

Plan oblique relief for web maps

Bernhard Jenny^{a*}, Jonas Buddeberg^{a,b}, Charlotte Hoarau^{c,d} and Johannes Liem^a

^aCollege of Earth, Ocean, and Atmospheric Sciences, Oregon State University, Corvallis, OR, USA; ^bInstitut für Geographie, University of Potsdam, Potsdam, Germany; ^cIGN, COGIT, Saint-Mandé, France; ^dUniversité Paris-Est, Paris, France

(Received 4 September 2014; accepted 19 December 2014)

Plan oblique relief shows terrain with a side view on a two-dimensional map, resulting in visualizations where the third dimension of the terrain is more explicit than on traditional two-dimensional maps. Existing plan oblique maps are static: the angle of terrain inclination is not adjustable and the orientation of plan oblique inclination does not change with the orientation of the map. This article introduces two complementary methods that address these issues by using the 3D graphics pipeline to render plan oblique relief for tile-based web maps. The goal is to allow users to adjust the terrain inclination and map rotation angles to better visualize the third dimension of the terrain. The first method pre-renders plan oblique tiles with a server-side application. The tiles are visualized with a standard web mapping framework. The second method renders plan oblique relief on-the-fly in a web browser using WebGL and a customized version of OpenLayers 3, which enables users to select arbitrary terrain inclination and map rotation angles. The second method uses a tiled digital terrain model that is loaded by the web browser. The browser applies the plan oblique transformation, computes a shaded relief, and texturizes the terrain with tiled map layers.

Keywords: plan oblique relief; 3D map; relief map; terrain map; WebGL; OpenLayers 3

Introduction

For many map users, it is easier to interpret landforms in three-dimensional panorama maps, such as those commonly used for visualizing mountain ski resorts, than in traditional two-dimensional orthographic maps. Three-dimensional maps are easier to read because the terrain is depicted with a side view, similar to how people see it in their everyday lives, instead of a top-down view.

Plan oblique relief is an intermediate style of map that aims to combine the advantages of both traditional two-dimensional orthographic maps and three-dimensional maps: the ground plane is shown in correct size and relief is shown with a side view. According to Jenny and Patterson (2007), plan oblique relief has several advantages over traditional orthographic maps. Plan oblique relief is more appealing and visualizes mountains with a familiar side view, distances between features at the same altitude are preserved and can be measured, and altitude differences can be gauged by the map reader. The disadvantages of plan oblique relief are that steep back slopes of mountains can be shortened or occluded and static plan oblique maps are not rotatable because mountain peaks do not point in the upward direction on a rotated map (Jenny and Patterson 2007).

Similar to an orthographic two-dimensional map, plan oblique relief uses a parallel projection where projecting rays, represented by arrows in Figure 1, are parallel. The image planes for orthographic (Figure 1(a)) and plan

oblique maps are both parallel to the ground plane; however, the image plane is shifted relative to the ground plane for a plan oblique map (Figure 1(b)). A parallel image plane results in a map where the distance between two points at the same altitude is rendered without distortion. This allows for measurements of distances between any two locations at the same altitude. Not all maps with parallel projections qualify as plan oblique. If the image plane is oblique relative to the ground plane, the parallel oblique projection compresses the ground plane space along one axis (Figure 1(c), see Appendix 1 for discussion).

Cartographers have not yet fully harnessed the rapid speed at which plan oblique relief can be rendered. We take advantage of modern graphics hardware to give users the option to explore terrain with alternative plan oblique inclination angles and to adjust the orientation of plan oblique maps. The goal is to allow users to adjust the terrain inclination angle and the map orientation angle to better visualize the third dimension of the terrain.

The following section discusses past manual and current digital construction techniques for rendering plan oblique terrain. The third and fourth sections introduce two complementary methods for creating plan oblique maps for the web. Both methods use the graphics pipelines for rendering plan oblique web tiles. The first method uses server-side rendering. The second method uses WebGL for client-side rendering, enabling users to select arbitrary terrain inclination and map rotation angles. The two

*Corresponding author. Email: jennyb@geo.oregonstate.edu

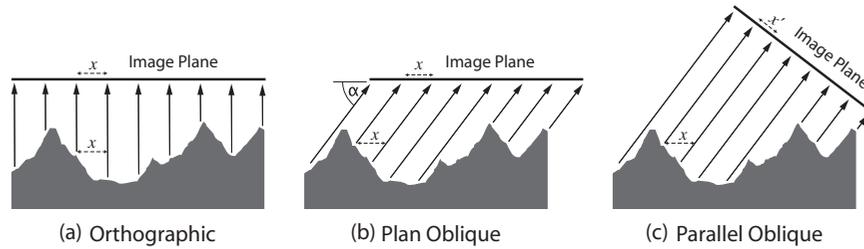


Figure 1. Projecting rays and image planes for (a) traditional orthographic maps, (b) plan oblique relief maps, and (c) parallel oblique maps. For orthographic and plan oblique maps, the distance x between two points at the same altitude is the same on the image plane as the ground plane. For parallel oblique maps, the distance x is compressed to x' on the image plane.

methods are critically analyzed in the Discussion section. Areas where further research is necessary are identified in the Conclusion. Appendix 1 points out the difference between plan oblique relief and general parallel oblique relief, and Appendix 2 discusses historical techniques for the manual construction of terrain visualizations with parallel projections.

