

# TREESVIS - A SOFTWARE SYSTEM FOR SIMULTANEOUS 3D-REAL-TIME VISUALISATION OF DTM, DSM, LASER RAW DATA, MULTISPECTRAL DATA, SIMPLE TREE AND BUILDING MODELS

H. Weinacker<sup>#</sup>, B. Koch<sup>#</sup>, R. Weinacker\*

<sup>#</sup>Institute for remote sensing and landscape information systems, 79106 Freiburg, Germany - (holger.weinacker; barbara.koch)<sup>#</sup>@felis.uni-freiburg.de, \*Rudi Weinacker Software Development, Ludwigshafen, Germany

**KEY WORDS:** LIDAR, Visualisation, DSM/DTM, Multispectral, Extended Superquadric, CLOD, Quad tree, ROAM, Geo-Mipmapping, billboards, culling, indicated triangles

**ABSTRACT:** A software package has been developed, allowing to visualise simultaneously several data-sets in real-time without the need of special graphic hardware. This is achieved by combining DirectX routines together with methods like CLOD, culling, indicated triangles, ROAM, Quad tree, GeoMorphing and Geo-Mipmapping. This has been done within the framework of the NATSCAN project.

## 1. INTRODUCTION

During the last years graphic cards used in standard computers and laptops have increased their power extremely, while the costs have decreased dramatically. This development is mainly caused by the game industry and millions of game user. Furthermore, free graphic software like OpenGL and DirectX have been extended to a wide range. These positive circumstances made it possible, that also the scientific community could draw profit from this development. TreesVis could not have been developed, if this background would not exist.

### 1.1 Background

The development of TreesVis was initiated by the NATSCAN project. One task within this still ongoing project was to visualise results in such a way that the potential user, e.g. employees of forest administrations or of power supply firms like RWE or Ruhrgas are able to work with the delivered data. In personal discussions it became obvious that the people, who finally have to work with this subject, were mainly interested to get a tool with which it is possible to solve their tasks in such a way, that they can use diagrams, but also include a powerful and easy to use visualisation tool. It was recognised, that such a kind of tool also ease the program development, as results can be checked directly after they have been calculated. These were the reasons, that the program has finally reached such a complexity.

### 1.2 Data

All laser and the multi-spectral data used were captured by TopoSys GmbH with the Falcon II system for the flights done in 2002 and 2003. In 2004 flights have been done with the new TopoSys scanner Falcon IIc, which is equipped with a "swinging" lidar system. The data from the four test areas within the NATSCAN project are representing totally different topographic structures, covering almost flat and very steep areas. Furthermore the test areas are covered by dense forest, by a power supply line and lying in a township. Therefore TreesVis could be tested with very different landscape situations. A detailed description of the used data can be found in (Weinacker 2004).

### 1.3 3D-Real-time Visualisation

Several visualisation methods have been integrated in the developed software package to be able to offer the user a comfortable user interface and prevent him with a systems which can visualise all objects as fast as possible without the need of special expensive hardware equipment. Therefore at least the most important methods used are shortly explained and their use in TreesVis is given. Those methods, which have been changed or expanded for a great extend and own developments are described in more detail in chapter 2.

**LOD - (Level Of Detail):** The aim is to visualise parts of a surface in great detail, which are lying near to the camera or consisting of a lot of detail. On the other hand flat areas or areas far away from the camera are shown with a small number of polygons. Introducing this method the account of the drawing triangles could be reduced to approximately 5% of the original triangles, without introducing less quality in the presentation.

**CLOD - (Continuous Level Of Detail) –** The switch from one LOD to another is visible very often, especially during camera movements, and create a strong popping. This effect can be avoided by continuous form changes (morphing). Therefore the level is fitted continuously, that's the reason why this procedure is called CLOD.

**Culling** - generally speaking, it means the masking of render objects. All these elements will not be transferred to the graphic processing unit (GPU).

**Backface-culling** - Masking triangles, which turn their back to the camera. This method is used, e.g. in case that the camera is moving inside a tree. Then the inner walls of the tree are not visible.

**View-frustum-culling** – Is a special type of culling. All objects, which are not inside the frustum created by the camera angle are masked. The technique is used for the presentation of the raw data and the terrain.

