

# INTEGRATION OF HIGH RESOLUTION DIGITAL ELEVATION MODELS IN 3D-GIS-APPLICATIONS OF THE ENVIRONMENTAL INFORMATION SYSTEM OF BADEN-WÜRTTEMBERG

D. Hilbring

Institute for Photogrammetry and Remote Sensing, University of Karlsruhe, Englerstr. 7, 76128 Karlsruhe, Germany, hilbring@ipf.uni-karlsruhe.de

Commission IV, WG IV/6

**KEY WORDS:** High resolution, DEM/DTM, Threedimensional, GIS, Application, Visualisation

## ABSTRACT:

This paper describes the integration of high resolution digital elevation models, currently acquired by the administration of Baden-Württemberg, a state in Germany, in two example 3D GIS applications of the Environmental Information System (UIS) of Baden-Württemberg.

The combination of the high resolution DEM with the area-wide data acquisition in Baden-Württemberg leads to a great amount of data, which needs to be handled. Thus the paper starts with a discussion about the data handling. The paper describes techniques for the preprocessing of the data as a prerequisite for the management of the data with the raster tiling approach in the database. The next part of the paper concentrates on techniques for the visualization of the data in a GIS system. These techniques are used in both example applications, discussed in the following part of the paper. Both examples show the successful integration of high resolution digital elevation models into 3D GIS applications.

## 1. INTRODUCTION

The administration of Baden-Württemberg is currently acquiring an area-wide high resolution Laser scanner digital elevation model. The resulting one and five meter DEM will be included into 3D GIS applications of the Environmental Information System (UIS) of Baden-Württemberg. The combination of the high resolution DEM with the area-wide data acquisition in Baden-Württemberg leads to a great amount of data. The goal is to include these masses of data in a usable manner into two different example applications of the Environmental Information System. The example applications are developed for the Ministry of Transport and Environment in the project AJA. The "Landesanstalt für Umweltschutz" is attending the technical aspects of the development (AJA 2003; LfU 2004).

The first example application, the DEMViewer, visualises textured digital terrain models in different sized regions. Depending on the visualising task the region can be small or as large as possible.

The second example application, GeoPro<sup>3D</sup> is a specific 3D application for analysing the ground water table in regional areas for planned construction sites. In this case the visualised region is usually fairly small.

Both examples are realised with the means of a 3D GIS service included in the base GIS GISterm Framework (Hofmann/Hilbring/Veszolka/Wiesel 2000). GISterm Framework is a Java based class library for the visualisation of space oriented data developed by the Institute for Photogrammetry and Remote Sensing and disy GmbH (IPF 2004; disy 2004). The 3D extension for GISterm, called 3D-Service, is based on Java and Java 3D and is developed by the IPF (Java 3D 2004).

The following sections will present methods and techniques for processing, managing and visualising high resolution DEM data.

## 2. DATA HANDLING

Both applications need terrain data in different sized regions. Thus the system needs techniques for providing a manageable and visualisable amount of data for every kind of bounding region. This means we need different data sets as basis.

### 2.1 Data Basis

Before the acquisition of the high resolution data started, Baden-Württemberg possessed digital elevation models derived from aerial views at a resolution of 50 m. This dataset is useful for applications using large regions. The new Laser scanner models provide high resolution data, which are necessary for applications using small regions.

Let us have a look at the data amount we are talking about in the high resolution Baden-Württemberg example. The raw DEM data is delivered in ASCII-files containing 1 km<sup>2</sup> in 1 m or 5 m DEM resolution. The area of Baden-Württemberg is 35 752 km<sup>2</sup>. That means we have to handle twice as many original ASCII-files. Surely the goal is to handle the data in a database. In our example the chosen database is the database of the Environmental Information System.

### 2.2 Concept

One commonality of all DEMs is the rasterised arrangement of the points. In many 2D-GI-Systems, techniques for visualising large amounts of raster data, like topographic maps, are already established. A common solution is dividing the raster data into tiles and store them as images in a specific format in the database (Hofmann, Veszolka, Wiesel 1999). The idea is to reuse this technique for managing digital elevation models. This section firstly describes techniques for the pre-processing of the data. Secondly it describes the raster tiling techniques. Thirdly it describes the access to the data in the data base.