Manual and digital creation of plan oblique relief maps

Plan oblique relief is not a new invention. Examples of manually produced plan oblique maps include Erwin Raisz' physiographic maps (Raisz 1931, 1956, 1962), Heinrich Berann's ocean floor maps (e.g., Berann 1967; Heezen et al. 1977; for a discussion of these maps, see Lawrence 1999), and Otto Betschmann's tourist maps (Figure 2; Betschmann 1937; for a discussion of these maps, see Jenny 2008). A variety of names have been used for this projection technique such as 3D planimetric relief, high oblique relief, oblique orthographic shaded relief, landform map, morphographic map, physiographic map, military projection, and proportional relief landform map (Jenny and Patterson 2007). This style of



Figure 2. Manually produced plan oblique map. Section of a state proof of *Reliefkarte Zentralschweiz – Berner Oberland* by Otto Betschmann (1937).

cartographic visualization is now commonly referred to as “plan oblique relief,” a term introduced by Jenny and Patterson (2007).

Manual construction of plan oblique maps is time consuming and arduous. Various techniques and instruments for constructing plan oblique relief maps have been developed to accelerate this process. When constructed manually, a conceptual ground plane is used to displace or “elevate” (Lobeck 1924) map features, such as contour lines, along the vertical axis of the map coordinate system. The distance of displacement is proportional to the vertical distance between the feature and the ground plane (Ridd 1963). Despite numerous attempts at simplifying and accelerating the construction of block diagrams and maps with various axonometric projections, the design and production of this style of visualization remained difficult and required considerable artistic training and geographic knowledge (see Appendix 2 for an overview of construction devices). Visiting any map library and randomly selecting a number of topographic maps produced with analogue tools would verify that plan oblique relief was only done occasionally during the era of manual cartography (Monmonier 1978).

A variety of terrain visualization software packages have been developed that support parallel projections to create axonometric and isometric surface visualizations. Cerny and Rudnicki (1972) provide an overview of early surface plotting software. An early example is the SYMVU software, initiated at the University of Michigan and then further developed and made available in 1969 at the Harvard Laboratory of Computer Graphics (Chrisman 2006). Many systems followed, leading to current advanced 3D rendering applications such as Cinema 4D or Blender. A noteworthy specialized terrain renderer is Natural Scene Designer, which makes it simple to create maps with plan oblique relief (Natural Graphics 2014). These applications allow for flexibility in configuring parallel projections and creating diverse rendering styles. Cartographers apply these tools to create plan oblique and axonometric terrain maps (Patterson 2008). Figure 3 is an example of a digitally produced map with plan oblique projection (Patterson 2014a, 2014b).