**Indicated triangle** - An index buffer is numbering the triangles in a vertex buffer. The contents of an index buffer could look like as following: 1, 32, 48, 90, 199, 2, 5, 4,... – which means,

that the first triangle consist of vertices 1, 32, 48 of the vertex buffer. The second one of the vertices 90, 199, 2 and so on. Using this method all coordinates only have to be stored once and the vertices at the touching point can be used several times. This method is used to show trees, buildings and SQ, but not for ellipses, polygons, planes and the raw data

**Roam** - (**R**eal-time **O**ptimal **A**dapting **M**eshes): The ROAM algorithm was introduced by Duchaineau (1997). The basic idea of this method is the use of a “binary triangle tree”. The greatest advantage of this method is, that the surface calculated is an almost optimal one, which means that a minimum of triangles are used to represent the searched surface. A disadvantage is that a lot computation is necessary.

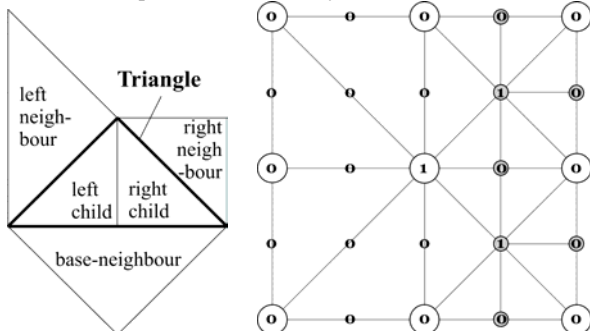


Fig 1: (a) Binary triangle structure; (b): Quad tree structure, one square with nine vertices. A father-square with two right hand children.

**Quad-tree based rendering** - A quad-tree is a well known data structure. Röttger (1998) based his terrain render algorithm on this structure (figure 1b). A triangulation based on this type of structure is faster as the binary tree oriented method and furthermore it is realisable using a matrix structure with the same size as the real surface. Therefore the memory requirements are constant and no triangle-buffer is needed. On the other hand the final network is not as good fitted to reality, as it is not possible to divide a single triangle, instead a complete rectangle has to be included. The caused pop-effect can be reduced by introducing morphing. The additional vertices are lying in the interpolated height between the original vertices and are moving continuously to their final position. Using this methodology changes are almost not visible.

**Geo-Mipmapping** – The two last methods presented above have the disadvantage, that a lot of calculation is necessary to decide whether a triangle or a square should be drawn or has to be further divided. This means, that the GPU has to wait for the CPU, which is not optimal as the GPU’s are very efficient machines. A better solution would be, to transfer several triangles to the GPU in one step. The Geo-Mipmapping algorithm works in this way. An area is divided into small patches with a size of 32x32 pixels. For each patch a vertex buffer is created and all vertex data are stored. Finally the whole area is stored in small patches in the storage of the GPU. A group of index buffers is created. An index buffer is a list of indices of vertices. In this way, it is possible to decide, which three vertices are used to draw a triangle. This is done for each LOD.

For a realistic presentation much more triangles have to be drawn compared to other methods, nevertheless this method is extremely fast, because the CPU will not thwart the GPU. A disadvantage of this method is the fix size of the patches used. Large areas would need a huge number of patches, which

overburden the GPU and therefore becomes inefficient at high resolution.

2. TREESVIS

The developed system consists of several components allowing the user not only to draw and visualise different sets of data, but also makes it possible to directly measure interesting points in 3D and showing the coordinates of these points. Furthermore, intersections of the terrain are also possible by using two of such points. All objects can be visualised in real-time. The programming has been done with Visual .Net 2003, VC++ and is based on DirectX 9.0c SDK. Some integrated mathematical fitting routines have been based on the public domain libraries TNT from NIST and NewMat. A screen shot of the GUI of TreesVis is shown in figure 2.

