

Challenges in adapting example-based texture synthesis for panoramic map creation: a case study

Helen Jenny* and Bernhard Jenny

College of Earth, Ocean and Atmospheric Sciences, Oregon State University, Corvallis, OR, USA

(Received 1 November 2012; accepted 9 February 2013)

Panoramic hiking and skiing maps are popular among tourists and map collectors. Such three-dimensional (3D) maps allow for easy orientation and provide the observer with an immersive impression of the landscape to be discovered or remembered. The most impressive panoramic masterpieces are almost exclusively painted manually or with minor digital tool support. Software packages that allow modern cartographers to create 3D maps of comparable visual quality (semi-) automatically are currently not available. Computer graphics has developed a number of methods for painterly rendering, including imitating the characteristics of a panorama artist's brush stroke and raster-based methods that synthesize new texture from examples. In this article, the latter approach is explored, and the idea of creating terrain textures for arbitrary regions by reassembling them from pieces of scanned hand-painted masterworks is pursued. Panorama painters vary the appearance of land cover depending on terrain characteristics and viewing parameters. This article suggests how the example-based texture synthesis approach could be adapted to accommodate such dependencies. A case study of transferring the appearance of H.C. Berann's hand-painted panorama of the Swiss Jungfrau region to a digital panorama of a different region is presented. The case study shows that a number of previously unanticipated hurdles are encountered when using hand-painted panoramas as input to an example-based texture synthesis algorithm. By identifying the challenges of applying texture-by-example to panoramic map making and by suggesting possible solutions, the authors aim to promote the creation of more visually appealing and legible digital panoramic maps.

Keywords: panorama maps; example-based texture synthesis; painterly rendering; terrain texture; 3D maps

Introduction

Panoramic hiking and skiing maps are a popular means of advertising a region's natural landmarks. Tourists enjoy these three-dimensional (3D) maps not only because they allow for intuitive orientation, but also because they immerse the observer in the beauty of the displayed landscape. The majority of highly impressive and admired masterpieces have been created manually or with only minor digital tool support (Tait 2010). Artists like Heinrich C. Berann (Austria), Hal Shelton (USA), Arne Rohweder (Switzerland), Winfried Kettler (Switzerland), and James Niehues (USA) are famous for their panoramic winter and summer maps.

The artist-cartographers show the landscape in an attention gripping abstracted realism that boasts a number of advantages over standard terrain texturing styles. Compared to draped aerial images, the perceptive and cognitive load on the observer in these hand-painted masterworks is reduced as the painters apply cartographic generalization principles to the landscape, for example, reinforcement of important and omission of unimportant detail (Patterson 2000). The map reader can rapidly recover the overall landscape structure while at the same time receiving an impression of the landscape's appearance in reality (Figure 1). Hand-painted maps can be

superior to mere renderings of digital elevation models with draped airborne images, where excess detail, clouds, shadows, undesired color variations, and interference with relief shading can be confusing, resulting in a representation that is ambiguous and difficult to interpret (Patterson and Kelso 2004).

Some panorama painters, for example, H.C. Berann, produce an immersive landscape impression by varying the depiction of land cover categories (e.g., forest, grassland, water bodies) with change in elevation, distance to the viewer, or lighting (Patterson 2000; Bratkova, Shirley, and Thompson 2009). Seasonal painted panorama series that advertise a region in summer and winter are sometimes encountered. They add vegetation variety and snow cover to the palette of possible land cover textures, which is typically not or only a very limited option in standard 3D rendering engines. Considering the time and effort needed to create a panorama by hand, the small number of professional panorama painters, and the popularity of landscape panoramas, many cartographers could be expected to welcome a (semi-) automatic method to create digital panorama maps in the look of hand-painted panoramas.

In the section "Current manual-digital approaches and related work in digital terrain texturing," we describe

*Corresponding author. Email: hjenny@coas.oregonstate.edu



Figure 1. Jungfraubahn (Switzerland) hand-painted by H.C. Berann (1947).

manual–digital methods currently used to create artistic panoramas, underlining the need for a less time-consuming (semi-) automatic method. An overview of related methods of (semi-) automatic terrain texturing that go beyond airborne image draping is subsequently presented.

