landscape interpretation. Therefore it is important that the updated heights points fit with the cartographic contours and is taken into consideration in the filtering process, to avoid modification of local extremes.

In order to use the generated contour lines in cartographic products, they need to be adjusted into a various set of data were they are presented.

To create visual cartographic contours for various map scales, it is necessary to reduce the amount of data and generalize the level of details, depending on required mapping scale and use. The amount of data should decrease as map scale becomes smaller.

To support the generalization, we used hydrological theme and named hilltops from our main databases, which are the most important landscape elements connected to elevation models. A simplified process flow of our contour line generation is shown in figure 1.

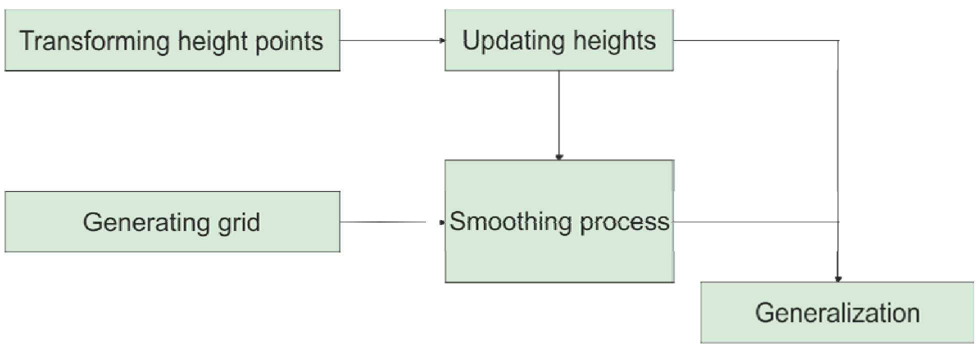


Figure 1: Process flow for production of cartographic contours and heights points based on DK-DEM. This is a simple flow; a more detailed process flow is shown in appendix A.

2 Methods

In Danish maps contours with an equidistance of:

- 1*m* is used in map scale 1:10.000.
- 2.5*m* is used in map scale 1:25.000.
- 5*m* is used in map scales 1:50.000 and 1:100.000.

Three sets of initial contour lines were generated from DK-DEM data, for use in different map scales.

The first generated set of contour lines in KMS was 2,5*m*. They were used for map scale 1:25.000 and 1:50.000, where every second contour line was used.

The second set was 1m contours, which was used for map scale 1:10.000.

The 5*m* contours generalized for 1:50.000 were far too detailed for use in 1:100.000, therefore a third set of contour lines were calculated with a different setup for 1:100.000 (see figure 2).

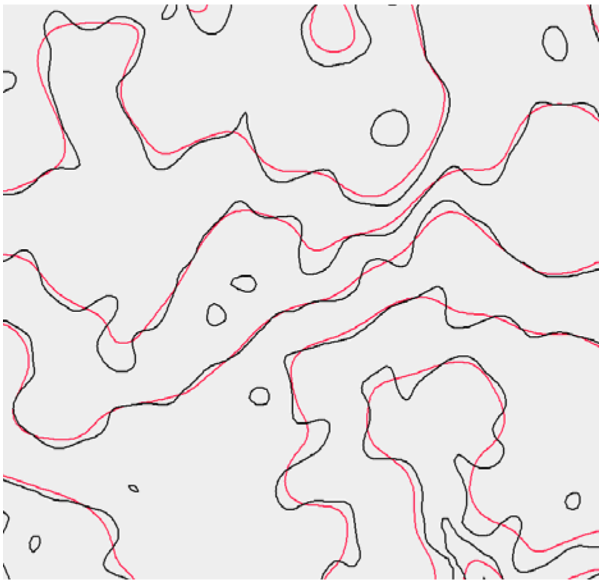


Figure 2: The figure shows the difference between the first generated 5m contours for map scale 1:50.000 (black color) and the later on generated 5m contours for map scale 100.000 (red color). There is less detail in the red contours.

2.1 Software

Each set of contours were separately generalized and each generalization processes were exceeded in 3 steps:

- Transformation and translation with FME
- Automatic cartographic treatment in GOTHIC
- Transformation and translation with FME

The software platform FME stands for Feature Manipulation Engine and is used for spatial data transformation and interoperability.

GOTHIC is a platform developed by 1. SPATIAL and is used for Cartographic Generalization. It's a module to detect and resolve conflicts between map objects for representation at the target scale.

2.2 Transformation and translation

The initial contours are calculated in 30 *km* x 30 *km* boxes, with a 1 *km* buffer around the boxes (32 *km* x 32 *km*), to make sure that they fit over the border line. Before cartographic treatment in GOTHIC, curves had to be snapped.

With a FME translation all 32 *km* x 32 *km* blocks are clipped to 30 *km* x 30 *km* blocks and snapped within a specified snap tolerance depending on map scale. Unfortunately some contours exceeded the snap tolerance and had to be edited manually. For contours with an equidistance of 2.5*m*, there were less than 100 loose endpoints, while the contours with an equidistance of 1m there was approximately 3000 loose endpoints.

Via FME transformation the contours were clipped and removed outside the borderline.

Furthermore the first set of contour lines (1:25.000, 1:50.000) were clipped at the coastline and removed if they were outside the coastline. This was a temporary solution for use in printed maps, where contours not are visible in the sea. In the subsequent datasets curves are not clipped. The conflict between coastlines and contours occurs, because the coastline and the contours are from two different datasets, where data are measured using different methods (see figure 3).

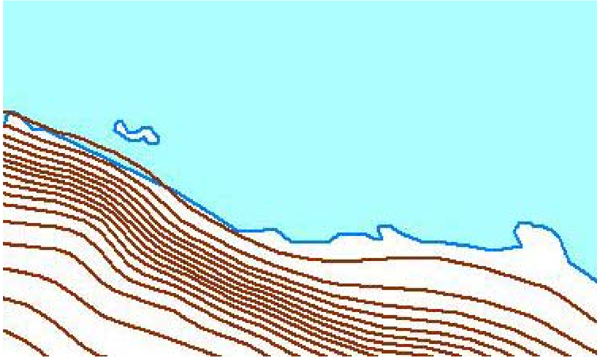


Figure 3: The picture shows the conflict between contours and coastlines. Contours are calculated on a laser scanned model and coastlines are photogrammetrically registered.

The new calculated height points were translated into the GOTHIC environment for cartographic treatment.

To support the generalization process in GOTHIC, we used coastline, lakes and watercourses from GeoDB, KMS geographic database, and named hilltops from SNSOR, the Danish Geoname database. (See figure 4).

