

Interactive Design of 3D Maps with Progressive Projection

Helen Jenny, Bernhard Jenny and Lorenz Hurni

Institute of Cartography, ETH Zurich, Switzerland
Email: helen.jenny@karto.baug.ethz.ch

The progressive projection is by origin a manual cartographic technique, traditionally used by panoramic landscape painters; however, it is rarely encountered in digitally created three-dimensional (3D) maps. In this article, the advantages of this specific projection when designing 3D maps are presented, the processes involved in its manual construction, as well as the existing and potential digital implementation approaches, are reviewed. A new algorithmic solution is described, allowing for user-friendly interactive bending of a terrain model into a progressive view, with options to add a curved horizon, to vertically exaggerate the terrain, and to create a 360° strip panorama. The resulting software, Terrain Bender, is freely available for download.

Keywords: panorama, terrain bending, surface deformation, cylindrical projection, 3D cartography

INTRODUCTION

This article focuses on the progressive projection, a specific projective construction technique traditionally encountered in hand-painted cartographic panoramas, and on its value and application to digital three-dimensional (3D) map design. This special technique depicts the panorama foreground in a two-dimensional (2D) map-like way as if looking straight down on the landscape from a high point of view; in contrast, the background is portrayed as if gazing towards the distant horizon from a low point of view; the midground is a transitional zone (Figure 1). Such a progressive view, resembling the way an airplane passenger perceives the landscape, can also be imagined as the result of combining a sequence of merged standard perspective views. This is in contrast to a standard central perspective projection with a single static camera where an often unsatisfactory compromise between the optimal display of the foreground and the best display of the background has to be found. The progressive projection thus extends the range of standard viewing possibilities.

In the context of this article, we understand ‘3D map’ as an obliquely viewed, three-dimensionally perceived cartographic representation. We include in this definition the cartographic panorama and the bird’s eye view, which show the landscape’s features, its topography and proportions in a topologically correct way. In hand-painted landscape panoramas, artists have used a number of refined construction techniques. They may, for example, select a specific projection, in order to bring important landscape elements or regions to the viewer’s attention by showing them from an optimal viewing angle. Yet, in digitally created 3D maps, this variety of available projections is missing since the projection options offered by

standard rendering software are usually very limited, and are often not geared towards the needs of cartographers. The hand-painted cartographic panorama can therefore serve as a source of inspiration for the design of digital 3D maps. The progressive projection is such a traditional projective technique, which is rarely encountered in digital maps, but has been widely used by well-known painters of panoramic maps, such as Heinrich C. Berann (Patterson, 2000), Max Bieder (Maggetti, 2000), Winfried Kettler (Kettler, 1986), James Niehus or Hal Shelton (Tait, 2008).

This article explores the advantages of designing 3D maps in progressive projection, as well as conventional manual methods of its construction. Other related methods to digitally create 3D maps with progressive views are critically reviewed. A new algorithmic solution, developed by the authors, is then presented, which allows user-friendly interactive bending of a terrain into a progressive view, with options to add a curved horizon, and to vertically exaggerate the terrain. We also introduce cylindrical progressive views, which combine a 360° cylindrical projection with a funnel-shaped deformation of the terrain. The Terrain Bender software, developed by the authors, implements these newly devised solutions and is freely available.

ADVANTAGES OF THE PROGRESSIVE PROJECTION

The progressive projection offers a number of advantages when designing a 3D map compared to using a standard central perspective projection. It enables the cartographer to create a ‘real’ horizon, to better convey the shape of the landscape, to focus user attention, to gain image display depth and to reduce occlusions.



Figure 1. Panorama of central Switzerland with progressive projection (Max Bieder, 1938)

Many panorama painters, such as Heinrich C. Berann, prefer the view from lowlands to highlands, depicting a horizon with high mountainous terrain in the map background (Patterson, 2000). In standard central perspective projection, the background of the 3D map is simply cut off where the terrain ends, without conveying the impression of a real horizon (Hölzel, 1963) (Figure 2, left). This undesirable cut-off line is the result of looking at the terrain from a steep viewing angle (or a high viewpoint). When a central perspective projection with a flat viewing angle (or low viewpoint) is chosen, a horizon is formed by mountains in the background, but unfortunately, a uniform flat viewing angle flattens the entire terrain and as a result compresses the foreground. The progressive projection solves this problem by progressively varying the viewing angle from steep in the foreground to flat in the background (Figure 2, centre).

On many hand-painted panoramas, the mountains in the background are shown from a lower point of view. According to Hölzel (1963), and Berann and Neugebauer (1987), showing the background under a flat viewing angle,

allows a better portrayal of the characteristic shapes of mountain ranges. In large- and medium-scale maps, a flat viewing angle for the background can also be used to draw the reader's attention to certain elements within the landscape.

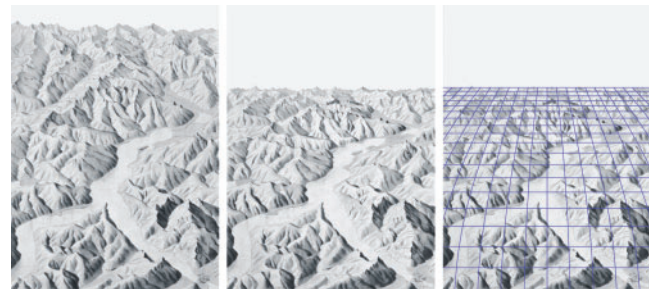


Figure 2. Lago di Como by Hölzel (1963) in central perspective projection (left) and progressive perspective projection (middle); construction grid for manual creation of progressive perspective added by Kern (right) (1986)

By applying the progressive projection, the image also gains display depth. A gain in display depth by means of the depiction of a horizon line is especially useful when portraying medium- and small-scale maps. This way, an entire country or continent can fit onto a single map sheet, while still retaining a pronounced 3D appearance (Figure 3). In the foreground and middle ground, landscape elements are depicted with a steeper viewing angle, which reduces the area of terrain features obstructed from view.

HAND-PAINTED TECHNIQUES FOR THE PROGRESSIVE PROJECTION

Manual construction of the progressive projection

In the past, when 3D maps in progressive projection were created manually, panorama painters drew a map construction grid on the canvas, which was then filled with horizontal strips of centrally projected landscape sections, with a gradually sinking horizon (Kern, 1986). Figure 2 shows the area of Lake Como in standard central perspective projection, in central perspective projection with progressive projection, and a construction grid for the latter. Hölzel (1963) observes that it is difficult to provide exact formulas for the manual construction of a progressive projection, since the construction grid, and the applied degree of foreshortening, need to be specifically adapted to the landscape situation that is to be displayed. He concludes that it is more convenient to customize the design of the progressive projection whilst sketching the panorama, in order to avoid the intricacies of pre-computations.