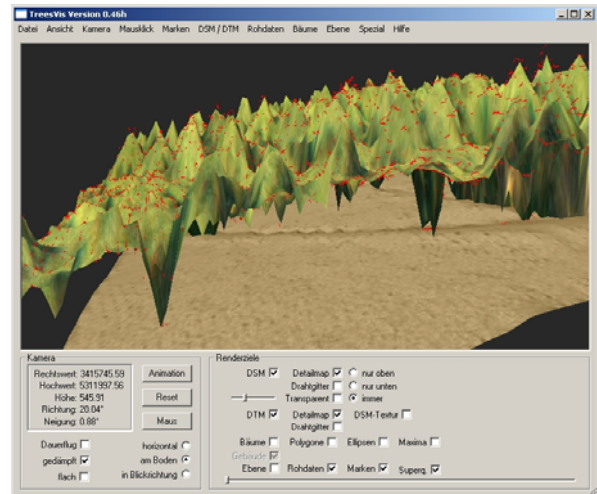


Fig. 2: Screen shot showing the main window of TreesVis.

2.1 Visualisation of large surface areas

To avoid time-delays in case, that large surface areas have to be visualised, it was decided to base the algorithm on a quad tree structure. Instead of drawing a square consisting of 3\*3 vertices as it has been done by Röttger (1998), a whole cluster consisting of at least 9\*9 vertices or even more are drawn simultaneously, this is similar to the geo-mipmapping method, see figure 3.



Fig 3: Clusters with (left) and without borders (right).

The used number of vertices depends to a great extent on the structure of the surface to be visualised. For the visualisation of the DSM, which especially in forest areas has a very rough texture, 33\*33 vertices per cluster are used. By this the power of the GPU is used to a high extent. To be able to draw also large areas with about 16000\*16000 vertices or even more, two vertex buffers are introduced. One is filled, while the data within the second one are used to be drawn on the screen by the GPU. Normally, as the clusters always have the same amount of vertices, the CPU has done its work, before the GPU starts and therefore the triangles can be drawn with an optimal filling rate.

Areas where a change in detail occur, the “plopping” of the additional vertices is very obvious. Therefore the geo-morphing method from Röttger (1998) is used too, as it also works on whole clusters.

A great advantage of the method introduced is the low memory requirement. Additionally to the height-field only the quad tree-matrix has to be stored. Our quad tree matrix needs much less storage space in comparison to Röttgers matrix. For a 33\*33 cluster only 1/256 of memory space is required for the quad tree matrix, this means that only 1/256 values of the height-field are needed.

A further advantage is that no pre-calculations at all are necessary. Only the height field and the quad tree-matrix has to be loaded. One time-delay exists, as during the uploading process a light texture is calculated simultaneously, which is necessary as without a perspective impression would not be visible. Fig. 4 shows a clip from a DTM from almost vertically above.



Fig.4: A vertical view upon a part of a visualised DTM covered with artificial texture.

Texture – different possibilities are included to integrate texture. One possibility is to use artificial texture to be able to recognize the visualised 3D surfaces. Additionally it is possible to use RGB-, CIR-GeoTif-images as texture. In our case the orthophotos delivered from TopoSys have been used as texture.

## 2.2 Visualisation of laser raw-Data Spots

**Data structure** - The laser raw-data delivered by TopoSys are stored in ASCII and/or in some binary format. Each flying strip is stored in a separate file. This is not ideal for the calculation of the DSM/DTM, as well for the visualisation process. Therefore the data have been restored in a special data structure.

To avoid that millions of raw data points are repeatedly sorted and splitted into blocks, both during the up-load process and especially during the visualisation, a block oriented binary data format has been introduced. Additionally wrong and identical raw data points are deleted during the conversion process. As the raw data are stored block wise, it is also possible also to cull the data.

**Visualisation** - Based on the data structure described above a fast visualisation of a certain amount of visible raw-data points separated in blocks has been achieved. These blocks stay around the position of the camera. It is possible to choose the amount of visible data blocks. Furthermore, the spots can be additionally represented as point sprites with a free choosable diameter and colour.

## 2.3 Visualisation of additional objects

**Trees** - are presented as stem and crown. Whereas the stem is calculated as a frustum. The crown of a conifer is shown as a dark green cone and the one of a broad-leaf tree as a light-green

oval. Therefore these bodies can be visualised in real-time, as they only need little video ram on the graphic card. The visualised trees have the height and diameter as the delineated trees (fig. 12 ).