We then introduce a new terrain texture synthesis method that focuses on extending an example-based texture synthesis approach from computer graphics, so that terrain textures for arbitrary regions can be reassembled from pieces of scanned hand-painted panorama masterworks. For this purpose, the choice of an example-based texture synthesis approach among other painterly rendering approaches is explained. Thereafter, we dive into the description of the basic example-based texture synthesis and transfer algorithm on which our method is built. We subsequently suggest how this approach could be extended to introduce variation in land cover textures based on terrain and viewing parameters, thus imitating techniques applied manually by panorama painters. A number of challenges arise when using example-based texture synthesis to transfer hand-painted panorama appearance. These hurdles are discussed in a case study that focuses on transferring the appearance of H.C. Berann’s panorama of the Swiss Jungfrau region (Figure 1) onto a digital panorama of a different region.

The bumpy road to semi-automatic terrain texture transfer

When we first started developing the presented method, we did not anticipate that the major challenge was not only developing the transfer method itself, but also using a hand-painted panorama as input. Hand-painted panoramas are often geometrically distorted. Thus, deriving elevation and land cover models for them is not easily done, yet

necessary for our texture transfer method. In addition to these challenges, we discuss the difficult task of handling cast shadows and locally rotated shadows on hand-painted panoramas that are used as input. We also reflect on using several hand-painted panoramas as input to the texturing algorithm to avoid overly repetitive appearance and to allow for interesting combinations.

While it may seem that more problems than solutions are discovered in this article, its major contribution is a time saving texture transfer method for panorama map creation. The full potential of this method can only be developed if it can be supplemented with equally time saving methods to create altitude and land cover information for hand-painted panoramas. We hope that our reflections on the encountered challenges will motivate other researchers in cartography and computer graphics to join us in tackling these remaining tasks.

Current manual–digital approaches and related work in digital terrain texturing

Common digital tools used by contemporary artists to create panoramic maps are raster graphics editors, such as Adobe Photoshop, providing imitations of analog pens and brushes. Digital drawing tools in Photoshop and other similar software accelerate the production process to a certain degree and provide a series of technical advantages compared to traditional painting. However, the production process has major disadvantages: drawing a map is still extremely time-consuming and requires considerable artistic talent, expertise, and a broad knowledge of physical geography for satisfactory results with manual–digital tools (Patterson and Kelso 2004).

Patterson (2000, 2001) suggested methods for creating medium and small-scale 3D panoramas using standard 3D rendering software combined with Photoshop. In contrast to the manual–digital painting approach described above, Patterson’s method requires much less artistic talent and knowledge of geographic detail. A number of beautiful digital panoramas created by Patterson with this combination of tools can be found at Patterson (2012). While Patterson did not explicitly pursue a hand-painted look in his digital panoramas, he analyzes H.C. Berann’s panoramas in detail and aims at transferring a number of characteristics. Yet as Patterson is working with a sequence of different tools and adapts his method to the demand of the region to be displayed, the method is still time-consuming and not semi-automatic or easily transferable to arbitrary regions.

Premoze (2002) developed a prototype terrain renderer where the user can interactively paint on the 3D terrain model. He suggests that colors should be adapted as in hand-painted panoramas, but does not provide a method description. In a semi-automatic approach, Dachsbacher, Bolch, and Stamminger (2006) use a clustering approach

to find color distribution in satellite images. They use the resulting color histograms to generate four classes of procedural land cover textures. Mantler and Jeschke (2006) experiment with enhancing forest representation on terrain by manipulating the underlying terrain model and adding special illumination effects. Premoze, Thompson, and Shirley (1999) developed an automatic method to simulate snow cover and seasonal illumination for very large-scale landscape views.

Bratkova, Shirley, and Thompson (2009) suggest an automatic painterly rendering approach for panoramas in the style of the famous panorama artist H.C. Berann. Bratkova's approach is similar to the one proposed in this contribution in that it addresses adapting land cover texture to terrain and viewing parameters using hand-painted panoramas as inspiration. However, her method differs from the one proposed in this project in that it is stroke-based instead of pixel-based.

