Cartography of Mars in a Virtual Reality Environment

Rolf Westerteiger[1,2]

1 **DLR Deutsches Zentrum für Luft- und Raumfahrt (German Aerospace)**
Lilienthalplatz 7, 38108 Braunschweig, Germany
`rolf.westerteiger@dlr.de`
2 **Computer Graphics and HCI Group, University of Kaiserslautern**
67653 Kaiserslautern, Germany

Abstract

This work aims to investigate the suitability of applying Virtual Reality techniques in the exploration of Mars terrain features in order to support the creation of topographic maps of the planet. Traditionally, these tasks are performed using Geo-Information-Systems (GIS) on desktop workstations, with a two-dimensional projection of the collected map data as the basis on which an operator performs manual feature extraction. After identifying features and characterizing them quantitatively using measurement operators, they are ultimately represented visually by so called geo-objects which are then entered into a GIS database. Within the scope of this project, a system will be developed which enables this workflow to occur entirely within a VR-environment, using appropriate navigation and interaction metaphors. The main goal of the project is to examine whether the more natural immersion in the VR-environment can help to improve the identification and spatial analysis of surface features.

Keywords and phrases Virtual Reality Environment, interaction metaphors, spatial analysis

Digital Object Identifier 10.4230/OASIcs.VLUDS.2010.100

1 Introduction

Nowadays the creation of topographic maps is supported by 2D Geographic Information Systems (GIS) running on desktop computers. For the experts in the field, working within a 2D environment is a natural choice because, depending on the chosen projection type, certain geometric properties (such as angles) can be preserved. Historically, this has been critical for performing spatial analysis on drawn or printed map material. However, using software systems the visual representation of the data can be decoupled from the measurement process. This gives rise to the question whether a virtual reality system, consisting of an immersive 3D display along with intuitive input and navigation methods, can support the user in identifying surface features by immersing him within a natural representation of the dataset and therefore closing the cognitive gap between the user and his data. Furthermore, since a perspective projection naturally implements a focus and context scheme, it has the potential to enable more data to be displayed at the same time while keeping visual clutter at an acceptable level. In order to test these hypotheses, a VR-GIS system is being developed in coorporation with cartography experts at the German Aerospace (DLR) whose ultimate goal is the creation of a topographic map of Mars.

2 Related work

The following section gives an overview of the state of the art in 3D-GIS systems, focusing on aspects of navigation, spatial analysis and manipulation of geo-objects which can be

Visualization of Large and Unstructured Data Sets– IRTG Workshop, 2010.
Editors: Ariane Middel, Inga Scheler, Hans Hagen; pp. 100–110
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

considered the core set of functions a GIS-system needs to implement. Furthermore, the terrain rendering algorithms and data structures being employed by these systems will be mentioned.

2.1 LandExplorer

LandExplorer[5] is a desktop 3D-GIS system, which allows the users to create 3D-maps by drawing both simple geometric primitives like landmarks or polygons as well as instantiate more complex pre-fabbed geo-objects such as buildings or bridges on top of a given DTM. In order to keep rendering work manageable, the approximation-tree LoD-Scheme[2] is employed, which uses a hybrid representation mixing triangulated irregular networks (TINs) and regular grids. Geo-referenced 2D-vector overlays are rendered on-the-fly to an off-screen texture using techniques described in [9] before being mapped onto the terrain geometry. This strategy avoids some of the problems such as clipping artifacts associated with the more traditional approach of displaying 2D-geometry as offset or extruded lines on top of the terrain. The LoD-adaptivity of this approach, however, only considers viewer distance and cannot take into account the slope of the terrain, which can warp the projected area of individual texels significantly. A more elaborate approach such as vector texture maps [13] could solve this problem and provide pixel-exact rasterization of projected 2D-geometry. To facilitate exploration of the dataset at hand, information visualization metaphors such as filter lenses are being provided in order to superimpose additional data within a user-defined region, while keeping the neighbouring region uncluttered to provide orientation. One drawback of this system, however, is that it does not support out-of-core rendering of large datasets (streaming) and furthermore requires long preprocessing times. Therefore, it is better suited for localized tasks like urban planning rather than explorative planetary research.

