Creating Panoramic Strip Format Maps Using Interactive Terrain Deformation

Helen JENNY, Bernhard JENNY, William CARTWRIGHT and Lorenz HURNI

The GI_Forum Program Committee accepted this paper as reviewed full paper.

Abstract

While digital panoramic photography taken along travelling routes is very popular, digital panoramic strip maps are virtually nonexistent. In this article, we explore the characteristics of panoramic strip maps and provide an interactive method for their digital creation, focusing on the challenge of straightening curved routes. The resulting software Terrain Bender implements a geometric deformation method for 2.5D digital elevation models based on moving least squares minimization. To demonstrate our method, we create a digital panoramic strip map.

1 Introduction

Panoramic maps in strip format are a rare form of cartographic route representation. They show panoramic depictions of the landscape along a linear, often straightened path. Although they have found virtually no entry into cartographic literature, panoramic strip maps have been produced by landscape painters throughout recorded history. For example, the painting from 12th century China in figure 1 (left) showing urban panoramic views along a river and the leporello (folding map) of Moselle river from about 1900 created for steamboat tourism (see figure 1 right) can be considered early examples of panoramic strip maps. These examples though are often unique specimens that are difficult to classify clearly as maps in the traditional sense.

With the advent of railway travel, hand-painted panoramic strip maps of train routes were published (see figure 2). These railway panoramas were created by different authors of train routes in different countries, often commissioned by railway authorities. They share common cartographic construction techniques and characteristics, including oblique mountain representations and the design of travel routes as central straight lines (see figure 2).

Considering the time and effort needed to paint a single cartographic panorama, it does not astonish that hand-painted panoramic strips corresponding to sequences of panoramas are rarely encountered today. However, the complete absence of digitally created panoramic strip maps in contrast to the abundance of panoramic photography on the Internet is striking. The public obviously enjoys representing excerpts of travelling routes in strip format. The limitations of panoramic leisure photography though are clear: it is usually restricted to very large scales, limited to photorealism, dependent on real-world conditions (e.g. vantage points and lighting), and requires physical presence. The panoramic photo-

Car, A., Griesebner, G. & Strobl, J. (Eds.) (2011): Geospatial Crossroads @ GI_Forum '11.
© Herbert Wichmann Verlag, VDE VERLAG GMBH, Berlin/Offenbach. ISBN 978-3-87907-509-6.

graph can serve as a piecewise document of a travelled route, but does not provide a continuous, generalized regional panoramic overview that could be used for orientation while travelling. We assume that limited awareness among cartographers of the panoramic strip map as well as lack of specialized 3D software to create such maps are responsible for their absence in computer-generated form.

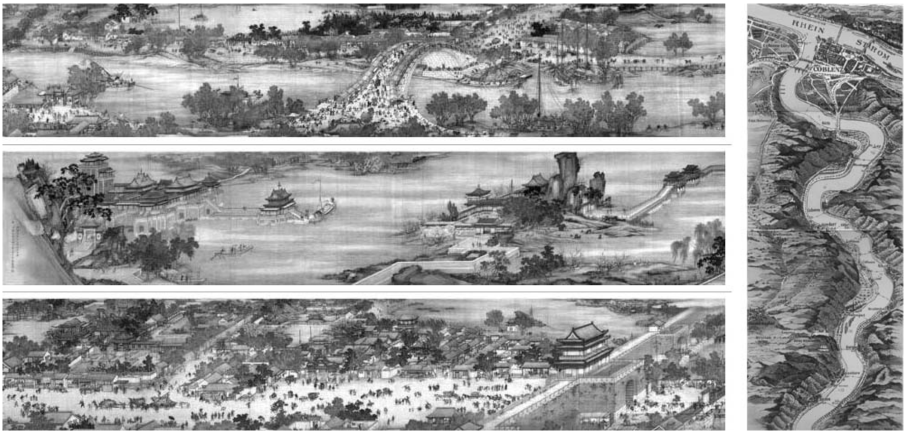

Fig. 1: 'Along the River During Qingming Festival' by Chinese painter Zhang Zeduan, 12th century (left); leporello of the Moselle river, Bremer (publisher), author unknown, about 1900, height 160 cm (right)

