# AUTOMATIC MESH GENERATION FOR 2D HYDRODYNAMIC FLOOD MODELS FROM HIGH-RESOLUTION DIGITAL ELEVATION DATA

STEFAN KURZBACH, NICO SCHRAGE

*Department of River and Coastal Engineering, Hamburg University of Technology, Denickestr. 22, 21073 Hamburg, Germany*

The Elbe estuary in Germany provides an example of a large and challenging area for two-dimensional hydrodynamic flood modeling. We will give our practical experiences in the flood model discretization process, which comprises managing and integrating high-resolution topographic and bathymetric data from different sources, dynamic tiling of the terrain model, automatic detection of structural features, and triangulation of a computational mesh. The discretization process has been automated as a chain of OGC-conforming Web services, which enables its integration with an existing spatial data infrastructure.

## INTRODUCTION

Creating high-quality unstructured meshes for two-dimensional, hydrodynamic flood simulation, e. g. as required in [1], is a comprehensive and tedious task. Specialized meshing software has previously been developed to automate this process based on an interpolation terrain surface and a skeleton of the mesh structure [2], [3]. Even though high-resolution digital elevation models (DEM) are now readily available using remote sensing methods (airborne, spaceborne, and shipborne), the mesh structure still needs to be defined mostly by hand. Furthermore, the combination of fluvial topography above and below the water surface often creates the need for integrating DEMs from different sources. A spatial data infrastructure (SDI) seeks to simplify discovery and use of such distributed data based on policies and standardized technologies, e. g. common data formats and Web services.

### Research Objectives and Overview

Our goal is to show a practical procedure for automatic unstructured terrain discretization used in two-dimensional flood simulation and mapping. In this paper, we focus on the technical aspects of the procedure including (1) data management and integration of multiple terrain data sets in a geographic information system (GIS) and (2) the detection of structural features (breaklines). The methodology incorporates the application of data and Web service standards issued by the Open Geospatial Consortium (OGC), which mandates DEM provisioning by means of a Web Coverage Service (WCS).

### Characteristics of the Study Area

The 140 km long Elbe estuary stretches between the city of Hamburg and the North Sea. Being a tidally influenced water body, the estuary is characterized by a combination of

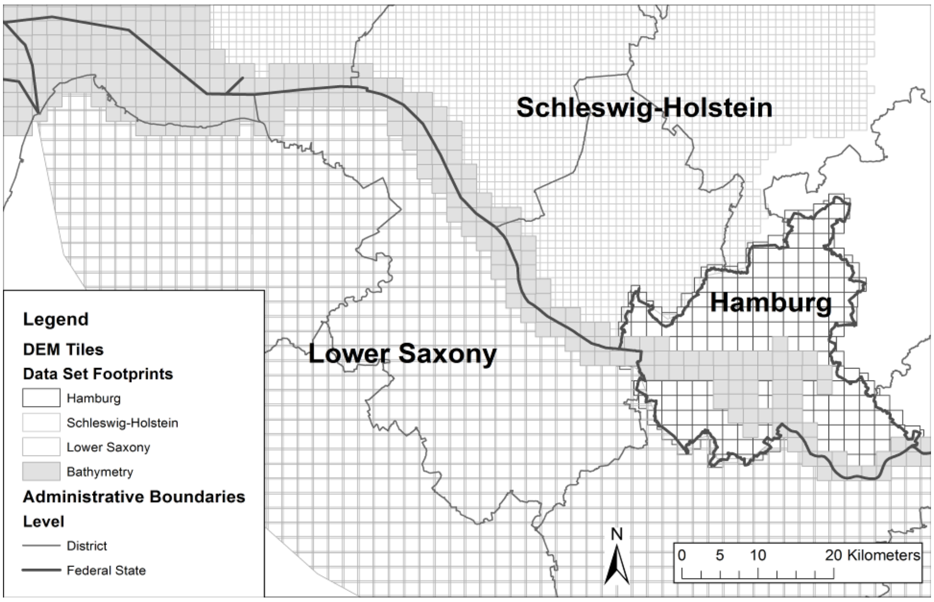

Figure 1: Terrain data repository of the Hamburg Metropolitan Region

open sea and river stretches, mud flats, large marshlands, floodplains and an inland river delta near Hamburg. The river is an important waterway, with substantial man-made modifications made to the system at the heart of the Hamburg Metropolitan Region. These include, for example, embankments, deepened shipping channels, ports, groyne fields, and artificial drainage ditches.