Vertical exaggeration and curved horizon

Small-scale hand-painted cartographic panoramas in progressive projection, are sometimes enhanced by vertically exaggerating the terrain features, and curving of the

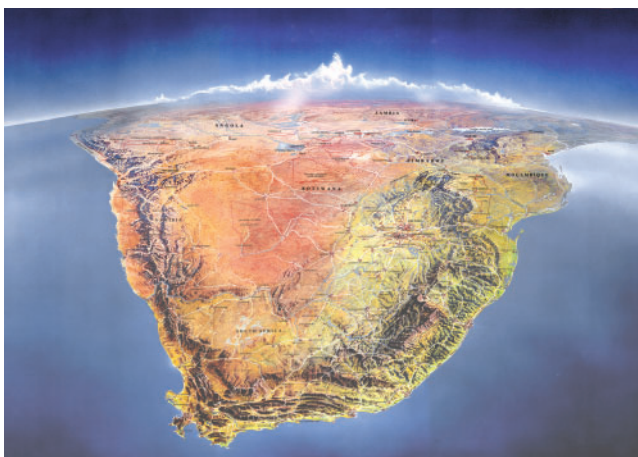


Figure 3. Small-scale depiction of southern Africa with curved horizon, progressive projection and vertical exaggeration of mountains (H. C. Berann)

horizon. The curved horizon adds more spatial depth to the map, complementing the effect of the progressive projection, and implying a globe-like shape (Figure 3). Without vertical exaggeration, mountain ranges on maps of very small scale are barely recognisable, since their altitude is very small in relation to the larger depicted area, so vertically exaggerating mountain ranges is a common means of giving the relief appropriate significance, especially on small-scale 3D maps and for low relief.

From manual to digital construction

When creating a progressive projection manually, most panorama painters have used a ‘designing-while-painting’ approach, which is both time consuming and insufficiently formalized for easy automation. To digitally construct the effect, a specific mathematical and algorithmic framework needed to be developed, making use of computational power, and allowing the cartographer to inspect and adjust different variants quickly and easily. In this article, we present such a framework, together with a software implementation that allows the cartographer to concentrate on the design aspects of progressive terrain bending, without needing mathematical knowledge. By interactively manipulating different adjustment parameters, the cartographer can create a variety of panoramic views of a terrain model in progressive projection. Before describing this new method in the section on ‘Interactive design of progressive bending for 2.5D terrain models’, however, the next section provides an overview of other possible digital approaches to digitally creating a progressive perspective, and previously implemented methods are critically reviewed.

DIGITAL IMPLEMENTATION APPROACHES AND PREVIOUS WORK

Rendering 3D maps in progressive projection involves projection methods that are unrelated to common 2D map projections, e.g. the Mercator or Mollweide projection. A projection in 3D computer graphics transforms a 3D model into a 2D image that can be visualized on a screen. Most images generated with computer graphics methods use the central perspective projection, or a parallel projection.

The central perspective projection corresponds roughly to the way in which the human eye perceives the world: parallel lines extending from the viewer into the image depth converge to a central vanishing point, while parallel lines which are perpendicular to the viewing direction, stay parallel. Thus objects of the same size appear larger in the foreground and are foreshortened in the background. When parallel projection is used, projection rays do not converge but stay parallel, and an object has the same size independently of its distance to the viewer. Each projection has clearly different applications. Cartographers also use the cylindrical projection, which generates a 360° strip panorama. It is possible to modify any of these three projections to a progressive projection.

When digitally implementing the progressive projection, cartographers and computer scientists have drawn from two major groups of approaches. They either modify the camera projection (group 1), or they deform the shape of the 3D

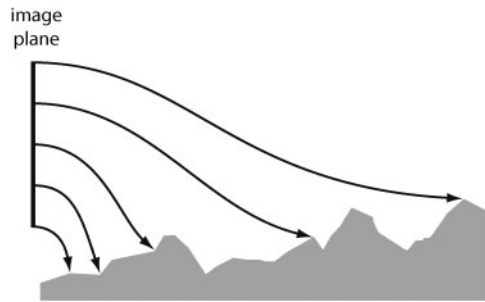


Figure 4. Curved rays for generating a progressive perspective image

object before rendering it with a standard camera (group 2). The first group includes the ray curving approach, and the interpolated camera approach; the terrain bending approach is part of the second major group. Table 1 shows an overview of implementation approaches for the progressive projection.

The ray curving approach

Whilst standard software in computer graphics uses straight projection rays for 3D representation, the ray curving approach, as its name suggests, applies curved rays. Since only the projection and not the shape of the rendered terrain is altered, the geometry of the terrain model is preserved. The ray curving approach is usually implemented using a ray tracer. Ray tracing is a technique for generating an image by means of tracing the path of light through pixels in an image plane. The pixel adopts the colour of the object that is first hit by the light ray; secondary rays can be used to add reflection, refraction or cast shadows. To create a progressive projection, the rays of light that intersect with the terrain are bent downwards in the foreground, resulting in a steep viewing angle, and upwards in the background, resulting in a flat viewing angle (Figure 4).

Falk *et al.* (2007) implemented a ray curving approach using a force field. They applied progressive projection to skiing panoramas, and their ray curving allowed them to avoid undesirable localized visual obstructions. Since the approach is computationally intensive, a GPU-accelerated implementation was deemed necessary. The algorithmic implementation is complex, since bent rays must not intersect; otherwise, the topology of the landscape can be falsified, and parts of the landscape could appear more than once in the rendered view. Interactively manipulating the rays in order to produce the desired effect demands a high level of abstract thinking from the user. An intuitive and easy to handle interface, one that gives the cartographer a

Table 1. Different ways to render a terrain with progressive projection. Previously proposed (*italic*), and unexplored methods for terrain models (*)

Camera modification	Surface deformation
<i>Ray curving</i>	<i>2.5D terrain bending</i>
<i>Interpolated camera</i>	3D free form deformation*
Projection surface modification*	3D vector field deformation*
Virtual projection lenses*	3D embedded deformation*

