

© ACM, 2009. This is the author's version of the work.
It is posted here by permission of ACM for your personal
use. Not for redistribution

Artistic Rendering of Mountainous Terrain

MARGARITA BRATKOVA

University of Utah

and

PETER SHIRLEY

University of Utah

and

WILLIAM B. THOMPSON

University of Utah

Panorama maps are aerial view paintings that depict complex, three-dimensional landscapes in a pleasing and understandable way. Painters and cartographers have developed techniques to create such artistic landscapes for centuries, but the process remains difficult and time-consuming. In this work, we derive principles and heuristics for panorama map creation of mountainous terrain from a perceptual and artistic analysis of two panorama maps of Yellowstone National Park. We then present methods to automatically produce landscape renderings in the visual style of the panorama map. Our algorithms rely on USGS terrain and classification data. Our surface textures are generated using perceptual metrics and artistic considerations, and use the structural information present in the terrain to guide the automatic placement of image space strokes for natural surfaces such as forests, cliffs, snow, and water.

Categories and Subject Descriptors: I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism—*Color, shading, shadowing, and texture*

General Terms: Algorithms

Additional Key Words and Phrases: Non-photorealistic rendering, terrain, texture synthesis

1. INTRODUCTION

Panorama maps are aerial view paintings of landscapes, commonly used to present terrain information to non-expert viewers [Imhof 1963; Board 1990; Dorling and Fairbairn 1997]. Panoramas are painted by artists with a strong affinity for cartography. The work involved is difficult, manual, and often takes months at a time (depending on the size and detail of the panorama). Nowadays, digital tools and topographic maps are often incorporated in the process, and have given artists faster turnaround times. However, the process is still complex and time consuming, and it is commonly believed that only artists can create such intricate paintings successfully.

In this paper, we present algorithms that take the first step in creating such renderings in a more automatic fashion. While traditional cartography has developed many principles

Permission to make digital/hard copy of all or part of this material without fee for personal or classroom use provided that the copies are not made or distributed for profit or commercial advantage, the ACM copyright/server notice, the title of the publication, and its date appear, and notice is given that copying is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers, or to redistribute to lists requires prior specific permission and/or a fee.

© 20YY ACM 0730-0301/20YY/0100-0001 \$5.00



Fig. 1. A panorama map rendering produced by our system for Yellowstone National Park, using Berann-style brushstrokes and colors. Refer to Figure 2 for the original Berann painting.

to guide the cartographer in the making of traditional maps, there are no such conventions when it comes to 3D cartography [Haeberling 2004]. Regardless of professional background, people who currently create aerial maps are by necessity self-taught [Patterson 2005].

In order to understand the process of panorama map creation, we examine two different panorama maps that depict an identical terrain: Yellowstone National Park. The panoramas are painted by two well-known and respected, yet stylistically different, panorama artists - Heinrich Berann and James Niehues. We analyze their paintings in terms of function and style, and derive heuristics and principles appropriate for landscape visualization of mountainous terrain. We then present algorithms that automatically generate images with a visual style reminiscent of the panorama map (refer to Figure 1 for a rendering of Yellowstone National Park produced by our system, in the style of Heinrich Berann).

Our system creates renderings that incorporate surface textures of natural surfaces. Specifically, we produce images of mountainous terrain that have cliffs, forests, snow, and lakes. Our textures are applied in image space and our algorithms are influenced by structural information present in the terrain, perceptual metrics, and artistic considerations.

2. BACKGROUND

2.1 Panorama Maps

Panorama map paintings, also called bird's eye view or aerial view maps, are artistic cartographic renderings of extensive geographic areas. Figure 2 is an example of two paintings of Yellowstone National Park painted by two accomplished artists - Heinrich Berann and

James Niehues. Panorama maps are significantly different from images produced by GIS, by photographs, or by direct rendering of the terrain. These maps are painted by artist-cartographers who manipulate the geometry, enhance the features of interest, and add texture details that are not present in the original data [Patterson 2000]. The cartographic and artistic techniques merge to produce an immersive landscape visualization that instills a strong sense of natural realism, while helping us understand the terrain better. To our knowledge, there are no existing algorithms that allow us to produce such paintings automatically.

2.2 Non-Photorealistic Rendering

Non-photorealistic rendering (NPR) algorithms commonly produce stylized images, rather than realistic renderings. Synthesized strokes play an important role in the final rendering result, e.g. pen-and-ink drawing, sketch, or oil painting. Strokes follow visually meaningful directions and can be applied in image space or object space.

Image space approaches often use an existing image as input. By processing it, certain meaningful quantities are produced and later used to guide the rendering. Hertzmann et al. [1998] compute the normals of the image gradients from the image and then use them as input. Some researchers manually specify the necessary vector fields [Salisbury et al. 1997], other rely on information provided by the eye movement of human subjects [DeCarlo and Santella 2002]. Image space algorithms tend to be simpler and usually more general. However, necessary information may not always be correctly computed or even available from the image and that can sometimes cause problems for the rendering.

When a model of the scene is present, the stroke placement and direction can be guided by geometric information. Examples include lines of principle directions of minimal and maximal curvature [Interrante 1997; Elber 1998; Hertzmann and Zorin 2000], lines preserving the angle between the surface normal and the view direction (isophotes) [Elber 1998], or plane-surface intersections [Deussen et al. 1999]. While such algorithms are more computationally intensive, they provide more flexibility since one can extract specific and visually relevant geometric information.

In addition to strokes, non-photorealistic shading models are also relevant to this work. Specifically, the model of Gooch et al. [1998] is built on practices of traditional technical illustration, and produces images that have better visual structure. More recently, Rusinkiewicz et al. [2006] propose a non-photorealistic shading model that is inspired by cartographic relief techniques. Their solution is designed to convey fine-scale detail, as well as overall geometry shape, regardless of surface orientation.

2.3 Digital Terrain Rendering

Algorithmic landscape sketching of terrain was first introduced by Visvalingam and Dowson [1998]. Their goal was to create sketches resembling artistic landscape drawings by computing terrain profile curves - curves that are perpendicular and orthogonal to the line of sight. A filtered set of points from these curves is then used to build a sketch of profile strokes (p-strokes). Whelan and Visvalingam improved on the p-stroke sketch by adding formulated silhouettes [2003]. These are silhouettes that would be brought into view if a higher viewpoint is to be selected. Lesage and Visvalingam [2002] compute terrain luminance maps and then extract sketch lines using four common edge detectors. Most recently, Buchin et al. [2004] advanced a line sketching technique incorporating slope lines and tonal variations.

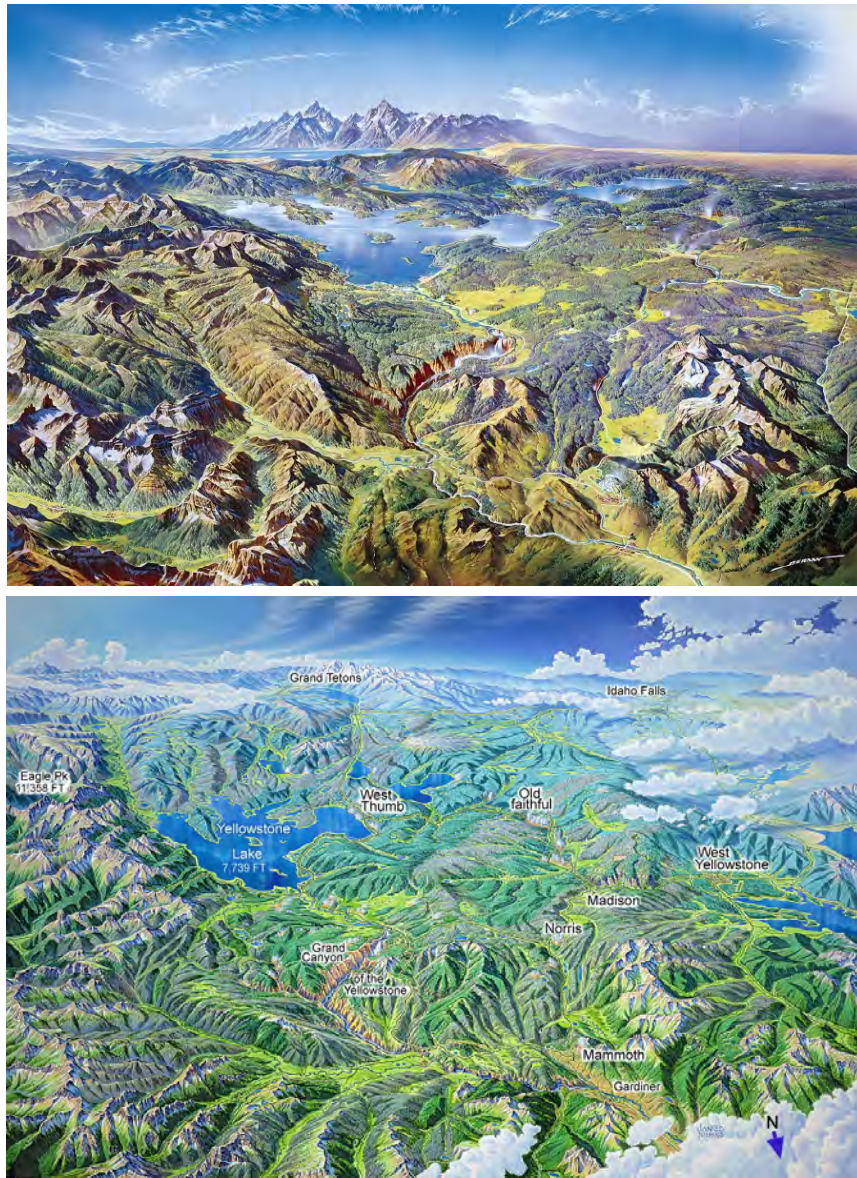


Fig. 2. Panorama Map Paintings of Yellowstone by Heinrich Berann (up) and James Niehues (down). (*Berann images used are in the public domain. Niehues images are used with permission*)

Saito and Takahashi [1990] developed methods for the visualization of topographic maps and provided an algorithm for producing an aerial view map that combined a contour image, a profile image, and a shaded relief image. While the produced image is an improved visualization of terrain over some GIS renderings, it is distinctly different from panorama map paintings.