Applying example-based texture synthesis to panoramic maps

To transfer the appearance of hand-painted panoramas to digital panorama maps, we chose to apply a texture-by-example approach. The term texture synthesis by example describes a family of methods that create visually similar textures based on an input exemplar without unnatural repetitions or artifacts (Wei et al. 2009). The approach is raster-based, which bears a number of advantages over stroke-based painterly rendering approaches. Regular grid data models are often used by GIScientists and cartographers when working with geospatial data sets, for example, orthorectified airborne images, derived land cover data sets, and 2.5D elevation models. For many areas of the world, these data sets are freely available. A raster-based approach can thus be conveniently extended to consider such grid-encoded parameters and can be applied to any region for which spatial data sets are available. Most stroke-based approaches to painterly rendering make the stroke and style explicit by defining stroke

parameters like position, density, orientation, width, length, etc. (Hegde, Gatzidis, and Tian 2012). The limited range of styles that can be expressed with stroke-based algorithms can be a problem (Hertzmann 2003). In contrast, raster-based texture synthesis by example is very flexible and works on many types of textures (Wei et al. 2009) as the style is expressed implicitly.

The basic synthesis-by-example algorithm

Many painterly rendering algorithms work with a cost function that expresses how well the rendered output is compared to a manually created exemplar or other input image. This applies to many stroke-based approaches (Hertzmann 2003) and to raster-oriented example-based methods:

$$C(I)_{\text{match}} = \sum_{u,v \in I} |I(u,v) - O(u,v)|^2$$

The most similar rendering is the one where the differences between the images at the locations u,v in the input image I and in the output image O are smallest – thus, where the value of the cost function is small. Similarity is expressed here as the sum of squared differences, but can be replaced by other similarity measures.

Simple versions of the texture synthesis-by-example approach aim at producing an arbitrarily large texture from a small texture input. The output texture is produced by taking pixels or patches of the input texture and reassembling them in a natural looking way. When creating a texture similar to the input exemplar, the method makes use of the cost function to decide which of the candidate pixels or patches (groups of pixels) in the input image is best to be placed in the output image. The difference in similarity that the algorithm aims to minimize is the similarity of L-shaped pixel neighborhoods or patch-borders (Figure 2, left). Thus, the average similarity of groups of pixels is examined as opposed to single pixel similarity. The neighborhood size is set based on the size of texture elements and is usually larger than the size shown in

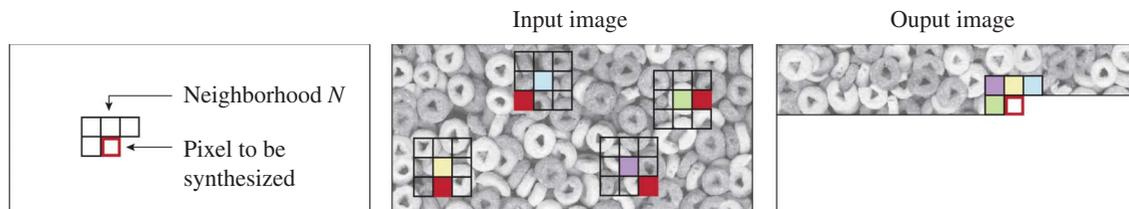


Figure 2. Illustration of the coherence principle to accelerate the basic texture synthesis-by-example algorithm. The location of pixels already synthesized (right image: green, violet, and yellow) in the neighborhood of the current pixel to be synthesized (red outline) is traced back to the color input (middle image: green, violet, and yellow). Candidate pixels (middle image: red fill) for the current pixel to be synthesized (red outline) are selected from the vicinity of these locations by considering the pixel in the color input with the same relative location as the neighborhood pixel has to the current pixel in the partly-synthesized output.

Figure 2. The method is inspired by the Markov Random Field texture model (Wei et al. 2009), which assumes that a pixel can be predicted from neighboring pixels and general similarity is apparent throughout the texture. One may recognize some similarity between this first assumption and Tobler’s first law of geography: “Everything is related to everything else, but near things are more related than distant things” (Tobler 1970).

The basic algorithm produces an output image by walking through the input pixels usually in scan line order and comparing the neighborhood N of each input pixel p_i in the input image to the neighborhood N of the pixel p_i (Figure 2, right) to be generated in the output image. The input pixel p_i with the most similar or similar enough neighborhood is transferred to the output image. This corresponds to a cost function that is evaluated for every pixel p_O to be generated in the output image O . For example, D_{color} finds the difference in RGB color between a neighborhood in the input and a neighborhood in the output as a distance in RGB color space.

$$\begin{aligned} D_{\text{color}} &= |N_{\text{color}}(p_i) - N_{\text{color}}(p_O)|^2 \\ &= |N_R(p_i) - N_R(p_O)|^2 + |N_G(p_i) - N_G(p_O)|^2 \\ &\quad + |N_B(p_i) - N_B(p_O)|^2 \end{aligned}$$

where N is the average of the pixel values in the neighborhood of pixel p .