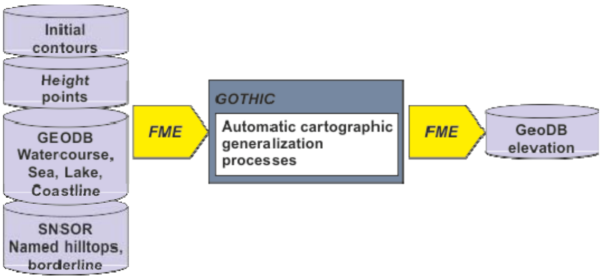


Figure 4: Workflow for generating automatic cartographic contours to products. (See Appendix 2)

2.3 Automatic cartographic treatment in GOTHIC.

The following automatic generalization processes were run on all 3 datasets, but with different parameters depending on map scales.

Small curves are removed, if the area of curve is less than the specified parameter for each scale, unless there is a top point, a bottom point or a named hilltop, which is described in SNSOR, inside the curve. (See figure 5)

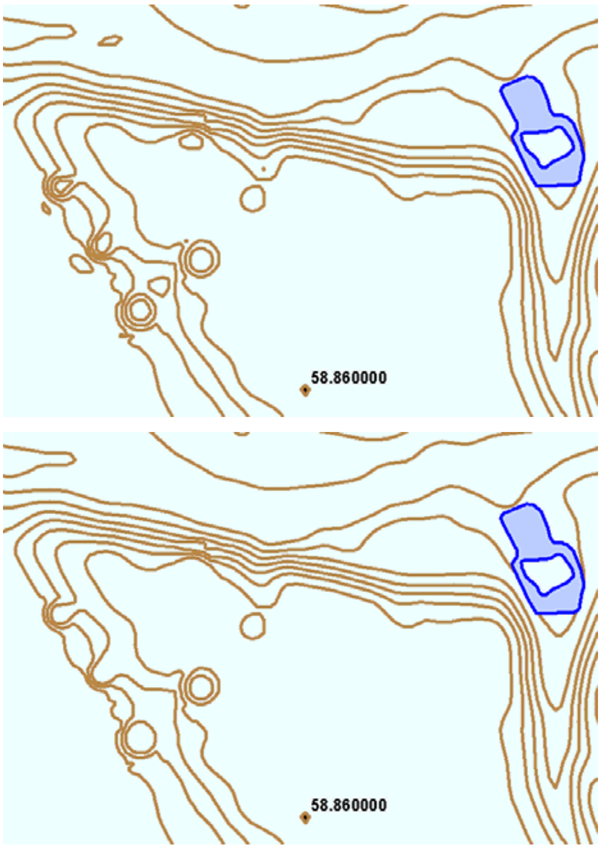


Figure 5: By comparing the upper panel to the lover panel approximately 11 small curves has been removed and the small curve around the height point is kept.

Points in local valleys are calculated and a bottom point is placed, except if it's inside lakes and watercourses. Points have no value; they are only visualized with a signature. This calculation process was extremely slow; therefore it was improved for 1:100.000. (The landscape in Denmark is very flat, but with a large number of small hills and valleys. We have a tradition to mark the small valleys).

Height points are removed, if there is a bottom point inside the same curve.

Small curves are extended, if it contains a top or bottom point, so that the signature for top and bottom points is visible inside the curve. Curves are not extended more than to a maximum limit parameter, specified for each scale, so they do not become too large relative to their original size, or will touch other curves. (See figure 6)

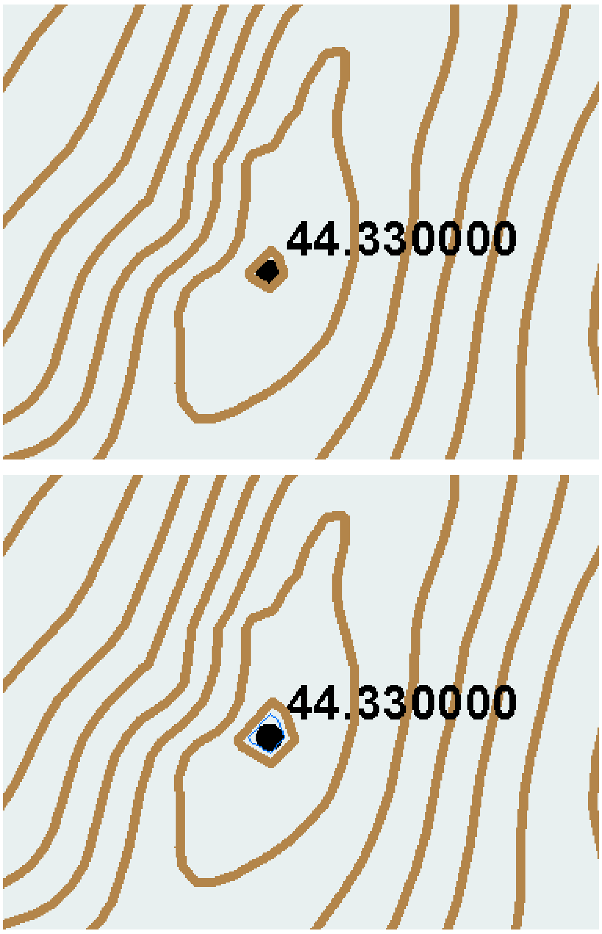


Figure 6: The curve around the height point has been extended, so that the symbol for elevation points becomes visible.

The height of top points is rounded to the nearest whole meter.

Height points are removed, if they have same value as the curve they are inside.

The following automatic process was run on the dataset 1:25.000-50.000 and 1:100.000.

Curves, height points and points in local valleys are removed, if they are entirely inside watercourses and lakes.

The following automatic processes were run on dataset 1:100.000.

Small curves outside the coastline are removed, if the curve is more than 50 % outside. (See figure 7)

Curves outside the coastline are moved to the coastline, so they follow the coastline. (See figure 7)

While the first contour lines (1:25.000, 1:10.000) were cut at the coast line, the generation of the 1:100.00 contour lines was adapted to the coast line.

Figure 7: By comparing the upper panel to the centre panel, two small curves in the sea and the small curve inside the lake has been removed. Lower panel: The

curves which were outside the coastline have been moved, so they follow the coastline.

2.4 Transformation and translation

With a FME translation the density of coordinates on the contours was reduced by a gentle Douglas-Peucker. Finally the contours were translated into a shape-format and stored, ready for use in products and delivery.

3 Final remarks

The contours and height points are implemented into the established workflow and we ended up with four sets of cartographic contours to four different map scales, which meet the demands of cartographers. The improvement made on the process generating point in local valleys is good enough, however from a production point of view it's not fast enough, a redesign of the algorithm should be considered.

Furthermore we have to reconsider, whether we could obtain a faster production if the hydrological theme was used in generating initial contours from the GRID, before the cartographic treatment.

All in all, we are very satisfied with the result of the contour generation. We have used these contour lines successfully in four different map products within the year 2011.

Acknowledgement: The process methods have been developed in cooperation with Peter West-Nielsen.