The Elbe estuary borders on three federal German states: Schleswig-Holstein, Lower Saxony, and the Free and Hanseatic City of Hamburg – each having their own SDI (see Fig. 1). The river center line forms part of the state boundaries. This creates the need to integrate different DEMs for the left and right river banks in a common repository. A recent effort resulted in the establishment of an SDI for the Hamburg Metropolitan Region. We expect that a DEM is going to be part of its data repository and provided via WCS in the near future. As there is currently no such WCS available, the next section shows our own experiences regarding DEM management and integration.

## MANAGEMENT OF THE DIGITAL ELEVATION MODEL

Three distinct data sources of topographic data with varying spatial extent, tiling, and resolution covering the Hamburg Metropolitan Region have been provided to us by the state authorities for use in the project KLIMZUG-NORD:

- DEM for the southern part of Schleswig-Holstein in tiles of 1·1 km (1 m raster, ca. 3500 km²)
- DEM for the state of Hamburg in tiles of 2·2 km (1 m raster, ca. 900 km²)

- DEM for the northern part of Lower Saxony in tiles of 2·2 km (5 m raster, ca 7300 km²)

Additionally, we have employed one source of bathymetry data of the Elbe estuary in the form of spot measurements (point cloud). It covers a total river area of ca. 1000 km² in a horizontal resolution ranging between 200 m and 50 cm.

**Integration of Bathymetric and Topographic Data**

All topographic data sets include the water surface elevation of the Elbe river at the time of measurement. However, due to the tidal nature of the river and different survey times, the water surface is neither flat nor continuous. This creates a difficulty in determining the boundary between the river and its foreland. As a guideline, we have decided that bathymetry information should have precedence over the topography where it is available.

Generally, the areas of the DEM which needed to be replaced by the bathymetry have been demarcated by a polygonal region. A typical scenario can be seen in Fig. 2. The region with false elevation values (shaded) has been delineated by a separation line derived from contours of the topography DEM. The contour level separating water from non-water ranged between −2 and +2 m, so we had to select the correct contours for the separation line manually for all areas of the DEM, simplify the line, and then connect the segments to form a closed polygon. As rivers have a natural slope from upstream to downstream, contour lines are not a good choice when dealing with long river sections. A fully automated method of finding the separation line between the two datasets would be of advantage, but this was out of scope for our research.

In the next step, the separation line was used to create a bathymetry TIN. It acts as a seamline between the two data sets by enforcing the topographic elevation at the boundary of the bathymetry data. Nevertheless, any other seamline could have been selected. Our experiments have shown that if the seamline is not a single contour line of the topography, its elevation should be interpolated from the DEM prior to the creation of the bathymetry TIN. Due to a current performance problem in ArcGIS when interpolating elevations to a line in the natural spacing of the DEM, we had to fall back to interpolation at equidistant points along the line, which did not give satisfactory results in all cases.

All spot measurements outside the boundary were discarded in order to obtain a smooth transition. Fig. 3 shows the bathymetry TIN we created together with the topography DEM. We conducted the data integration step for all DEM tiles shown in Fig. 1, but the study area for the subsequent feature detection process was limited to the extent shown in Fig. 2 and 3.

**Data Management in ArcGIS**

The terrain data repository was managed inside an ArcGIS 10 geodatabase. All land surface DEM tiles were loaded into a *mosaic* data set, which stores a catalog of rasters and their outline polygons (*footprints*) that are dynamically assembled into a single big raster. A mosaic provides the possibility to define *seamline* polygons for all tiles. We erased the footprint polygon areas intersecting our separation boundary polygon and set the resulting

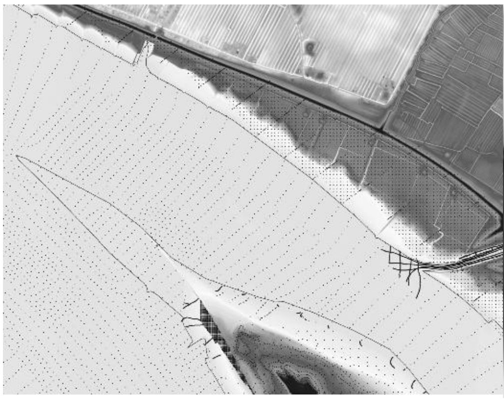

Figure 2: Separation line between land and water (spot measurements).

Figure 3: Topography and bathymetry integrated into one DEM.

polygons as seamlines for the DEM tiles. In this way, all raster cells inside water regions were marked as invalid and could be replaced by the bathymetry.