Premože [2002] proposed an interactive tool for panorama map creation. The editor
ACM Transactions on Graphics, Vol. V, No. N, Month 20YY.

provides a 3D perspective view of the terrain drawn from a selected viewpoint, as well as a 2D contour view. The user can perform selective vertical exaggeration and vertical axis rotation. Orthoimage texture mapping can be draped over the terrain, and the user can selectively paint over the terrain skin. Sky color and atmosphere effects can also be specified. While the editor provides certain improvements over traditional terrain visualization packages, it does not allow the resulting maps to have a stylized quality, nor does it modify the geographical information in ways that improve its perceptual legibility and is not automatic.

3. PANORAMA ANALYSIS

We are interested in developing algorithms that would allow us to automatically render images of terrain that have a similar visual quality to panorama map paintings. To our knowledge, there is very limited cartographic and artistic theory we can directly apply to this problem.

In order to derive some principles and heuristics for panorama map automation, in this section we analyze how visual elements can affect the perception of panorama paintings and help convey the stylistic preferences of artists. We support our analysis with examples from two panorama maps of the same terrain - Yellowstone National Park, painted by two stylistically very different, yet highly renowned and sophisticated panorama artists: Heinrich Berann and James Niehues. Figure 2 shows their Yellowstone panorama paintings.

Throughout this section, all supporting Yellowstone Panorama Map example close-ups are from Berann, to the left, and Niehues, to the right. We limit our exploration to five categories of terrain textures, the ones most commonly used in mountainous maps: trees (evergreens and deciduous), cliffs, snow, lakes, and grass.

First, we provide an overview of pictorial cues and how they can be relevant to the perception of depth in panorama maps. Then, we examine how the visual elements used by the two artists serve a functional role by helping the visual system perceive surface shape and orientation. We also explore how the visual elements serve a stylistic role and allow the artists to make their paintings subjectively expressive and pleasing. Finally, we summarize a list of principles and heuristics that can be used in the automation of panorama map rendering. While we base our analysis on only two examples, we feel they are different enough to extend to a range of styles.

3.1 Visual Perception and Pictorial Cues

Maps visually convey a sense of the spatial structure of the terrain, as well as the spatial context of natural and cultural features located on the terrain. In perceptual terms, this means conveying information about distances and surface orientations [Marr 1982; Gibson 1950].

When considering visual perception, distance information is often divided into *depth* (or *egocentric distance*) and *exocentric distance*. Depth is the distance between the viewer and a visible location in the environment, while exocentric distance is the interval between two arbitrary environmental locations, expressed in an environmental frame of reference. Conventional plan view (overhead view) maps explicitly represent exocentric distance. Some plan view maps augment this with an implicit indication of surface orientation using contour lines or shading.

Unlike maps, landscape paintings, such as panorama maps, create a sense of directly viewing three-dimensional terrain and are dominated by perceptual information for depth

<i>Pictorial Cue</i>	<i>P</i>	<i>F</i>	<i>I</i>	<i>S</i>	<i>L</i>
Convergency of Parallel Lines	●				
Position Relative to Horizon	●				
Relative Size	●				
Absolute Size	●				
Texture Gradients	●	●			
Projection of 3D Texture Elements	●	●	●		
Edge Interpretation			●		
Surface Contours				●	
Shading					●
Aerial Perspective					●

Fig. 3. Pictorial cues for depth and surface orientation arising from perspective scaling (*P*), foreshortening (*F*), interposition (*I*), intrinsic surface markings (*S*), and light transport (*L*).

and orientation. The depth and orientation information in panorama maps is conveyed through *pictorial cues*, which do not involve binocular stereo, ocular information, or motion. Pictorial information arises due to perspective scaling, foreshortening, interposition (occlusion), intrinsic surface markings [Stevens 1979; Knill 1992], and light transport [Knill 1992; Leung and Malik 1997; Palmer 1999]. Figure 3 lists the pictorial cues most relevant to the perception of depth and surface orientation.

3.2 Functional Elements

For the purposes of panorama map paintings, the potentially most important pictorial cues are texture element distributions affected by foreshortening, the partial occlusion of the texture elements in forested areas (though not the variation in the amount of occlusion, as used by Leung and Malik [1997]), surface contours [Stevens 1981], shading, and silhouetted terrain features. Panorama maps covering more extensive areas could use all of these cues, together with position relative to the horizon, relative and familiar size, texture gradients, and aerial perspective.

TEXTURE GRADIENTS

Panorama maps of mountainous terrain typically represent outdoor areas in which forest is common. The three-dimensional nature of forests results in an appearance of forest textures in projected images that is quite distinct from textures generated by surface markings. Forests are unusual in that the elements that form them (the trees) are not tangential to the surface. The image plane projection of forests forms 3D image textures. Such textures have the potential to provide the human visual system with a number of distinct perceptual cues, because the projection of the texture elements (texels) to the image plane varies based on distance and surface orientation.

Shape-from-texture theory [Gibson 1950; Witkin 1981; Garding 1993; Rosenholtz and Malik 1997; Malik and Rosenholtz 1997] suggests using the texture distortions caused by projection for the purpose of recovery of surface shape and orientation. Such projective distortions can affect the *texel shape* (compression or foreshortening, e.g. a circle projects into an ellipse), the *texel size* (perspective scaling), and the *texel density* of texel elements. In addition to these, 3D textures also introduce *texel occlusion*. We examine these four cues and develop guidelines we believe are specifically applicable to the visualization of

forested textures. Our goal is to generate forest textures that provide as much functional 3D information in terms of perceptual cues as possible.

Texel shape foreshortening in 3D textures has been mostly unexplored in the shape-from-texture literature. Since the shape of tree texels does not change much in the projection, we believe that in this context foreshortening probably does not serve a useful purpose, as the small shape variations would be hard to detect.

True perspective projection dictates that *texel size* of elements vary with distance. However, since panorama maps often represent large distances, we believe that in doing so, texture elements in the distance will become too small to serve a visual purpose or to fit the resolution of the painting. Instead, we propose that texel size variations should be based on the extent of the visualized terrain. The chosen perspective scaling should guarantee that texture elements present even in the far distance are discernible in the final rendering. Both panorama artists seem to follow this rule (see Figure 4).

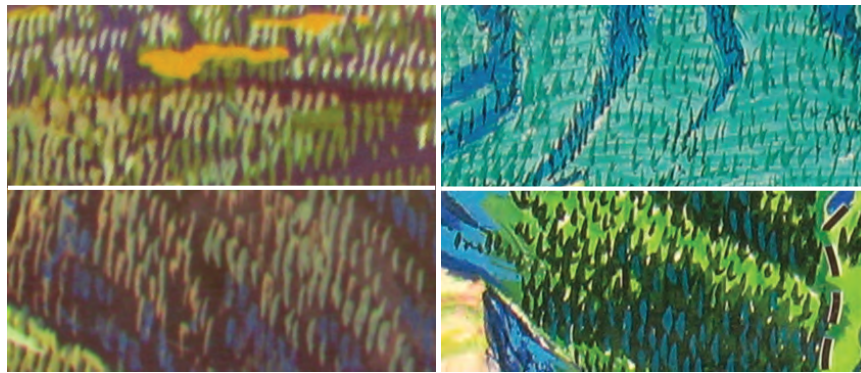


Fig. 4. Trees in the distance (top) and foreground (bottom). If size of trees varied with true perspective, pixel size of trees in the background would be a small fraction of their foreground size. Clearly this is untrue, especially in Berrann’s painting (left).

The projection distortion of *texel density* is a function of both perspective scaling and foreshortening [Stevens 1979]. The effects of scaling and foreshortening can be partially separated under a first order homogeneous assumption (i.e. the texture pattern has a constant area or density). Under this assumption, Aloimonos [1988] showed that density foreshortening can be used in the recovering of surface orientation. Malik and Rosenholtz provide evidence that even under the stronger homogeneity assumption (i.e. second and higher order statistics are translation-invariant), density foreshortening can be used to recover surface orientation [1997].

If we are to benefit from visual information present due to the projection distortion, we must ensure our texels are homogeneously distributed on the 3D surface. This argues either for an object-based rendering approach or for an image space approach that takes into account certain 3D information so that, as far as it is practical, homogeneity is reasonably approximated. In addition, we believe that texel distributions should be further constrained, so as to imply additional structural information, as we believe is done in landscape paintings via the application of tree strokes in the direction of fall lines. We address this issue when we explore surface contours.