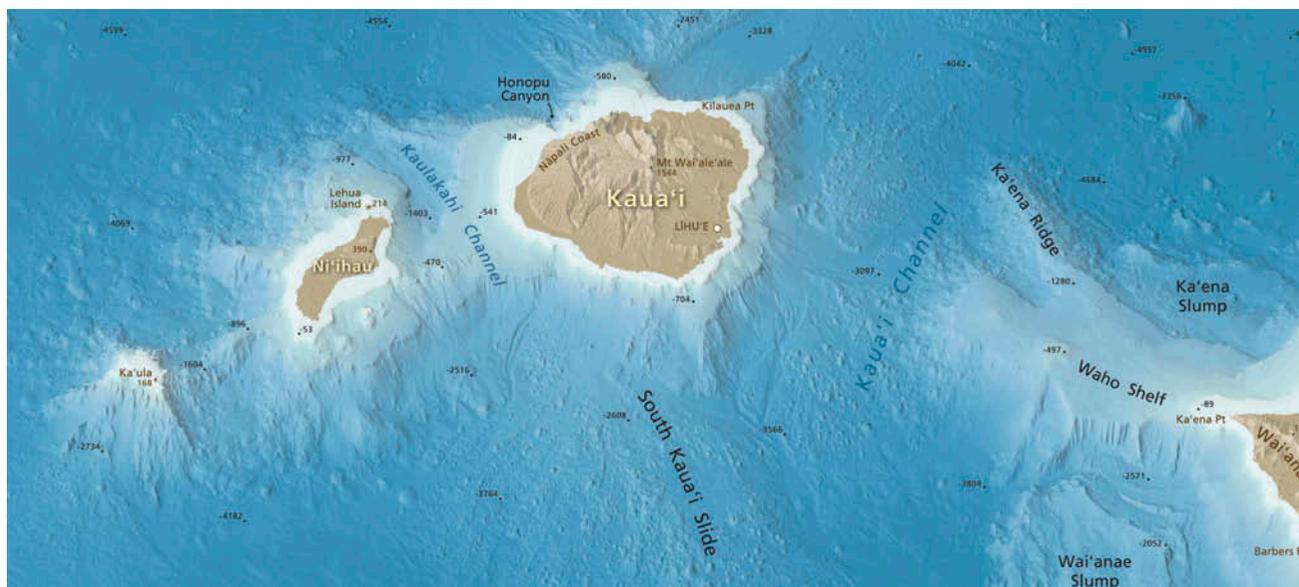


Figure 3. Digital map with plan oblique relief. Scale is identical in all directions for points at the same elevation. Section of *Seafloor Map of Hawai'i*, courtesy of T. Patterson (2014a, 2014b).

There are two primary methods for rendering three-dimensional objects: ray tracing and the graphics pipeline. Ray tracing, where the path of light is traced through pixels of the image, can simulate a variety of optical effects including reflection and refraction. Professional high-end rendering software uses ray tracing for the generation of still images and animated movies with realistic illumination effects. Current software for producing plan oblique relief, such as Natural Scene Designer, uses ray tracing (Casebolt, personal communication, July 2014). However, ray tracing is computationally expensive and considerably slower compared to the graphics pipeline, the other primary rendering method. With the graphics pipeline, a three-dimensional model is transformed to image space through geometric operations applied to each model vertex. The two most common graphics pipelines today are the OpenGL and Direct3D application programming interfaces (APIs). The cross-platform OpenGL was originally developed by Silicon Graphics, Inc. (Segal and Akeley 1992) and is now maintained by the non-profit technology consortium Khronos Group (www.khronos.org). Direct3D is a similar 3D rendering API by Microsoft for Windows and Xbox. Video games and other applications with interactive or animated three-dimensional geometry use the graphics pipeline through these APIs to access graphics processing units (GPUs) for hardware-accelerated graphics. Terrain rendering is an important component in many applications, and a plethora of methods and algorithms for the effective treatment of terrain data have been developed in the past.

This article discusses two complementary methods for plan oblique terrain rendering for web maps using the

graphics pipeline. The fast rendering speed of the hardware-accelerated graphics pipeline offers new possibilities for the adjustment of the rendering parameters in real time. The first method for creating plan oblique web maps uses conventional tile-based web frameworks, such as Leaflet (<http://leafletjs.com/>), for displaying zoomable and “pannable” web maps. The parallel projection is applied to each tile of the map in order to create a new set of tiles where each tile has plan oblique geometry. This method uses server-side rendering, meaning the plan oblique projection is applied by an OpenGL-based software application running on a server.

The second method also uses tiles, but the geometry is not stored in a plan oblique projection. Instead, tiles store orthographic elevation values that are used to create a plan oblique map on the web browser. This allows the user to rotate the plan oblique map and adjust the terrain inclination angle (α in Figure 1) in real time.

Server-side rendering of plan oblique map tiles

Server-side rendering adds a plan oblique projection to two-dimensional web maps using two input web tile sets. The first tile set contains a tiled digital terrain model. The digital terrain model is tiled according to the standard web tile schema (OGC 2010; Batty et al. 2010; Peterson 2012) after it has been projected to the web Mercator coordinate system. We use a modified version of the GDAL2Tiles open source script for generating map tiles (Přidal 2008). GDAL2Tiles is a Python script based on the Geospatial Data Abstraction Library, an open source library for handling geospatial raster data

(Warmerdam 2008; GDAL 2014). We modified the script to support floating-point digital elevation models, and write elevation in Esri ASCII or binary grid formats. The second tile set consists of standard web map image tiles. The rendering software creates a plan oblique view from the terrain model tiles and drapes the web map image tiles on top of the terrain. The result is a new set of map tiles with plan oblique geometry and optional shaded relief. These tiles can be viewed with standard web mapping libraries such as Leaflet or OpenLayers.

The custom rendering application for creating map tiles with a plan oblique projection is based on the OpenGL graphics pipeline. First, we define an orthographic camera in the OpenGL library. We load each terrain tile and construct a triangle mesh from the elevation values. The triangle mesh is loaded onto the GPU and the graphics pipeline applies plan oblique projection to the vertices of the mesh. A custom vertex shader program that runs on the GPU is used for the projection. The vertex shader program alters the vertical y coordinate of the vertex relative to the altitude of the vertex. Vertices at higher elevations are shifted by a larger amount than pixels at lower elevations. The shifted vertical coordinate is computed with $y + k \times z$, where y is the original vertical coordinate, k is a constant derived from the selected inclination angle (α in Figure 1), and z is the altitude of the vertex (Willett et al. 2015). Shifting the horizontal x coordinate is also possible, but it is not practical for a north-oriented map because horizontally shifted mountain peaks would not point in the upward direction. The vertex shader program also computes a shading value using a standard illumination model.