**Extended superquadrics (ESQ)** – this type of geometric body was introduced by (Jaklic 2000) and used to model trees recorded by airborne laser scanning in (Weinacker 2004) and to classify a tree as broad-leaf tree or as conifer. This 3D body is flexible, thereby enabling to calculate a more precise volume of a tree in comparison to using a generalized ellipsoid of revolution (GER) (Gong 2002, Straub 2003, Vögtle 2004). Finally it is also comfortable to use it for the visualisation of trees, as all necessary surface points and normal vectors can be calculated.

The implicit mathematical expression of a superquadric (SQ) is given with equation 1. In figure 5 a number of different SQ shapes, depending on the variation of the parameters  $e_1$  and  $e_2$ , are shown. The parameter  $e_1$  represent the curvature perpendicular to the xy-plane and the curvature  $e_2$  for the plane parallel to the xy-plane.

$$\left[ \left( \frac{x}{a_1} \right)^{\frac{2}{e_2}} + \left( \frac{y}{a_2} \right)^{\frac{2}{e_2}} \right]^{\frac{e_2}{e_1}} + \left( \frac{z}{a_3} \right)^{\frac{2}{e_1}} - 1 = f(x, y, z) \quad (1)$$

Equation 1: Implicit mathematical expression of a superquadric.

The parameters stand for:  $a_1$  = radius in x;  $a_2$  = radius in y;  $a_3$  = radius in z;  $e_1$  = curvature perpendicular to xy-plane;  $e_2$  = curvature for the plane parallel to the xy-plane.

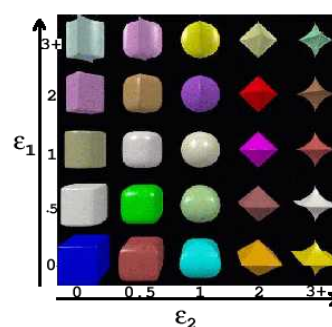


Fig. 5: Changes in shape of a superquadric (SQ) caused by the variation of the parameters  $e_1$  and  $e_2$  (Chevalier 2003).

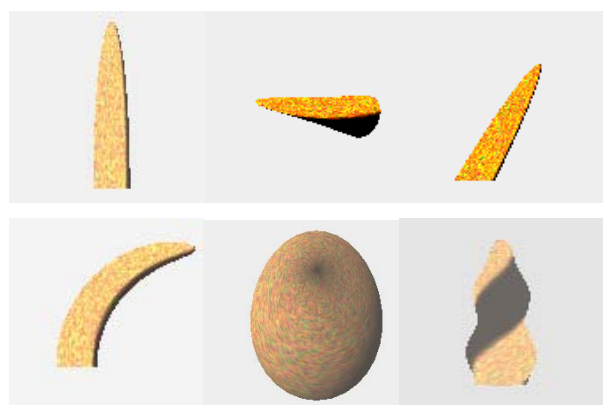


Fig. 6: This illustration is showing SQ's which have been extended with global deformations, resulting in ESQ's.



As a SQ still has a very symmetric shape additionally global deformations have been introduced. These deformations are tapering, bending, torsion, shearing in xy-direction and cavity (fig. 6). For a detailed description see (Weinacker 2004). Figure 13 shows a fitted ESQ representing a tree and the corresponding raw data points.

**Buildings** – the information about the extracted buildings is not yet very detailed, therefore it was decided to show only simple block models of the buildings (fig. 14). The form, size and orientation corresponds with the values extracted from the laser raw data or from the DSM. The walls of a building are drawn as a triangular strip. For the roof a separate model is used, where a special algorithm called “cutting ear” has been implemented in order to draw concave areas

## 2.4 Further capabilities of TreesVis

Additional to the described possibilities the developed software package offers the user the following:

- Tool for the DTM/DSM calculation (fig. 15)
- Screen-shot registration as BMP file
- Comfortable creation of AVI-film
- Visualisation of vector data (polygons, ellipses)
- Ray-tracing used for 3D point measurement and visualisation with bill-boards (fig. 16, fig. 17).
- Calculation and visualisation of intersections between two freely chosen points
- Reading/Storing of Geotiff images considering rotations in the xy-plane
- Clipping of textures (RGB, CIR images) and DSM/DTM areas

## 3. RESULTS

Screen shots have been done from the TreesVis program to show the results of the visualisation of the objects based on the methods described in the previous chapter.