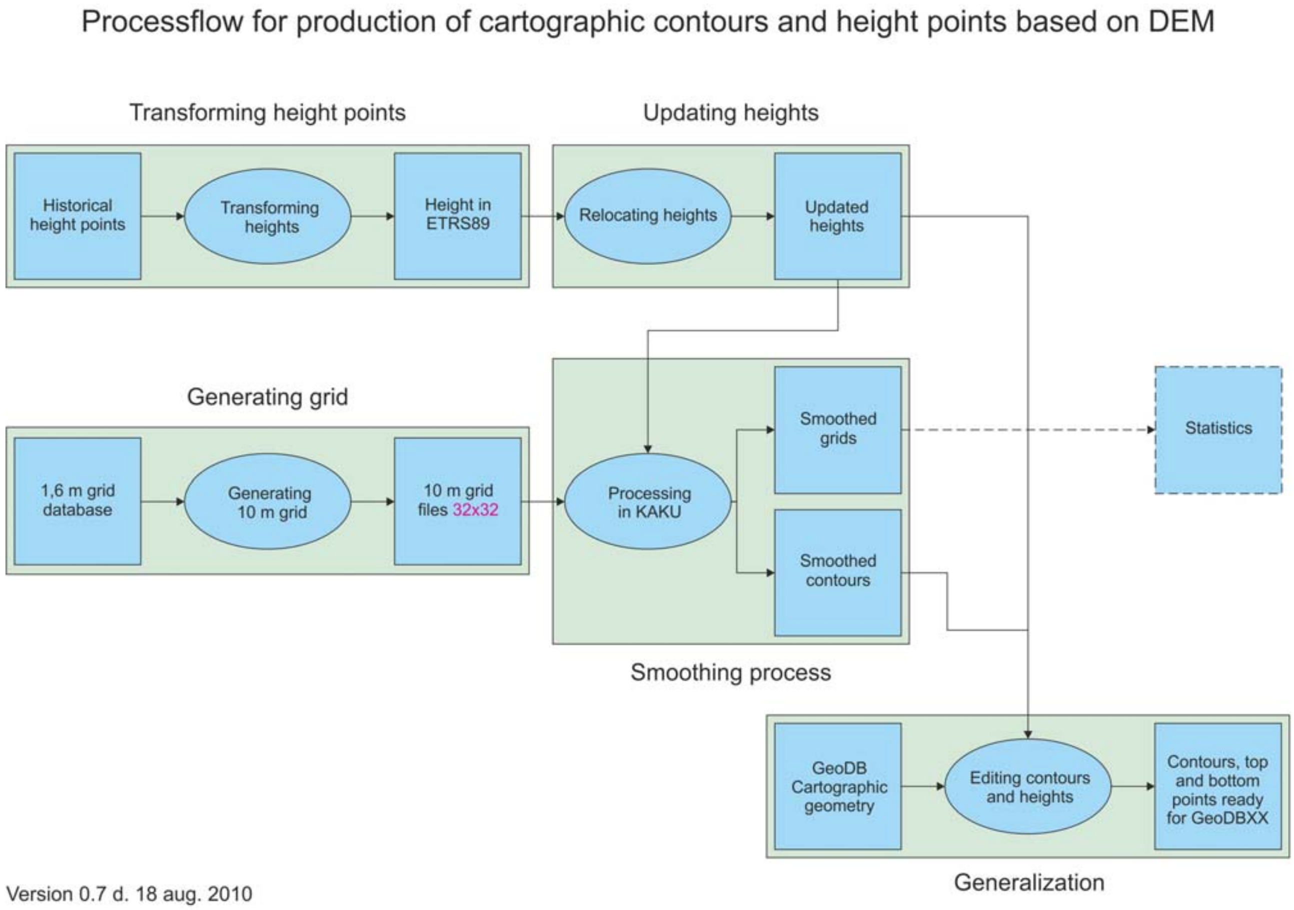
References

Brigitte Rosenkranz. Update of historical contours by automatic generated contours, based on the laser scanned Digital Terrain Model DK-DEM. Technical Report 13, National Survey and Cadastre KMS, 2011.

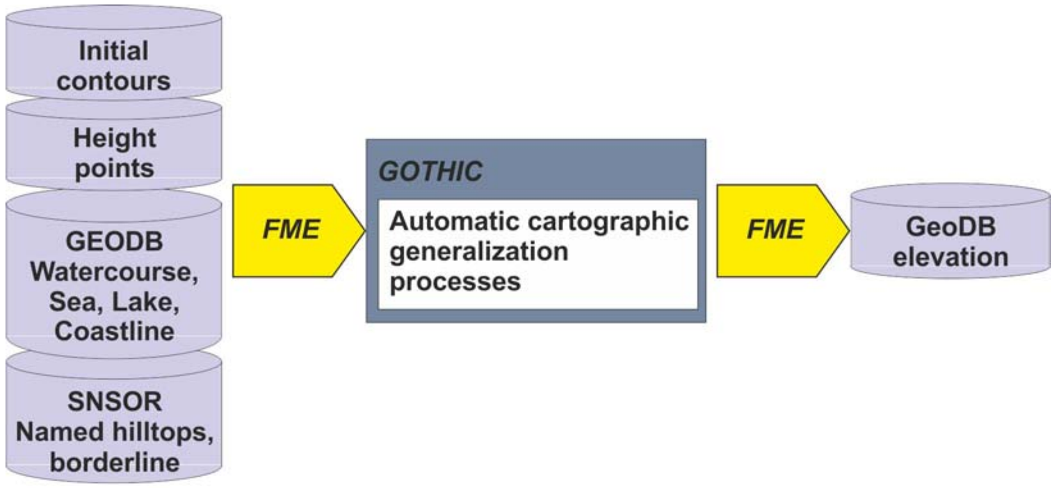
Thomas Knudsen. An algorithm for production of cartographically optimized height contours. Technical Report 13, National Survey and Cadastre KMS, 2011.

Peter Michaelsen. Højdegengivelse på topografiske kort – en historisk redegørelse. Technical Report 13, National Survey and Cadastre KMS, 2011.

APPENDIX A


Processflow for production of cartographic contours and height points based on DEM
Transforming height points
Historical height points
Transforming heights
Height in ETRS89
Updating heights
Relocating heights
Updated heights
Generating grid
1,6 m grid database
Generating 10 m grid
10 m grid files 32x32
Processing in KAKU
Smoothed grids
Smoothed contours
Smoothing process
Statistics
GeoDB Cartographic geometry
Editing contours and heights
Contours, top and bottom points ready for GeoDBXX
Generalization
Version 0.7 d. 18 aug. 2010

APPENDIX B



Initial contours
Height points
GEODB Watercourse, Sea, Lake, Coastline
SNSOR Named hilltops, borderline
FME
GOTHIC
Automatic cartographic generalization processes
FME
GeoDB elevation

Højdegengivelse på topografiske kort – en historisk redegørelse

Peter Michaelsen
pbm@kms.dk

Denne artikel handler om hvorledes højdeforhold er gengivet på trykte danske topografiske kort udarbejdet af Videnskabernes Selskab, Generalstabens topografiske Afdeling, Geodætisk Institut og Kort & Matrikelstyrelsen.