A fragment shader program, which runs in tandem with the vertex shader program on the GPU, combines the projected and shaded pixels with an optional texture. The texture is also a standard web map tile set in the web Mercator coordinate system. Our server-side rendering software can load the texture tiles from a hard disk or retrieve them from a web server using the HTTP protocol. An overview of the server-side rendering pipeline is illustrated in Figure 4.

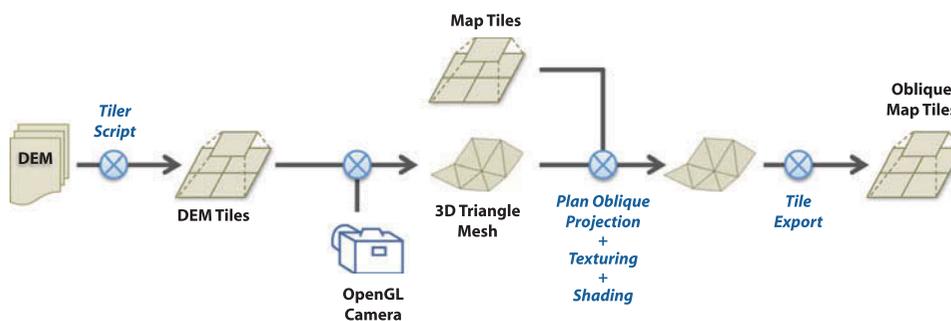


Figure 4. Per-tile server-side rendering pipeline.

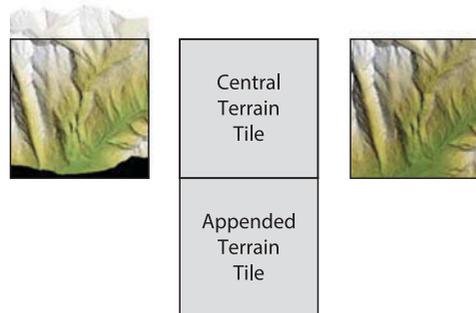


Figure 5. Plan oblique shifting creates an empty lower tile border and shifts terrain beyond the upper border (left). Attaching a second terrain tile to the lower border (center) fills the empty space when rendering the tile (right).

When each terrain tile is rendered individually, visual artifacts occur along the lower border of tiles (Figure 5). The dark irregular area along the lower border of the resulting tile is undefined because the plan oblique projection vertically shifted the vertices of the triangle mesh. The inverse issue happens along the upper border when vertices are shifted beyond the upper border of the tile and do not appear in the resulting image. Instead, these vertices should appear on the next upper tile to fill the lower void border of this neighboring tile.

To solve this problem, we first attach the immediate lower tile to the current tile, which doubles the size of the terrain tile. The enlarged terrain tile is converted to a triangle mesh, rendered with the graphics pipeline, and clipped to the extent of one tile. The resulting tile image does not have the black area along its lower border. A further complication arises when the shading algorithm, which is part of the vertex shader program, requires elevation values around a tile for computing terrain normal vectors for shading computations within the tile. The terrain tiles must be enlarged further before they are converted to a triangle mesh and loaded onto the GPU. To accomplish this, additional tile neighbors must be loaded and combined with the central terrain tile.



Figure 6. Section of a server-side generated plan oblique web map (Crinkled Watercolor Map available at <http://cartography.oregon-state.edu/CrinkledWatercolor.html>).

Figure 6 is a section of a plan oblique map created with the server-side method. We combined Stamen's Watercolor map tiles (Watson 2012) with a terrain model for the State of Oregon.

Client-side rendering of plan oblique maps with WebGL

Client-side rendering of plan oblique web maps offers more customization options and provides the user additional tools for interacting with the map. Rather than loading image tiles, the web browser loads a tiled terrain model. Similar to the server-side rendering method, the plan oblique projection is applied to each tile. However, with client-side rendering, the parameters for plan oblique projection can be adjusted by the user. Figure 7 is a section of a web map created using client-side rendering. When the user rotates the map, the orientation of plan oblique shifting is adjusted to the direction of rotation in order to orient inclined mountains in the upward direction. The user can define the amount of plan oblique inclination (angle α in Figure 1). Depending on the local steepness of the terrain, this angle can be increased (for flat terrain) or decreased (for mountainous terrain). Azimuthal and zenithal illumination directions for relief shading are adjustable. The tiled terrain model can be used to identify flat areas and colorize them with a blue tint to indicate bodies of water (Figure 7). Hypsometric tinting can be applied in the browser by mapping elevation values to a color ramp. The color ramp is adjustable by the user. All user-controlled changes are reflected in the map in real time.