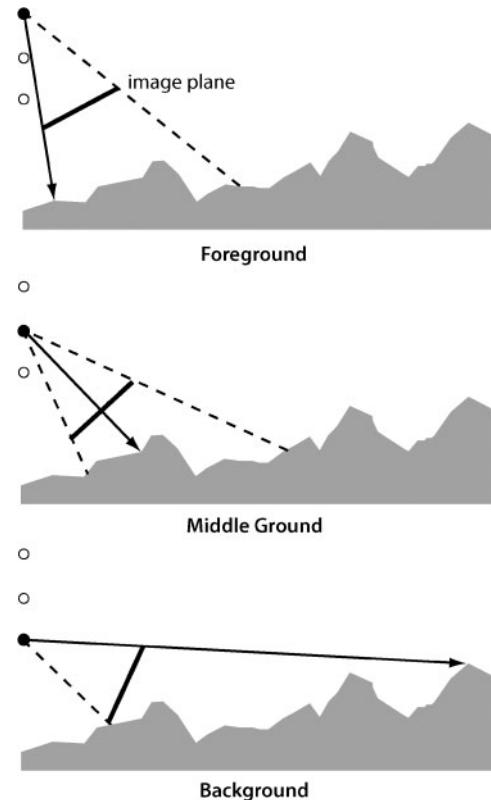


Figure 5. Progressive perspective by camera interpolation: camera for bottom row showing the foreground (top), intermediate camera (middle) and camera for top row showing the background (bottom). To simplify this figure, camera positions only change vertically

proper level of control, seems to be difficult to develop for ray curving approaches.

The interpolated camera approach

The interpolated camera approach implements the progressive projection by interpolating between two imaginary cameras (Jenny, 2004). The user defines the geometry of two central perspective cameras (Figure 5). The first one is used for the bottom zone in the rendered image, i.e. for the foreground of the view, the second camera is used for the top-most zone of the image, i.e. for the background of the view.

For each image zone in between, separate intermediate camera parameters are interpolated. The user freely positions the two extreme cameras and adjusts their vertical inclination angles; the axis through the two camera centres defines the azimuthal direction of view. This approach also preserves the original terrain geometry.

When testing this approach, we found it difficult for the user to foresee what effect the manipulation of the camera parameters would have on the resulting image. As with the ray curving approach, giving the cartographer sufficient control to design the progressive projection poses a problem.

The terrain bending approach

The terrain bending approach creates a progressive perspective view by means of surface deformation. Two

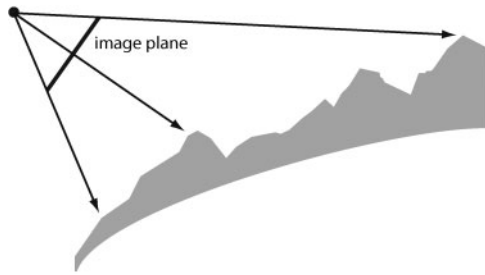


Figure 6. Progressive view by terrain bending

approaches are possible: bending the foreground and applying a flat viewing angle (Figure 6), or bending the background and using a steep viewing angle. This approach manipulates the terrain geometry and usually uses either a central perspective or parallel projection for display.

Surface deformation methods have been devised for 3D models as well as for 2.5D models with regular grid structure. The regular 2.5D grid structure contains only one altitude value per grid cell. Some 3D shapes requiring two altitude values per cell (e.g. overhanging cliffs) cannot be represented with a 2.5D model. Yet, it is the preferred model in cartography since it is widely used and very easy to process. Degener and Klein (2009) automatically deform a 2.5D terrain surface in order to maximize the visibility of a specific set of features. Patterson (2001) works with commercial software applications, and uses a workaround to bend 2.5D terrain to create a progressive view. He first imports a digital elevation model as a greyscale image into a raster graphics editing program, where it is merged with a greyscale gradient image representing an arc. The modified image is then written to a file and imported into a 3D rendering program to inspect the result. As Patterson (2001) observes, this method contains some inconveniences: working with greyscale images may introduce terracing artefacts on the terrain, due to limited resolution of the greyscale image (especially when using 8-bit images); editing a greyscale image representing elevations is not intuitive; results cannot be immediately verified; and numerous conversions, and import and export procedures delay the work process. Nevertheless, the simplicity of deforming terrain by adding or subtracting elevation values is intriguing, because it is fast to apply, not algorithmically complex, and easy to grasp and control for the user. The new algorithm presented in the section on 'Interactive design of progressive bending for 2.5D terrain models' of this article uses the principle devised by Patterson

without burdening the user with the previously mentioned inconveniences.

Unexplored approaches

There exist a number of additional implementation methods from computer graphics that either modify the projection, or deform the shape of the object to be rendered. These methods have been implemented in experimental systems but are not available in standard rendering engines. We think that they could be applied to create a progressive projection (Table 1), but to our knowledge, they have not been used for this purpose.

Among the first major group of approaches using modification of the camera projection are: Levene (1998), generating non-linear, non-realistic projections, allowing for user control of the projection surface's shape; Yu and McMillan (2004), using non-planar projection surfaces; Yang *et al.* (2005), placing a virtual projection lens in front of a projection plane to alter the projection rays.

The second group consists of methods for surface deformation. Within this group, methods for 3D shape deformation (in contrast to 2.5D) are numerous. Among the 3D deformation methods are free form deformation (Sederberg and Parry, 1986) that applies coarse deformation to complex shapes, algorithms that use a vector field to deform objects (von Funck *et al.*, 2006) and algorithms deforming space through the direct manipulation of objects embedded within it (Sumner 2007). We are not aware of surface deformation methods for 3D meshes that have been specifically developed to create specialized projections for cartographic landscape panoramas.

INTERACTIVE DESIGN OF PROGRESSIVE BENDING FOR 2.5D TERRAIN MODELS

Our new approach of creating a progressive view shares with Patterson's method (2001) the principle of manipulating the altitude values of a 2.5D elevation model. However, unlike Patterson, we can create, adapt and evaluate the progressive view interactively, within a single software application. Our algorithm creates a terrain base that the user can interactively shape: the altitude values of the terrain model and those of the terrain base are combined by adding them together to create a deformed terrain model (Figure 7).