### 3.1 Terrain

Based on the data captured within the NATSCAN project, the terrain rendering algorithm could be tested with raster images providing different resolutions, namely 0.5m and 1.0m, but also raster sizes down to 0.25m and up to 20m were tested and showed satisfying results.

Some figures clarify the capacity of the program for terrain rendering of large areas (fig. 7, 8, 9). Unfortunately its strength, the real-time capacity, can not be presented in a paper.



Fig. 7: The illustration shows the calculated DTM of the test area Günterstal in the black forest covered with an artificial texture. The raster resolution is 1m.

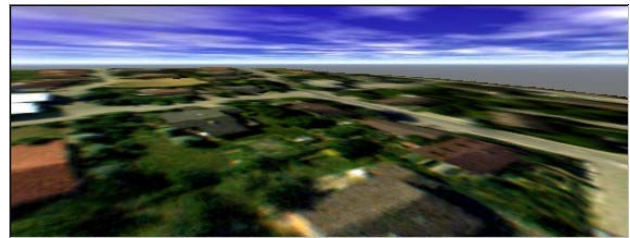


Fig. 8: A skew view of the DTM of Engen town covered by a RGB scene registered by the line scanner of TopoSys. Ground resolution is 1m for the DTM and 0.5m for the RGB image.



Fig. 9: Engen power supply line above the DSM.

### 3.2 Laser Raw Data

Figure 10 shows an example where the raw data belonging to different flying strips are presented with separate colours. The complete raw data set is visualised.

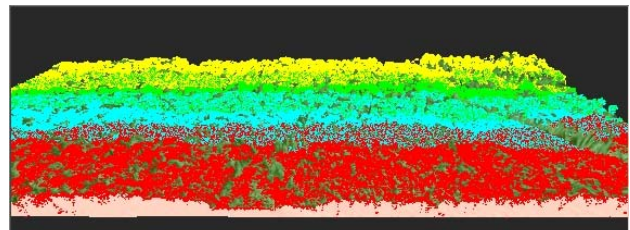


Fig. 10: This figure shows the raw-data and the DTM (light area in the bottom). Raw-data belonging to different flying strips are visualised by differently coloured point sprites.

In contrast figure 11 illustrates, that the user also has the possibility to choose a special number of raw data patches, which are visualised around the intersection point of the camera ray and the DSM. Using this alternative real time visualisation is still possible even if the total number of raw data is huge.

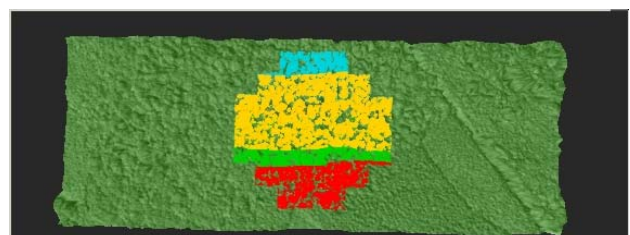


Fig. 11: Selectable number of raw data patches, visualised together with the DSM of the Mooswald test area.

### 3.3 Additional Objects

**Trees** - Figure 12 is showing deciduous trees placed upon the DTM of the Mooswald test area. The height, as well as the ellipse used for the construction of the border line of the light-crown of the trees corresponds to the extracted values. Upside-down cones marking the top of each tree.

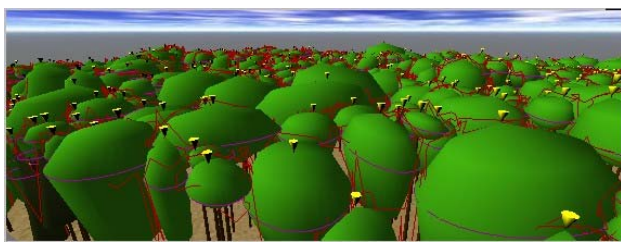


Fig. 12: Showing DTM covered with simple tree models of the delineated trees. The upside-down cone is marking the top of each tree.

**Extended superquadrics (ESQ)** – can be visualised in solid or net mode. Additionally the upper part of the ESQ is shown in a different colour as the lower half. Furthermore also the data used for the modelling can be shown. It is even possible to visualise them in different colours, depending of their weights within the adjustment process, see figure 13.