The output pixel to be generated p_O will take on the value of the input pixel p_i , where D_{color} is the smallest.

Optimization of the basic algorithm

The basic algorithm as described above is rather slow as it works through the input image in scanline order. For our panorama texture transfer method discussed in the following section, we used an optimized variation of the basic algorithm suggested by Wei et al. (2009). Wei et al. introduced the coherence concept (Figure 2), which is based on the idea that pixels that are close in the input image are more likely to be close in the output image.

The improved algorithm does not consider all pixels in the input pixels as candidates to be potentially transferred to the output image, but only specific candidates. These specific candidates come from the vicinity of the input neighborhood of those pixels that have already been synthesized around the current output pixel to be generated.

Transferring textures from panoramic maps

In panoramas, more than one type of land cover is present, for example, forest, water, and grass. The distribution of these land cover classes differs between the input panorama and the output panorama to be generated. Such global structures (Figure 3, top row) that vary over the image cannot be reproduced with the basic algorithm. To introduce variation in global structure, Hertzmann et al. (2001) and Zalesny et al. (2005) suggest guiding the basic algorithm by providing one or several control masks. Hertzmann et al. (2001) called this *painting by numbers* as the output and input control masks restrict copying of input pixels from one type of label only to output areas

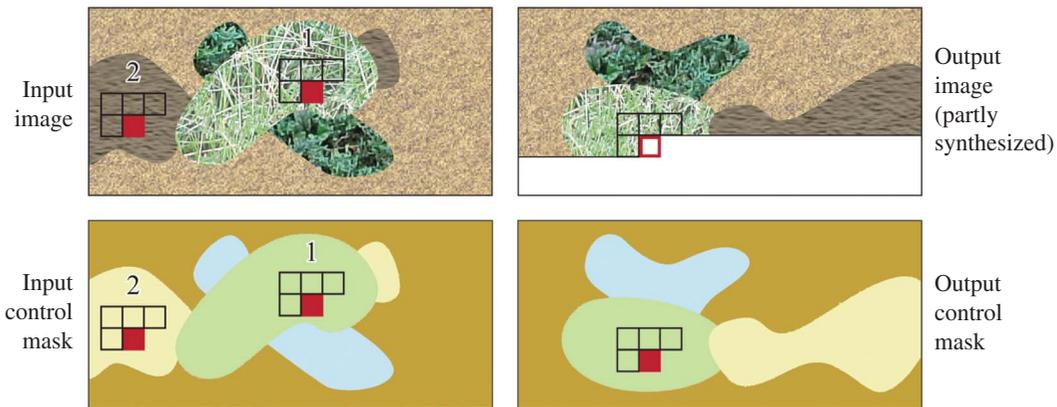


Figure 3. Illustration of *painting by numbers* concept using a control mask. Candidate pixels are filled in red; the current pixel to be synthesized is shown with a red outline. Candidate pixel 1 will be preferred to candidate pixel 2 because it shares the same control mask category with the current pixel to be synthesized. Input image, input control mask, and output control mask are the information provided to the algorithm to produce the output image.

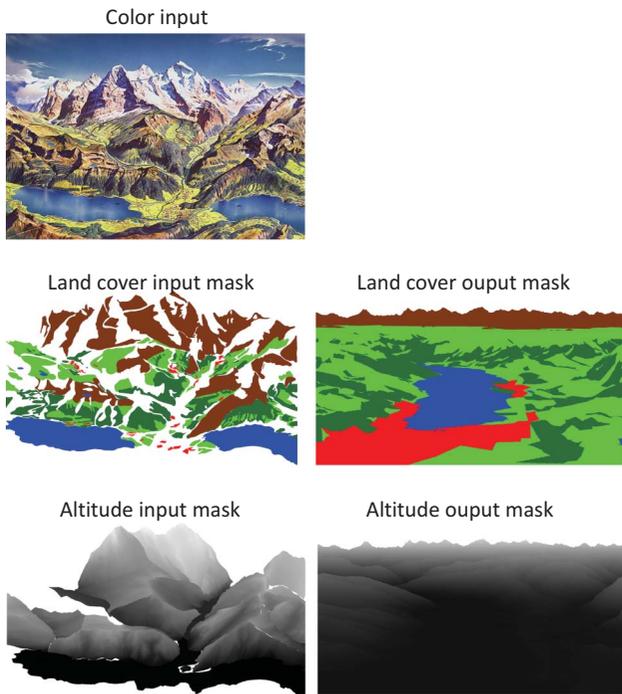


Figure 4. Information used by the adapted algorithm. Top row: Jungfraubahn by H.C. Berann (1947); middle row: land cover masks; bottom row: altitude gradient. All images serve as input to the algorithm to produce Figure 5.