The following step consisted in the conversion of the bathymetry TIN to a raster. We were free to choose a tiling suitable for loading the data set into the mosaic. Similarly to the topography DEM, the separation boundary was intersected with the bathymetry footprints, keeping the overlapping areas, and setting them as seamline polygons for the bathymetry tiles. The result was an integrated DEM that can be accessed, e. g. via an ArcGIS data service. Our terrain data repository, altogether, contains more than 5 billion raster cells.

**Dynamic Re-Tiling by a Web Coverage Service**

The OGC is an international voluntary consensus standards organization developing open standards for the geospatial community. WCS is an abstract interface description for serving coverage data, such as raster data, over the internet using common Web protocols. This standard has found acceptance in national implementations of the European INSPIRE Directive, e. g. in Germany, setting the framework for a European SDI. WCS is thus the technology of choice for providing and accessing raster data in an interoperable way.

A simple coverage request consists of an HTTP URL with encoded parameters describing the requested data set and its output format. A bounding box parameter delineates the raster extent in a given coordinate system. Additionally, either the width and height of the raster (in cells) or the resolution in X and Y is specified. ArcGIS mosaic data sets can be served as a WCS in ArcGIS Server without an additional effort. Free alternatives of geodata servers with WCS functionality are GeoServer and MapServer.

We set up a WCS for the contiguous, integrated DEM mosaic. Using WCS it was possible to query arbitrary raster tiles dynamically, on-the-fly, independent from the original terrain data set boundaries, and in any desired resolution. This feature was important as we wanted to perform computationally intensive geoprocessing operations on the data, such as the detection of structural features.

## AUTOMATIC BREAKLINE DETECTION AND MESHING

Rath [2] has developed an automatic discretization methodology for unstructured 2D hydrodynamic models that bases mesh structure on breaklines detected in a DEM. He created a software library for this purpose, Gaja3D, which we modified to work with a large number of DEM tiles. We extended Gaja3D by a new feature detection method based on the Canny edge detector and applied this new method to derive breaklines from the Elbe estuary DEM.

### Service Chaining and Data Flow

The Gaja3D library functions were captured as OGC-conforming geoprocessing services by application of the Web Processing Service (WPS) standard. WPS specifies three mandatory operations: *GetCapabilities* and *DescribeProcess*, for metadata about the offered processes, their inputs, and their outputs, and *Execute*, for starting a process. Complex WPS inputs can either be included in the message payload data or specify the reference to a web-accessible resource. The feature detection process can also easily be parallelized given a number of Gaja3D WPS instances and a load-balancing WCS. The feasibility of this concept has been demonstrated in [4] using Grid computing technology.

Our focus was on the service chain and data flow when a WCS and a WPS are used in conjunction. Fig. 4 shows the sequence of messages in the process. It was initiated by an *Execute* request using the identifier *DetectBreaklines*. The following input items (with the given type) were part of the request:

- A digital elevation model (e. g. GeoTIFF raster) and its boundary (optional), and
- feature detection parameters (type of gradient filter, smoothing filter, feature detection high and low thresholds, breakline simplification tolerance)

We obtained the raster in GeoTIFF format from the WCS, which allowed us to use the WCS reference URL (including extent and resolution) in the request. In this way, no DEM data had to be transferred as part any message sent from the client to the WPS. Instead, the WPS requested coverage data from the WCS.

### Parameterization of the Feature Detection Process

Due to the large size of the DEM and difficulties to process it all at once, it was divided into a number of tiles. The tile rasters had to overlap by a few cells for correct feature detection across tile boundaries. A suitable tile size has been experimentally found to be between 500 by 500 and 1000 by 1000 cells, i. e. a height and width of about 500 m to 1000 m at an analysis resolution of 1 m. Given our small study area boundary, Gaja3D created a simple partitioning by overlaying a rectangular grid of tiles, which was automatically clipped at the boundary polygon. We chose a grid of 5 by 4 tiles having a width and height of about 650 m and an overlap of 50 m. The resulting boundaries were used to query the WCS for tile rasters in 1 m resolution. For each tile the same set of feature detection parameters was applied.

As the first step, a Gaussian smooth filter removed noise in the input data. This eliminated false breaklines, but it also introduced a small displacement in the breakline location. Next was a surface gradient filter representing the slope of the terrain as the 1st derivative approximation of the elevation magnitude in x and y direction. Rath [2] found the one-over-distance gradient filter as most suited for breakline detection, so it was used here as well. The geometric mean of the gradients in x and y was then taken as a measure of the local change in surface elevation (intensity gradient). The direction, or aspect, was not considered.