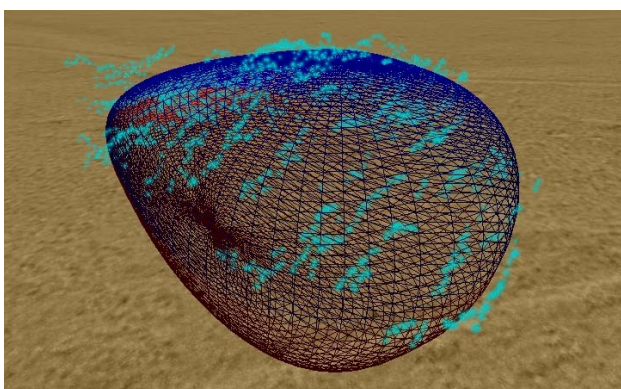


Fig. 13: This figure shows an ESQ visualised in net mode and the corresponding raw laser data points. The ESQ is representing a fitted tree.

**Buildings** – are visualised as simple cuboid without a roof and only covered with artificial texture. This simple kind of presentation is used until more detailed information about the buildings is made available. A view of the test area Engen town is shown in Figure 14.

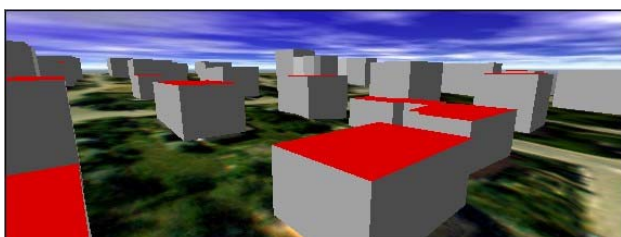


Figure 14: Simple block models are representing the extracted buildings, which are placed upon the DTM of the test area Engen town, whereas the RGB image is used as texture.

### 3.4 Further Capabilities

It is not possible to show screen shots of all the further capabilities listed above. As examples two tools are presented using figures. Figure 15 shows the window for the DSM/DTM calculation, where arbitrary rectangular, rotated areas can be chosen for the DSM/DTM calculation.

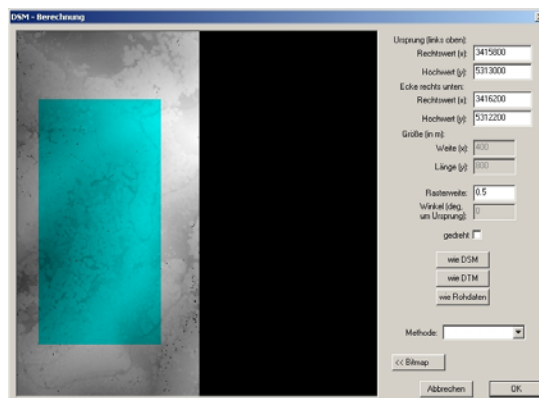


Fig. 15: DSM/DTM calculation window. The light grey area is the area covered by laser raw data and the smaller one is the DSM, which will be calculated by the method described in (Weinacker 2004).

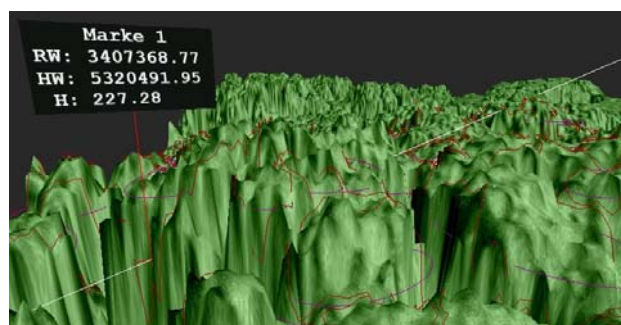


Fig. 16: Ray tracing – the white ray is the line from the camera to the measured point. The measured and marked 3D point is the first intersection on the ray with the DSM.

Figure 16 shows the result of the measuring of a 3D point on the DSM. The coordinates of the points are shown on the blackboard and the used light ray from the camera is illustrated by a white ray. Finally figure 17 shows another measured point, but this time placed on the DTM of the test area Günterstal.