When projecting 3D elements to the image plane for certain viewpoints, *texel occlusion* [Leung and Malik 1997; Saunders 2003] will occur. The occluded element is visually perceived as positioned behind the occluding element. The visual overlap is a very strong relative depth cue. This occlusion never happens for 2D textures. However, it often occurs in forest textures incorporated in landscape visualizations. Therefore, it is important that rendered texture elements align such that they partially occlude each other, especially for steeper slopes, where such occlusions are more likely to naturally occur. In agreement with perceptual theory, panorama artists use occlusion in their depiction of forests (see Figure 5).

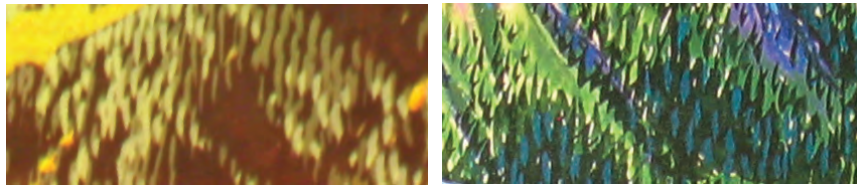


Fig. 5. Tree strokes in forest textures are laid down so that they partially overlap each other. The visual system interprets that to mean that the overlapped stroke is behind the overlapping one.

SURFACE CONTOURS

Surface contours were first introduced by Stevens [1979] and arise from the projection of extended surface markings to the image plane [Knill 1992]. Such surface markings are formed by a variety of physical processes, such as erosion, etc. Stevens suggests surface contours help us perceive shape, because the visual system attributes a strict geometric relationship to the surface contours and the curvature of its underlying surface. Knill believes surface contours help us perceive shape, because people have prior assumptions about the constraints present in such surface markings.

In this work, we are interested in the ways surface contours aid landscape visualization. Artistic line drawings of terrain shape often incorporate *fall-lines* (sometimes called slope lines), the lines drawn “downhill” the path of steepest decent [Imhof 1982; Buchin et al. 2004]. They provide much information about terrain shape, as seen in the line integral convolution [Cabral and Leedom 1993] image in Figure 6, and are believed to describe the essential structure of relief [Koenderink and van Doorn 1998].

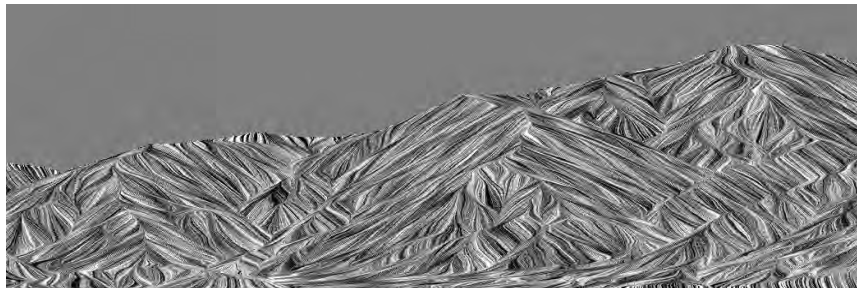


Fig. 6. This rendering is a visualization of the flow along fall lines using line integral convolution.

Considering how vital fall lines are to relief, it is not surprising that panorama artists consistently use them as a directional guide for stroke placement in the painting of cliffs and other mountainous forms (see Figure 7, bottom). In addition, the directions of fall lines is commonly used as imaginary lines along which to place vertical tree strokes, spacing them according to slope (see Figure 7, top). We refer to such strokes as *meta-strokes*. We believe that when artists incorporate fall lines in their paintings, they combine surface contours with texture cues for surface orientation.

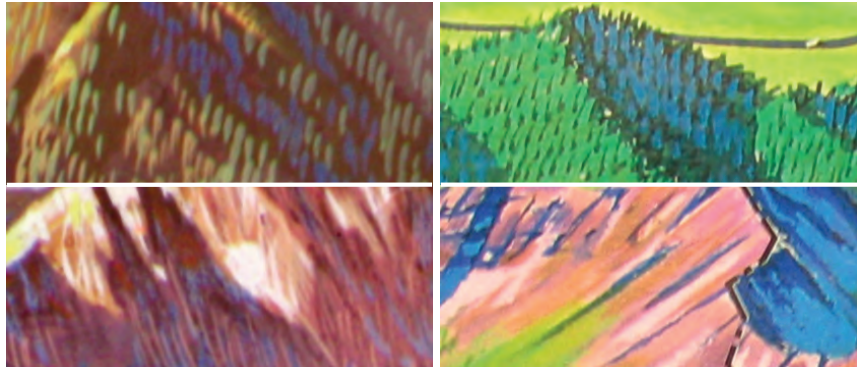


Fig. 7. Tree strokes are placed along the imaginary fall lines of the geometry (top images). Cliffs and snow are painted with strokes that follow the direction of fall lines (bottom).

SHADING

Shading is vital in depicting terrain surface form [Horn 1981] and is an effective and very useful perceptual depth cue. Imhof emphasizes the importance of using midtone to help the light and dark areas express the form of the terrain [1982]. It is commonly recommended that the tone of shading gradation covers the spectrum of value - light, medium, and dark, because by doing so, a painting helps the viewer perceive the form better [Sullivan 1997; Da Vinci 2003]. We note the difference between standard rendering practices where continuous shading is the norm, and painting, where artists explicitly blend palettes of discrete colors.

Light direction plays an important role in successful landscape visualization. Imhof [1982] recommends that light comes diagonally and at an acute angle to the more dominant direction of valleys and mountain ranges. It is also important to use light coming angularly from the front, so as to achieve a good balance between the light and shade of the terrain in the image [Haeberling 2004]. Panorama artists also seem to follow these rules (see Figure 8).

SILHOUETTES

Silhouettes are the contours formed at the boundary of an object and its background. They are perceptually useful as they indicate edges where depth changes. Such edges result from an image-based spatial discontinuity and signal real-world distance between the surfaces separated by the silhouette curve. Silhouette edges are especially appropriate for expansive mountainous terrain, where the landscape commonly overlaps in the distance (e.g. rolling hills). It is therefore not surprising that artists commonly use that cue to increase the perception of depth in their paintings (see Figure 9 for an example of the use of silhouettes).

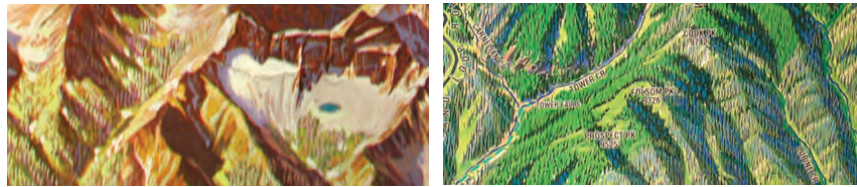


Fig. 8. Light direction is diagonal to the dominant direction of valleys and mountain ranges.



Fig. 9. Silhouetted curves are used to separate an object from its background and are used in panorama maps to imply depth.

3.3 Stylistic Elements

We now analyze the basic stylistic variations present in the two panorama maps of Yellowstone National Park we have been examining so far. We observe that the physical act of painting landscapes is achieved by layering color shapes and brush elements that vary in color and size depending on the natural elements portrayed [Patterson 2000].

We believe our stylistic analysis should examine variations in three basic parameters - *base color*, *brushstroke color*, and *brushstroke shape*. The base color provides the basic shading of the terrain, and varies with classification type, while the brushstroke color and brushstroke shape create the visual appearance of the texture elements.

BASE COLORS

The base color for all texture regions is chosen such that its interaction with the tree strokes provides enough contrast and makes the brushstrokes stand out. That usually implies that the colors tend to be darker and often less saturated. The base color also varies with surface orientation, as is to be expected. Colors in the light tend to be brighter and warmer in tone, while colors in the shade, darker and colder in tone. Figure 10 provides samples for light, medium, and dark tone in lighted, as well as shaded areas, for both panoramas. We observe that background colors get desaturated and often have a blue shift, as would be the case when the effects of Aerial Perspective are taken into effect.

BRUSHSTROKE COLORS

Brushstroke colors play an important role, as they add vibrancy to the rendered image. As expected, they change with respect to the orientation of the surface towards the light (see Figure 10). Niehues tends to vary mostly the value, while Berann also adds a variation in hue. A good distribution of strokes in light, medium, and dark tones achieves a stronger articulation of the form. Both artists add brushstrokes for areas that represent forest, snow, cliffs, or water. They don't seem to use strokes for grassy areas.