### 2.3 Data Pre-processing

The raw data of all three digital elevation models is organised in a grid structure, where every point has an east, north and height value. The distance in east and north direction is the same. For reusing the GI-System techniques it is necessary to create images including the information of a digital elevation model. We need to create a georeferenced image with each image pixel holding the height information of a grid point of the DEM. This height information is represented by the colour value of the image pixel. Thus it is necessary to find an image format, which colour depth value is large enough for handling the height resolution of the digital elevation model. It is necessary to define an interval including the height range of the digital elevation model to calculate the resolution.

Calculation of height resolution:

$$\text{height resolution} = \frac{(\max_{\text{height value}} - \min_{\text{height value}})}{(\text{depth value} - 1)}$$

One colour value should be reserved for grid points without defined data in the raw data set. All other colour values can be used for encoding the height information. The following shows the calculation of the colour value for one image pixel.

Calculation of colour value:

$$\text{factor} = \frac{(\text{depth value} - 1)}{(\max_{\text{height value}} - \min_{\text{height value}})}$$

$$\text{colour value} = \text{height value} * \text{factor}$$

Using this technique it is possible to create images including the height information. Additionally one file for each image is created including the bounding region of the image for the georeferencing.

Using this technique for the integration of the digital elevation models in the UIS of Baden-Württemberg we can consider the following conditions:

- The accuracy of the given height information in all raw datasets (50m, 5m and 1m) is +/- 0.1 m.
- All possible heights of Baden-Württemberg fit in an interval from -500 m to 3500 m.
- That leads to a grey value depth of the image of 16 bit (65 535).
- PNG-images with 16 bit depth are used to store the digital elevation model.

### 2.4 Raster Tiling

It is possible to treat the created images like other raster data: Use the given tiling, storing and accessing techniques for the management of the DEM data.

In our example we use the already existing tiling algorithm of the base GIS GISterm Framework. GISterm includes classes and methods for tiling images, storing them into the database and accessing them. These methods include possibilities for managing data in raster pyramids in different resolutions (Hofmann/Veszeka/Wiesel 1999). This techniques is called *raster tiling*. It is possible to totally reuse these parts of GISterm.

We can use the existing techniques for storing all three DEM data sets into the database.

### 2.5 Memory Consumption of the Example

Reviewing the data handling section up to this part of the paper we firstly looked at the data basis (Raw ASCII data of digital elevation models in 50 m , 5 m and 1 m resolution). Secondly we created PNG-image files from the raw data. Thirdly we stored these images in the database using the raster tiling approach. The following table summarises the amount of memory consumption in all three states of the high resolution digital elevation model data needed for the entire area of Baden-Württemberg:

resolution / file type	1 m	5 m
ASCII	~ 933 GB	~ 38 GB
PNG	~ 11 GB	~ 1 GB
PNG in database	~ 21 GB	~ 2 GB

Tab. 1: Memory Consumption for high resolution DEM data of Baden-Württemberg

Thus, we can easily store the complete high resolution DEM of Baden-Württemberg into the database. The next part concentrates on accessing of the data in the database.

### 2.6 Access to the Data in the Database

DEM data are used in different applications. Thus we need a generic interface for getting the data out of the database. This interface is called Height-Service and offers classes and methods for providing the data for further processing (Hilbring 2002). The Height-Service is realised as newly created extension of the base GIS GISterm for accessing DEM data in the database.

The basic information needed by the Height-Service for accessing the data is the bounding region of the requested DEM part. Based on the size of this bounding region the Height-Service looks for an appropriate DEM in the database.

#### 2.6.1 What is an appropriate DEM?

On the one hand the decision includes the aspects of the final displaying resolution. For example it is not sensible to load the entire DEM of Baden-Württemberg in 1 m-resolution into a 3D scene, because the screen resolution is not high enough for displaying. On the other hand, loading such a large DEM needs huge amounts of memory and creates performance problems while handling the 3D object in a virtual scene. Practical tests showed that 500 000 base points can be handled by the application with reasonable response time.