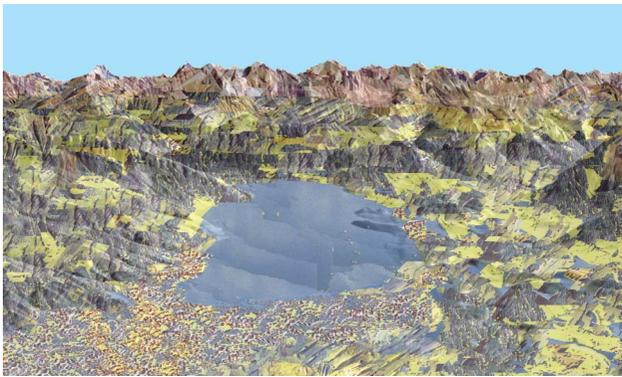


Figure 5. A tentative first result of the adapted algorithm showing a landscape texture for the target output region rendered with the adapted algorithm. It was blended with a terrain shading to add additional three-dimensional appearance.

with the same type of label or from transition area to transition area (Figure 3). For the purpose of panorama creation, an example would be copying pixels from forest areas in the input panorama only to forest areas in the output panorama (and border areas) and not to lake areas.

The method for transferring scanned hand-painted panorama textures for panoramic map making presented in this article builds on the *painting by numbers* approach. We use the scanned, hand-painted panorama of the Jungfraubahn by H.C. Berann (1947) shown in Figure 1 for a case study. In Figure 4 (top row), it is also shown as color input image for the algorithm. The output image is the digital panorama of a different region to be generated in Berann's style (Figure 5). To guide the placement of land cover areas, control of global structures is necessary and additional control masks need to be included. For the land cover control mask, a segmentation of the hand-painted panorama into land cover classes (Figure 4, middle row) is used. As no satisfying automatic segmentation method was available, the input land cover control mask was manually drawn with Adobe Illustrator. The output land cover control mask is an oblique terrain rendering showing land cover information for the output region. It was produced by draping a land cover data set onto a digital elevation model of the target region and exporting an oblique view as an image file. The inclusion of an altitude gradient as a control mask (Figure 4 bottom row) to vary the composition of textures by elevation is discussed in the section "Challenges encountered."

For a first test, a pixel-based version of the algorithm with coherence candidate pixel selection was implemented. The cost function C_{diff} that is minimized to select the best matching pixel is adapted to include the land cover similarity:

$$C_{\text{diff}} = D_{\text{color}} + w_{lc} D_{lc}$$

where D_{lc} is small when compared pixel neighborhoods contain approximately the same number of pixels from the same land cover class and punishes large differences in land cover provenance. w_{lc} is a scalar weight parameter that allows controlling the trade-off between color similarity and land cover similarity.

For a first tentative test (Figure 5), a constant neighborhood size pass was executed to create a panorama with the adapted algorithm. A multiresolution implementation is expected to avoid visible patches in the future. The result in Figure 5 appears somewhat darker than the Jungfraubahn panorama because it was blended with a digital terrain shading of the output region to add three-dimensional appearance. Fine-tuning of parameters, for example, neighborhood size and influence of land cover provenance through the weighting parameter w_{lc} is also a work in progress. This preliminary result helps identify the challenges that need to be met when one wants to use texture-by-example for digital panorama creation. It indicates how the algorithm should be improved to provide satisfactory results.

Discussion of encountered challenges

Improving input acquisition

The output example in Figure 5 shows a lot of repetitiveness. The algorithm selected the same input pixel groups (patches) many times when assembling the output image. This is especially apparent in the settlement areas designated in red on the land cover input and output maps (Figure 4, middle row). The few settlement texture areas in the input image do not provide enough texture variation to satisfactorily fill large settlement areas in the output image. The repetitiveness is also apparent in other land cover classes, where part of the input areas needed to be excluded from the algorithm because of shadow and lighting issues. As the lighting of the output region differs from the lighting of the input region, shadow regions in general were excluded for the first test run. Even if a digital elevation model and camera parameters could be generated for the hand-painted panorama (see section “Altitude”), automatic suggestion of where shadow areas are located would be difficult as panorama painters sometimes use nonuniform lighting including cast shadows and possibly locally rotated light sources (Patterson 2000).