Begreber ved højdegengivelse på kort

Bakkestreger er linjer som benyttes til at vise en hældning i terrænet. Linjer vises fra top mod bund.

Lehmannske bakkestreger er et særligt måde at gengive hældninger på. Den saksiske militærgeograf Johann Georg Lehmann (1765-1811) præsenterede i 1799 et særligt bakkestregssystem. Stregerne tegnes i hældningsretningen. Stregernes tykkelse angiver hældningen i 9 step fra 5^0 til 45^0.

Kote. En kote er et punkt, hvis højde er bestemt ved opmåling. På de originale målebordsmålinger er der mange flere koter end på de trykte kort. Koteværdier er normalt med én decimal og målt i fod, halve meter eller meter. På trykte kort er koteprik og kotetal normalt gengivet i sort og tallet er ofte uden decimal.

Højdekurver er linjer i en given højde over havet. Ækvidistancen (højdeforskellen) kan være 5 fod, ½ m, 1 m, 1 ¼ m, 2 m, 2½ m eller 5 m. Nogle kurver kan være fremhævet f. eks. 0 m, 10 m, 20 m osv. eller 0 m, 25 m 50 m osv. Kurverne er tegnet på grundlag af koterne i forbindelse med opmålingen. Til at fastlægge kurvers forløb er anvendt indtegningsstokke (rød/hvide landmålerstokke) og håndniveau, der er et simpelt ”sigteinstrument” af træ eller metal med en libelle.
På kort trykt i flere farver er højdekurver normalt brune.

Kurvetal er tal på højdekurver for at hjælpe læsningen af kurvebilledet. Tallene har samme farve som højdekurverne. Tallene er normalt placeret, så de står op ad bakkens sider.

Kurveskrænt er en signatur, som er anvendt på en del kort. Hvor højdekurverne ligger meget tæt vises et murstensmønster. På nyere kort føres kurverne blot sammen eller en del mellemkurver udelades.

Lodskud er dybdeinformationer i havet og er overført fra søkort.

Dybdekurver er aftegnet fra søkort ved brug af en pantograf. Kurverne kan have en ækvidistance på 1 favn (6 fod) eller være kurver for dybde på 2, 4, 6 og 10 m.
De ældst kendte kort med dybdekurver viser floden Merwedes udmunding i Nordsøen og er udarbejdet i 1728 af Samuel Cruquius.

Dansk Normal Nul er indført i 1902. I Århus Domkirke er indmuret en bolt med værdien 5,615 m over Dansk Normal Nul. Før 1902 brugtes middelvandstand i nærmeste havn til fastsættelse af 0-værdi i et opmålingsområde samt Den Danske Gradmålingens foreløbige nivellement.

Interpolering
For at ænder kote- og kurveværdier fra fod til meter er der foretaget en manuel interpolering. Arbejdet er sket på tegnestuerne hos Generalstabens topografiske Afdeling og Geodætisk Institut. Interpoleringen er sket ud fra de oprindelige opmålinger og ved brug af omsætningstabeller mellem fod og meter og mellem meter og fod. Se "Bestemmelser for interpolering af kurver fra 5 fod til 2½ m" Topografisk afdeling november 1955 og "Instruks for kotering af 4-cm kort med samtidig overgang fra fod- til meterkoter" Topografisk afdeling oktober 1955.

Gentagne gange indskærpes det skriftligt at "interpolatørerne" (oftest topografelever, topografmedhjælper og topografassistenter) skal være omhyggelige med arbejdet. Det er klart at små toppe eller lavninger i terrænet enten udelades eller overdrives, når der rent grafisk skal findes mellemværdier. Der vil også være sket fejltolkninger og forglemmelser ved den manuelle omarbejdelse af højdekurverne.

Fod til Meter.

	0	1	2	3	4	5	6	7	8	9
0	0.000	0.031	0.063	0.094	0.125	0.157	0.188	0.220	0.251	0.282
1	0.314	0.345	0.377	0.408	0.439	0.471	0.502	0.534	0.565	0.596
2	0.628	0.659	0.690	0.722	0.753	0.785	0.816	0.847	0.879	0.910
3	0.942	0.973	1.004	1.036	1.067	1.098	1.130	1.161	1.193	1.224
4	1.255	1.287	1.318	1.350	1.381	1.412	1.444	1.475	1.506	1.538
5	1.569	1.601	1.632	1.663	1.695	1.726	1.758	1.789	1.820	1.852
6	1.883	1.915	1.946	1.977	2.009	2.040	2.071	2.103	2.134	2.166
7	2.197	2.228	2.260	2.291	2.322	2.354	2.385	2.417	2.448	2.479
8	2.511	2.542	2.574	2.605	2.636	2.668	2.699	2.731	2.762	2.793
9	2.825	2.856	2.887	2.919	2.950	2.982	3.013	3.044	3.076	3.107
10	3.138	3.169	3.201	3.232	3.264	3.296	3.327	3.358	3.390	3.421
11	3.452	3.483	3.515	3.546	3.577	3.609	3.640	3.672	3.703	3.734
12	3.766	3.797	3.829	3.860	3.891	3.923	3.954	3.986	4.017	4.048
13	4.080	4.111	4.143	4.174	4.205	4.237	4.268	4.300	4.331	4.362
14	4.394	4.424	4.457	4.488	4.519	4.551	4.582	4.614	4.645	4.676
15	4.708	4.739	4.771	4.802	4.833	4.865	4.896	4.928	4.959	4.990
16	5.021	5.053	5.084	5.116	5.147	5.179	5.210	5.242	5.273	5.304
17	5.335	5.367	5.398	5.430	5.461	5.493	5.524	5.555	5.587	5.618
18	5.649	5.680	5.712	5.743	5.774	5.806	5.837	5.869	5.900	5.931
19	5.963	5.994	6.026	6.057	6.088	6.120	6.151	6.183	6.214	6.245
20	6.277	6.308	6.340	6.371	6.402	6.434	6.465	6.497	6.528	6.559
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

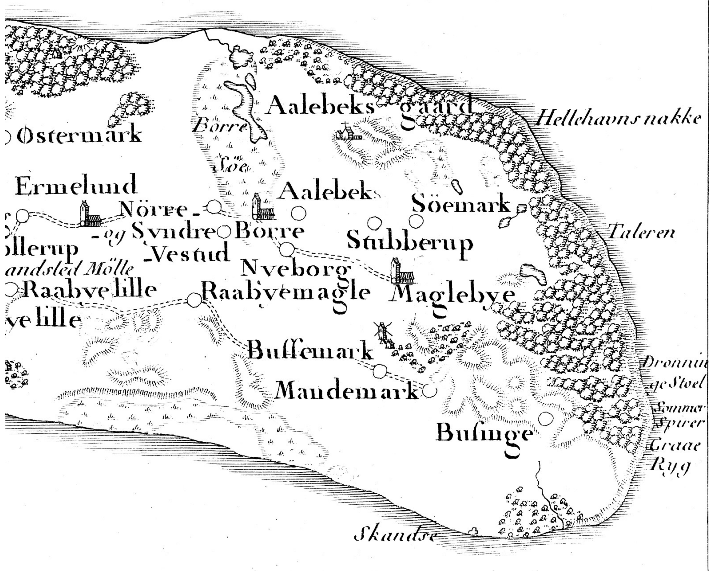
Del af omsætningstabel: fod til meter.