2.2 Google Earth

Google Earth is a popular desktop application which enables the user to browse a high-resolution digital terrain model (DTM) of Earth (and nowadays, Mars) at interactive rates by using Level-of-Detail (LoD) techniques and on-demand streaming of DTM data. In spite of a well-performing implementation, LoD-transitions are readily apparent, owing to the bandwidth and latency limitations of the internet. By eliminating some degrees of freedom in camera movement, the software successfully unifies the virtual trackball metaphor with traditional pan & zoom interaction, depending on whether the view axis is tangential or perpendicular to the planet's surface, enabling a natural transition between the two modes. Even though Google Earth is not geared towards the expert user, it nevertheless provides some tools resembling simplified versions of tools provided by GIS systems, like a simple ruler tool to measure distances. Furthermore, 3D objects (geo-objects) can be constructed and geo-referenced within the system, mainly to provide the user with the means to augment Google's database by adding a visual representation of buildings in urban areas, which are not representable within the DTM model due to undersampling.

2.3 RIMS

The real-time interactive mapping system[3] is a 3D-GIS research prototype which uses a quadtree data-structure to support a level-of-detail rendering scheme. While screen-space projected size of tiles is being considered in rendering, the primary criterion for LoD selection is a user-defined region of interest (ROI). Geometric detail is highest at the center of the ROI while gradually decreasing with distance. The motivation for this is that spatial analysis tasks are commonly performed locally at a pre-selected site.

To support this spatial analysis, the sytem provides a so-called virtual geological compass to measure the orientation of surface features, which is visualized in the form of a plane which is tilted and translated manually in order to fit to surface features. The orientation of the plane is then displayed on-screen and can be used to annotate features quantitatively, which can be represented by polylines drawn directly on top of the DTM.

Furthermore, curviplanar surfaces can be created based on the polyline control points in order to visualize hypotheses about the original configuration of geological features before erosion. User studies were performed, showing that the 3D interaction with the data indeed can improve the identification phase of feature extraction.

3 Research goals

3.1 Cartography workflow

3.1.1 Identification and classification of features

The primary motivation for moving the cartography workflow into a VR environment is to investigate whether the immersive effect of VR in combination with adequate navigation interfaces can improve the perception of surface features that are of interest to the expert users. As opposed to the top-down projection used in traditional 2D-GIS, where the height attribute can only be incorporated indirectly, a perspective projection with stereoscopic rendering has the potential to close the gap between the user and the data by allowing him or her to perceive the actual topography of the terrain in a natural way, leveraging the potential of the human brain to recognize structures.

3.1.2 Instantiation of geo-objects

The process of creating a topographic map involves the creation of so-called geo-objects, which are symbolic representations of the surface features that have previously been identified. The most trivial geo-object considered here is a point object, marking a location on the planet's surface. These can be considered waypoints designating areas of interest that the user might wish to return to later for presentations or further analysis. Extended surface features are traditionally labelled using polygon primitives tracing the outline (footprint) of the feature.

Accurate placement of these geometric primtives on a workstation can be considered a simple task when performed using a mouse or a similar pointing device. However, input devices within a VR environment have different characteristics. The natural equivalent of a mouse cursor in VR is a picking ray, which is controlled by the user using a so-called wand, a input device resembling a pointer. The selected point on the surface is given by the intersection of the picking ray with the terrain geometry. Whether this method of input can improve upon traditional techniques will be investigated within the scope of this work.

Even though the final goal is implementation of the full GIS workflow within the proposed system, it is still necessary to be able to interface with common GIS databases. This will eventually enable a user to quickly switch from his workstation to a VR environment to perform a detailed analysis of a questionable terrain configuration, for example. On the other hand, geo-objects created during a VR-session could be taken to a desktop GIS in order to further characterize and organize them.

3.1.3 Spatial analysis

To help fully understand surface processes, virtual measurement tools will be provided to quantify geometric properties of features previously identified. For example, given a crater, a central quantity of interest is the volume of dislocated material. This can be computed by using a surface integral over the height field data, constrained to the inside of a user-specified polygon which represents the crater boundary. This capability can be extended to include the computation of statistic properties of other data channels, such as hydration, within a polygonal footprint.

3.2 Use as collaboration platform

Since most VR setups are multi-user capable by design, the question arises how this can be leveraged for a VR cartography system. A realistic use-case would be a geologist wanting to present a hypothesis to his or her peers. After visualizing it using geo-objects and other glyphs, a camera trajectory could be recorded which clearly documents the area in question and can serve as an anchor for an (interactive) presentation.

3.3 Public relations

In addition to its purpose in supporting geoscientific research, a secondary role of the proposed system will be to present the data and results in a way that is both accessible and appealing to a general audience. To support this usage scenario, additional rendering effects such as atmosphere, shadowing and advanced shading will be provided. Synthetic ground detail will be overlaid on top of interpolated source data in order to hide low sample-density in views close to the surface (see 9.1).