The Canny method [5], with various improvements by other authors, represents the state-of-the-art in edge detection. It can be regarded as the detection of local maxima in the intensity gradient. Typically, it is applied together with a noise-reduction filter, in the same way as stated above. The problem with this approach is that only a single breakline is detected at the central line of a sloped area. Such a line is not a good structural line for surface reconstruction in the mesh creation process because the upper and lower banklines of the slope will be discarded.

Instead of detecting edges in the image itself, we applied the Canny detector to the intensity gradient of the surface elevation. Gaja3D was extended to use the VIGRA computer vision library [6] for image processing. A second smoothing step was used to reduce noise in the gradient image, whereupon we computed the 2nd derivative, or intensity curvature, in the same manner as deriving the gradient. It was then possible to obtain the "edgels" of maximum local curvature by non-maximum suppression and detection of zero-crossings of the 3rd derivative. Edgel detection with sub-pixel precision was conducted using VIGRA by fitting a parabola to the intensity curvature. Hysteresis thresholding was applied to filter some noisy edges.

The resulting edgels were linked and projected from image coordinates to raster coordinates to construct breaklines in vector form. Our approach locates features where there is a sudden change in the gradient, e. g. from flat to sloped. Fig. 5 shows the grid of

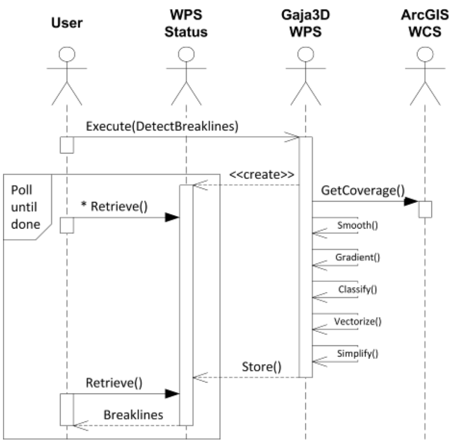


Figure 4: UML sequence diagram of the breakline detection service chain.

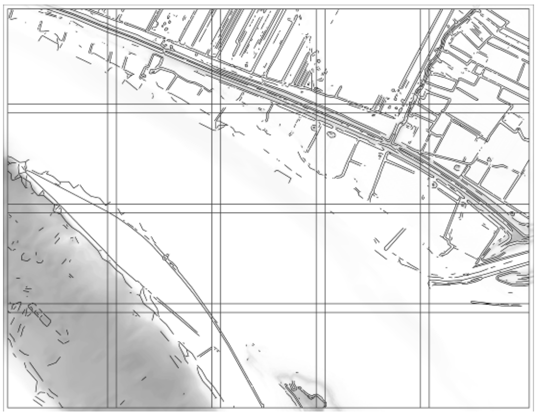

Figure 5: Gridded boundary and breaklines for Canny edge detection on the gradient image (σ = 1.5 and threshold 0.5).

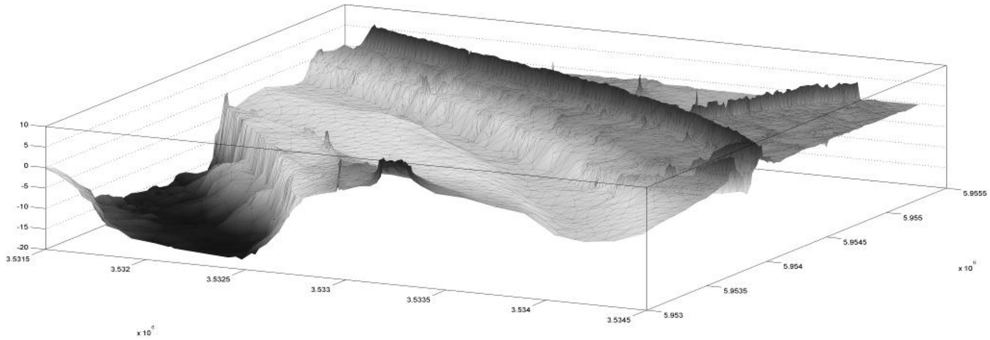

Figure 6: Perspective view of the mesh created by Delaunay triangulation of the breaklines (maximum triangle area 5000 m², minimum angle 22.5°, ca. 20.000 triangles). Node elevations have been assigned from raster tiles.

raster tiles together with the detected breaklines. From this display it becomes obvious that breaklines are found at the bottom and top of slopes, such as the dike at the upper river bank or at the inlet at the right side of the figure. If desired, the small ditches in the hinterland could be detected using a higher raster resolution and an appropriate smoothing filter. However, for current hydrodynamic simulations, such a precision would lead to many small triangles that increase the simulation time unnecessarily in most use cases.