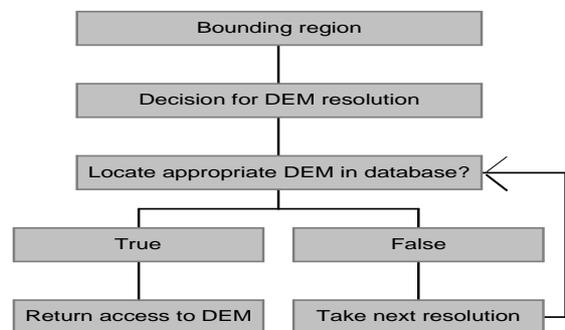


Fig. 1: Choice of appropriate DEM

Taking these aspects into account, the Height-Service decides from the size of the bounding box and the number of expected points on an appropriate resolution of the DEM and

looks for available data in the database, beginning with the highest possible resolution. One aspect we have to consider is, that the database has no or only fragmented data for the requested region. Possible reasons are:

- The acquisition of the laser scanner data is still going on. That means the requested data is not yet fully available.
- The processing of the laser scanner data is not yet fully completed.

In this case the Height-Service will search for a lower resolution digital elevation model. Now we are ready for the access of the DEM data.

### 2.6.2 Accessing the DEM data

The Height-Service includes methods for accessing the data from the selected DEM. The class diagram shows the most important methods.

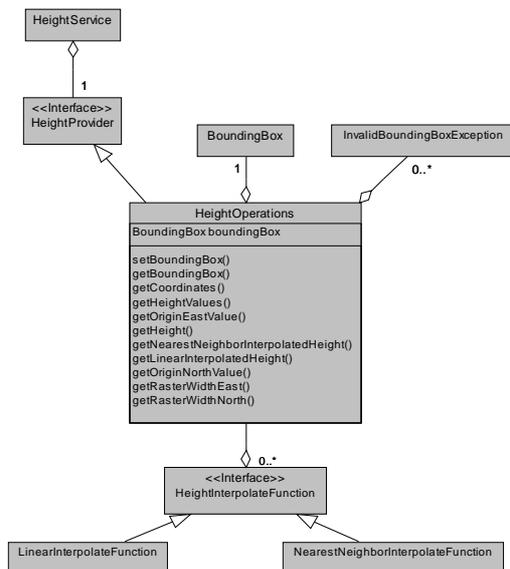


Fig. 2: Height-Service class diagram

The methods:

- `getCoordinates(double[] eastValues, double[] northValues)`
- `getHeightValues()`

are accessing the grid structured base points of the DEM, while the methods:

- `getNearestNeighbourInterpolatedHeight(double east, double north)`
- `getLinearInterpolatedHeight(double east, double north)`

are able to interpolate height values between the base points.

## 2.7 Summary

Reviewing the data handling section of this paper we began with a look at the data basis. Then we created PNG-image files from the raw data and stored them in the database using the raster tiling approach. Finally we considered the access to the data in the database via the Height-Service interface. That means we are now ready for the next part of the paper: the visualisation.

## 3. VISUALISATION OF THE DATA

This part of the paper concentrates on the visualisation of the DEM data. The presented concepts of this part are not limited to the visualisation of our Baden-Württemberg example. The techniques can be used for all kinds of rasterised terrain visualisations.

Most 3D applications concerning environmental aspects need digital elevation models as a data basis. The visualisation of these models leads to a three-dimensional terrain-object presentation in a virtual reality.

### 3.1 Creation of the Virtual Universe

For the integration into the base GIS GISterm, the virtual universe is created with the help of the 3D-Service, which uses the means of Java 3D. This virtual universe includes common functions and properties needed for every 3D-GIS application:

- Management of semantically grouped object content in different 3D layers.
- Means for orientation and navigation.
- Possibility for changing common scene properties, like the disbanding of the height axis, the change of the lighting or the background colour.

For the visualisation of the digital elevation models we need a 3D Layer, holding the terrain object generated from the DEM data out of the database. This is described in the next section.

### 3.2 Creating Three-Dimensional Object Content

For visualising the DEM data in the virtual universe, a 3D Layer called `DEMLayer3D` is created. This Layer holds the actual 3D object content. The `DEMLayer3D` is added into the virtual scene created by the 3D-Service, which visualises the terrain in the 3D-viewer component.