4 Reference dataset

The reference dataset which will be used in the development and initial evaluation of the system is a data product obtained by the Mars Express (MEX) mission, the first european mission to Mars. The probe, which was launcehd in June 2003, consists of an orbiter for remote sensing as well as a lander (Beagle 2) for geological surveys on-site. One of the instruments aboard the orbiter is the High Resolution Stereo Camera (HRSC), which was developed by the Institute for Planetary Research (IFP) of the German Aerospace (DLR) in Berlin. It consists of an array of line-sensors, capturing different wavelengths simultaneously while being swept across the surface (known as pushbroom-configuration). Due to this mode of operation, the resulting dataset fragments each represent long strips of the planet's surface. The sensor array consists of 2 spectrometers, 4 color sensors (red, green, blue, infrared), as well as 5 panchromatic (sensitive within the whole visible spectrum) black-and-white sensors. The latter are primarily used to acquire image data under different viewing angles in order to reconstruct height information by correlation of the 5 channels. This process is known as photogrammetry and results in a digital terrain model (DTM). The multi-spectral channels will not be available directly at first. Instead, derivative products will be used for visualization, which, for example, includes a scalar field encoding surface water density.

Eventually, the DTM as well as the individual color channels are de-warped, inverting the perspective projection. This process yields so called ortho-images, which are characterised by a constant ground sample distance (GSD), which is the distance between adjacent pixels of the image (10 to 40 meters in this case). For details about the camera operation and data processing pipeline, see [7] and [14].

Since the MEX mission is still ongoing, as of this writing only two thirds of the planet's surface have been captured. Therefore, the NASA MOLA (Mars Orbiter Laser Altimeter) dataset, which provides much less resolution but full coverage of the surface, will be used to fill in the gaps in order to provide a complete view of the planet.

5 System requirements

5.1 Data streaming

One of the requirements for the system is that the whole dataset has to be explorable as a continuum, without noticeable loading times. Considering the large size of the dataset (around 500 GiB), this calls for a streaming approach, where required data are requested from the storage backend as required. Because of the inevitable latencies involved in this data access, it is ncessary to predict user movements in order to anticipate which data tiles will most probably be required for rendering future views and issues requests to load that data in advance (prefetching).

5.2 Interactivity

In order to achieve a full immersion of the user in the virtual environment and to prevent so-called simulator sickness it is imperative to guarantee a certain bound on framerate and latency (delay between user input and effect of that input). Since the rendering of the terrain data can reasonably be expected to be the most time-consuming operation in updating the display, this translates into certain requirements on the chosen algorithm. Most importantly, given a target framerate, the algorithm has to restrict the choice of LoD-levels as to not violate this bound. For the same reason it has to be possible to suspend inter-frame coarsening / refinement operations, which can become problematic if the user moves across the surface at low altitude and high velocity, which calls for frequent reloading of data from the backend. Until requested data is made available to the frontend, the next lower available LoD-level for each region will be displayed instead.

5.3 Accuracy of rendering

In order to faithfully retain the content of the source data, it is necessary to visualize it with high accuracy. This demand is in conflict with the requirement of maintaining interactive framerates, however, since LoD rendering techniques decrease the resolution of terrain geometry with increasing distance to the viewer. This is acceptable as long as this decrease in resolution stays below the visual threshold. This can be guaranteed by algorithms which employ a screen space error threshold as a metric for controlling the LoD selection. If this threshold is set to a value smaller than a single pixel, the output of the algorithm will not be distinguishable from a full-resolution rendering. However, if the required data is not available at sufficient detail to satisfy these demands because of disk- or network-related delays, the LoD criterion might be violated, resulting in visible artifacts. Since these situations cannot, in general, be avoided, a compromise solution is to indicate within the environment whether the view represents the converged result.

An additonal challenge in faithfully representing the data lies in the fact that remote sensing data frequently contains redundancy due to overlap between adjacent datasets. Since, in the general case, the structured grids of the individual datasets are not aligned, handling these situations presents a challenge. A straightforward approach would be to resample the data on a common grid spanning the whole surface, however this would imply an undesirable low-pass filtering of the data. A possible solution to be explored is to merge the two grids involved, giving a new grid which is not regular anymore but still has a simple topology. However, the resulting data cannot be directly rendered by the LoD terrain rendering algorithms considered in this scope. On the other hand, as long as the overlap regions are not too large, a brute-force approach can be applied by rasterizing the undecimated mesh directly.