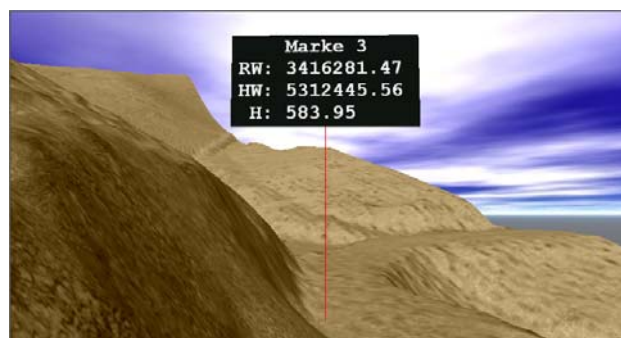


Fig. 17: This screen-shot shows a small part of the DTM of the Günterstal area nearby Freiburg and a flag (bill-board) presenting information about the measured 3D point.

## 4. CONCLUSIONS

TreesVis is a software package, which makes it possible to simultaneously handle both laser raw and multi-spectral data. The program provides the developer and the forester with effective tools for 3D real time visualisation.

Up to now the package does not have the capacity to rebuild a virtual reality landscape, but this was not the aim in the NATSCAN project.

In the future, it is intended to expand the program in such a way, that it can also handle huge data sets, which cover areas of up to 2000qkm and more, with a resolution of about 0.5m. The simple visualisation methods used for 3D objects will be exchanged by more advanced ones, which make it possible to visualise trees and buildings in a more realistic manner. Finally also dynamic processes, like the visualisation of wind streams, noise and pollution events will be integrated.

## 5. ACKNOWLEDGEMENTS

TreesVis has been developed during the NATSCAN project, which is financially supported by the German Ministry of Education and Research (BMBF). We like to express our gratitude to this organisation and to the Verein der Deutschen Ingenieure (VDI).

## 6. REFERENCES

Chevalier, L., Jaillet, F., Baskurt, A., 2003: Segmentation and superquadric modeling of 3D objects; Journal of WSCG, Vol. 11, No. 1, ISSN 1213-6972, WSCG2003, February 3-7,2003, Plzen, Czech Republic.

Duchaineau, M., Wolinsky, M., Sigeti, D., Miller M., Aldrich, C., Mineev-Weinstein, M., 1997: ROAMing Terrain: Real-time Optimally Adapting Meshes, <http://www.llnl.gov/graphics/-ROAM/>, (accessed August 2003)

Gong, P., Sheng, Y., Biging, G, 2002: 3D Model Based Tree Measurement from High-Resolution Aerial Imagery. Photogrammetric Engineering&Remote Sensing, 11(68), pp.1203-1212.

Jaklic, A., Leonardis, Solina, F.: Segmentation and Recovery of Superquadrics, Kluwer Academic Publishers, Dordrecht, 2000

Roettger, S., Heidrich, W., Slusallek, P., Seidel H.-P., 1998.: Real-Time Generation of Continuous Levels of Detail for Height Fields. In V. Skala, editor, *Proc. WSCG '98*, pages 315-322

Straub 2003 Automatische Extraktion von Bäumen aus Fernerkundungsdaten; Fachbereich Bau- u. Vermessungswesen, Universität Hannover; DGK Reihe C, Heft Nr. 572, ISSN 0174-1454, pp. 97

TopoSys, 2003. FALCON LIDAR Sensor System, <http://www.toposys.de/system/FALCON%20Outline%20-%20Jan%202003.pdf>, (accessed 1 Feb. 2004).

TopoSys, 2004, Faclon IIc, <http://www.toposys.de> (accessed Juli 2004)

Vögtle, T., Steinle, E.: Detektion und Modellierung von 3D-Objekten aus flugzeuggetragenen Laserscannerdaten; Photogrammetrie, Fernerkundung, Geoinformation; Heft 4, Jahrgang 2004, pp. 315 - 322

Weinacker, H., Koch, B., Heyder, U., Weiancker, R., 2004: Development of filtering, segmentation and modelling modules for lidar and multispectral data as a fundament of an automatic forest inventory system; In: Proceedings of the international Conference.Laser-Scanners for Forest and Landscape Assessment - Instruments, Processing Methods and Applications. Freiburg im Breisgau. Germany