Niehues uses brighter and less natural colors. The colors he chooses for areas in shade have a strong blue shift, both in the foreground and in the background. Berann uses more

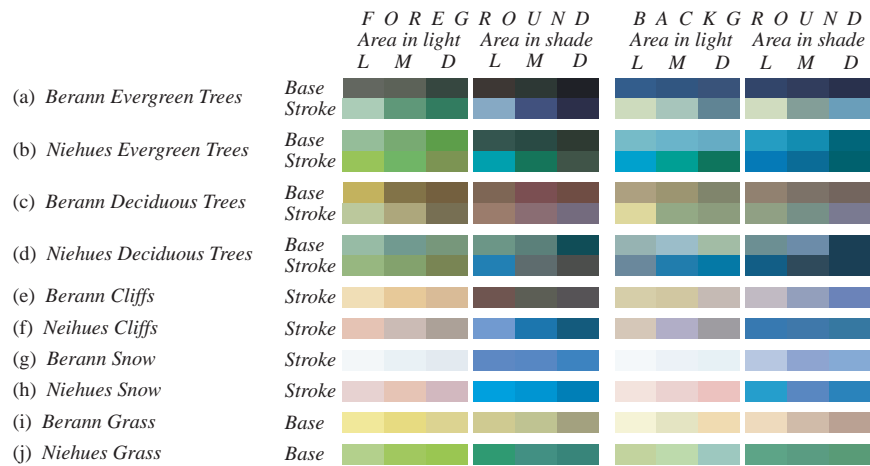


Fig. 10. Sample of light (L), medium (M) and dark (D) colors used by Niehues and Berann in their Yellowstone painting for areas in the foreground and the background.

realistic tones. The brushstroke colors are more natural in hue and more diverse in value. Snow and cliffs is mostly painted by the use of strokes, and the base color does not come through much.

Water textures use base colors that vary with viewing angle. Berann uses the brushstroke color to emphasize the viewing angle even further. In Niehues' work, the base color varies little with viewing angle, and the stroke colors mimic the effect of physical water reflection.

BRUSHSTROKE SHAPES

Niehues varies the way he paints brushstroke shapes for trees, depending on the type painted - evergreen vs deciduous. Berann does not seem to make such a distinction. The brushstroke shapes for trees are very simple, elongated, perhaps a little bit fatter at the bottom, and orientated in the vertical direction. Figure 11 provides a sample of evergreen and deciduous brush strokes from both artists.

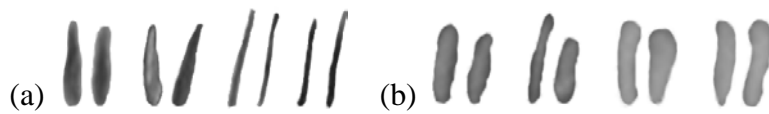


Fig. 11. Sample of the shapes for tree strokes used by (a) Niehues and (b) Berann. For each artists, the two samples to the left correspond to strokes used for evergreen trees, while the two to the right - for deciduous trees.

Berann's cliff and snow strokes are applied in various scales - small, long and distinct, or applied sweepingly and semi-transparently over large areas so as to unify them. Niehues uses much simpler, uniform brushstroke lines. They seem to alternate between chosen colors, following the direction of fall lines.

Water strokes seem to be very simple and applied horizontally to the image plane.

3.4 Principles and Heuristics

In the hope of reverse-engineering certain conventions used by panorama artists, we analyzed the visual elements of two panorama map landscape paintings in terms of function and style.

An additional useful heuristic from cartography is *vertical exaggeration*. It is often used when subtle topographic features need to be made more prominent, or when the terrain covers large areas and features get lost due to scale - as would be the case for panorama maps. Exaggeration scales often vary with the type of depicted terrain, but scales typically range from 1:1.5 to 1:5.

SUMMARY

Here is a summary of our proposed heuristics and principles for panorama map creation:

- The image texel size of surface textures that represent 3D elements (e.g. forest) should vary with distance, but should not match true perspective (*texel size* in Section 3.2, Texture Gradients).
- The image space distribution of texel elements of 3D textures (e.g. forest) should mimic one that would result from the projection of homogeneously distributed surface elements (*texel density* in Section 3.2, Texture Gradients).
- Image space texel spacing of 3D textures should ensure that texels overlap, especially in steep areas (*texel occlusion* in Section 3.2, Texture Gradients).
- Fall lines follow essential structures of terrain. They act as surface contours, and are used by panorama artists to paint cliff and snow textures (*fall lines* in Section 3.2, Surface Contours).
- Fall lines are used as imaginary lines along which tree strokes are placed, acting as texture *meta-strokes* (*meta-strokes* in Section 3.2, Surface Contours).
- Shading tone should have a good distribution of light, medium, and dark values (*shading* in Section 3.2, Shading).
- Light position should be placed so that the rendering of the terrain exhibits a good balance of light and shade, as seen from the selected viewpoint (*light direction* in Section 3.2, Shading).
- For extended terrain areas, indicating silhouettes, especially between occluding hills, is useful (*silhouettes* in Section 3.2, Silhouettes).
- Water surfaces should reflect the environment (*water textures* in Section 3.3, Brushstroke Colors).
- Geometry should be emphasized by use of vertical exaggeration (*vertical exaggeration* in Section 3.4).

4. PANORAMA AUTOMATION

The goal of this work is to develop methods for meaningful, automatic landscape visualization, with an emphasis on natural landscape panoramas. Specifically, we present a set of algorithms that produce images in the style of the panorama map. We concentrate on elements present in mountainous terrain. We do not address geometric feature enhancements, though such techniques are very valuable, and are used by panorama artists [Patterson 2000]. Our algorithm proceeds in three stages: preliminary setup, base shading, and surface textures (refer to Figure 12 for an overview).

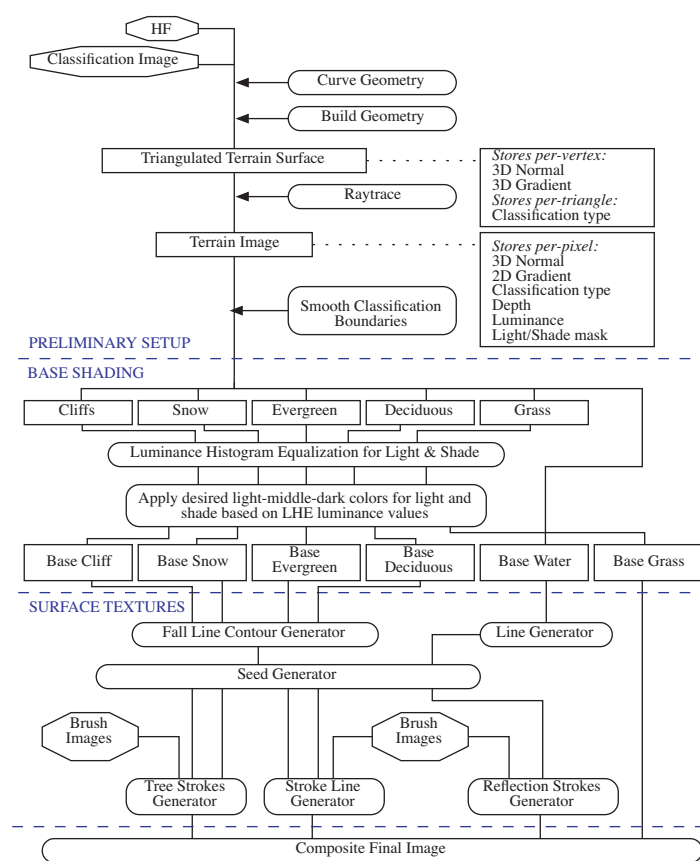


Fig. 12. An overview of the steps taken by our algorithm.

4.1 Preliminary Setup Stage

TERRAIN GEOMETRY AND CLASSIFICATION

The geometry of the terrain is specified by Shuttle Radar Topography Mission (SRTM) elevation data. The data is stored in a floating point format and is a heightfield. For the purposes of rendering, the heightfield is converted into triangulated mesh.

We also read-in a classification image that corresponds to the terrain data. It is based on the National Land Cover Database (NLCD) land cover data compiled by the Multi-Resolution Land Characteristics (MRLC) Consortium, freely available from the U.S. Geological Survey (USGS).

VERTICAL EXAGGERATION

Since panoramas depict terrain that covers large areas, vertical features are often lost in the vast horizontal extent of the geometry. Cartographers address this problem by increasing the vertical scale of the geometry in relation to the horizontal scale, thus improving the recognition of such features. This is a common visualization technique, and is referred to as “vertical exaggeration”.

We add exaggeration in a similar manner by allowing users to specify two exaggeration parameters: one for lowest elevation terrain and another for the highest elevation (peaks). For a given elevation value, we determine the exaggeration factor by linearly blending in between the two exaggeration factors given. This allows for control over both exaggeration in small scale details such as small hills (that would otherwise be unnoticeable) and peaks independently of each other. A more crude constant multiplier for the full terrain tends to exaggerate the geometry unacceptably.

FIRST-ORDER TERRAIN DERIVATIVES

A heightfield is a parametric surface of two variables, $\mathbf{S}(u, v)$. Let $f(u, v)$ be the elevation of the heightfield at a given parametric location $\mathbf{P} = \langle x(u, v), y(u, v), z(u, v) \rangle$. In the simplest mapping, $x(u, v) = u$, $y(u, v) = v$, and $z(u, v) = f(u, v)$. However, more complex mappings are allowable: the surface can be transformed in any manner, yet will remain a parametric surface of two variables. In parametric space, ∇f points in the direction of maximum elevation change. $-\nabla f$ is thus the fall line vector that we need to guide our texture placement. However, we desire the fall line in cartesian space instead of parametric space so that we may eventually project that value into image space.