Our application allows the user to preview the deformed terrain and to apply simple texturing and lighting options, and the deformed model can then be exported and rendered

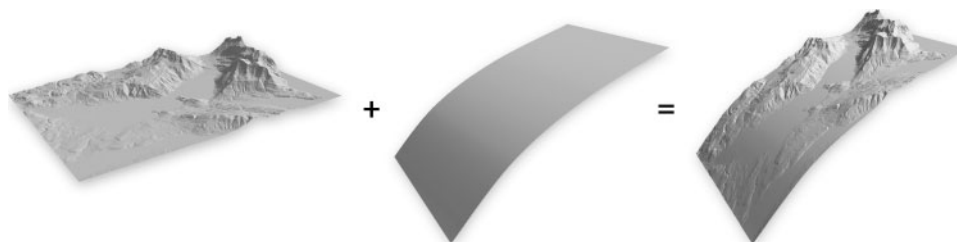


Figure 7. Deforming a terrain model (side view) by combination with a bent base; the camera for rendering the view is placed on the left looking towards the right

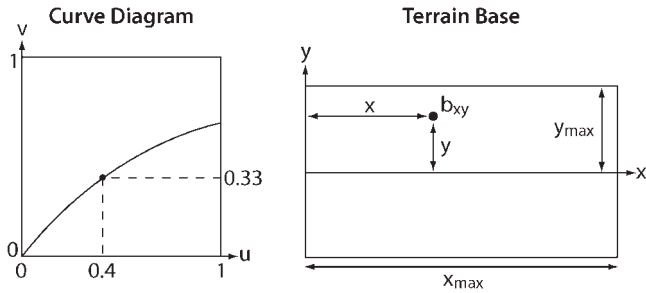


Figure 8. Interactive curve diagram and its influence on the bending of the terrain base

using a more sophisticated renderer, e.g. with a ray tracer. In the following sections, the different algorithmic steps and control options of our method are explained: constructing the terrain base, curving the horizon, and vertically exaggerating the terrain. We also explain how we deal with large terrain models, and compare our 2.5D terrain deformation method to 3D deformation methods.

Terrain base construction

A terrain base is constructed as a 2.5D grid that is independent of the original terrain model, the base grid having the same number of grid cells as the terrain model. The user can influence the degree of bending by interactively manipulating a bending curve. Control points can be added to the curve and dragged, to form the curve into the desired shape. This interactive graph resembles the gradation curve for brightness correction in images, which is well known to cartographers from imaging applications, such as Adobe Photoshop. In our method, the curve diagram (Figure 8, left) corresponds to the profile of the bent terrain base, looking from the foreground (left) to the background (right) of the model.

The vertical axis of the graph represents the base-bending factor: no bending (bottom) to maximum possible bending (top). The curve is interpolated through the control points using Catmull-Rom splines, a type of smooth C1 continuous piecewise curve. For every value on the horizontal axis u , the corresponding vertical value $v = g(u)$ can be computed with high precision, which avoids the creation of terracing artefacts, a problem that afflicts Patterson's method. For each cell in the terrain base grid, the relative distance to the foreground is computed: $u = x/x_{max}$ (Figure 8). The deformation value v is then derived from the curve: $v = g(u) = g(x/x_{max})$. Since the deformation value is a relative measure (e.g. between 0 and 1), it needs to be multiplied with a scaling factor s_b . As the scaling factor, we chose the longer side (length or width) of the terrain model, in metres. This empirical scaling factor proved to be adequate for all models tested. The scaled value b_{xy} is then stored in the terrain base grid: $b_{xy} = g(x/x_{max}) \times s_b$ with $s_b = \max(x_{max}, 2y_{max})$. A 3D terrain model preview is computed on the fly whenever the user manipulates the curve graph, so that changes can be inspected and corrected immediately.

Curved horizon

An optional curved horizon can be added to the 3D map by bending the terrain model perpendicular to the progressive

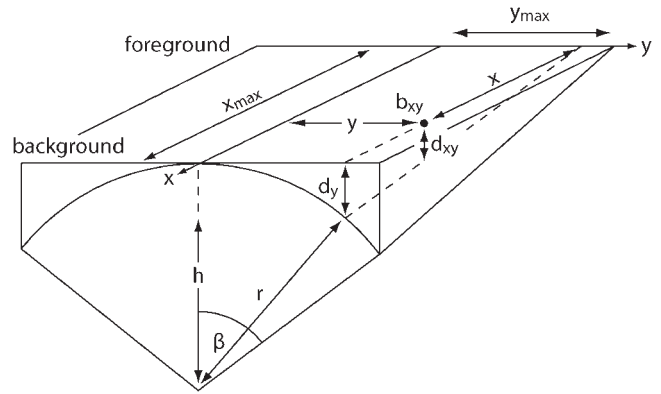


Figure 9. Scheme for adding a curved horizon to the terrain base (inverted view from background towards foreground)

bending direction. The user can interactively select the amount of curvature for the horizon, which is fully applied to the background and fades out, linearly, towards the foreground.

The algorithm progresses from background to foreground, and alters the terrain base values. The horizon curvature can be visualized as a tunnel, touching the terrain base in the middle along the x -axis, and becoming flatter from background to foreground (Figure 9). For each row of the terrain base, the radius r of the tunnel arc is computed as $r = y_{max} / \sin \beta$. For each cell in the background row, the distance d_y between the tunnel arc and the terrain base is calculated as

$$d_y = r - b = r - (r^2 - y^2)^{1/2}$$

To let the curvature fade out linearly towards the foreground, the distance d_{xy} between the tunnel arc and the terrain base is linearly interpolated for every elevation value as $d_{xy} = d_y \times x/x_{max}$. The distance d_{xy} is then subtracted from the terrain base grid value, so that for every grid cell in the terrain base, the new value is $b_{xy} = [g(x/x_{max}) - d_{xy}] \times s_b$. Horizon curvature is only applied to the terrain base, which is in turn combined with the terrain model (Figure 10).

Vertical exaggeration of the terrain

The terrain is vertically exaggerated by multiplying the altitude values of the terrain model with a user-defined factor between 0 (terrain base only, without terrain details) and 10. The user can set two exaggeration factors: s_{fg} for the foreground and s_{bg} for the background. The scale factor s_t is then linearly interpolated as $s_t = (1 - x/x_{max}) \times s_{fg} + x/x_{max} \times s_{bg}$ (Figure 8) and applied to each cell t_{xy} of the terrain model: $t_s = s_t \times t_{xy}$. The exaggerated terrain model is then combined with the terrain base: $c_{xy} = b_{xy} + t_s = [g(x/x_{max}) - d_{xy}] \times s_b + s_t \times t_{xy}$.

Often, for large-scale 3D maps, no vertical exaggeration is needed. Larger exaggeration factors are appropriate for small-scale maps in order to increase the visibility of the proportionately small terrain elevations.