A possibility to increase the number of available input pixels would be to use several hand-painted panoramas as input to the algorithm. If one land cover class, for example, forest, is not sufficiently represented in the input image, it could be completely replaced by forest from a different input panorama. As transitions between the replacement texture and the remaining textures are not present in a single image, these transitions would need to be created by a different method, for example, texture knitting (Zalesny et al. 2005). Sometimes panorama artists

paint a series of panoramas in very similar stroke style. Possibly the neighborhood search to find a fitting pixel could be extended to looking for the best fit in several panoramas. The algorithm would need to be adapted to allow for considering candidate pixel neighborhoods in several input panoramas. A color-harmonization of the two input panoramas as preprocessing step may be necessary.

Introducing texture variation by terrain and viewing characteristics

Altitude. Panorama painters sometimes vary the depiction of land cover depending on terrain characteristics. For example, in Figure 1, lower-elevation forest is sprinkled with yellow color spots that imply summer season. Similarly, grassland in lower and higher elevations is shown in different color shades and snow is only present in the highest elevations. The cost function for matching neighborhoods can be extended to include a term for altitude similarity D_{alt} , where w_{alt} is a scalar to control the influence of altitude similarity on the matching process.

$$C_{diff} = D_{color} + w_{lc}D_{lc} + w_{alt}D_{alt}$$

To find pixels of approximately equal altitude, altitude information is needed for the hand-painted panorama. For a first approximation, an elevation grid of the region was encoded as orthogonal 2D grayscale image, segmented into pieces and overlaid on the panorama scan in Adobe Photoshop. The individual 2D pieces were distorted to

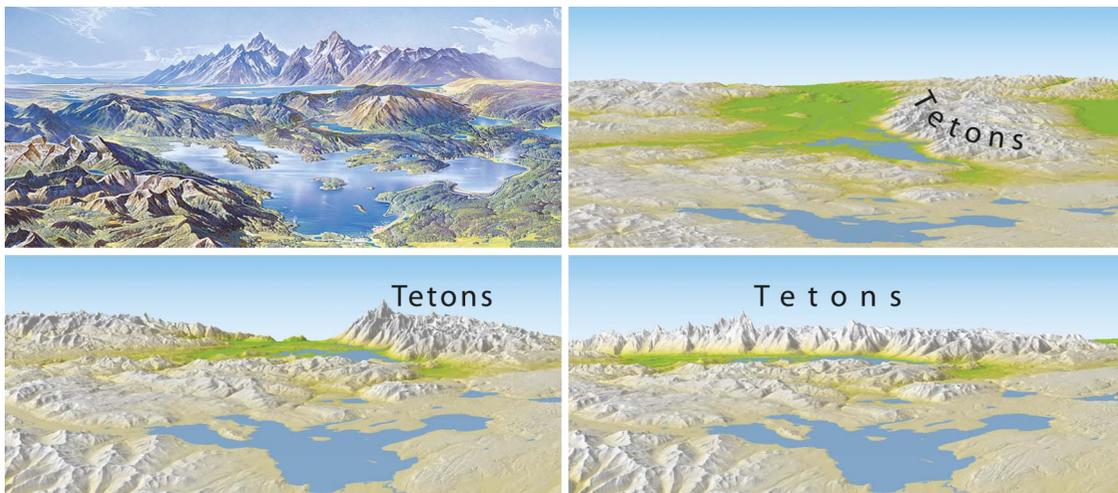


Figure 6. Deformation of a digital elevation model to match the geometry of a hand-painted panorama. Top left: excerpt of H.C. Berann’s Greater Yellowstone National Park panorama; top right: digital rendering of the same region undeformed; bottom left: digital rendering with progressive deformation; bottom right: digital rendering with progressive and local deformation (Jenny et al. 2011 adapted).

match the oblique distorted 3D display of the panorama landscape (Figure 4, bottom row left). This approach was inspired by a technique by panorama painter Arne Rohweder who uses obliquely viewed digital terrain shadings, which he cuts, distorts, and pastes into a collage to serve as orientation for the painting to be created (Holzgang 2005).