To compute the fall line in cartesian space, we first need to determine the ∇f vector in cartesian space, and then negate it. Starting at (u, v) in parametric space and moving along the surface in the direction ∇f brings us to $(u + \frac{\partial f}{\partial u}, v + \frac{\partial f}{\partial v})$. It is this new parametric location that corresponds to an increase in the height function. The corresponding point on the surface \mathbf{S} in cartesian space is reached by starting at $\mathbf{S}(u, v)$ and moving by $\frac{\partial f}{\partial u}$ along the first tangent ($\frac{\partial \mathbf{P}}{\partial u}$) and $\frac{\partial f}{\partial v}$ along the second tangent ($\frac{\partial \mathbf{P}}{\partial v}$). Therefore, ∇f in parametric space corresponds to $\frac{\partial f}{\partial u} \frac{\partial \mathbf{P}}{\partial u} + \frac{\partial f}{\partial v} \frac{\partial \mathbf{P}}{\partial v}$ in cartesian space. Conveniently, we also need these tangents to compute the normal to the parametric surface: $\mathbf{N} = \frac{\partial \mathbf{P}}{\partial u} \times \frac{\partial \mathbf{P}}{\partial v}$.

To compute surface derivatives on the terrain in either the u or v parametric directions, we use central differences when possible. On the boundaries, we use either forward or backward differences as appropriate.

GEOMETRY CURVING

Viewpoint selection affects the final rendered image. Selecting a low elevation view point produces an image that includes the horizon. However, features in the foreground occlude features in the background. Selecting a high elevation point generates an image where we see larger parts of the terrain. However, the view, in the extreme case, resembles the traditional two-dimensional cartographic map and the image loses its 3D spatial quality.

Since visibility of both the horizon and the terrain features is desirable, some panoramists like Heinrich Berann emulate the view from a high-elevation viewpoint and curve the base of the terrain [Patterson 2000]. This makes it possible to see both non-obscured features in the foreground as well as the horizon and sky in the distance.

To provide a similar effect, we transform the original terrain surface into a curved one, so that rendering it now combines multiple perspectives into one. The geometry in the background is from a low elevation view point, while the geometry in the foreground appears as seen from a high elevation viewpoint (refer to Figure 13).

We compute the curved surface as follows. First, for a given view, we curve the hypothetical base plane of the original terrain (at sea level) along the view direction into a curved base plane via a power function $y = a * x^2$. The arc length of the curved plane in the direction of view is kept the same as the distance between the nearest and furthest points

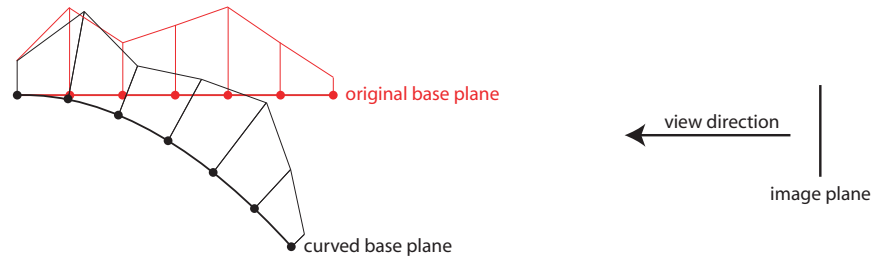


Fig. 13. When the base of the terrain is curved, the image will include both non-obscured features in the foreground, as well as horizon and sky in the distance.

on the hypothetical base plane of the terrain, d , in the direction of view, to maintain the original (u, v) parametrization. The scalar value a is determined by $a = b/d$, where b is the desired vertical offset. x varies from $[0 - 1]$ and corresponds to the normalized value of the distance along the line of view from the edge of the terrain to the point on the grid we are currently curving.

To find the new 3D curved position for each heightfield grid point, we first find the 3D position on the curved base surface that corresponds to the same (u, v) . The new position is then an offset $h = f(u, v)$ in the normal direction to the curved plane at the (u, v) location. The new offset surface places all the geometry within view without self-intersection or obvious foreground occlusion.

Modifying the base plane of the geometry requires computing new surface tangents (as the mapping from u, v to 3D has changed) and updating the gradients and normals accordingly. As pointed out previously, the only requirement is that the surface is still a parametric surface of two variables; the mapping from (u, v) to points in cartesian space is much more complicated now, but the fall lines are still expressed in the same manner.

In Figure 14 we compare leaving the geometry as is, geometry curving, and moving the view point to a higher elevation. Since rotating the geometry is equivalent to rotating the view, we keep the view the same for all the comparisons. We simulate the view from higher elevation by instead rotating the geometry.

PRELIMINARY TERRAIN IMAGE

We use a raytracer to render an image of the triangulated surface model for a specified light position (single distant point light) and viewing direction. We treat the terrain surface as perfectly diffuse. If the pixel is in light ($\mathbf{N} \cdot \mathbf{L} > 0$), we use simple direct lighting as our resulting luminance value. Otherwise, we compute a 1 bounce diffuse path tracing solution with 25 samples to lighten up the otherwise perfectly black pixels. If we hit a triangle that is classified as water, we send reflection rays to determine the values of the reflected point. Because of the curved geometry, we find that we need to lower the reflected ray (by increasing the angle between the reflected ray and the normal by 10-15 degrees) so that the reflection values match our visual expectations.

For each pixel in the rendered image, we store six variables: luminance, normal, 3D gradient projected into 2D screen space, depth, classification type and the light/shade mask. This results in 33 bytes per pixel assuming single precision floating point values.

CLASSIFICATION BOUNDARIES

As a result of the rendering projection, the boundaries of the classification types in image

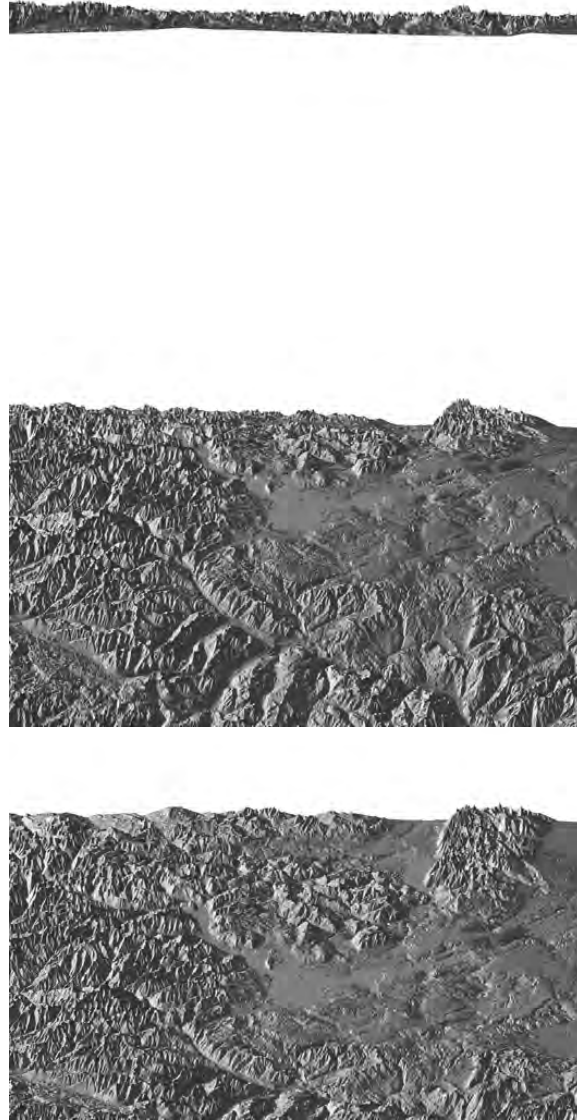


Fig. 14. Comparison of the same terrain rendered from the same viewpoint and camera parameters, but with different geometry modifications: leaving the geometry as is (top), our geometry curving method (middle), rotating the geometry (instead of rotating the view) to show corresponding view from above (bottom). Geometry curving preserves the 3D properties of the geometry, while allowing us to see both the horizon and the terrain features of interest, that would have otherwise been lost due to occlusion.

space are not smooth, especially if the resolution of the terrain is lower than the resolution of the final rendering. We find that smoothing these boundaries improves the visual quality of the final rendered image.

We smooth the boundaries by treating each pair of bordering classification types independently. We assign 0.0 to the first type, 1.0 to the second, and then run an averaging filter. All pixels with value below 0.5 get assigned to the first classification type. All pixels with value above 0.5 get assigned to the second classification type. We repeat until all pairs of bordering classifications are smooth.

4.2 Base Shading Stage

In our analysis, we concluded that the shading tone should have an even distribution of light, medium, and dark values. This step ensures that we extend the luminance value ranges in a manner independent of the brightness of the light source or the original image statistics of the raytraced image. In effect we abstract the terrain shading so it has the minimum amount of detail, while we retain important geometric and shape information. Refer to Figure 15 for a rendering of Yellowstone National Park using only Berann base colors.



Fig. 15. In this rendering of Yellowstone National Park, textures are using only the base Berann colors.

Based on this convention, we now describe a method for terrain base shading. First, we label individual pixels as being one of three tonal types - Light, Medium, or Dark. Our algorithm relies on luminance, classification type, and light/shade mask. To determine the light/medium/dark choice, we take all pixels for a given classification and mask types, and compute a three-bin histogram equalization. This step ensures that we have an equal distribution of light, medium and dark tones.

Then, we compute the color for each pixel in the scene. Per classification type, the user specifies 12 base shading palette colors corresponding to the combinations of (Light / Medium / Dark Tones) x (In Shade / In Light Masks) x (Near / Far Variations). For a tonal type and mask type, we blend between the Near color and the Far color given by the

palette. The blend factor is the fraction computed by dividing the depth stored in the current pixel by the maximum depth of the terrain for the given viewpoint.