Dealing with large terrain models

Large terrain models are memory intensive, and depending on the hardware available, they may take a long time

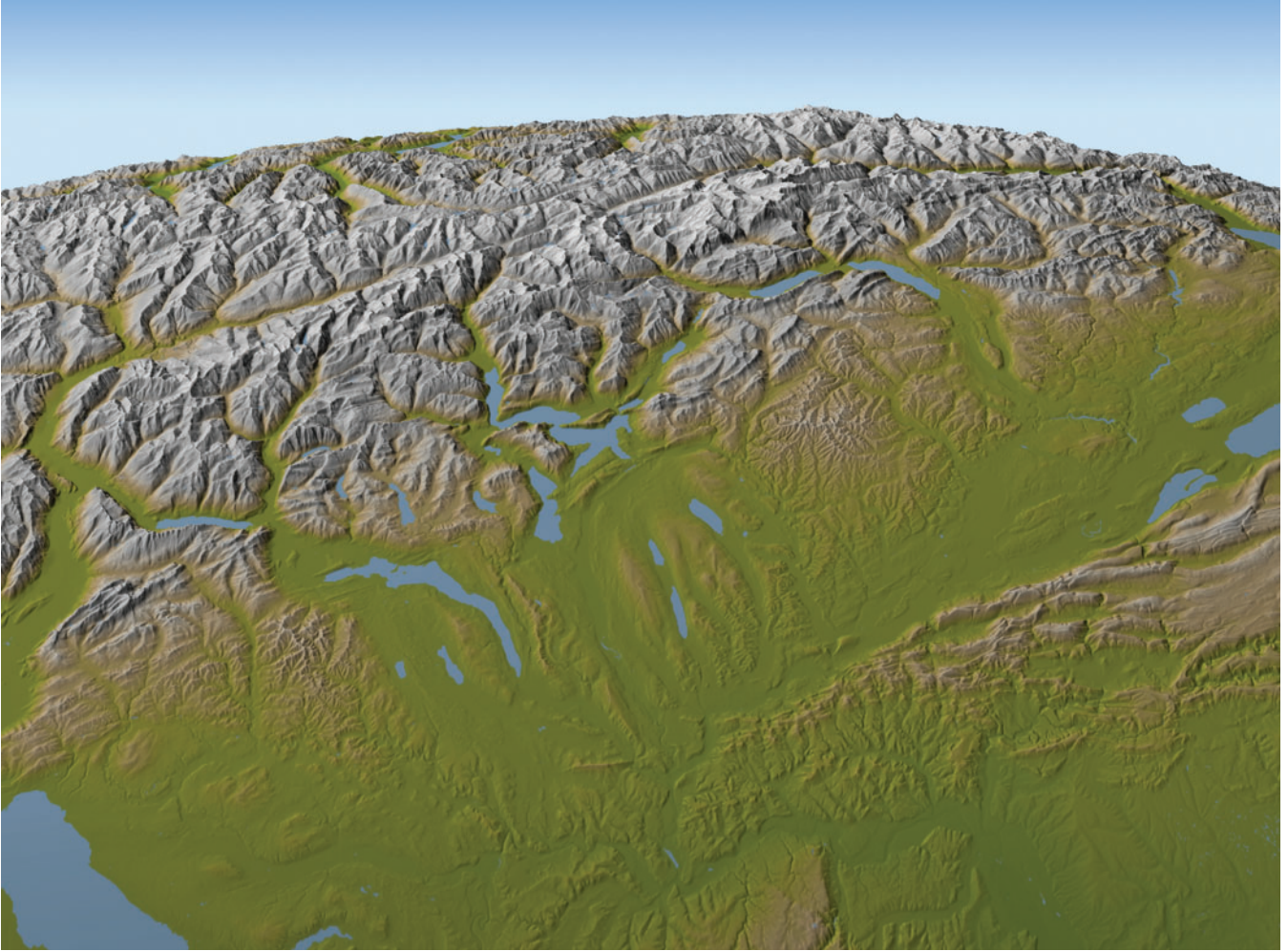


Figure 10. Progressive perspective with curved horizon (south-oriented view of Switzerland)

to display, which can disturb the workflow. To allow the user to rapidly deform large terrain models, a Gaussian pyramid is constructed in our application. A Gaussian pyramid is a stack of downsampled terrain models consisting of successively smaller grids. The original terrain model can be imagined to be at the bottom of the pyramid, and every subsequent model is half as long and half as large as the previous level, its grid cell number being four times smaller. Each grid cell contains a local average that corresponds to a grid cell neighbourhood on a more detailed level of the pyramid. The memory overhead needed to store the pyramid is only one-third of the original terrain model. To find a local average for the subsequent pyramid level, we use a 5×5 cells Gaussian convolution filter, giving close cells a greater influence on the averaged value than cells that are further away. Gaussian filtering has the effect of removing fine (high-frequency) details from the terrain model, yielding a series of generalized models that are well suited for previewing. The user can choose the pyramid level to display. When exporting the bent terrain model, the deformations are executed on the original full-resolution model, and written directly to the output file.

2.5D versus 3D terrain deformation

In contrast to a 3D terrain model, a 2.5D terrain model is somewhat limited in the landscape forms it can represent. For example, a 2.5D terrain model cannot accurately portray overhanging cliffs because it would need to store at least two altitude values per grid cell. Yet 2.5D regular elevation models are the structures that cartographers are most often confronted with when working with digital elevation models, and their simple structure facilitates many tasks. For this reason, our algorithm is specifically adapted to bending 2.5D terrain models. When bending terrain with our method, elevation values are shifted only vertically, hence overhanging formations, which would require a 3D terrain model format, cannot be created, and extreme bending of the horizon over 180° , as shown on Berann's panorama in Figure 11, is not possible with a 2.5D model. However, we consider our algorithm to be suitable for most cases where cartographers need to apply a progressive projection. Our algorithm offers the advantage of keeping the elevation model in geo-referenced form, i.e. the horizontal x - y position of each cell does not change, keeping the regular 2.5D grid structure intact. As a consequence, draping geo-referenced textures and vector