Meter		Fod	Meter		Fod
2½	=	8.0	47½	=	151.3
5	=	15.9	50	=	159.3
7½	=	23.9	52½	=	167.3
10	=	31.9	55	=	175.2
12½	=	39.8	57½	=	183.2
15	=	47.8	60	=	191.2
17½	=	55.8	62½	=	199.2
20	=	63.7	65	=	207.1
22½	=	71.7	67½	=	215.1

Del af omsætningstabel: meter til fod.

Historisk forløb

På **Videnskabernes Selskabs** kort (1757-1842) gengives bakker og slugter ved brug af simple bakkestreger. Kortene er uden koter og højdekurver. En følge af Napoleonskrigene var en omorganisering af den danske hær og større interesse for brug af geografiske informationer i hæren. I 1808 oprettedes Generalstaben, hvor en af underafdelingerne var en Generalkvartermesterstab, som bl.a. skulle tilvejebringe topografiske og statistiske oplysninger til hæren. Det erkendtes hurtigt at Videnskabernes Selskabs kort ikke var tilstrækkelige for en moderne hær. Kortene skulle være i større målestoksforhold, have informationer, der var relevante for hæren og have højdeinformationer, der var vigtige ved fremrykning i et terræn.



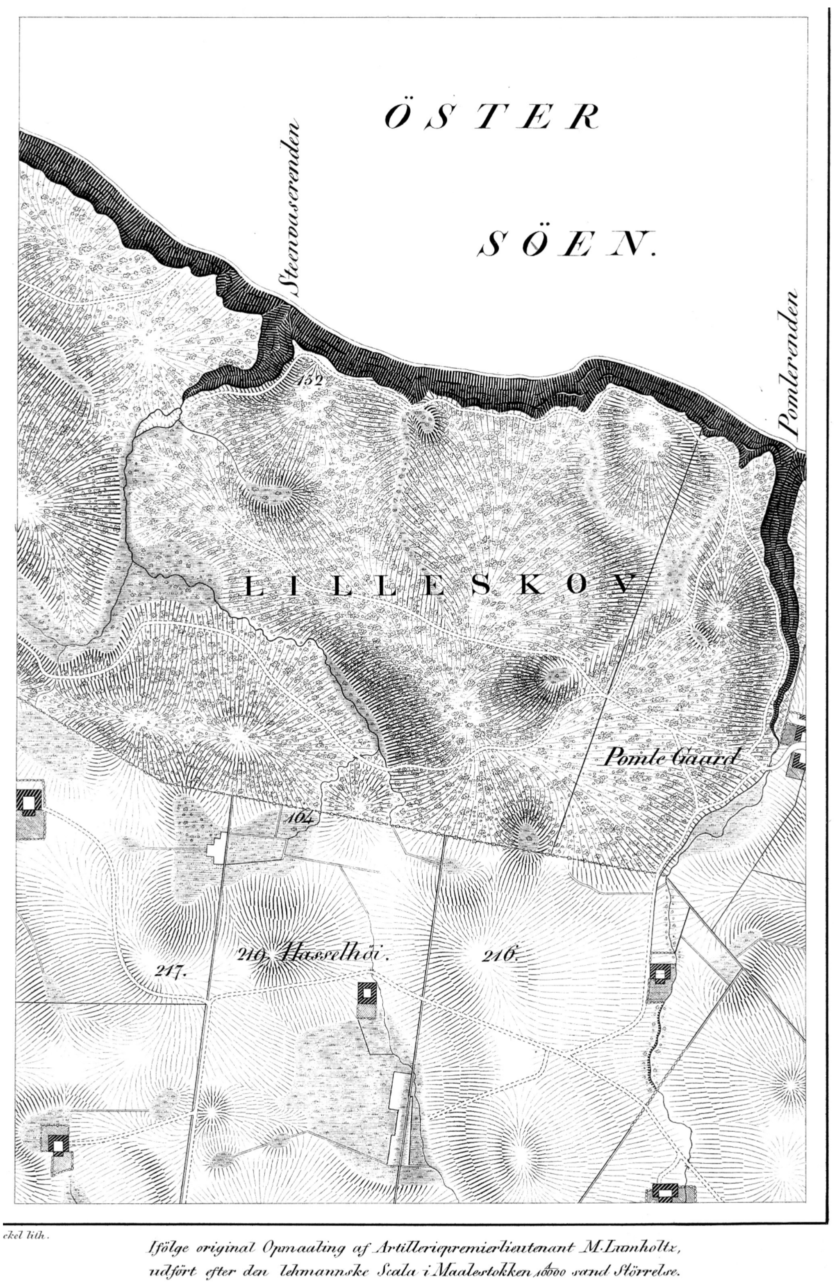
Bakkestreger på kort over det østlige Møn. Kort 1:120 000 fra Videnskabernes Selskab 1776.

Generalkvartermesterstaben gik i 1808 i gang med en opmåling og rentegning af Sjælland i 1:20 000. Grundlaget for opmålingen var Videnskabernes Selskabs kort. Opmålingen er udført med måleborde og ved afskridtning i perioden 1808-1841 med Videnskabernes Selskabs målinger som grundlag. Højdegengivelse er med lehmannske bakkestreger. Der er ingen koter. På den sydlige del af Lolland måles der dog højder ind med koter og 5 fods kurver 1840-1842.

I 1842 omorganiseredes forsvaret. Generalstaben (GS) oprettedes og under den **Generalstabens topografiske Afdeling**, der fik ansvar for topografisk kortlægning. Opmålingen fortsattes. I perioden 1842-1860 måles der videre på opmålinger fra 1831-1841. Der koteres og indtegnes 5 fods højdekurver, dog ikke i større skove.

I 1846 udsender Generalstaben et ”Indledningshæfte til det topographiske Atlas over Danmark med hertugdømmet Slesvig”. Atlasset vil blive udgivet som kortblade i 1:80 000. Blad 13 Nysted er udarbejdet i 4 forskellige versioner: ”Terrainfigurering efter den Lehmannske Vinkelmetode, uden Terrainfigurering, Terrainfigurering udført med horisontale equidistante Kurver og uden Terrainfigurering med Sogneillumination.” Generalstaben er villig til at fremstille alle 4 varianter af kortbladene, men man vil vurdere ”Publicums Dom og Afsætningen” af hensyn til økonomi og mindre nødvendige eller endog overflødige opgaver. ”Bestillinger modtages alene af den Schubothske og Reitzelske Boghandel i Kjøbenhavn, hvor Heftet forefindes. Til Eftersyn vil det ogsaa søges henlagt i Odense, Aarhuus, Flensborg og Kiel”.