The blend factor also determines how much aerial effects should be added to the pixel. Colors in the far distance have an added bluish cast and are further desaturated to match the effect of Aerial Perspective. Finally, we composite a sky image underneath our base shading.

4.3 Fall Lines

All classification types in the panorama paintings but the grass type have additional textures applied on top of their base shading. How should such textures be created automatically?

As discussed in Section 3, Panorama Analysis, fall lines are an appropriate guiding direction for applying painterly strokes or for the placement of painterly textures. Figure 16 compares the placement of tree textures along fall lines to placing them via jittered sampling or random sampling. We note that fall lines are clearly superior at communicating geometric and perceptual properties.

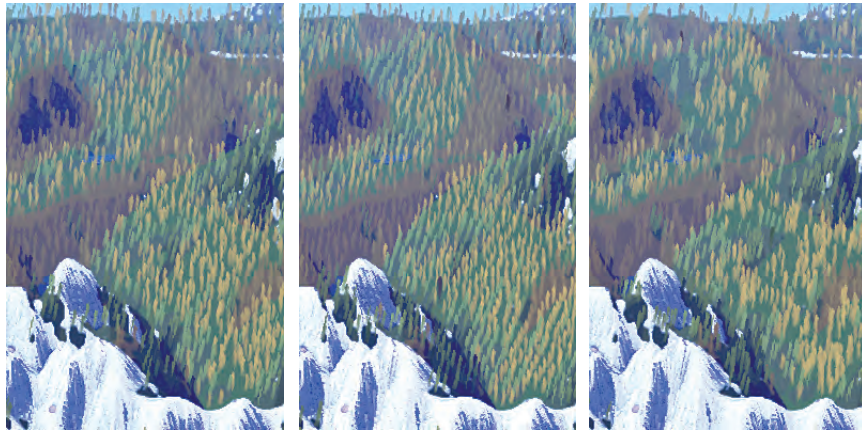


Fig. 16. Our method of tree distribution uses placement of trees along fall lines (center). As a comparison, you can see the same region with trees placed using jittered sampling (left) and random sampling (right).

Since not all classification types require surface textures (e.g. grass), and since we would like to give the user control over the length of the texture strokes, we build fall line guided paths for each classification type independently.

For each pixel in the image, p_i , we store the image-space negative gradient of the visible geometry, $v_i = -\nabla f$. From this discrete vector field V , where $v_i \in V$, we need to reconstruct a set C of smooth image-space curves that follow these gradients (the fall lines projected in image-space). $c_i \in C$, is the path starting at pixel p_i , with desired length l , and has the same classification type as p_i .

Each curve is the solution to the initial value problem, specified as follows:

$$\begin{aligned} \frac{dc_i}{dt} &= g(t, c_i(t)) = -\nabla f \\ c_i(t_0) &= p_i. \end{aligned}$$

To compute c_i we must solve this differential equation, and can do so with any method. In our implementation, we use a 4th order Runge-Kutta method for its accuracy and stability.

After building all the paths for our image, we are likely to have far too many paths. More importantly, many of these paths will converge and diverge so that many paths go through the same pixels. These are unnecessary as only one final set of strokes can actually be drawn within a single pixel. To prune the resulting paths, we iterate over all pixels that contain more than one path and only allow the longest one to continue. The other paths are clipped to stop before the current pixel. This keeps longer, flowing paths instead of many short paths (see Figure 17).

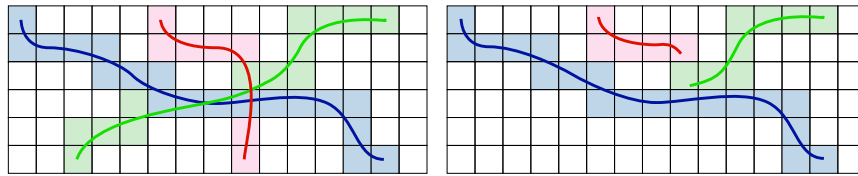


Fig. 17. Paths before (left) and after (right) pruning.

Because of numerical precision issues with projecting into integer pixels, paths may go through the same pixel twice. We detect these loop cases and trim the paths accordingly. Once done with this step, we have unique paths that run through every pixel in each classification region.

4.4 Surface Textures Stage

We have painted all the base colors. Now we need to place the visual elements in order to enhance the panorama perceptually and stylistically. Each classification type is handled individually, as the visual elements are different. To examine the effect of stroke textures, compare Figure 15, where we only have base shading, to Figure 1.

FOREST TEXTURES

Since forests are 3D textures formed by fairly regular 3D texture elements, we try to maximize their perceptual effects by following the rules we developed about image space texel size, texel distributions, and texel overlap (see Section 3.2). We make their effects even stronger, by combining the texture cues with surface contour directions, forming forest meta-strokes.

As mentioned in Section 3, we would like to place our tree strokes so that they are reasonably spaced, follow the fall lines of the terrain, and give the illusion of occlusion. We choose to place our trees by first computing "seed" positions in image space by walking along the fall lines by an amount determined by the slope of the surface projected into image space. Projecting the slope values is a useful approximation of texel spacing foreshortening. To keep tree spacing reasonable, we also mark pixels surrounding a chosen seed point as unavailable (see Algorithm 1). We use the longest paths first to ensure that the most important fall lines are most visible.

Once all the seed positions have been chosen, we paint the actual tree strokes in a "back-to-front" order. Using back-to-front ordering provides the illusion of 3D occlusion. Our tree brushes are defined as alpha values and we pick the per pixel brush color by multiplying each of the brush alpha values by the desired color.

Algorithm 1 *PlaceTreeSeeds* (*paths*, *treeSpacing*, *radius*)

```

mark pixels ∈ FOREST as AVAILABLE
for all paths ∈ FOREST do
  for all currPath ∈ paths, starting with max length do
    length = 0
    while length < max path length of currPath do
      pixel = currPath[length]
      if CircleAvailable (pixel) then
        mark pixel as SEED
        mark circle of radius pixels around pixel as USED
      else
        length += 1
        continue
      stride = treeSpacing
      stride *= Dot (-viewDir, surfNormal[pixel])
      length += stride

```

The desired stroke color is determined by the (Light / Medium / Dark Tones) x (In Shade / In Light Masks) x (Near / Far Variations) combination. This is very similar to what we did for the Base Shading Stage, except now the chosen color corresponds to surface brushstrokes, not background. To add additional variety to the colors (as Berann does in his strokes), we stochastically choose a blending factor (between 0% and 30% percent) and use it to blend between the chosen color and the next darkest tree color. This forms the final color for each tree stroke.

To allow the user control of the perspective scaling of brushes, we provide a single scale factor that describes how small texels in the farthest distance should be relative to the original brush size. At points in between the nearest point and the farthest point, we use the relative distance ($\text{dist}/\text{maxdist}$) to scale the user parameter.

CLIFF AND SNOW TEXTURES

Surface textures for cliff and snow enhance the perceived terrain structure. As discussed in Section 3, fall lines follow the essential structure of terrain. Therefore, using the fall line paths as imaginary lines along which brush strokes for cliff and snow are placed will make that geometry more structurally expressive.

The application of strokes is made independently in each of the classification regions (snow and cliff). The user has control over four parameters - *stroke color*, *stroke width*, *stroke overlap*, and *stroke image definition*.

Similarly to forest strokes, the user provides 12 *stroke colors* per brushstroke type (cliff or snow). Again, the tonal classification and the mask type indicate the appropriate color for Near and Far. The two palette colors are blended using a weight related to the relative depth of the pixel we are shading, to form the selected color. Color variety is incorporated by blending the selected color and its closest darker color (here blend weights vary from 0% to 50%). The most significant difference for these strokes is that the color selection is made based on the luminance segmentation for the starting point (origin) of the path, and not on the current position along the path, as the path's tonal classification can change. This is necessary to maintain the visual integrity of individual strokes.

Stroke overlap and *stroke width* determine the visual density of strokes. Overlap specifies the number of pixels we allow strokes to overlap on. Width is randomly selected, per stroke, from the set of widths specified by the user. The less dense the strokes are, the more the background will come through. We use the width and overlap settings to resize the stroke footprint for each point along the stroke path. If the current stroke footprint is still unoccupied in the image, we mark it as occupied, and indicate that we should draw part of the stroke there. If it is already occupied, we move on to the next path.

The *stroke image definitions* are similar to Hsu et al.'s skeletal strokes [1993]. For each stroke, we read the brush stroke image definition and resample it to match the width of our desired stroke and the length of our currently selected path. We then walk along each pixel that makes the current path. When the footprint is marked as available for drawing we alpha blend the selected brushstroke color with the base shading, based on the weights provided in the brushstroke image.

WATER

Water surfaces reflect light from its surroundings, providing the observer with a distorted and muted image of the terrain as well as the sky. Such reflections undoubtedly contribute to the appeal and natural appearance of the water body. Water reflections were also painted in both panorama maps we analyzed in Section 3.