This article explores the characteristics of panoramic strip maps taking panoramic railway route maps as a model. Specific to this map type is the challenge to represent curved routes as approximately straight lines. We review the possible approaches to digitally create distorted panoramic landscape representations. An algorithm based on moving least squares minimization is presented, which allows the cartographer to interactively deform a 2.5D digital elevation model in the process of creating panoramic strip maps. We also give a pseudo code version of the algorithm in Appendix A to facilitate reproduction by the reader. To demonstrate our approach, we create a digital panoramic strip map using Terrain Bender, a user-friendly software implementation of our algorithm. Our digital method takes only a fraction of the time necessary to manually paint a strip panorama and does not require any artistic talent.

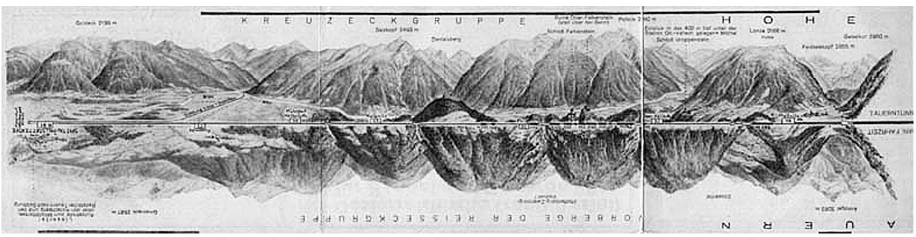

Fig. 2: Leporello of the Tauernbahn, Österreichisches Verkehrsbureau [Railway of the Tauern Range, Austrian Transportation Department], 1933, length 90 cm

2 Characteristics of the Panoramic Strip Format Map

Taking railway route panoramas as model, we observe that the panoramic strip map consists of three components: an (approximately) linear center feature representing a schematic route with stops and two panorama strips showing the immediate landscape along the route.

The viewing directions of the panorama strips are perpendicular to the tracks. One panorama is mirrored around the travelling direction, so that the traveller can follow the route along the stops and can identify the landscape seen through the left and right windows by tracing perpendicular from a point on the tracks. If one rotates the map in such a way that the mountain tops point up, the travelling direction is from left to right in one panorama strip and from right to left in the other (see figure 3), an unfamiliar task for the Latin alphabet reader. The panoramic strip map can thus serve in both travel directions.

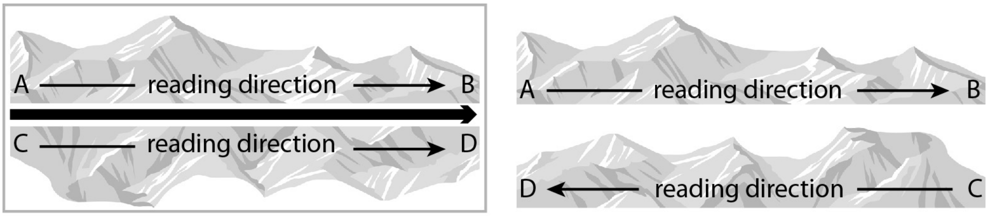

Fig. 3: Assuming the travel direction represented by the thick arrow, the top panorama is read left to right and the bottom panorama right to left