#### 3.2.1 Structure of Terrain Object

This part of the paper describes the structure of this terrain object. Due to texture size issues explained deeply in the next section the terrain objects need to be divided into several parts. The number of terrain parts depends on the size of the texture parts. By dividing the terrain size by the texture size the number of needed terrain parts is calculated. The size of the terrain parts corresponds with the size of the texture parts. The arrangement of the terrain parts starts in the lower left corner increasing into east and north direction. The terrain parts at the upper north and east border are smaller than the texture size. The following image shows the arrangement of the terrain parts.

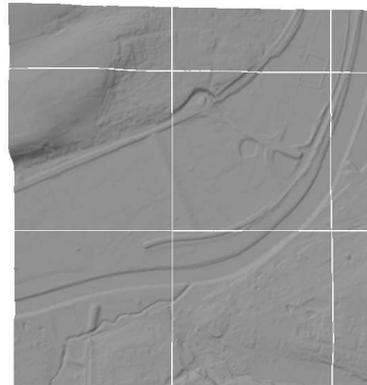


Fig. 2: Arrangement of terrain parts

Each terrain part is created as DEMFeatureProxy3D object. Afterwards all DEMFeatureProxy3D objects are assembled in the DEMLayer3D to build the entire terrain. Fig. 3 shows the final terrain object.

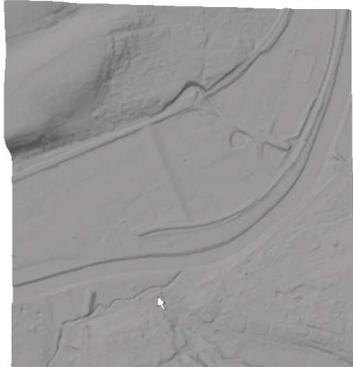


Fig. 3: Created terrain object

### 3.2.2 Creation of Object Geometry with VisAD

The actual 3D object content of one DEMFeatureProxy3D object is created with the means of VisAD, a Java component library for the visualization and analysis of numerical data (Hibbard, 2004).

The following explains the reason for using VisAD in this combination. The VisAD library is pure Java, including Java 3D for the visualisation of 3D object content. Thus VisAD fits well into the GIS term Framework and the 3D-Service. This includes all basic concepts for the handling of 3D object content in the 3D-Service.

The data for the creation of the terrain content is provided by the Height-Service. Afterwards the object geometry is created with the means of VisAD. The resulting objects contain pure Java 3D object content. This Java 3D object content is fetched from VisAD and included in the DEMFeatureProxy3D as geometry.

## 3.3 Texturing Object Content

By now we created the geometry of the terrain and included it in the virtual universe. *Texturing* techniques provide the possibility of adding a realistic impression or adding more information to the scene.

### 3.3.1 What is Texturing?

“Texturing, also called *texture mapping*, is a way to add visual richness of a surface without adding the fine geometric details. The visual richness is provided by an image, also called *texture*, which gives the appearance of surface detail for the visual object.” (Java 3D Tutorial 2004). The image is mapped on to the geometry of the object content.

### 3.3.2 Goal

We can use texture mapping in our example for two different reasons:

- One goal is to add a realistic impression of the terrain in the scene. This can be achieved by using satellite images or aerial views as texture.
- Another goal is to add additional information into the scene. This can be achieved by adding specific images as textures. Topographic maps for example add much naming information, while geological maps can give hints about the underground situation in the presented region.

### 3.3.3 Texturing with Java 3D

Java 3D provides classes and methods for adding texture images on top of the geometry objects, created with Java 3D. Thus, the 3D-Service uses Java 3D for texture mapping. Images used for texturing must fulfil two conditions:

- The image must be saved in a file format, supported by Java3D. Examples for readable image formats are GIF or JPG.
- “Java 3D requires the size of the texture to be a mathematical power of two (1,2,4,8,16,...) in each dimension” (Java 3D Tutorial 2004).

Image fulfilling these conditions can be used as texture for Java 3D geometry.