**Mesh Creation and Quality Assessment**

We created a TIN of the breaklines inside the study area boundary by quality constrained Delaunay triangulation. For this purpose, Gaja3D employs the widely used Triangle meshing library [3]. We defined meshing parameters that limit the maximum allowed triangle area and the minimum internal angle. A 3D view of the created mesh can be seen in Fig. 6. The general impression is that the main dike crest and terrain breaks are well represented. Larger triangles are located in the river channel, whereas most of the triangles can be found near breaklines. Fig. 7 shows a cross-sectional profile of the river and the main dike. The feature detector sometimes failed to locate the top of a ridge or it missed either the upper or lower breakline of a slope. The main dike should have had an elevation of about 8 m, but the meshing process took away 40 to 60 cm on average. Slowly changing slopes in the river bed were mostly correct, but due to the large triangles, holes or

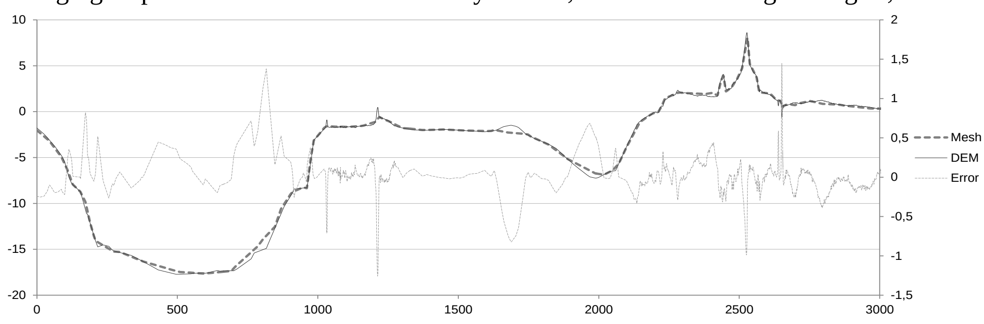


Figure 7: Comparison of elevation profiles along exemplary cross section.

bumps were cut off in some places.

Compared to a DEM with 20 m horizontal resolution (similar data volume), the accuracy of our mesh was much higher. However, the overall volume of the original DEM (below a reference plane at 8 m) amounted to 77.8 million m³, while our mesh only had a volume of 74.6 million m³ (minus 4.1 %). Apparently the method is biased towards losing volume.

## CONCLUSION AND OUTLOOK

We have presented and approach to DEM data management and integration compatible with existing OGC data services for use in a spatial data infrastructure. A Web Coverage Service was employed to access topographic and bathymetric data of the Elbe estuary. We showed how breaklines can be detected in this DEM using a Web Processing Service and developed an improved feature detection scheme based on the Canny edge detector. Especially near feature edges, our mesh provides a good representation of the terrain.

Some of the errors described, such as ridges being too low, could be reduced if the horizontal resolution is increased, e. g. to 0.5 or 0.25 m. Although this would increase the processing time dramatically, such an optimization could easily be tested due to the dynamic tiling and resampling capabilities of WCS. This supports the fact that WCS is a useful abstraction for handling large DEM data sets in computationally intensive geoprocessing scenarios.

## REFERENCES


[1] Schrage N., Antanaskovic D., Jung T. and Pasche E., "KALYPSO – An Open Source Software Tool for Flood Studies in Rivers", *Proceedings of the 8th International Conference on Hydroinformatics (HEIC)*, Concepción, Chile, (2009).

[2] Rath, S., "*Model Discretisation in 2D Hydroinformatics based on High Resolution Remote Sensing Data and the Feasibility of Automated Model Parameterisation"*, Dissertation, Hamburger Wasserbau-Schriften, No. 8, Department of River and Coastal Engineering, TUHH, Hamburg, (2007).

[3] Shewchuk J. R., "Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator", *Applied Computational Geometry: Towards Geometric Engineering*, Lecture Notes in Computer Science, Vol. 1148, Springer-Verlag, (1996), pp 203–222.

[4] Kurzbach S. and Pasche E., "A 3d Terrain Discretization Grid Service for Hydrodynamic Modeling", *Proceedings of the 8th International Conference on Hydroinformatics (HEIC)*, Concepción, Chile, (2009).

[5] Canny J., "A Computational Approach To Edge Detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 8, No. 6, (1986), pp 679–698.

[6] Köthe U., "Reusable Software in Computer Vision", in: Jähne B., Haußecker H. and Geißler P., "*Handbook on Computer Vision and Applications",* Vol. 3, Academic Press, (1999).