Panoramic strip maps can be considered a special 3D type of strip map sharing a number of characteristics with 2D strip maps (MACEACHREN 1986; MACEACHREN & JOHNSON 1987). Like their 2D equivalent, they focus on a linear geographic feature (e.g. railway, river, road, coastal border) and may lack attention to cardinal directions. If the linear feature is not straight, it may be deformed and straightened out, giving the map an elongated shape. In contrast to routing applications, the strip map is user-centered: it shows only the immediate surroundings of the travelling route and thus cannot serve for route planning and selection. Though it was used in the past for navigation with additional symbols (e.g. Ogilby's strip maps see HÜTTERMANN 2002), the deformation of an otherwise curved route may be misleading. Like the 2D strip map, the panoramic strip map is well suited for sequential route following, similar to following a travel narrative where the route to be taken is clearly lined out. Like the panorama map, it can immerse the map reader in the landscape, giving a preview of what one may see when travelling along the depicted route. The panoramic strip map is therefore most suitable for information on landscape appearance before travel (e.g. touristic promotion) and for orientation during travel along a given route. It could also provide the basis for personalized documentation of a trip.

3 A Method for Digital Generation of Strip Format Panoramas

3.1 Approaches for deforming panoramic strip maps

To our knowledge, no specific digital method to create panoramic strip maps has so far been suggested. As the rendering of panoramic maps in general can be achieved using standard 3D modeling and rendering software, we focus on terrain deformation necessary to straighten curved routes for panoramic strip map creation. Two major approaches to achieve such deformed panoramic representations can be identified: camera parameter modification for rendering and surface deformation before rendering.

Camera parameter modification is used by computational photography working with images as well as by computer graphics working with models. Methods from computational photography assemble a multiperspective photographic panorama by taking pictures or filming along a curved route (ROMAN et al. 2004 and 2006; ZHENG 2003). Methods aim at minimizing distortion by adapting camera parameters, ray sampling and image mapping procedures when reassembling the final panoramic image. Methods from computer graphics working with 3D models modify ray geometry (FALK et al. 2007) to reduce local occlusions in panoramas and combine different projections (MÖSER et al. 2008) of a model in a single image.

Surface deformation is an alternative when creating distorted panoramic views. The geometry is first deformed and then rendered in standard perspective or parallel projection. While surface deformation has been intensively researched by computer graphics, to our knowledge, it has not been specifically applied to strip panoramas. To interactively deform an elevation model for panoramic strip map creation, we build on a method, which we originally developed to solve conflicts in panorama design (JENNY et al. in print). It is based on an algorithm by SCHAEFER et al. (2006) who used moving least squares minimization for 2D image deformation. In contrast to interactive camera modification, control of interactive surface deformation with our software is mastered easily as the cartographer does not need to acquire in-depth concepts of scene rendering.

3.2 Interactive route straightening using moving least squares minimization

With our Terrain Bender software, the cartographer can import terrain models and deform them interactively. Our deformation algorithm is especially geared at regularly spaced 2.5D elevation models, where for every grid cell only one elevation value is available. Representation of certain landforms requiring two or more altitude values per grid cell (e.g. overhanging cliffs or arcs) is not possible with this data structure; yet, 2.5D elevation models are standardly used by cartography, GIS and surveying institutions as they are easy to process and can be combined with other raster datasets.

Terrain Bender was written in Java and uses JOGL, a Java implementation of OpenGL to render a 3D preview of an imported digital elevation model. Terrain Bender runs on Windows and Max OS X platforms. The user can add deformation handles in the form of control points by clicking on the rendered terrain. By dragging the control points to a new position, the model can be deformed. Based on the displacement of the control points, a displacement vector is computed for every grid cell in the elevation model and the deformed model is rendered. The number of elevation values in the model does not change.

For interactive display, the user can select a downsampled elevation grid. As we only store the displacement of the control points, it is easy to recompute the deformation for different model resolutions.