6 System architecture

The initial development and evaluation platform for the software will be a Powerwall at the DLR (German Aerospace) in Braunschweig. This is an active stereoscopic display (using shutter glasses) equipped with infrared cameras for tracking the head or input devices. The rendering frontend consists of three nodes driving one projector each as well as a master node, using an InfiniBand interconnect. The storage backend will be realized by a server offering the data to the frontend by means of a network filesystem.

6.1 Distributed execution

The system will be a distributed application executing on the frontend nodes. A SPMD (single-program, multiple-data) programming model will be employed, which is facilitated by the usage of MPI as a network stack. In order to distribute the rendering and handle frame-synchronization, OpenGL Equalizer[6] will be used, which is a lightweight minimally invasive library supporting all common VR display configurations as well as providing built-in drivers for head-tracking devices. Any input devices will be connected to the master node, which collects input events and broadcasts them to the rest of the nodes. Since the program state at every node is only a function of the processed events, this strategy, in combination with synchronization barriers at each frame swap, will ensure that all the nodes share a common, global state at any time.

6.2 Scripting environment

In order to make the system easy to extend, non-performance critical modules of the system are implemented in the Python language while rendering and other costly operations are performed within a library written in C++. Since Python is easily learned and already widely popular, this design choice will allow even non-professional programmers to easily adapt the system to different usage scenarios without sacrifying much performance.

7 Navigation

To enable the user to move within the virtual world, intuitive navigation metaphors have to be provided. As proposed in [11], independent of the navigation mode, a 2D overview map showing the current location and view direction will always be available for reference. The user will be able to jump to any location on the map by using a pointer device or by selecting a landmark from the GIS database. In order to prevent disorientation, the transition from the previous location to the target location must not be abrupt but should follow a smooth interpolating trajectory, as demonstrated by Google Earth. To improve immersion, the system will support head-tracking, adapting the view frustum to the user's position relative to the screen. Optionally, the camera trajectory will be recorded during navigation for later playback or export to a third party system for offline rendering of high quality movie sequences.

7.1 Virtual walk

Keeping the camera height fixed at eye-leve above the planet's surface by sampling the DTM below the viewpoint position, the hypothetical scenario of an astronaut walking the surface of Mars will be simulated. To prevent the user from being distracted by having to learn an unknown input system, a gamepad will be used as the primary input device in this mode. This mode of input is accessible to a wide audience while at the same time providing expert users with a large set of controls that can be mapped to functions within the VR environment. Keeping the up-vector of the camera perpendicular to the surface, the two remaining degrees of freedom defining the viewing direction can be directly mapped to an analog stick, while the two additional degrees of translatory freedom will be controlled by the direction pad.

7.2 Flyover

To get an initial impression of the topography of a larger area, the viewer can fly through the atmosphere at medium altitudes, looking down onto the terrain. This will be especially useful for surveying a larger area before beginning detailed analysis or for exploring elongated features like ridges or canyons. The parallax effect experienced due to the movement across the terrain can be expected to improve the perception of depth even further in addition to the stereoscopic rendering. By relaxing the requirement that the up-vector be perpendicular to the planet's surface, even tilting the camera would be possible, however the necessity to handle three degrees of freedom simultaneously would surely overburden most casual users, unless it was coupled to a flight dynamics simulation.

7.3 Top-down view

A top-down view of the planet, resembling the view from a satellite's position, will be provided for navigation at large scales and public presentations (flights towards the planet). A virtual trackball metaphor enables the user to rotate the planet by picking a surface point using a wand and dragging it to a new location. Alternatively, the analog stick of the gamepad can be used to control rotation along the two axes perpendicular to the view direction. Upon approaching the planet, a seamless transition to the flyover mode will occur (as in Google Earth). This means that the view direction, which initially is perpendicular to the surface, will gradually be tilted to lie in a tangential plane.

8 Terrain rendering

In the following some terrain rendering algorithms will be presented which are considered suitable for this project, based on the criteria previously given in section 5. The algorithms presented here are optimized to run efficiently on modern hardware, which due to the relative increase of the performance of GPUs as opposed to CPUs favors simple data-structures which can be represented as textures and do not perform fine-grained LoD-adaption as in previous approaches. Storage as textures implies that the input data have to be sampled on a regular grid, which is given for both the MEX and MOLA datasets considered in the reference application. Regular grids also facilitate the implementation of spatial analysis functionality, since most queries on the dataset can be trivially implemented, given that the finest LoD-level is available in memory. Applying these rendering algorithms to unstructured (e.g. LIDAR) data directly is not possible and necessitates a resampling onto a regular grid, which for irregular data implies a tradeoff between undersampling, which removes data content by low-pass filtering, or oversampling which incurs a large memory overhead. For a more general survey of terrain rendering algorithms, see [12].