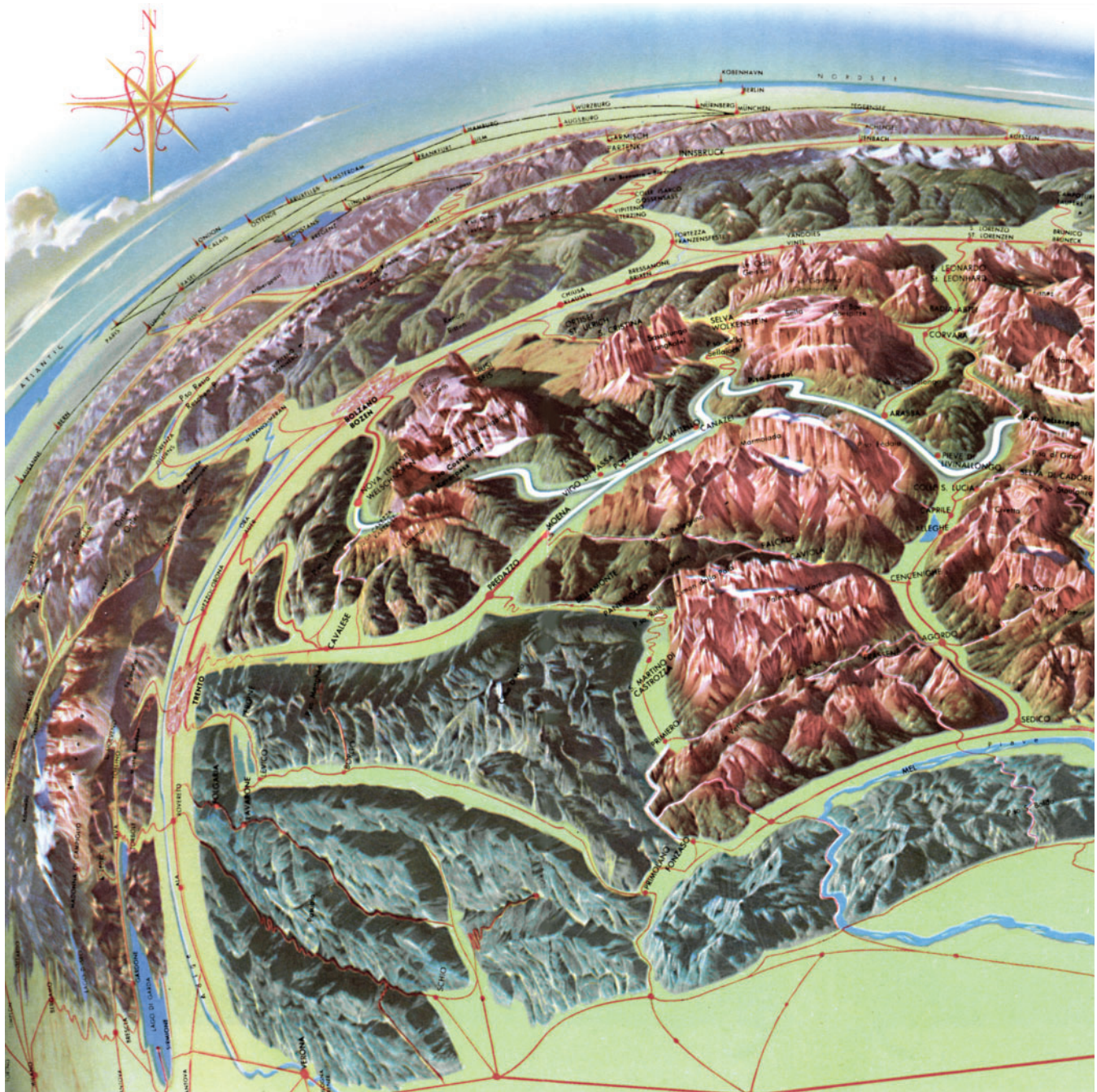


Figure 11. Extreme deformation of the horizon in H. C. Berann's panorama 50 Jahre Dolomitenstrasse (excerpt, 1958)

data onto the terrain model, as well as post-processing with other 2.5D specific software, can be done with ease.

THE PROGRESSIVE-CYLINDRICAL PROJECTION

With our terrain bending software, the user can preview the deformed model in both central perspective projection and parallel projection. Additionally, we developed a new type of progressive terrain rendering, by applying a funnel-shaped deformation to the terrain and rendering the deformed model using a cylindrical projection. The result is an unwrapped 360° strip panorama, where the

landscape's foreground is progressively bent towards the observer.

Cylindrical projection

A standard cylindrical projection generates a 360° strip panorama by first projecting the surrounding landscape onto a virtual vertical cylinder, and then unwrapping the cylinder, generating a plane image (Figure 12). Cylindrical panoramas are often displayed on the summit of mountains frequented by tourists, giving information about the surrounding scenery. They can be constructed manually, or rendered by specialized ray tracing software.

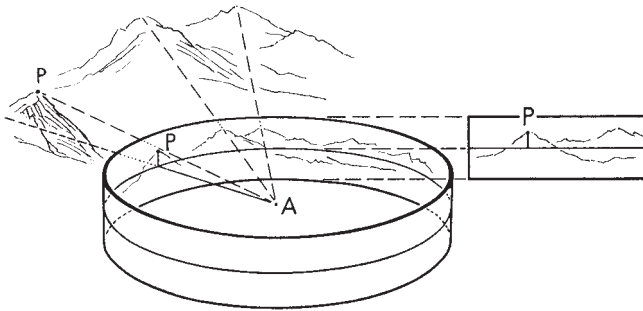


Figure 12. Cylindrical projection (Imhof, 1963)

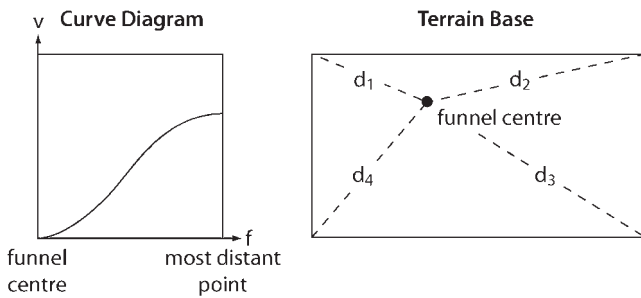


Figure 13. Schema for computation of funnel deformation in the terrain base grid

A new combination: the progressive-cylindrical projection

To produce a cylindrical strip panorama with progressive view using our digital method, a funnel-shaped deformation is applied to the terrain model. The user selects a location for the funnel centre on the terrain, which corresponds to the centre of the projective cylinder. The

user can adapt the bending of the terrain using the curve diagram, as described in the section on ‘Interactive design of progressive bending for 2.5D terrain models’. The difference between this and the centrally projected model is that bending is not applied in one direction only. Instead, one could imagine standing in the centre of the funnel and turning on the spot, while at the same time, sweeping the curve diagram (or rather its impact on the terrain) in a full circle. The left hand side of the curve diagram’s horizontal axis is placed at the centre of the funnel, corresponding to the foreground or the lower part of the strip panorama; the right end of the horizontal axis is the point farthest away in the $x-y$ plane, corresponding to the background or top of the image.

Algorithmically, the funnel is created by applying a distance-weighting factor to the base grid values. First, the distance between the funnel centre and the most distant cell in the $x-y$ plane of the terrain is evaluated: $d_{ref} = \max(d_1, d_2, d_3, d_4)$; in Figure 13, the maximum distance is d_3 . Terrain parts that are equidistant from the funnel centre receive the same degree of bending. The distance weighting factor f is calculated for each cell in the terrain base by dividing its distance to the funnel centre by the maximum distance: $f = d/d_{ref}$. On the curve diagram, the user can then assign a deformation value v to cells with a certain distance weighting factor f . Cells on the left of the horizontal axes are close to the funnel centre, while cells on the right are located far from the funnel centre. Figure 14 shows a terrain model with a funnel deformation (bottom), and the resulting panoramic view (top).