To straighten a curved route usually following a valley, we only need to deform the model in the horizontal plane. To compute a horizontal displacement vector for each grid cell, we use a moving least squares minimization approach based on SCHAEFER et al. (2006). We solve for the optimal warping function for all elevation model points that minimizes the weighted sum of squared distances between the control points' mapped and dragged positions. As the weights depend on the distance between control point and elevation grid point, the mapping function is different at every grid point and small distances have larger impact. For a mathematical formulation of the algorithm, please refer to JENNY et al. (in print). An application oriented pseudo code version is given in Appendix A. After deformation, the elevation model has still the same number of grid cells, but the cells' geometry is not necessarily square anymore. Extreme distortion can lead to degenerate geometry, but usually this is easily spotted in the 3D preview.

The deformed model is rendered in Terrain Bender with parallel oblique projection. In parallel oblique projection, the projector rays are parallel, but other than perpendicular to the plane of projection. Equally large objects are displayed at the same size at different distances. We chose this projection because we consider it adequate to represent the center of view of a moving person and best for a panoramic strip format.

4 An Example of a Digital Railway Strip Panorama

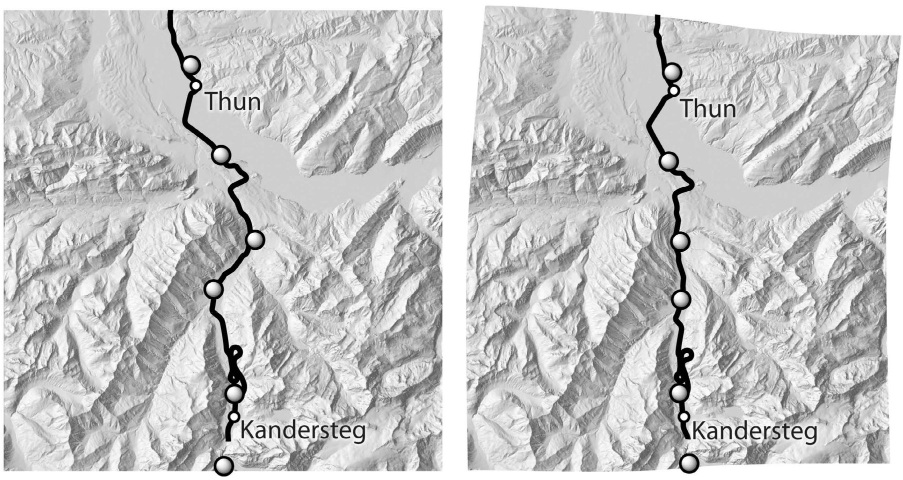

Fig. 4: Orthogonal view of the test region: undeformed (left), deformed (right) using our algorithm with 6 control points (grey spheres)

To demonstrate our method, we created a panoramic strip map along the railway route from Thun to Kandersteg (Switzerland). In figure 4 (left) the curved shape of the original railway path is marked. We used our deformation algorithm to interactively distort an elevation model of the region so that the railway route formed an approximate straight segment (see figure 4 right). We then rendered panoramic views perpendicular to the travel direction in parallel oblique projection and assembled them into a strip panorama, adding a schematic central piece representing the railway tracks (see figure 5).

5 Conclusion and Outlook

To this date, panoramic maps in strip format have not received any attention by digital cartography. We hope to promote the production and circulation of such maps with our method for interactive creation of panoramic strip maps. As we identified straightening curved routes as one of the major challenges in the creational process, we focused in this article on the interactive terrain deformation step. Digital panoramic strip maps could become valuable for touristic promotion of famous hiking trails (e.g. Camino de Santiago) or popular public transportation overland routes. For the hobby cartographer, they could also serve as digital basis for personalized or collective travel documentation.

Extreme terrain deformation can produce degenerate geometry using our method. Control lines or curves instead of control points may yield better results. While improving our method in this regard, it would be also interesting to explore if extreme deformations (e.g. circle to straight line) are reasonable for panoramic strip map creation or if other specialized representation forms may be better suited. Future work could also focus on vertical terrain levelling for panoramic strip maps. The few exemplars of hand-drawn panoramic strip maps available to us do not show a mountain profile when the terrain rises; instead, the landscape base is level. Levelling the terrain perpendicular to the viewing direction may be an interesting extension to our method. Also, a fully automatic approach to route straightening should be considered. However, the interactive version has the advantage of visually controlling distortion.