1845-1872 tryktes der 29 atlasblade i 1:80 000 over Sjælland, Fyn og øerne. Højdegengivelsen er 10 fods kurver, dog ingen kurver i skove.



Lehmannske bakkestreger på kort i 1:10 000 over Lilleskov ved Møns Klint. Hvor bakkestregerne er mørkest, er hældningen på mere end 45^0.

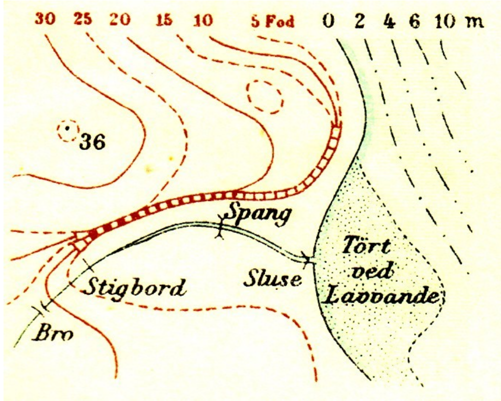
Målebordsblade
6 kort i 1:20 000 ved København måltes 1847-1853 og tryktes 1854-1857 med rettelser til1855. Højdetal og 5 fods højdekurver uden for skove.

I 1846 begyndte man en opmåling på grundlag af matrikelkort i 1:4 000 pantograferet ned til 1:20 000, de såkaldte byblade, som omfatter et ejerlav. Koter måltes i fod med én decimal og der indtegnedes 5 fods højdekurver. Udgivelsen af målebordsblade i 1:20 000 indledtes i 1866. Kotetal er i fod uden decimal og der er 5 fods kurver. 5, 15, 25 osv. stiplede kurver. Fulde linjer for 0, 10, 20 osv. Højder, z-værdier, bestemtes ud fra vandstandsmålinger i havne.

I erkendelse af at især de sjællandske målingers nøjagtighed ikke var egnet til udgivelse som kort i 1:20 000 påbegyndtes en ny måling i 1887. Z-datum: GS nivellement fra vandstandsmåler på Nordre Toldbod i København. Lamberts konformkoniske projektion, GS-projektionen, anvendes. Målebordsmålingen skete på blankt papir med indkonstruerede planfikspunkter og kortbladshjørner for ¼ højt målebordsblad. Koter måltes i fod med én decimal. Der måltes 5 fods ækvidistante højdekurver. Målingen begyndte i 1887 på Sydfalster.

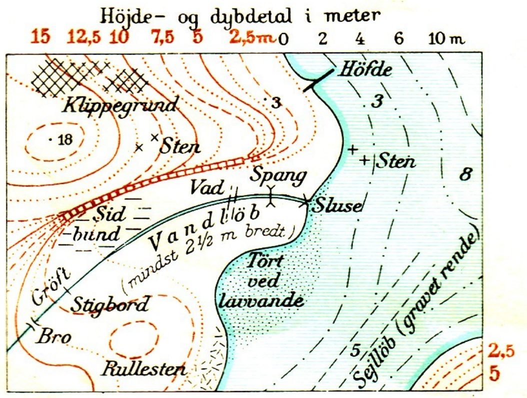
I 1889 gik man over til at måle i metersystemet. Koterne blev målt i ½ m med én decimal, altså med 5 cm nøjagtighed. Kurveækvidistancen blev 5/2 m = 2½ m. Ved rentegningen til tryk anførtes koteværdier i meter. På nogle kortblade med én decimal på andre uden decimal. 2½, 7½ osv. kurver vistes med stiplet linje, 0, 5, 10 osv. vistes med fuld linje. Helsingør og København og omegn (Landbefæstningen) måltes i 1:10 000 og med en kurveækvidistance på 1¼ m. Mellemkurver med prikket linje.

Anden nymåling fortsatte på den sydlige del af Fyn i 1902. Kurveækvidistancen ændredes til 2 m. Der benyttes nu DNN som z-datum. Målingen sker i 1:15 000. "Koteringen sker i Meter med 2 Decimaler, Kotetallene paaskrives Bladet med 1 Decimal. Kikkerthøjden og de Koter, ved Hjælp af hvilken den beregnes, bestemmes med 2 Decimaler." (Instruks for Maaling og Revision i 1:15 000, uden år, ca. 1930.) Kurverne vises på de trykte kort med fulde linjer for 2, 4, 6 og 8 m, 0, 10, 20 osv.



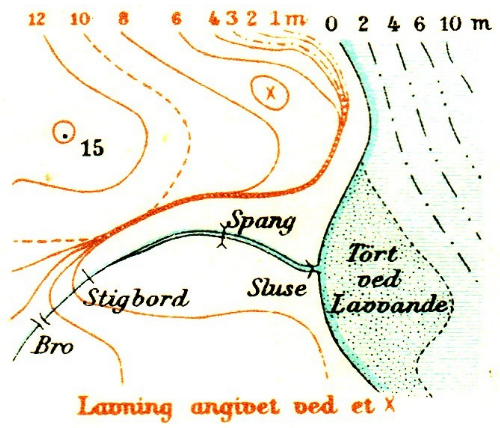
5 fod højdekurver. Vignet på målebordsblad M 3703 Mandø (Sydvestjylland). Målt 1870. Rettet 1938. Vejrevision 1946. Enkelte rettelser 1954. Trykt 1967. Stiplede kurver for 5 fod, 15 fod, 25 fod. Kotetal i fod.

Illustrationerne er en del af signaturforklaringerne på målebordsblade.



1,25 m højdekurver. Vignet på målebordsblad M 3129 Glostrup (København). Målt 1897. Rettet 1965. Trykt 1966.
Højderne er målt i meter. Prikkede kurver for 1,25 m, 3,75 m, 6,25 m osv Stiplede kurver for 5 m, 10 m, 15 m osv. Kotetal uden decimal.

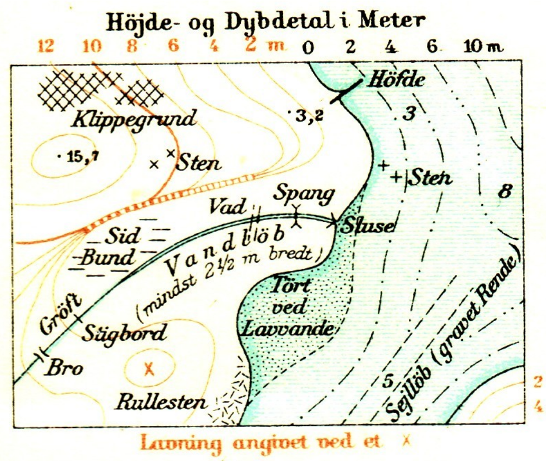
vises med stiplede linjer. På de trykte kort angives koter med én decimal. Sønderjylland måles på samme måde 1924-1938, dog måles der også 1 og 3 m kurver i det vestlige Sønderjylland. 1 og 3 m kurverne vises på trykte kort med streg, prik, streg, prik.