The user can slide the camera up and down on the z -axis in order to select an altitude for the point of view. The terrain model with the embedded funnel deformation can

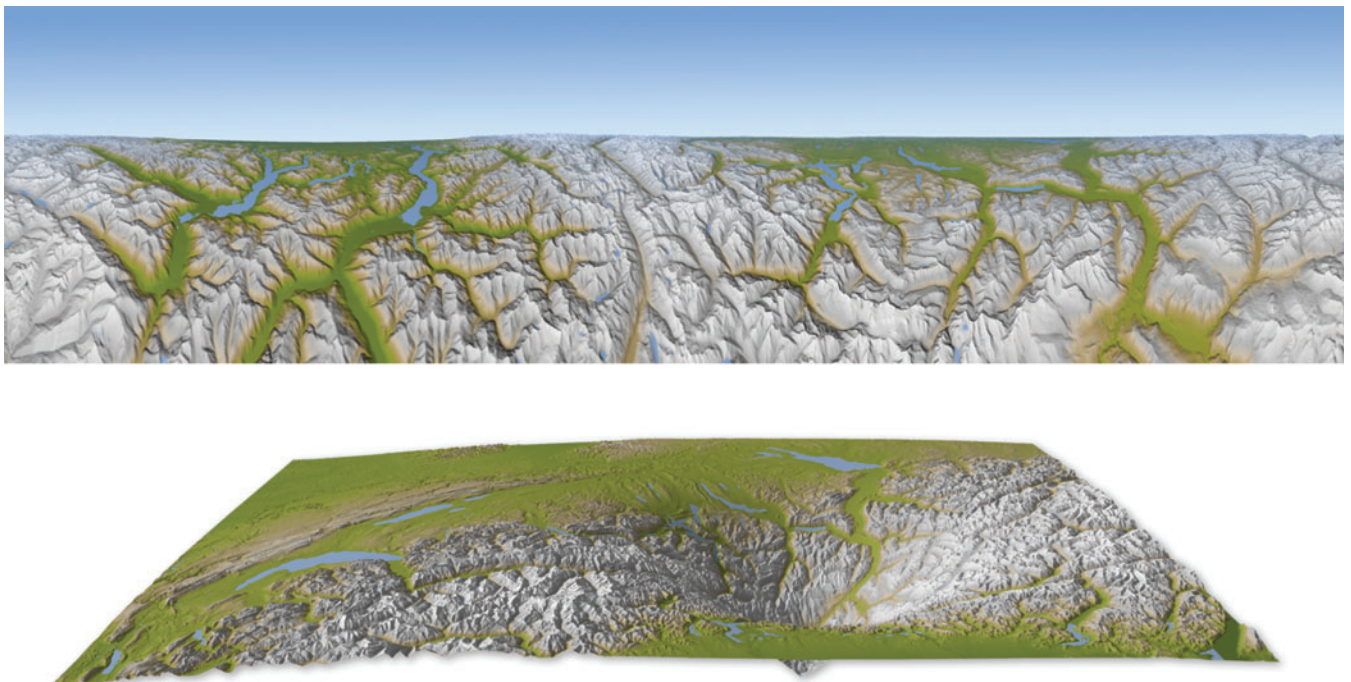


Figure 14. Panorama in progressive-cylindrical projection (top) derived from funnel deformation applied to the terrain model (bottom)

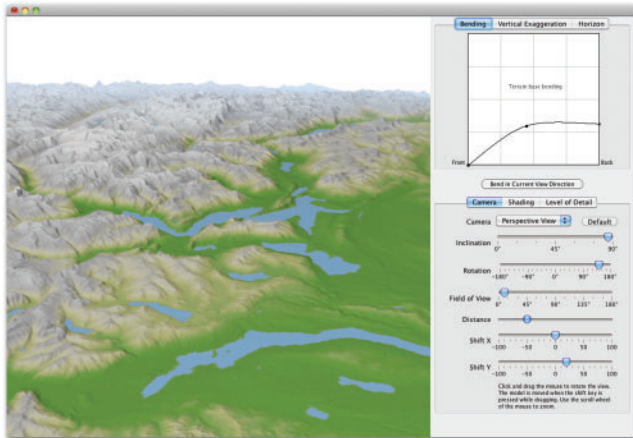


Figure 15. Screenshot of the Terrain Bender interface

be exported for rendering with a ray tracer, offering more sophisticated cylindrical rendering.

TERRAIN BENDER SOFTWARE

In this section, we briefly introduce Terrain Bender, the software that was developed by the authors, implementing the algorithms described above to create progressive views. Terrain Bender was written in Java and runs on Windows as well as on Mac OS X platforms (Figure 15). For rendering previews of the terrain model, Terrain Bender builds on JOGL, a Java implementation of OpenGL (JOGL, 2009). OpenGL is a standard specification, defining an application programming interface for generating 2D and 3D computer graphics (OpenGL, 2008). Terrain Bender is free open-source software, and can be downloaded from <http://www.terraincartography.com/terrainbender>.

Terrain Bender offers a user-friendly interface, allowing the cartographer to create progressive views from imported terrain models as intuitively as possible. Controlling the degree of bending, by dragging and dropping points on the curve diagrams, and inspecting on-the-fly generated previews of the bent terrain, greatly facilitates this task. The user can choose to preview the bent terrain in central perspective projection, parallel projection on inclined image plane, and cylindrical projection.

For previewing the terrain model in cylindrical projection, a workaround had to be implemented, since standard graphical rendering pipelines like OpenGL cannot readily generate images in cylindrical projection. Our software mimics a cylindrical panorama by creating eight images in central perspective projection, and stitching them together to a single 360° panorama strip. This workaround allows us to take advantage of the faster rendering speed of the OpenGL pipeline. The eight image tiles connect seamlessly, but some distortions appear towards the image tile borders.

CONCLUSIONS

Our terrain bending algorithm allows the user to intuitively create and manipulate a progressive view for a digital terrain

model within a compact workflow. The cartographer can immediately assess and adapt the bending of the terrain. The results are rendered as a 3D view in real time. Trying out different parameter settings in order to show a landscape in an optimal way can therefore be done very quickly. The algorithm for terrain bending is easy to implement and performs quite fast. For very large terrain models, Terrain Bender uses a downsampled version to compute a preview, but in the future, a GPU-accelerated version would be welcome, so that large terrain models can be previewed in their original resolution without too much delay.