### 3.3.4 Creating the Texture Image

The question is now, where does the texture image come from for our terrain example? The user should have the possibility to design the texture image. The user can individually create his own 2D map, including several 2D layers with appropriate content, in the base GIS GIS term. This 2D map is used as texture image for our terrain example. Using the printing technique of GIS term we can fetch a GIF or JPG file of the current display region. These image files can be saved in different resolutions (Normal, Middle or High). The better the resolution of the image is the higher is the number of pixel in the image and the bigger is the memory consumption. This is important because, depending on the image memory of the graphic card, the largest possible size of the texture image is limited. The main memory is not used. For example using a decent graphics chip (Nvidia GeForce4 Ti 4200) limits the size of the texture to 2048 \* 2048 pixel (Virtual Terrain Project 2004). Thus the decision for the resolution of the texture image is a compromise between good resolution and memory consumption. However, images fetched from GIS term do not match the condition of power of two dimension sizes.

Thus we need to do image manipulation before we can use the fetched image as texture. The ratio of the image dimension of the fetched image depends on the ratio of the displaying region in GIS term. And usually both image dimensions have an unacceptable size. The dimension of the fetched image in our example is 1224 \* 1036 pixel.



Fig. 4: Textured terrain

The idea to solve this problem is to divide the entire terrain into parts fulfilling the requested condition of the power of two dimension for a texture image. That means we decide on a texture part dimension, matching the power of two dimension. In our example we took a texture size of 512 \* 512 pixel. Then we split the terrain into parts (see section:

3.2.1 Structure of Terrain Object) and split the image into the corresponding texture parts. These texture parts are mapped as texture images on top of the geometry of the appropriate terrain part. Due to the fact that the terrain parts at the north and east boundary are smaller than the other parts, the texture, which still has the 512 \* 512 pixel size, overlaps the geometry. This is not a problem, because the overlapping texture parts are not displayed in the virtual universe. Fig. 4 shows the textured terrain of our example.

### 3.3.5 Memory Issues

However, it does not matter for the memory consumption if the texture is actually visible or not. That means overlapping parts are not useful, but they consume needed texture memory. That is the reason, why we divided the terrain into parts.

Looking at the original size of the fetched image (1224 \* 1036) in our example we would have needed a Java 3D corresponding texture size of 2048 \* 2048 = 4194304 pixel. Dividing the terrain into parts we need 9 texture images with a dimension of 512 \* 512 pixel. That adds to a number of 2359296 pixel. Thus we could save a significant amount of texture memory.

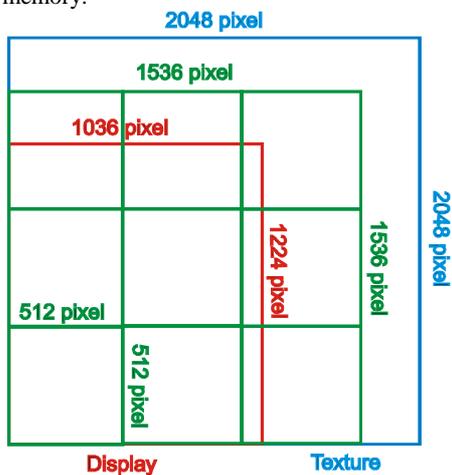


Fig. 5: Memory consumption of textures

### 3.3.6 Outlook

Techniques for reloading texture parts with a better resolution when the user zooms into the 3D scene for analysing details will be a big improvement for the visualisation in the future.

### 3.4 Summary

The visualisation part of the paper described the creation of the terrain object and the corresponding texture. The presented techniques can be used for terrain visualisations in 3D GIS example applications.

## 4. INTEGRATION OF DEM IN EXAMPLE APPLICATIONS

In the last part of the paper we return to our Baden-Württemberg example. We describe two use cases using the presented terrain visualisation techniques to include the DEM data of Baden-Württemberg in 3D GIS example applications.

### 4.1 DEMViewer

The goal of the DEMViewer is the visualisation of textured digital terrain models. The user has the possibility to individually design the appearance of the texture. Depending on the user request the visualised region can be very small or large. Depending on the size of the region the appropriate DEM resolution (50m, 5m or 1m) in the database will be chosen.