1 m, 2 m, 3 m, 4 m, 6 m osv højdekurver. Vignet på målebordsblad M 3804 Rejsby (Sønderjylland). Målt 1936. Vejrevision 1946. enkelte rettelser 1966. Trykt 1968. Højderne er målt i meter. Prik streg kurver for 1 m og 3 m. Stiplede kurver for 10 m, 20m osv. Kotetal uden decimal.

Mellem 1910 og 1921 er kortene over den nordlige del af Fyn omarbejdede, så der er 2 m kurver og koter med højder i meter. 25. oktober 1945 (GI 3/45) bestemmes det, at for de nyreviderede målebordsblade over Bornholm skal der udarbejdes 2 m kurver (og kotetal i meter.) I 1949 er det planen at ændre fodkurverne på de jyske målebordsblade til 2 m kurver. (Se bogen: Det danske Geodætiske Instituts Kort, 1949, side 5.)

På grund af den langsomme fremdrift i produktionen af 4 cm kort fortsattes der med rekognoscering til målebordsblade (Rettet 19--) til 1958. (M 3129 Glostrup er dog som et forsøg rettet 1965 på grundlag af fotogrammetri.) Der tryktes målebordsblade med "Enkelte rettelser" til omkring 1970. På nogle kort i Midtjylland ændredes højdeinformationerne ved "Rettet" eller "Enkelte rettelser" fra fod til meter. Kurveækvidistancen på disse kort er 2½ m. Der er forskel på hvorledes kurverne er vist. Nogle målebordsblade har stiplede kurver for 2½ og 7½ m, andre har kun fulde kurver med ekstra tykke kurver for 0, 25, 50 m osv. Atter andre har tynde kurver for 2½, 7½ m osv, medium kurver for 5, 10, 15 og 20 m og tykke kurver for 0, 25, 50 m osv. Årsagen til højdekurvernes uensartede udseende er skift fra tegning med tusch på papir til gravure på glasplader med emulsion og senere skift til pvc-folie med gravurehinde. Det kunne spares arbejdstid ved at vise flest mulige kurver som fulde linjer. Der er kort, som har koter i meter med én decimal, andre kun i hele meter.



2 m højdekurver. Vignet på målebordsblad M 5336 Neksø (Bornholm). Målt 1886. Rettet 1946. Enkelte rettelser 1961. Trykt 1963. Højderne er målt i fod og siden interpoleret til meter. Tykke kurver for 10 m, 20m osv. Kotetal med én decimal.

4 cm kort

Danmarks indtræden i NATO i 1949 fik stor indflydelse på de topografiske kort. Målestoksforhold skulle ændres fra 1:20 000 og 1:40 000 til 1:25 000 og 1:50 000. I 1953 påbegyndtes rekognosceringen til 4 cm kort på Djursland. Produktionen af 4 cm kort var meget ambitiøs. Kortene skulle nytegnes og trykkes i 6 farver, det geometriske grundlag var i Jylland stadig de gamle byblade. Endeligt i 1957 tryktes de første 4 cm kort. Kurver og koteværdier var interpoleret fra bybladene. 2½ og 7½ m kurver stiplede, 5, 10, 15, 20 m fulde linjer og 0, 25, 50 m fulde tykke linjer med kurvetal. I Sønderjylland interpoleredes 2 m kurver til 2½ m kurver. Alle 4 cm kort har 2½ m kurver. Kotetal er i meter med én decimal eller uden decimal.

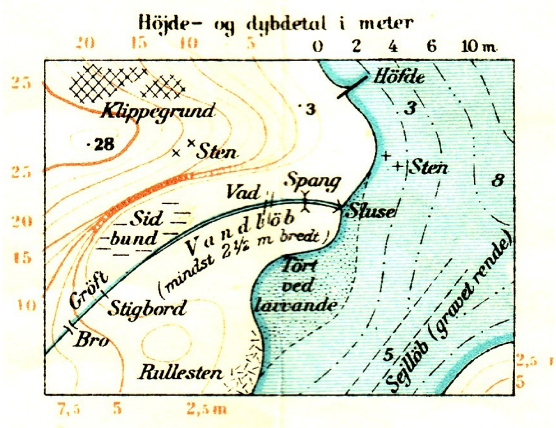
Fotogrammetrisk udtegning

I 1965 påbegyndtes den fotogrammetriske opmåling af Danmark i 1:10 000 som grundlag for 4 cm kort. Ved den fotogrammetriske opmåling måltes koter med placering som på de foreliggende trykte kort. Der måltes ikke nye højdekurver, da et sådant arbejde ville være mindst lige så tidskrævende som den lineære udtegning. Kun særlige tilfælde udtegnedes der nye højdekurver. Det skete ved større udgravnigsområder (grusgrave, brunkulslejer mm) og i nogle klitområder. "Ved større rettelser kontaktes gruppelederen eller kartografen, som herefter træffer bestemmelse, om det foreliggende billedforelæg er tilstrækkelig, eller en fotoflyvning i lav højde kan blive aktuel" (Fotogrammetrisk sektion 9. aug. 1973).

Det tidligere kurvebillede genanvendtes på alle kortblade. På Fyn og Bornholm, hvor der ikke forelå 2½ m kurver, interpoleredes 5 fod kurver eller 2 m kurver til 2½ m kurver. I områder, hvor der var interpoleret fra 5 fod til 2 m, interpoleredes der 2½ m kurver fra grundmaterialets 5 fod kurver. I forbindelse med rekognosceringen af kvarterne (1/4 målebordsblad = 1/9 af et 4 cm kort) tilpassedes kurver til vandløb, søer mv.

I nogle få områder (skove og slugter) konstateredes der så stor mispasning mellem kurver og det øvrige kortbillede, at kurverne måtte tilpasses partielt. Tilpasningen skete ved at kvartens kurvebilag klippedes op langs veje, jernbaner og lignende. De mindre dele tilpassedes så det udtegnede kort, så slugter kom til at passe med vandløb og dale og lavninger med søer.

Den fotogrammetriske opmåling kom til at omfatte hele landet bortset fra Sønderjylland.



2½ m højdekurver. Vignet på målebordsblad M 2211 Vejerslev (Midtjylland). Målt 1877. Rettet 1952. Enkelte rettelser 1957. Trykt 1963.
Højderne er målt i fod og siden interpoleret til meter. Tynde kurver for 2½ m, 7½ m 12½ m osv. Medium kurver for 5 m, 10m, 15 m osv. Tykke kurver og kurvetal for 25 m, 50 m, 75 m osv. Kotetal uden decimal.