Most publications concerning terrain rendering due not address the use-case of a high-altitude viewpoint, which reveals the curvature of the planet, with the extreme case requiring a spherical rendering of the data. It can be expected that this issue can be solved in a straightforward way by performing projection to a spherical surface within a vertex shader. However, care must be taken to make sure that the LoD-metrics used by the chosen algorithm are adapted to be aware of this projection.

8.1 Geometry Clipmaps

The Geometry Clipmaps[10] scheme by Losasso and Hoppe renders concentric, rectangular rings of geometry centered at the viewer's position, with a factor two decrease in linear resolution between adjacent rings as distance increases. The on-disk representation of the dataset consists of a simple mipmap pyramid which is constructed from an original full-resolution image by successive filtering and downsampling. The simplicity of this data-structure allows the application of image compression algorithms and delta-encoding of finer LoD-levels against coarser ones, since the latter is required to be already in memory when the finer level is being loaded. A compression factor of 100 for the DTM of the United States being used as reference in the publication is reported.

At runtime, the algorithm maintains a LoD-pyramid in RAM, with each level storing a square subregion (window) of the corresponding level on disk. When the viewpoint moves, these windows are shifted and the data are updated correspondingly. By making use of toroidal memory addressing, copying operations are eliminated and only the L-shaped region of newly acquired data needs to be written to RAM. If the required data cannot be acquired in time for the next frame, during rendering the next lower LoD-level is used in its place. Therefore, data structure updates can be budgeted in order to guarantee a given minimal framerate.

8.1.1 GPU implementation

The original Geometry Clipmaps algorithm as described above uses vertex buffers to present heightfield data. These are updated each time the viewer positions shifts and have to be re-transferred to the GPU in full. Soon after the original publication, a newer generation of hardware enabled vertex shaders to access texture memory directly. This has been exploited by Asirvatham and Hoppe in their follow-up work, which unifies the handling of DTM and color data by storing both as textures. For rasterization, a proxy geometry is used which does not contain z-values but only texture coordinates. Thus, for each level of resolution only one of these proxy geometries is needed which is scaled and translated as required. For each vertex (x, y) within the geometry a vertex shader procedure then reads the corresponding z-value (height) from the currently bound texture. In order to update the L-shaped region within a texture, this revised algorithm employs render-to-texture functionality, reducing bus-traffic even further.

8.2 RASTeR

This algorithm[4] makes use of essentially the same disk and memory structure as the previous algorithm, however it uses a more complex proxy-geometry constructed using a subdivison scheme based on triangular patches. In order to control the level of subdivision, a screen-space error threshold is guaranteed by evaluating the size of bounding volumes of subtrees when projected to the screen. The detail levels of the texture pyramid correspond with the level of geometry subdivision and are selected accordingly, with the triangle vertices being centered on the texels, avoiding low-pass filtering due to bilinear interpolation. The advantage to the Clipmaps approach is that a more fine-grained LoD adaptation occurs.

9 Future work

9.1 Rendering

Once basic rendering of the geometry including texture-mapping and normals for lighting works and has been optimized to fulfill interactivity and accuracy demands, further work will including improving the perceived visual quality in order to enhance immersion.

In order to make the navigation metaphor of a virtual walk visually pleasing, once must consider that the ground resolution (ground sample distance) of the available dataset is 10m at best. Because the camera is very close to the ground, this would result in large areas of flat shading. As proposed in [10], additional geometric detail (that is not present in the original data) could be synthesized by generating detail maps to modulate the normal vector.

Further ideas include rendering the sun (possibly using HDR rendering to floating point buffers followed by tonemapping) and shadow mapping. Going even further, a global illumination solution could be precomputed for a given, fixed sun position and rendered by modulating the terrain brightness using lightmaps. The motivation for this is that global illumination has been shown to improve perception of geometric configurations and might therefore simplify the identification of terrain features.

Another idea to improve depth perception in the distance, where stereoscopic separation becomes close to imperceptible, is to visualize the planet's atmosphere. Enabling OpenGL fog (fading towards a constant color based on z-distance) is a straightforward solution that has the added benefit of hiding the geometry cutoff at the z-far clipping plane, which can be used to reduce the rendering workload by culling the distant parts of terrain. On the opposite end of complexity, a weather model could be employed, which gives density distributions of gases and particles within the atmosphere. Their effect on the light arriving at the viewer could then be simulated by tracing a ray from the eye-point to the geometry for each pixel, integrating atmospheric effects such as attenuation and scattering along the path.