In order to simulate surface variation, we adopt Berann's approach of adding strokes horizontal to the image plane (refer to section 3.3). We render textures for water surfaces by generating strokes of varying lengths in the horizontal direction. Based on the luminance of the base water color, we pick a stroke color that is a blend between the lightest and darkest color of the sky. To add some variation to the brush appearance, for each brush, we modify the luminance value of the brush pixels by 10%. Then, as we did for the other textures, we alpha-blend the brush with the background computed in the Base Shading stage, resulting in water strokes.

5. RESULTS

As evidence that our proposed algorithms are successful at producing panorama maps reminiscent renderings, we render two panorama maps of Yellowstone National Park. The first is in the style of Heinrich Berann (see Figure 1), the second in the style of James Niehues (see Figure 18). The renderings allow us to do a direct comparison with the original work. Examine the similarities between our renderings and the actual paintings (see Figure 2 for the original paintings).

For the Berann-style rendering of Yellowstone, we used Berann-style tree strokes and samples of his colors for near and far distance (refer to Figure 10). Similarly, for the Niehues-style rendering, we used Niehues-style tree strokes and color samples. Cliff and snow strokes used are shown in Figure 19. Stroke length is set to 50 pixels for tree strokes and 70 pixels for cliff and snow. Stroke width for snow and cliff strokes for both images varies between 3 and 6 pixels. In the application of the tree strokes, we pad the shape by 3 pixels, to control spacing.

We render our images on an Intel Mac Pro 3.0 GHz with 8GB of RAM. The image size matches the physical dimensions of the original paintings, assuming 150 dpi. For our chosen image size, rendering consumes from 1.5 to 2.4 GB RAM. For rendering statistics, refer to Table I.

To show the applicability of our algorithms to different mountainous terrains, we create a



Fig. 18. Yellowstone National Park, rendered with rock, snow, water, and two kinds of tree textures with Niehues-style strokes and colors.



Fig. 19. Cliff strokes used by our system for (a) Yellowstone (b) Rocky Mountain National Park.

Panorama Map	Image Dimensions	Datset Dimensions	Rendering (Path Tracing)	Texture Generation	Total Rendering Time
Yellowstone (Berann Style)	5284 x 3587	1889 x 1880	23.65 min	4.83 min	28.48 min
Yellowstone (Niehues Style)	5284 x 3587	1889 x 1880	24.03 min	4.95 min	28.98 min
Rocky Mountain NP	5284 x 3699	927 x 767	12.05 min	4.26 min	16.31 min
White Sands NM	6400 x 3050	661 x 657	5.73 min	4.98 min	10.7 min
Mt Rainier	6400 x 3054	3470 x 1470	9.53 min	3.95 min	13.48 min

Table I. Rendering Statistics for Our Example Panoramas

panorama map rendering of Rocky Mountain National Park, Colorado. We view the terrain from a distance, but with a narrow field-of-view, to reduce the perspective distortion.

The tree strokes are based on the shapes and dimensions of the strokes we sampled from the Berann Yellowstone Panorama (refer to Figure 11). The cliff and snow strokes we use are shown in Figure 19. Each stroke is assigned randomly from one of the four available shapes. Stroke length is set to 50 for tree strokes, and 70 for cliff strokes. Stroke width for

snow and cliff strokes for all images again varies between 3 and 6 pixels. We control tree spacing using a 4 pixel stride.

Since we do not have a reference panorama map to compare to, and we do not have colors that have been manually selected by artists, we lazily select the 12 color palette for base and stroke colors for each classification type from images of the region. Unfortunately, this approach does not produce a very attractive panorama (see Figure 20, top). Despite the fact that our colors are sampled from real images, corresponding to the same terrain, the colors of the rendered image are not acceptable. Guessing the effects of tweaking the base and stroke colors on the final rendering, in a predictable manner, is a difficult task, aggravated by the effects of simultaneous contrast.

A satisfactory solution to this problem can be achieved by performing a color transfer of image statistics from a desirable image to the rendered panorama, using Reinhard et al. [2001]. The result of the color transfer, using our original rendering as source and Berann's panorama map of Yellowstone as color target, can be seen in Figure 20 (bottom).

To demonstrate the applicability of our algorithm to a different type of terrain, we generate an example panorama map of White Sands National Monument (refer to Figure 21). We treat sand as we would treat cliffs. We view the terrain from the east, again from a distance, and with a narrow field-of-view. We reuse the Berann strokes (refer to Figure 11), as well as the cliff and snow strokes used for our rendering of Rocky Mountain NP. Stroke length is set to 50 for both tree, sand, and cliff strokes. Tree spacing is set to 4 pixels. Once again, we pick our 12 color palette for base and stroke colors for each classification type from real photographs of the region. We didn't have to do anything special to produce this panorama, except for categorize the sand as snow (so that stroke handling for snow is applied to regions where there is sand). Our rendering accentuates the mountains in the background and retains the appearance of the sand dunes.

Finally, we produce a rendering of the area surrounding Mount Rainier and Mount St. Helens (see Figure 22). We view the terrain from the west, in order to see the Pacific Ocean and Columbia River with the mountains as a backdrop. We reuse Berann strokes (refer to Figure 11) for cliffs and snow, and Niehues' strokes for trees. Stroke length is set to 50 for both tree, sand, and cliff strokes. Tree spacing is set to 4 pixels. We mix cliff and snow colors from the Berann palette with tree and water colors from Niehues.

6. CONCLUSION

We have demonstrated a novel method for landscape visualization applicable to mountainous terrain. Our main contributions are: an analysis of the functional and stylistic role played by visual elements in panorama maps and an automatic technique that allows us to generate images whose visual style is reminiscent of those in panorama maps of mountainous terrain. We have chosen to operate mainly in image space because that is the natural space in which one generates painterly strokes. The downside of using image space is that an animated change of viewpoint is difficult. This is an example of the classic tradeoff between image quality and frame coherence seen in almost all NPR work.

The most obvious limitation of our work is that we have mostly concentrated on the rendering, and only partially on the geometry manipulation (and mostly for visibility purposes) of panorama maps. We have not modified the geometric information itself to make it clear and easier to understand, and we have not proposed any ways in which we can do geometric feature enhancements.

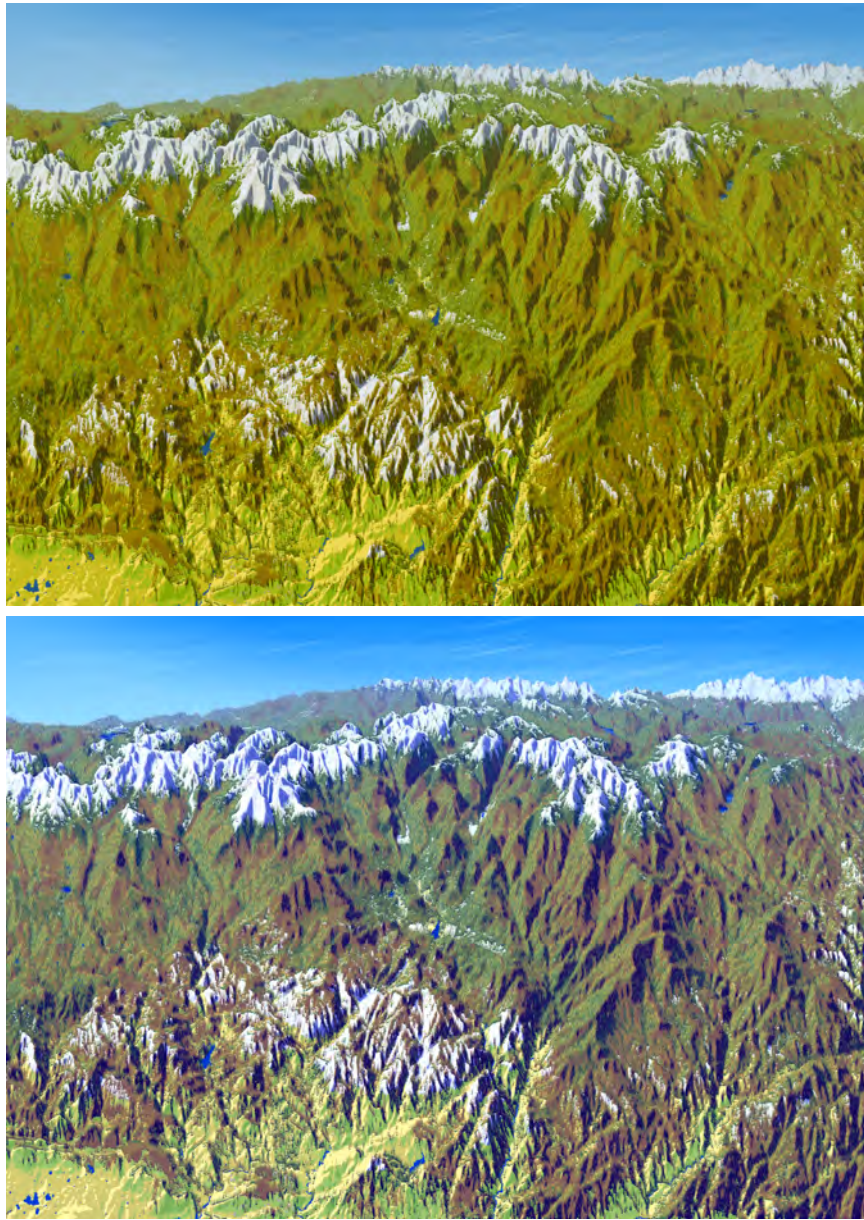


Fig. 20. Rocky Mountain National Park terrain, rendered with rock, snow, water, and two kinds of tree textures. Rendered with colors from photographs of the terrain (top) and with a color transfer (bottom).