WebGL, the browser version of the OpenGL graphics pipeline, is used for client-side rendering. WebGL (Web Graphics Library) is a JavaScript API for rendering 2D and 3D graphics based on OpenGL ES 2.0, a subset of OpenGL (Khronos Group 2014). The WebGL standard

was released in 2011 and is currently the only standardized 3D graphics pipeline that is integrated in all major desktop and mobile web browsers. The OpenLayers 3 (<http://openlayers.org/>) web mapping library integrates WebGL for rendering standard two-dimensional web maps using raster tiles. We modified the vertex shader and fragment shader programs used by OpenLayers 3 to add support for plan oblique terrain. Terrain data are transferred to the browser using two color channels of the PNG file format (Auer 2012). We use the red and blue RGB channels to store 16-bit elevation values. For each elevation tile, the vertex shader program assigns elevation values to a triangle mesh by decoding the 16-bit elevation values to a floating-point elevation. The vertex shader program applies the plan oblique projection to the triangle mesh by shifting vertex coordinates as for server-side rendering. Additionally, the shader program computes shading values, hypsometric tints, and identifies water bodies using the decoded elevation values.

Imagery can be used to texture the terrain model if desired (Figure 8). The imagery is stored in the blue and the alpha channels of the PNG image tiles. To create the PNG tiles with terrain and imagery data, server-side scripts execute a series of preprocessing steps. We first combine multiple terrain models in a GDAL virtual format (VRT) composition. Next, we use a modified version of the tiler-tools, a collection of Python scripts based on GDAL (available at <https://code.google.com/p/tilers-tools/>). Elevation values are encoded in the red and green channels of the PNG tile using the Python Image Library (PIL). The script also converts the optional RGB imagery from 24-bit to 16-bit encoding and writes the 16-bit color information to the blue and alpha channels of the PNG tile. The result is a PNG map tile that encodes both elevation and image data.

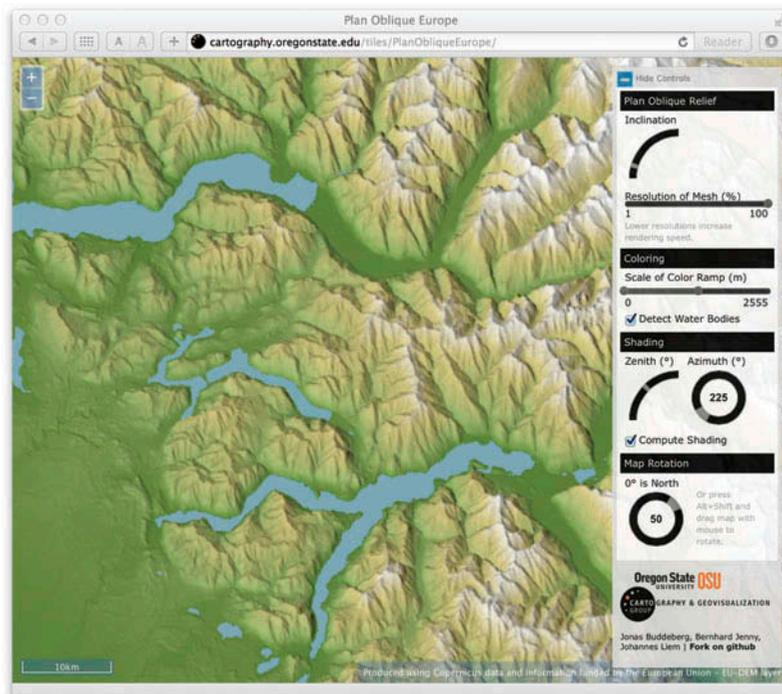


Figure 7. Plan oblique relief map rendered with WebGL. Plan oblique projection, shading, hypsometric tinting, detection of water bodies, and map rotation are applied in real time by the web browser (Plan Oblique Europe, available at <http://cartography.oregonstate.edu/tiles/PlanObliqueEurope/>).



Figure 8. Plan oblique map rendered in the web browser with relief shading and texturing. Terrain elevation data and imagery are stored in 4-band RGBA image tiles.

The plan oblique projection for rotated maps has to shift the terrain upward in the vertical direction (Figure 5). Because the vertical direction in a rotated map does not generally align with north, the terrain vertices are shifted along the tile's local x - and y -axes. We illustrate the effects of vertical shifting in Figure 9 by not rendering four tiles from the tile set. The borders of the rotated tiles in Figure 9 are no longer straight lines as in Figure 5. Instead, the plan oblique projection shifts terrain vertices along the map's vertical axis such that the tile borders