The user loads his region of interest into the 2D map of GISterm. The individually created 2D map is used as texture for the terrain object created by the DEMViewer. Calling the function "Create digital elevation model" in the 3D menu of GISterm starts the DEMViewer. The DEMViewer fetches the currently displayed bounding region of the 2D map and the texture image of the map. Afterwards it selects an appropriate DEM with the means of the Height-Service. The DEMViewer creates a virtual universe including a textured DEMLayer3D out of the selected data using the terrain visualisation techniques. Fig. 6 shows a virtual universe in GISterm created with the DEMViewer.

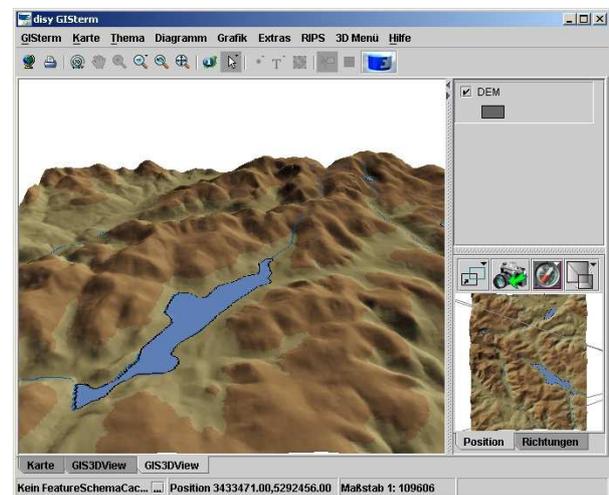


Fig. 6 50m DEM in DEMViewer

Now the user has the possibility to navigate inside the 3D scene and examine the loaded terrain object. The main view of the 3D-Service displays the object content seen from the current viewer position, which can be changed by navigation. The navigation handling of the 3D-Service is realized with the means of the mouse. The left mouse button leads to rotation of the viewer around the objects in the 3D scene (rotation), while the middle mouse button moves the viewer closer to or farther from the objects (zooming). Finally the right mouse button results in panning. The resulting viewer positions from the navigation handling are displayed in the overview window, which displays the entire scene together with the viewer position represented by a camera.

Due to the amount of computer memory the size of the loaded terrain part is limited. Usually the user is interested in neighbouring parts of the displayed terrain. Thus the user needs to reload new terrain parts, which were not displayed before. For this purpose the overview window has a button for interactive reloading of terrain parts. Clicking that button opens a submenu with arrow-buttons for all cardinal points. Choosing one of the arrow-buttons deletes the current terrain part in the scene and loads the neighbouring part in the arrow direction. The newly displayed part is partly overlapping the former terrain part.

## 4.2 GeoPro<sup>3D</sup>

In our next example application, GeoPro<sup>3D</sup>, the visualisation of the terrain is not the primary goal of the application, but it is still a very important part which adds understanding to the created scene.

The goal of GeoPro<sup>3D</sup> is the visualisation of the ground water table and hydro geological layers in combination with planned construction sites. The visualised region is usually fairly small.

When starting a GeoPro<sup>3D</sup> project the user loads the region he is interested in into the 2D map of GISterm. There he must select ground water measuring points, which are available through the ground water data base. Additionally he needs to digitise the planned buildings in the 2D map. With this information GeoPro<sup>3D</sup> creates a virtual scene with the following content:

- The ground water table is visualised in three ground water layers with information from the ground water database.
- Hydro geological layers are additionally created from the information of the ground water database.
- The data for the terrain layer is selected with the Height-Service out of the database. Usually the 5 m high resolution DEM of Baden-Württemberg has the appropriate resolution.
- A building layer is created with the user digitised information from the 2D view. The building is placed on top of the terrain.

Fig. 7 shows a virtual scene created with GeoPro<sup>3D</sup>.