Fotogrammetrisk udtegnede højdekurver ved Karlstrup Kalkgrav, 1:25 000. Udtegnet 1987 til brug for 4 cm kort 1513 II NV. Kotetal er skrevet med syd opad.

4 cm kortene er i nederste venstre hjørne forsynet med oplysningen ”Kurveplan efter ældre bordmåling”. Det dækker altså over at højdeinformationerne er meget gamle i forhold til de andre topografiske informationer på kortet.

Kort25

Første version af Kort25 kom i 2003, da der var behov for et digitalt kort med udseende og indhold som 4 cm kortet. Kortet opdateres årligt på grundlag af grunddata. Kortets højdekurver er digitaliserede og vektoriserede kurver fra seneste udgave af 4 cm kort. Det betyder at kurverne i det meste af Jylland og dele af Fyn har rod i fodkurver fra 1862-1887, og at kurverne på Sjælland er 2½ m kurver fra omkring 1900! Kort25 er fra 2010 med 2½ m højdekurver og koter fra den digitale højdemodel.

1:50 000 og mindre målestoksforhold

Det skal nævnes at Danmark 1:50 000, 2 cm kort, i perioden 2001-2008, hvor kortet fremstilledes som digitalt vektor produkt, havde 5 m kurver, der var genereret ud fra et højdegrid. Grundlaget for højdegridet var z-værdier på veje og andre overfladelinier i TOP10DK samt digitaliserede koter fra 4 cm kort. Der var problemer med at få kurvebilledet til at ligne landskabet og tidligere korts højdegengivelse. Terrænet blev udjævnet i højdemodellen. Stejle skrænter blev til jævne bakker og små top- og bundkurver forsvandt fra kortet. Alle øvrig trykte kort har haft målebordsbladenes højdekurver og koter som grundlag for højdegengivelser.

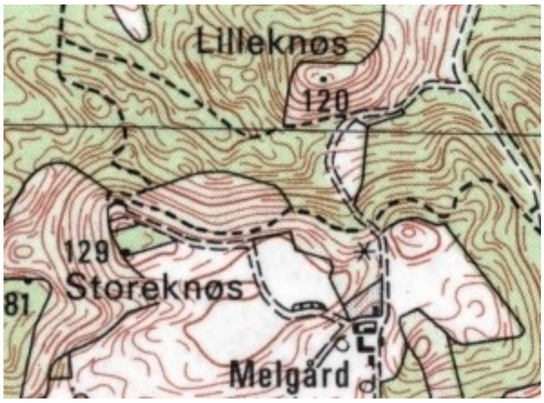
Kort 1:5 000 og 1:10 000

Generalstabens topografiske Afdeling udgav 1905-1911 bykort over Esbjerg, Horsens, Randers, Svendborg og Aalborg i 1:5 000. Kortene er med ½ m højdekurver og med koter i meter med én decimal. I samme periode tryktes kort i 1:10 000 med 1 m højdekurver over Randers og omegn og over Borris skydeterræn.

Om højdeinformationerne fra disse kort har været grundlag for 2½ m kurver til målebordsblade og 4 cm kort, er der ikke informationer om i arkivmaterialet.

Lilleknøs og Storeknøs på målebordsblad, 4 cm kort og Kort25

Målebordsblad N 13 Borresø. Målt 1872 og 1876. 5 fod højdekurver.



4 cm kort 1214 I SV Salten Langsø. Fotogrammetrisk udtegnet, flyvefotografering 1985. Kurver efter ældre bordmåling. Kompletteret i marken 1987. 2½ m højdekurver og kotetal i meter uden decimal. (Interpoleret efter målingerne fra 1872 og 1876.)

Kort25 udgave 2010 med højdegengivelse på grundlag af DHM. 2½ m højdekurver og kotetal i meter uden decimal.

Konklusion

Højdetemaet havde stor bevågenhed på de topografiske kort i slutningen af 1800-tallet og begyndelsen af 1900-tallet. Højdeinformationerne var essentielle for en hær, der bevægede sig frem til fods eller på heste. Højdekurverne var vigtige i forbindelse med Københavns Landbefæstning, hvor området mellem Lyngby og Øresund skulle oversvømmes af vand fra Furesø. Målebordsbladene have også stor civil anvendelse ved planlægning af jernbaner, veje og landvinding omkring forrige århundredskifte.

4 cm kortet blev aldrig et hærkort på samme måde, som målebordsbladene havde været det. 2 cm kortet med 5 m højdekurver blev det væsentligste hærkort efter Danmarks indtræden i NATO. Forsvarets interesser for højdeinformationer nedtrappedes, antageligt som følge af at hæren motoriseredes, og det må nok ses som en at årsagerne til at højdeinformationerne på 4 cm kort aldrig fik en kvalitet, som de andre informationer på kortene. De gamle kurver var gode nok, til det kortene forventedes brugt til, nemlig orientering i terrænet.

I løbet af 1980-erne kom der en stigende interesse for digitale højdeinformationer bl.a. i forbindelse med opsætning af master til mobiltelefoni. Med retablering af naturområder og risiko for stigende vandstrand er der forøget interesse for højdeinformationer. Der er i 2007 fra fly foretaget en laserscanning af hele Danmark, og ud fra den er det nu muligt fremstille terrænmodeller og overflademodeller med stor nøjagtighed. Terrænmodellen vil fremover være grundlaget for nye koter og højdekurver på de topografiske kort.

Kilder:

Bøger:
Generalstabens Kort, 1910
Den Danske Generalstabs Kort, 1921
Det Danske Geodætiske Instituts Kort, 1940
Det Danske Geodætiske Instituts Kort, 1949
Geodætisk Institut 1928-1978, 1978

”Jørgensen” oversigt over opmålinger 1809-1899 udarbejdet 1940-1944 af fhv. depotforvalter H. Jørgensen, Geodætisk Institut

Instrukser, bestemmelser, befalinger, cirkulærer, korrespondancer mm. i LANs arkiv i KMS:
Ringbind: D/20 målebord, D/40 atlasblad
Ringbind: D/25 før 1973
Ringbind: D/25 1974-1979

Specifikationer D/25 (G) Danmark 1:25 000, april 1977, med rettelser til 1989
Instruks for komplettering D/10, 1979, med rettelser til 1989


Agri
137
Agri Baunehøj
MOLS BJERGE
132
44
Trehøje
(Dyrehøje)
127
Bogens Hoved
Viderup
111
41
Fuglsø
KOLSKÆR GRUND
EGSMARK GRUND
PIKKELGRUND
Egsmark
Dråby
Karls
53
Skelhøj
61
EBELTOFT
Havn 4,5m
PLADEN
Fl.G.5s
Fl.R.5s
Sp