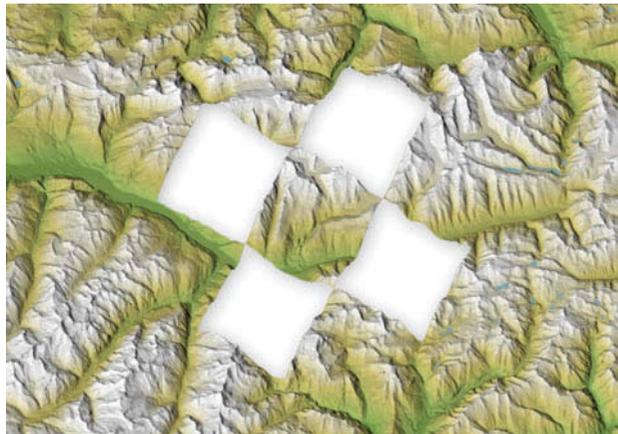


Figure 9. The upper and lower borders of tiles are moved vertically with plan oblique projection resulting in irregular tile borders. Four tiles are not rendered to illustrate the irregular tile borders.

form irregular lines. Unlike server-side rendering, combining multiple terrain tiles to fill the undefined border areas of a tile is not required for client-side rendering because terrain tiles are rendered one after the other into a common destination image. We shift vertices in both directions: the new horizontal coordinate is $x + k_x \times z$ and the vertical coordinate is $y + k_y \times z$, where x and y are the original coordinates, k_x and k_y are constants derived from the

selected inclination angle (α in Figure 1) and the current map rotation angle, and z is the altitude of the vertex.

Discussion

The presented server-side and client-side rendering of plan oblique relief maps both use the graphics pipeline's geometry transformation for projecting terrain. The server-side method is conceptually similar to filling a checkerboard with square tiles, whereas client-side rendering is similar to assembling a jigsaw puzzle with irregularly shaped pieces. The plan oblique web maps created with both methods can cover a large geographic extent and span multiple zoom levels.

With server-side rendering of plan oblique map tiles, standard web map libraries, such as Leaflet or OpenLayers, can be used to create interactive and zoomable web maps with a plan oblique projection. These libraries are available for all current web browsers and supported by a variety of mobile devices. The plan oblique tiles are currently created in a server-side batch process, but on-the-fly rendering by a web service would also be possible. The disadvantage of server-side rendered plan oblique maps is that it is not possible to combine multiple tiled layers in the web browser unless a server process first transforms the tiles to the plan oblique projection on the server.

Client-side rendering of plan oblique maps addresses this limitation. By using GPU-accelerated rendering with WebGL, multiple orthographic web map layers can be combined and transformed to plan oblique relief on-the-fly. Because rendering with WebGL is fast enough for interactive frame rates, client-side plan oblique rendering could be used with GIS-like "hide and show" functionality for layer visibility. Our implementation uses an alternative approach where both the elevation data (using the red and blue channels) and the image data (using the blue and alpha channels) are stored in a single, compact tile set. However, with this encoding schema, texture images can only use a reduced color space (16-bit instead of 24-bit), which reduces the number of representable colors from millions to only thousands of available colors. The combined encoding of elevation and color values also limits flexibility because a new tile set must be generated if either the elevation or image data change.

The client-side approach for rendering plan oblique provides more interactivity and flexibility. Plan oblique rendering with WebGL enables the user to adjust parameters for terrain obliqueness, relief shading, texturing, and hypsometric tinting in real time. When the map is rotated, the rotation of inclined terrain is adjusted on-the-fly to the map orientation to ensure that mountains are always pointing in the upward direction.

For server-side rendering alternative architectures are feasible. On Windows systems OpenGL could be replaced with Direct3D. If higher rendering quality is required, the graphics pipeline could be replaced with slower ray tracing.

For client-side rendering, WebGL is available on all major platforms and web browsers, and is currently the only platform-independent API for hardware-accelerated rendering of 3D graphics. One of the disadvantages of using WebGL is that graphics hardware of some mobile devices is limited, which may result in reduced rendering speed or a limited number of texture images accessible via WebGL.

Conclusion

This article introduces two complementary methods for rendering plan oblique relief for web maps using the graphics pipeline. The browser-based graphics pipeline allows for real-time interactive adjustments of rendering parameters such as terrain inclination or map rotation, but important usability aspects have yet to be explored. The implicit assumption of the presented research is that plan oblique relief shows terrain in a more immediate and easier-to-grasp manner than standard, two-dimensional orthographic maps. This assumption remains to be verified.

Another question is whether map-readers have a preference when given the choice between plan oblique and standard orthographic maps. It would also be interesting to explore whether interactive options for adjusting illumination parameters and terrain inclination help users better perceive landform structures and whether these parameters are easy to set by novice users. Lastly, it remains to be explored whether plan oblique relief reduces the terrain inversion effect (Bernabé Poveda and Çöltekin 2014), which causes landforms to appear inverted when a north-oriented map is illuminated from the south.

Acknowledgments

The authors would like to thank William McNulty of Google for his support and Google for a Faculty Research Award. The authors would like to give special thanks to Brooke E. Marston, Oregon State University, for editing the text of this article, to Tom Patterson and Roger Smith for the permission to reproduce their maps, and to the anonymous reviewers for their valuable comments.

Funding

This work was supported by the French National Research Agency [grant number ANR-12-CORD-0025] and Google through a Google Faculty Research Award.