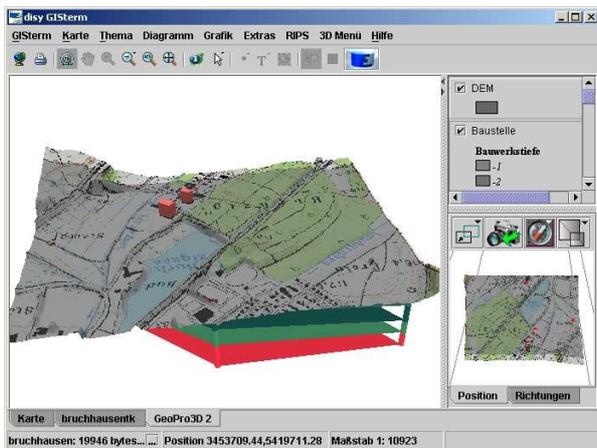


Fig. 7 Scene created with GeoPro<sup>3D</sup>

GeoPro<sup>3D</sup> provides several analysis possibilities of the scene, like clipping the construction site for detecting conflicts with the water table or creating 2D view profiles for visualising the run of the hydro geological layers. The visualisation of the textured terrain adds a realistic impression into the scene, which helps the user to understand the visualised analysis tasks.

## 5. CONCLUSIONS

This paper discussed data handling and visualisation techniques for the integration of high resolution digital elevation models into 3D GIS systems. These techniques were adopted successfully for two different visualisation applications of the Environmental Information System of Baden-Württemberg. The used techniques are generally transferable to other examples.

## 6. REFERENCES

AJA, 2004, Projekt AJA (Anwendung JAVA-basierte Lösungen in den Bereichen Umwelt Verkehr und Verwaltung), <http://www.lfu.baden-wuerttemberg.de/lfu/uis/forschung/>, (accessed April 2004)

disy, 2004, disy, [http://www.disy.net/de/start\\_de/index.html](http://www.disy.net/de/start_de/index.html), (accessed April 2004)

Hibbard, 2004, VisAD, <http://www.ssec.wisc.edu/~billh/visad.html>, (accessed April 2004)

Hilbring 2002, Projekt AJA, Anwendung Java-basierter Lösungen in den Bereichen Umwelt, Verkehr und Verwaltung, Phase III 2002, GISterm3D – Weiterentwicklung von GeoPro<sup>3D</sup> und Neuentwicklung des Height-Service für die Integration digitaler Geländemodelle, S. 37-56 in Mayer-Föll, Keitel, Geiger (Hrsg.), Wissenschaftliche Berichte, FZKA 6777, Forschungszentrum Karlsruhe, Technik und Umwelt

Hofmann, Veszelka, Wiesel, 1999, Projekt GLOBUS, Von Komponenten zu vernetzten Systemen für die Nutzung globaler Umweltsachdaten im Umweltinformationssystem, Baden-Württemberg und anderen fachübergreifenden Anwendungen, Phase VI 1999, GIStermFramework – das flexible komponentenbasierte Geodatenzugriffssystem, S. 57-79 in Mayer-Föll, Jaeschke (Hrsg.), Wissenschaftliche Berichte, FZKA 6410, Forschungszentrum Karlsruhe, Technik und Umwelt

Hofmann, Hilbring, Veszelka, Wiesel, 2000, Projekt AJA, Anwendung Java-basierter Lösungen in den Bereichen Umwelt, Verkehr und Verwaltung, Phase I 2000, GISterm – Weiterentwicklung des flexiblen Frameworks zur Analyse und Visualisierung von raumbezogenen Daten, S. 147-169 in Mayer-Föll, Keitel, Geiger (Hrsg.), Wissenschaftliche Berichte, FZKA 6565, Forschungszentrum Karlsruhe, Technik und Umwelt

IPF, 2004, Institut für Photogrammetrie und Fernerkundung, <http://www.ipf.uni-karlsruhe.de/>, (accessed April 2004)

Java 3D, 2004, Java 3D API, <http://java.sun.com/products/java-media/3D/>, (accessed April 2004)

Java 3D Tutorial, 2004, Dokumentation Collateral, Chapter 7, <http://java.sun.com/products/java-media/3D/collateral/> (accessed April 2004)

LfU, 2004, Landesanstalt für Umweltschutz, <http://www.lfu.baden-wuerttemberg.de/>, (accessed April 2004)

Virtual Terrain Project, 2004, Virtual Terrain Project, Hardware Support for Large Texture Maps, <http://www.vterrain.org/LargeTextures/index.html>, (accessed April 2004)