This process of visually distorting pieces of an elevation raster turned out to be too time-consuming and error prone. The major reason was that the geometry of hand-painted panoramas can be very strongly deformed. This is illustrated by Figure 6, which shows a digital elevation model that was deformed using Terrain Bender software (Jenny et al. 2011) to match an excerpt of H.C. Berann's Yellowstone panorama. The Tetons were rotated around the vertical axis by about 50° compared to the undeformed digital elevation model.

The Terrain Bender software does not allow making a connection between a terrain model and other geometry. A work in progress is to develop an algorithm that deforms a digital terrain model of the region shown in the painting. The algorithm is based on oblique viewing parameters and painting-model control point pairs suggested by the user. As a result, altitude information and viewing parameters could be used as information to feed into the terrain-texturing algorithm to match equal altitude areas from the input and output images.

Viewing distance and silhouettes. Viewing parameters, especially viewing distance and terrain silhouettes, could also be extracted from a terrain model distorted to approximate the geometry of a hand-painted panorama. Some panorama painters vary forest textures based on the distance to the viewer. Figure 7 shows excerpts of forest textures taken from H.C. Berann's Greater Yellowstone National Park panorama from the panorama foreground (left) progressively to the background (right).

In the distance, tree strokes have approximately the same size as tree strokes in the foreground but stand

symbolically for a larger number of trees. In contrast, standard rendering engines aim at photorealism instead of potentially more illustrative hyperrealism. Such engines would let the trees become smaller with increasing viewing distance to mimic perspective foreshortening (Bratkova, Shirley, and Thompson 2009).

As Bratkova, Shirley, and Thompson (2009) observe, the trees as texture elements on Berann's hand-painted panorama are not shown tangential to the surface, as one would expect from digital image draping. Instead, they stand upright, parallel to the up-axis of the image plane. Bratkova, Shirley, and Thompson (2009) also note that the tree trunks are often positioned along the fall lines of the terrain. Although such placement is possible with a stroke-based algorithm, it seems to be difficult to imitate with a raster-based texture-by-example approach.

Another challenge noticeable in our case study is that the terrain texture transfer algorithm may plant a patch of coherent pixels across landscape elements at different viewing depths (see Figure 5 in the mountains west of the lake). Thus, mountain ranges that should look disconnected appear connected, which disturbs depth perception and coherence. A possible solution would be to extract terrain silhouettes, which indicate edges where depth changes occur, from a digital elevation model of the painted region. This way, the proposed algorithm could be extended to respect viewing depth by introducing such necessary coherence breaks.

In the Jungfraubahn panorama (Figure 1), small valleys with cast shadows are a challenge for the proposed algorithm. Shadowed pixel patches from the input image may be placed unintentionally into lighted areas of the output image. Our method currently does not detect cast shadows or emphasized fall lines in forested and rock areas in the hand-painted input. As these areas cannot be avoided, they are placed in the output image at wrong locations, disturbing the appearance of terrain relief in Figure 5. Using pattern detection to identify these shadowed valleys and fall lines in the input could be an improvement. A far goal would be to reassemble



Figure 7. Excerpts from H.C. Berann's Greater Yellowstone National Park panorama.

shadowed regions in the output from shadowed areas in the input.

Conclusion

Very few artist-cartographers are able to paint panoramic maps by hand. Manual–digital approaches are still very time-consuming and require multistep processing. Hand-painted panoramas are very popular and often appraised for their clarity in conveying an easily readable, immersive impression of the landscape. A (semi-) automatic method to render digital panoramas in hand-painted style would enable the modern cartographer to apply this sophisticated style without need for artistic talent. In this article, a raster-based texture-by-example method was extended to accommodate transferring land cover texture from a scanned hand-painted panorama onto a digital elevation model. To produce texture variation as observed in hand-painted panoramas, it was suggested to include land cover, altitude, viewing distance, and silhouettes into the algorithm. Many challenges remain before an entirely satisfactory result can be expected. Most of these challenges are connected to missing terrain, lighting, and viewing information for the hand-painted input panorama. Panorama artists sometimes use lighting from multiple direction and distort terrain geometry, which makes it difficult to adjust a digital elevation model to match the hand-painted panorama and may make parts of the panorama unusable as a texture source. An input image without geometry distortion and inhomogeneous lighting could be achieved by having a panorama artist texturize an undistorted, standardly lit digital elevation model. While this would serve to better prove the worth of our terrain texture transfer method, the method's real potential lies in its flexibility to transfer a variety of panoramic map styles to panoramas of arbitrary regions. To fully develop this potential, the aforementioned challenges need to be met.