Our software is a specialized application for cartographic 3D mapping, permitting the user to import and export geo-referenced 2.5D terrain models. While the altitude values are altered to create a progressive projection, the x - y coordinates remain geo-referenced, making it possible to combine the deformed model with textures or vector layers, using a GIS or standard rendering engine.

A curved horizon and vertical exaggeration can be added to the terrain. In small-scale 3D maps, this increases the impression of spatial depth and allows the cartographer to give certain landscape elements, e.g. high mountain ranges, the necessary weight. The progressive-cylindrical projection is a new type of projection that unwraps a 360° panoramic view as a 3D strip map, while still offering the advantages and design options of a progressive projection.

Our software builds on the manual techniques of the panorama painters. Yet with our digital solution, the cartographer is no longer burdened with the details of complicated computations, and can focus on the design of the 3D map projection. The Terrain Bender software is freely available for download at <http://www.terraincartography.com/terrainbender>.

BIOGRAPHICAL NOTES



Helen M. Jenny is a PhD student and a member of the research staff at the Institute of Cartography, ETH Zurich, Switzerland. She received a MS in Physical Geography and GIS from the University of Stuttgart, Germany. Her current research interests are 3D map design, distortions and painterly rendering of digital landscape panoramas, web mapping and map portals.

ACKNOWLEDGEMENTS

We thank H. R. Bär, ETH Zurich, for sharing his framework for 3D rendering of terrain models; Julia Everwin, Stuttgart, for her help with some mathematics;

Jerry Huxtable for sharing his code for interactively designing spline curves; Tom Patterson, US National Park Service, for evaluating Terrain Bender; Marino Maggetti, University of Fribourg, Switzerland, for providing Figure 1; and Matthias Troyer, ETH Zurich, for providing Figure 3. We would also like to thank Ramamurthy Suganthan, Gregg Verutes, Hugh Stimson, Boris Stern and the other software testers for their feedback.

REFERENCES

- Berann, H. C. and Neugebauer, G. (1987). 'Die Kartenprobe "Brenta" in vogelschaubildlicher Darstellung', in **Brenta-Monographie: Grundlagenforschung auf dem Gebiet der Hochgebirgskartographie**, ed. by Neugebauer, G., pp. 173–182, Universität der Bundeswehr München, Neubiberg.
- Degener, P. and Klein, R. (2009). 'A variational approach for automatic generation of panoramic maps', **ACM Transactions on Graphics**, 28, pp. 1–14.
- Falk, M., Schafhitzel, T., Weiskopf, D. and Ertl, T. (2007). 'Panorama maps with non-linear ray tracing', in **5th International Conference on Computer Graphics and Interactive Techniques in Australia and Southeast Asia (GRAPHITE 2007)**, pp. 9–16, Perth, WA, Australia, Dec 1–4.
- Hölzel F. (1963). 'Perspektivische Karten', in **International Yearbook of Cartography**, ed. by Imhof, E., pp. 100–118, Bertelsmann, Gütersloh.
- Imhof, E. (1963). 'Kartenverwandte Darstellungen der Erdoberfläche. Eine systematische Übersicht', in **International Yearbook of Cartography**, ed. by Imhof, E., pp. 54–99, Bertelsmann, Gütersloh.
- Jenny, B. (2004). 'Bringing traditional panorama projections from the painter's canvas to the digital realm', in **4th ICA Mountain Cartography Workshop**, pp. 151–157, Barcelona, Spain, Sep 30–Oct 2.
- JOGL. (2009). 'Java™ Binding for the OpenGL® API Wiki', <http://kenai.com/projects/jogl/pages/Home> (accessed 29 September 2009).
- Kern, H. (1986). 'Kreisingpanorama und progressive Perspektive', in **Karlsruher Geowissenschaftliche Schriften**, A4, ed. by Herrmann, C. and Kern, H., pp. 153–164, Fachhochschule Karlsruhe, Karlsruhe.
- Kettler, W. (1986). 'Das Panorama – ein unentbehrliches Informations- und Werbemittel für den Tourismus', in **Karlsruher Geowissenschaftliche Schriften**, A4, ed. by Herrmann, C. and Kern, H., pp. 105–112, Fachhochschule Karlsruhe, Karlsruhe [see also for Kettler's manual progressive perspective construction grid: Winfried Kettler Panorama-Studio: 'Das Panorama – ein unentbehrliches Werbemittel für den Tourismus', <http://www.kettler-panorama.ch/panorama.htm> (accessed 29 September, 2009)].
- Levene, J. (1998). 'A framework for non-realistic projections', MSc thesis, Massachusetts Institute of Technology, Cambridge, MA, USA.
- Maggetti, M. (2000). 'Leben und Werk des Vogelschaubilder-Malers Max Bieder (1906–1994)', **Cartographica Helvetica**, 22, pp. 11–18.
- OpenGL.org. (2008). 'OpenGL – the industry's foundation for high performance graphics', <http://www.opengl.org> (accessed 29 September 2009).
- Patterson, T. (2000). 'A view from on high: Heinrich Berann's panoramas and landscape visualization techniques for the U.S. National Park Service', **Cartographic Perspectives**, 36, pp. 38–65.
- Patterson, T. (2001). 'DEM manipulation and 3-D terrain visualization: techniques used by the U.S. National Park Service', **Cartographica**, 38, pp. 89–102.
- Sederberg, T. W. and Parry, S. R. (1986). 'Free-form deformation of solid geometric models', **SIGGRAPH Computer Graphics**, 20, pp. 151–160.
- Sumner, R. W., Schmid, J. and Pauly, M. (2007). 'Embedded deformation for shape manipulation', **ACM Transactions on Graphics**, 26, article 80.
- Tait, A. (2008) 'Mountain ski maps of North America – a preliminary survey and analysis of style', in **6th ICA Mountain Cartography Workshop**, pp. 219–225, Lenk, Switzerland, Feb 11–15.
- von Funck, W., Theisel, H. and Seidel, H.-P. (2006). 'Vector field based shape deformation', **ACM Transactions on Graphics**, 25, pp. 1118–1125.
- Yang, Y., Chen, J. and Beheshti, M. (2005). 'Nonlinear perspective projections and magic lenses: 3D view deformation', **Computer Graphics and Applications**, 25, pp. 76–84.
- Yu, J. and McMillan, L. (2004). 'General linear cameras', in **8th European Conference on Computer Vision**, pp. 14–27, Prague, Czech, May 11–14.