It would also be interesting to find out how the deformation affects the viewer's understanding of the depicted landscape. To our knowledge, there are no studies focusing on the perception of distorted landscapes. D'ANGELO et al. (2010) suggest a method for assessing the perceptual quality of geometrically distorted images, which may provide some input for a perceptual analysis of landscape deformation.

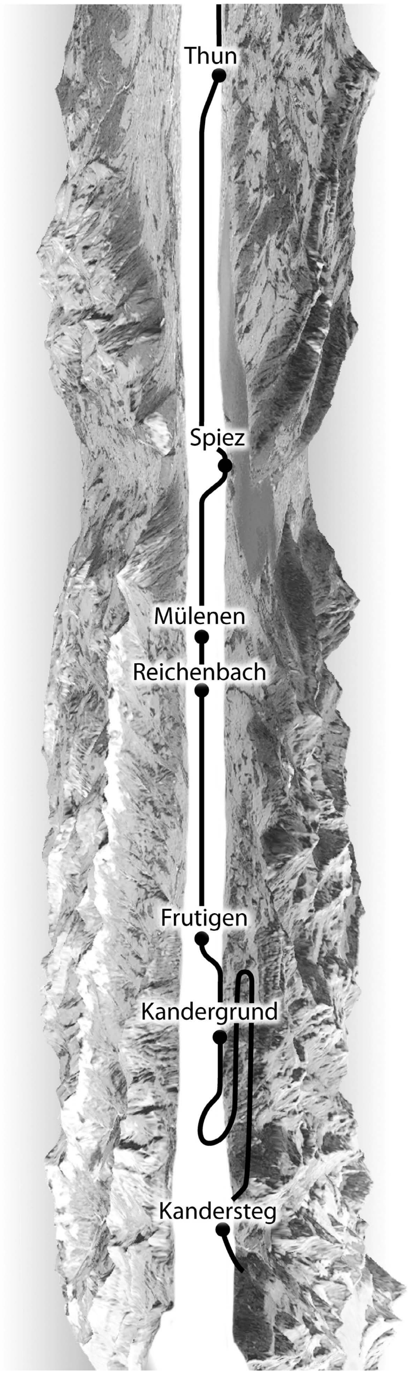

Fig. 5: Digital panoramic strip map of the railway route from Thun to Kandersteg created with our algorithm and textured with Landsat images.

6 Appendix A – Pseudo Code

To facilitate reproduction of our algorithm based on SCHAEFER et al. 2006, we provide a pseudo code version of the deformation algorithm. We use a vertex shading approach where we pass the positions of the elevation model grid cells as vertices to the vertex processor. The vertex processor is a programmable unit that operates on incoming vertex values, which define the geometry of the rendered object. The vertex processor usually performs vertex transformation (i.e. the vertex geometry is manipulated), normal transformation (the normal is used for subsequent shading), texture coordinate manipulation, and lighting and colour application (ROST et al. 2010). For the purpose of terrain deformation, we focus on the manipulation of the vertex geometry.

Vertex shaders are programs that run on the vertex processor. A vertex shader can only access the geometry of a single vertex, i.e. the other vertices defining the rest of the object geometry are not simultaneously accessible. This allows for the concurrent processing of multiple vertices, and is not a limitation for our application. The vertex shader for the 2D deformation of terrain models using the moving least squares method receives a single undeformed vertex, a set of control points in the undeformed terrain *p*, as well as a set of corresponding control points defining the deformed positions *q*. In our implementation, the main function of the vertex shader calls the *warp* method, passing the undeformed x/y position of the vertex:

```
function warp(vertex)
        [pCentroid, qCentroid] := calculateCentroids(vertex)
        m := calculateM(vertex, pCentroid, qCentroid)
        displacedVertex:= qCentroid + m * (vertex - pCentroid)
        return displacedVertex
```

The warp method first computes the weighted centroids *pCentroid* and *qCentroid* of the two point sets *p* and *q* by calling *calculateCentroids*. In a second step a rigid transformation matrix *m* is computed by calling the method *calculate*, which could be replaced to use a different affine transformation.