References

- Alpha, T. R., and R. E. Winter. 1971. "Quantitative Physiographic Method of Landform Portrayal." *Cartographica: The International Journal for Geographic Information and Geovisualization* 8 (2): 126–136. doi:10.3138/374X-71K6-171T-W54R.

- Auer, M. 2012. "Real-Time Web GIS Analysis Using WebGL." *International Journal of 3-D Information Modeling* 1 (3): 49–61. doi:10.4018/ij3dim.2012070105.
- Batty, M., A. Hudson-Smith, R. Milton, and A. Crooks. 2010. "Map Mashups, Web 2.0 and the GIS Revolution." *Annals of GIS* 16 (1): 1–13. doi:10.1080/19475681003700831.
- Berann, H. C. 1967. *Indian Ocean Floor* [Map]. Washington, DC: National Geographic Society.
- Bernabé Poveda, M.-A., and A. Çöltekin. 2014. "Prevalence of the Terrain Reversal Effect in Satellite Imagery." *International Journal of Digital Earth* (web preview), 1–16. doi:10.1080/17538947.2014.942714.
- Betschmann, O. 1937. *Reliefkarte Zentralschweiz – Berner Oberland* [map]. Luzern: Publisher: Nietlispach Verlag, Zofingen : Ringier.
- Cavelti Hammer, M., K. Spichtig, N. Von Flüe, T. Germann, P. Caminada, H.-U. Feldmann, and T. Glatthard. 2006. *Xaver Imfeld 1853–1909: Meister der Alpentopografie*. Sarnen: Ah Druck.
- Cerny, J. W., and R. E. Rudnicki. 1972. "On Three-Dimensional Surface Plotting Algorithms." *Proceedings of the Pennsylvania Academy of Science* 46: 103–105.
- Chrisman, N. 2006. *Charting the Unknown: How Computer Mapping at Harvard Became GIS*. Redlands, CA: Esri Press.
- Coradi, O. 1934. Pantographic map drawing apparatus. US Patent No. 1,943,205. Washington, DC: U.S. Patent and Trademark Office.
- Dept. of the Army (1960). Preparation of Terrain Diagrams. Engineer Intelligence Guide No. 33, Chief of Engineers, Washington, DC :Department of the Army.
- Dufour, P. T. 1917. "Nouveau procédé permettant d'obtenir les perspectives-reliefs des formes géographiques représentées sur les cartes hypsométriques." PhD diss., Faculté des sciences, Université de Paris, N°: 99, Delagrave.
- GDAL. 2014. "GDAL—Geospatial Data Abstraction Library." <http://www.gdal.org/>
- Heezen, B. C., M. Tharp, H. C. Berann, and H. Vielkind. 1977. *World Ocean Floor* [Map]. Washington, DC: United States Navy.
- Imfeld, X. 1887. *Reliefkarte der Zentralschweiz* [Map]. Winterthur: Topographische Anstalt von Wurster, Randegger & Cie.
- Jenny, B. 2008. "Otto Betschmanns Aerovue-Karten." *Cartographica Helvetica* 38: 13–20.
- Jenny, B., and T. Patterson. 2007. Introducing Plan Oblique Relief. *Cartographic Perspectives* 57: 21–40, 88–90. doi:10.14714/CP57.279.
- Johnston, W. D., and T. B. Nolan. 1937. "Isometric Block Diagrams in Mining Geology." *Economic Geology* 32 (5): 550–569. doi:10.2113/gsecongeo.32.5.550.
- Khronos Group. (2014). "WebGL—OpenGL ES 2.0 for the Web." <http://www.khronos.org/webgl>
- Lawrence, D. 1999. "Mountains under the Sea: Marie Tharp's Maps of the Ocean Floor Shed Light on the Theory of Continental Drift." *Mercator's World* 4 (6): 36–43.
- Lobeck, A. K. 1924. *Block Diagrams and Other Graphic Methods Used in Geology and Geography*. New York: Wiley.
- Monmonier, M. S. 1978. "Viewing Azimuth and Map Clarity." *Annals of the Association of American Geographers* 68 (2): 180–195. doi:10.1111/j.1467-8306.1978.tb01189.x.
- Natural Graphics. 2014. "Natural Scene Designer." <http://www.naturalgfx.com>
- Open Geospatial Consortium. 2010. "OpenGIS Web Map Tile Service Implementation Standard (version 1.0.0)." <http://www.opengeospatial.org/standards/wmts>
- Patterson, T. 2008. Creating a National Geographic-Style Physical Map of the World. In *Proceedings of the ICA Commission on Mountain Cartography Workshop, Lenk, Switzerland*. <http://www.mountaintcartography.org>
- Patterson, T. 2014a. "Seafloor Map of Hawai'i (Version 1.4, June 20, 2014) [Map]." <http://www.shadedrelief.com/hawaii/>
- Patterson, T. 2014b. "Mountains Unseen: Developing a Relief Map of the Hawaiian Seafloor." *Cartographic Perspectives* 76: 5–18. doi:10.14714/CP76.859.
- Peterson, M. P. 2012. "The Tile-Based Mapping Transition in Cartography." In *Maps for the Future*, edited by L. Zentai and J. Reyes Nunez, 151–163. Berlin: Springer.
- Přidal, P. 2008. "GDAL2Tiles Project." <http://www.klokan.cz/projects/gdal2tiles/>
- Raisz, E. 1956. "Landform Maps." *Petermanns Geographische Mitteilungen* 100 (2): 171–172.
- Raisz, E. 1962. *Principles of Cartography*. New York: McGraw-Hill.
- Raisz, E. J. 1931. "The Physiographic Method of Representing Scenery on Maps." *Geographical Review* 21 (2): 297–304.
- Ridd, M. K. 1963. "The Proportional Relief Landform Map." *Annals of the Association of American Geographers* 53 (4): 569–576. doi:10.1111/j.1467-8306.1963.tb00466.x.
- Robinson, A. H., and N. J. Thrower. 1957. "A New Method of Terrain Representation." *Geographical Review* 47: 507–520. doi:10.2307/211862.
- Segal, M., and K. Akeley. 1992. *The OpenGL Graphics System: A Specification*. Silicon Graphics.
- Smith, R. 2008. "Arthur's Pass National Park, New Zealand [map]." <http://geographx.co.nz/>
- Warmerdam, F. 2008. "The Geospatial Data Abstraction Library." In *Open Source Approaches in Spatial Data Handling*, edited by G. B. Hall and M. G. Leahy, 87–104. Berlin: Springer.
- Watson, Z. 2012. "Watercolor Process." http://content.stamen.com/watercolor_process
- Willett, W., B. Jenny, T. Isenberg, and P. Dragicevic. 2015. "Lightweight Relief Shearing for Enhanced Terrain Perception on Interactive Maps." Proceedings of the 33rd ACM Conference on Human Factors in Computing Systems (CHI 2015), Seoul, April 18–23. doi:10.1145/2702123.2702172.
- Wilson, W. H. 1936. "Tri-Dimensional Mapping of Extensive Mine Workings, with Special Reference to Those of the Kolar Gold Field." *Transactions of the Institution of Mining and Metallurgy* 45: 365–393.