Acknowledgments

The authors thank Prof. Matthias Troyer for providing Figure 1, Juliane Cron for preparing Figures 2 and 3, Linda Lamb for proof reading, and the reviewers for their helpful comments.

References

- Bratkova, M., P. Shirley, and W. B. Thompson. 2009. "Artistic Rendering of Mountainous Terrain." *ACM Transactions on Graphics* 28 (4): 1–17.
- Dachsbacher, C., T. Bolch, and M. Stamminger. 2006. "Procedural Reproduction of Terrain Textures with Geographic Data." In *Proceedings of the Vision Modeling and Visualization*, edited by L. Kobbelt, T. Kuhlen, T. Aach and R. Westermann, Aschen, November 22–24, 105–112.
- Hegde, S., C. Gatzidis, and F. Tian. 2012. "Painterly Rendering Techniques: A State-of-the-Art Review of Current Approaches." *Computer Animation and Virtual Worlds*. doi:10.1002/cav.1435.
- Hertzmann, A. 2003. "A Survey of Stroke-Based Rendering." *Computer Graphics and Applications* 23 (4): 70–81.
- Hertzmann, A., C. Jacobs, N. Oliver, B. Curless, and D. Salesin. 2001. "Image Analogies." In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '01)*, 327–340. New York: ACM.
- Holzgang, R. producer. (2005) "Panoramamaler." (in German). SF VideoPortal podcast. Accessed May 9, 2013. <http://www.srf.ch/player/tv/schweiz-aktuell/video/panoramamaler?id=639c0bd0-a1ee-4d3a-aa4f-fa7c40cf3aae>
- Jenny, H., B. Jenny, W. E. Cartwright, and L. Hurni. 2011. "Interactive Local Terrain Deformation Inspired by Hand-Painted Panoramas." *The Cartographic Journal* 48 (1): 11–20.
- Mantler, S., and S. Jeschke. 2006. "Interactive Landscape Visualization Using GPU Ray Casting." In *Proceedings of the 4th International Conference on Computer Graphics and Interactive Techniques in Australasia and Southeast Asia (GRAPHITE '06)*, 117–126. New York: ACM.
- Patterson, T. 2000. "A View from on High: Heinrich Berann's Panoramas and Landscape Visualization Techniques for the US National Park Service." *Cartographic Perspectives* 36: 38–65.
- Patterson, T. 2001. "DEM Manipulation and 3-D Terrain Visualization: Techniques Used by the U.S. National Park Service." *Cartographica* 38 (1): 89–101.
- Patterson, T. 2012. "Shaded Relied – Maps and Data." Accessed June 30, 2012. <http://www.shadedrelief.com>
- Patterson, T., and N. V. Kelso. 2004. "Hal Shelton Revisited: Designing and Producing Natural-Color Maps with Satellite Land-Cover Data." *Cartographic Perspectives* 47: 28–55.
- Premoze, S. 2002. "Computer Generation of Panorama Maps." *Proceedings of ICA Mountain Cartography Workshop 2002*, Mt. Hood, OR, May 15–19.
- Premoze, S., W. Thompson, and P. Shirley. 1999. "Geospecific Rendering of Alpine Terrain." In *Proceedings of the Eurographics Workshop*, edited by D. Lischinski and G. W. Larson, Granada, June 21–23, 107–118. New York: Springer Vienna.
- Tait, A. 2010. "Mountain Ski Maps of North America: A Preliminary Survey and Analysis of Style." *Cartographic Perspectives* 67: 5–17.
- Tobler, W. 1970. "A Computer Movie Simulating Urban Growth in the Detroit Region." *Economic Geography* 46 (2): 234–240.
- Wei, L.-Y., S. Lefebvre, V. Kwatra, and G. Turk. 2009. *State of the Art in Example-Based Texture Synthesis*. Eurographics 2009, State of the Art Report, EG-STAR, 93–117. Accessed May 9, 2013. <http://diglib.org/EG/DL/conf/EG2009/stars/093-117.pdf>
- Zalesny, A., V. Ferrari, G. Caenen, and L. Van Gool. 2005. "Composite Texture Synthesis." *International Journal of Computer Vision* 62 (1–2): 161–176.