The method *calculateCentroids* uses the function *weight* to compute a distance dependent weight between the vertex position and each control point in *p*.

```
function calculateCentroids(vertex)
        pCentroid := 0
        qCentroid := 0
        weightSum := 0
        for each control point p and q
                w := weight(vertex, p)
                weightSum:= weightSum + w
                pCentroid:= pCentroid + w * p
                qCentroid:= qCentroid + w * q
        return pCentroid / weightSum and qCentroid / weightSum
```

```
function weight (vertex, control_point)
        l := length of vector between vertex and control_point
        return 1 / (l * l)
```

The matrix *m* for a rigid transformation is finally computed. For the two-dimensional case *m* is a square 2 × 2 matrix:

```
function calculateM(vertex, pCentroid, qCentroid) {
        m11 := 0
        m12 := 0
        for each control point p and q
                w := weight(vertex, p)
                dp := vector from pCentroid to p
                dq := vector from qCentroid to q
                m11 := m11 + w * (dp.x * dq.x + dp.y * dq.y)
                m12 := m12 + w * (dp.y * dq.x - dp.x * dq.y)
        mu := sqrt(m11 * m11 + m12 * m12)
        m11 := m11 / mu
        m12 := m12 / mu
        elements of matrix m := m11, m12, -m12, m11
        return m
```

References

D'ANGELO, A., ZHAOPING, L. & BANRI, M. (2010), A full-reference quality metric for geometrically distorted images. IEEE Transactions on Image Processing, 19 (4): 867-881.

FALK M., SCHAFHITZEL, T., WEISKOPF, D. & ERTL, T. (2007), Panorama maps with non-linear ray tracing. 5th international Conference on Computer Graphics and Interactive Techniques in Australia and Southeast Asia, pp. 9-16.

HÜTTERMANN, A. (2002), Tobias Mayer und seine Reisekarte von 1751. Cartographica Helvetica, (25-26): 13-22.

JENNY, H., JENNY, B., CARTWRIGHT, W .E. & HURNI, L. (in print), Interactive local terrain deformation inspired by hand-painted panoramas. The Cartographic Journal.

MACEACHREN, A. M. (1986), A linear view of the world: strip maps as a unique form of cartographic representation. The American Cartographer, (13) 1: 7-25.

MACEACHREN, A. M. & JOHNSON, G. B. (1987), The evolution, application and implications of strip format travel maps. The Cartographic Journal, (24): 147-158.

MÖSER, S., DEGENER, P., WAHL, R. & KLEIN, R. (2008), Context aware terrain visualization for wayfinding and navigation. Computer Graphics Forum, (27) 7: 1853-1860.

ROMAN, A., GARG, G. & LEVOY, M. (2004), Interactive design of multi-perspective images for visualizing urban landscapes. Proc. of the Conference on Visualization '04, IEEE Computer Society: 537-544.
ROMAN, A. & LENSCH, H. P. A. (2006), Automatic multiperspective images. Eurographics Symposium on Rendering, pp. 161-171.
ROST, R. J., LICEA-KANE, B., GINSBURG, D., KESSENICH, J. M., LICHTENBELT, B., MALAN, H. & WEIBLEN, M. (2010), OpenGL Shading Language (3rd edition). Boston: Addison-Wesley.
SCHAEFER, S., MCPHAIL, T. & WARREN, J. (2006), Image deformation using moving least squares. ACM Transactions on Graphics, (25) 3: 533-540.
ZHENG, J. Y. (2003), Digital route panorama, IEEE Multimedia, (10) 3: 57-68.