Appendix 1. Plan oblique maps vs. maps with oblique parallel projection

Plan oblique maps use an image plane that is parallel to the ground plane, resulting in an undistorted rendition of flat objects, such as lakes or planes. If the image plane is oblique relative to the ground plane (as in Figure 1(c)), the ground plane is compressed along one axis. The compressed axis in parallel oblique maps is usually aligned with the vertical axis of the map. For a north-oriented map, the result is a map that uses a larger scale in the west-east direction and a smaller scale in the north-south direction. With this type of projection, distances can only be measured and easily converted to ground distances if they are parallel to the direction of either of the two axes. An exceptionally well-done manual map with a parallel oblique projection is Xaver Imfeld's *Reliefkarte der Zentralschweiz*, a map of central Switzerland published in 1887



Figure 10. Digital map with parallel oblique projection. The vertical north-south scale is smaller than the horizontal east-west scale. Section of *Arthur's Pass National Park*, New Zealand, courtesy of Roger Smith (2008).

(Imfeld 1887; for a discussion of this map, see Cavelti Hammer et al. 2006; figure in Jenny and Patterson 2007).

Figure 10 is a map by Roger Smith (2008) with parallel oblique projection produced with digital tools. The scale along the vertical north-south axis is smaller than along the horizontal west-east axis, which results in vertical compression of space. According to Smith, using two scales creates a map that looks “more real and natural” when compared to plan oblique relief (personal communication by Roger Smith, Geographx Ltd, Wellington, New Zealand).

Appendix 2. Construction techniques for analogue plan oblique relief and block diagrams

Before digital tools and digital elevation models became available, several auxiliary devices have been used to construct plan

oblique relief and other related terrain representations, such as block diagrams and other axonometric representations that foreshorten the ground plane along selected directions. Examples for auxiliary devices include isometric paper (Johnston and Nolan 1937), the use of inclined contours (Robinson and Thrower 1957), and a mechanical instrument called isometrograph. The isometrograph, invented by Dufour (1917), is composed of a linkage system, transverse bar, and a track. It resembles a pantograph, but converts planimetric contours into axonometric drawings (Lobeck 1924; Wilson 1936; Johnston and Nolan 1937; Raisz 1962; Alpha and Winter 1971; for a picture of the instrument, see Dept. of the Army 1960). A more sophisticated “pantographic map drawing apparatus” for the construction of plan oblique maps was developed and patented by the Swiss Coradi company in 1934 (Coradi 1934). However, this instrument has not been widely used (Jenny 2008).