9.2 Extension to Earth

After evaluation of the proposed system for topographic mapping of the Mars surface using the reference dataset, it is planned to extend the functionality to different usage scenarios on earth. Considering that earth harbours a more differentiated set of surface features than Mars as well as vegetation and man-made artifacts, the corresponding GIS representations can be expected to be much more complex, which will certainly require extension of the database interfaces. Also the available remote sensing data of Earth covers a wider range of the EM spectrum, including multi-spectral images, which will at the very least complicate the user interface for selecting the mapping of data to its visual representation.

10 Conclusion

A visualization framework was presented which enables geological studies and mapping of the Mars surface within a VR environment. Interaction methods were defined on the basis of a simplified cartography workflow, with a focus on different navigation modes and corresponding input devices. From a description of two different usage scenarios, requirements for the terrain visualization component were derived which, together with assumptions about the nature of the source data, motivated the selection of a LoD terrain rendering algorithm. Furthermore, a hardware architecture was presented, consisting of a PowerWall driven by a PC cluster as a rendering frontend as well as a backend server providing data access.

References

1. Ingo Assenmacher. *Low latency technology for interactive virtual environments*. PhD thesis, RWTH Aachen, 2009.
2. Konstantin Baumann, Juergen Doellner, Klaus Hinrichs, and Oliver Kersting. A hybrid, hierarchical data structure for real-time terrain visualization. *Computer Graphics International Conference*, 0:85, 1999.
3. T. Bernardin, E. Cowgill, R. Gold, B. Hamann, O. Kreylos, and A. Schmitt. Interactive mapping on 3-d terrain models. *Geochemistry, Geophysics, Geosystems (G-cubed)*, 7, 2006.
4. Jonas Boesch, Prashant Goswami, and Renato Pajarola. Raster : Simple and efficient terrain rendering on the gpu. In *Proceedings EUROGRAPHICS Areas Papers*, pages 35–42, 2009.
5. Jürgen Döllner, Konstantin Baumann, and Oliver Kersting. Landexplorer - ein system für interaktive 3d-karten. *Kartographische Schriften*, 7:67–76, 2003.
6. Stefan Eilemann, Maxim Makhinya, and Renato Pajarola. Equalizer: A scalable parallel rendering framework. *IEEE Transactions on Visualization and Computer Graphics*, 15:436–452, 2009.
7. K. Gwinner, F. Scholten, M. Spiegel, R. Schmidt, B. Giese, J. Oberst, C. Heipke, R. Jaumann, and G. Neukum. Derivation and validation of high-resolution digital terrain models from mars express HRSC data. *Photogrammetric Engineering & Remote Sensing*, 75(9), 2009.
8. Lok M. Hwa, Mark A. Duchaineau, and Kenneth I. Joy. Adaptive 4-8 texture hierarchies. In *VIS '04: Proceedings of the conference on Visualization '04*, pages 219–226, Washington, DC, USA, 2004. IEEE Computer Society.
9. Oliver Kersting and Jürgen Döllner. Interactive 3d visualization of vector data in gis. In *GIS '02: Proceedings of the 10th ACM international symposium on Advances in geographic information systems*, pages 107–112, New York, NY, USA, 2002. ACM.
10. Frank Losasso and Hugues Hoppe. Geometry clipmaps: terrain rendering using nested regular grids. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, pages 769–776, New York, NY, USA, 2004. ACM.
11. Verbree E.and Maren G.V.and Germs R.and Jansen F.and Kraak M-J. Interaction in virtual world views-linking 3D GIS with VR. *International Journal of Geographical Information Science*, 13(4), 1999.
12. Renato Pajarola and Enrico Gobbetti. Survey on semi-regular multiresolution models for interactive terrain rendering.
13. Nicolas Ray, Thibaut Neiger, Xavier Cavin, and Bruno Lévy. Vector texture maps. Technical report, INRIA - ALICE, 2005.

14 F. Scholten, K. Gwinner, T. Roatsch, K.-D. Matz, M. Wählisch, B. Giese, J. Oberst, R. Jaumann, and G. Neukum. Mars express HRSC data processing - methods and operational aspects. *Photogrammetric Engineering & Remote Sensing*, 71(10), 2005.