In the future, we would like to improve the speed of our algorithms, specifically the path tracing part. We anticipate that making our system parallel and utilizing an irradiance cache would speed up our system drastically. In addition, we would also like to address modifications that might be necessary when extending the rendering to different types of terrain or when adding cultural features.

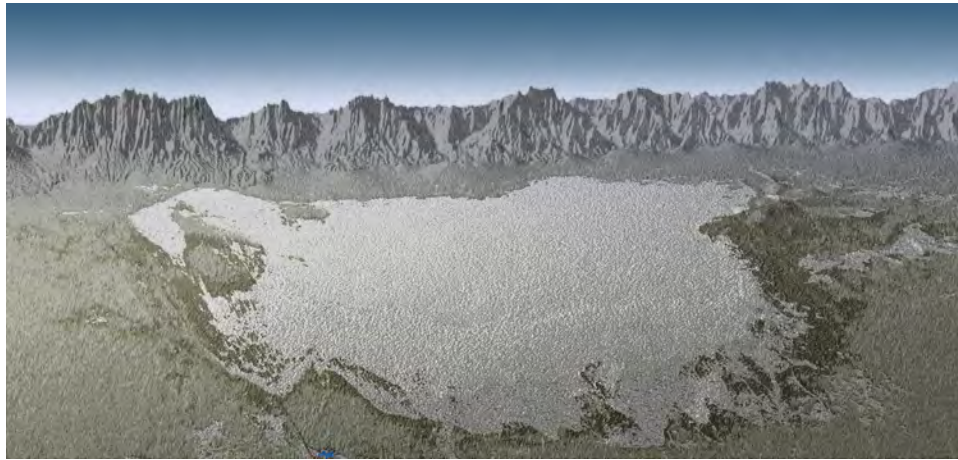


Fig. 21. White Sands National Monument, rendered with brush tree, sand, and grass texture. Colors were selected from real images of the terrain.



Fig. 22. Mount Rainier and Mount St Helens, rendered with tree, snow, grass, and water texture. Colors were selected from a combination of Berann and Niehues colors.

Acknowledgements

This work was supported by NSF grant 03-12479. We thank Solomon Boulos for constructive comments and help with mathematical derivations. We thank Simon Premože for useful discussions. We thank the Trail Tracks company (www.trailtracks.com) for allowing us to use images of the map by James Niehues. We thank Bill Martin for sharing his LIC code.

REFERENCES

- ALOIMONOS, J. 1988. Shape from texture. *Biological Cybernetics* 58, 345–360.
- BOARD, C. 1990. Report on the working group on cartographic definitions. *Cartographic Journal* 29, 65–69.
- ACM Transactions on Graphics, Vol. V, No. N, Month 20YY.

- BUCHIN, K., SOUSA, M. C., DOLLNER, J., SAMAVATI, F., AND WALTHER, M. 2004. Illustrating terrains using direction of slope and lighting. In *4th ICA Mountain Cartography Workshop*.
- CABRAL, B. AND LEEDOM, L. C. 1993. Imaging vector fields using line integral convolution. In *SIGGRAPH '93: Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*. ACM Press, 263–270.
- DA VINCI, L. 2003. *The Notebooks of Leonardo Da Vinci*. Konecky, William S. Associates, Inc.
- DECARLO, D. AND SANTELLA, A. 2002. Stylization and abstraction of photographs. In *SIGGRAPH '02: Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*. ACM Press, 769–776.
- DEUSSEN, O., HAMEL, J., RAAB, A., SCHLECHTWEG, S., AND STROTHOTTE, T. 1999. An illustration technique using hardware-based intersections and skeletons. In *Proceedings of the 1999 Conference on Graphics Interface*. Morgan Kaufmann Publishers Inc., 175–182.
- DORLING, D. AND FAIRBAIRN, D. 1997. Mapping ways of representing the world. In *Insights into Human Geography*. Addison Wesley Longman.
- ELBER, G. 1998. Line art illustrations of parametric and implicit forms. *IEEE Transactions on Visualization and Computer Graphics* 4, 1, 71–81.
- GARDING, J. 1993. Shape from texture and contour by weak isotropy. *Journal of Artificial Intelligence* 64, 243–297.
- GIBSON, J. J. 1950. *The Perception of the Visual World*. Riverside Press, Cambridge, MA.
- GOOCH, A., GOOCH, B., SHIRLEY, P., AND COHEN, E. 1998. A non-photorealistic lighting model for automatic technical illustration. *SIGGRAPH '98: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, 447–452.
- HAEBERLING, C. 2004. Selected design aspects and graphic variables for 3d mountain maps. In *Proceedings, 4th Mountain Cartography Workshop, Vall de Nuria, 30th September - 2nd October 2004*. Vol. Monografies tecniques, Num. 8. 109–118.
- HERTZMANN, A. 1998. Painterly rendering with curved brush strokes of multiple sizes. In *SIGGRAPH '98: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*. ACM Press, 453–460.
- HERTZMANN, A. AND ZORIN, D. 2000. Illustrating smooth surfaces. *SIGGRAPH '00: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*.
- HORN, B. 1981. Hill shading and the reflectance map. *Proc. IEEE* 23, 1.
- HSU, S. C., LEE, I. H. H., AND WISEMAN, N. E. 1993. Skeletal strokes. In *UIST '93: Proceedings of the 6th Annual ACM Symposium on User Interface Software and Technology*. ACM Press, 197–206.
- IMHOF, E. 1963. *Zurich, Vorhof der Alpen*. Chapter Zurcher Kartenkustler und Panoramazeichner, 105–138.
- IMHOF, E. 1982. *Cartographic Relief Presentation*. Walter de Gruyter, Berlin.
- INTERRANTE, V. 1997. Illustrating surface shape in volume data via principal direction-driven 3d line integral convolution. In *SIGGRAPH '97: Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*. ACM Press, 109–116.
- KNILL, D. 1992. Perception of surface contours and surface shape: From computation to psychophysics. *JOSA A* 9, 9.
- KOENDERINK, J. J. AND VAN DOORN, A. J. 1998. The structure of relief. *Advance in Imaging and Electron Physics* 103, 65–150.
- LESAGE, P. L. AND VISVALINGAM, M. 2002. Towards sketch-based exploration of terrain. *Computers & Graphics* 26, 309–328.
- LEUNG, T. AND MALIK, J. 1997. On perpendicular texture or: Why do we see more flowers in the distance? In *IEEE Conference on Computer Vision and Pattern Recognition*. 807–813.
- MALIK, J. AND ROSENHOLTZ, R. 1997. Computing local surface orientation and shape from texture for curved surfaces. *IJCV* 23, 2, 149–168.
- MARR, D. A. 1982. *Vision*. W. H. Freeman, San Francisco.
- PALMER, S. 1999. *Vision Science – Photons to Phenomenology*. Cambridge, MA: MIT Press.
- 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 (Spring).

- PATTERSON, T. 2005. Looking closer: A guide to making bird's-eye views of national park service cultural and historical sites. *Cartographic Perspectives* 52.
- PREMOZE, S. 2002. Computer generation of panorama maps. In *3rd ICA Mountain Cartography Workshop*.
- REINHARD, E., ASHIKHMIN, M., GOOCH, B., AND SHIRLEY, P. 2001. Color transfer between images. *Computer Graphics and Applications*, 2–9.
- ROSENHOLTZ, R. AND MALIK, J. 1997. Surface orientation from texture: Isotropy or homogeneity (or both)? *Vision Research* 37, 16, 2283–2293.
- RUSINKIEWICZ, S., BURNS, M., AND DECARLO, D. 2006. Exaggerated shading for depicting shape and detail. In *SIGGRAPH '06: Proceedings of the 33th Annual Conference on Computer Graphics and Interactive Techniques*.
- SAITO, T. AND TAKAHASHI, T. 1990. Comprehensible rendering of 3-d shapes. In *SIGGRAPH '90: Proceedings of the 17th Annual Conference on Computer Graphics and Interactive Techniques*. Vol. 24. 197–206.
- SALISBURY, M., WONG, M., HUGHES, J., AND SALESIN, D. 1997. Orientable textures for image-based pen-and-ink illustration. In *SIGGRAPH '97: Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*. ACM Press, 401–406.
- SAUNDERS, J. A. 2003. The effect of texture relief on perception of slant from texture. *Perception* 32, 211–233.
- STEVENS, K. 1979. Surface perception from local analysis of texture and contour. Ph.D. thesis, MIT.
- STEVENS, K. 1981. The visual interpretation of surface contours. *AI* 17, 47–73.
- SULLIVAN, C. 1997. *Drawing the Landscape*. John Wiley and Sons, Inc.
- VISVALINGAM, M. AND DOWSON, K. 1998. Algorithms for sketching surfaces. *Computers & Graphics* 9, 213–228.
- WHELAN, J. AND VISVALINGAM, M. 2003. Formulated silhouettes for sketching terrain. In *Theory and Practice of Computer Graphics*. IEEE, 90.
- WITKIN, A. P. 1981. Recovering surface shape and orientation from texture. *Journal of Artificial Intelligence* 17, 17–45.

Received January 2007; Revised April 2008; Revised February 2009; Accepted April 2009;