Expressive Rendering of Mountainous Terrain UUCS-07-001

MARGARITA BRATKOVA
University of Utah
and
PETER SHIRLEY
University of Utah
and
WILLIAM B. THOMPSON
University of Utah

Painters and cartographers have developed artistic landscape rendering techniques for centuries. Such renderings can visualize complex three-dimensional landscapes in a pleasing and understandable way. In this work we examine a particular type of artistic depiction, panorama maps, in terms of function and style, and we develop methods to automatically generate panorama map reminiscent renderings from GIS data. In particular, we develop image-based procedural surface textures for mountainous terrain. Our methods use the structural information present in the terrain and are developed with perceptual metrics and artistic considerations in mind.

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

Image understanding can be aided by methods that reduce the perceptual and cognitive efforts required in human image interpretation [Tufte 1990; Zeki 1999; Santella and DeCarlo 2004], such as omission of extraneous detail, abstraction of elements, or emphasis on features of importance. Visual artists have developed specific techniques for many types of scenes. The one of interest here is the landscape, in particular, the *panorama map*, which is a bird's eye view painting of a landscape and is commonly used to present terrain to non-expert viewers [Imhof 1963; Board 1990; Dorling and Fairbairn 1997].

In this work we examine the panorama map in terms of function and style, and we derive heuristics and principles appropriate for landscape visualization. Based on these, we develop algorithms to automatically generate images with a visual style reminiscent of

Fig. 1. A rendering produced by our system for Yellowstone National Park, using Berann-style brushstrokes and colors.

the panorama map. The renderings incorporate surface textures that model natural surfaces. Specifically, we produce renderings of mountainous terrain that have rocky areas, forests, snow, and lakes. Figure 1 is a representative example. Our algorithms are image based, and are influenced by the structural information present in the terrain.

2. BACKGROUND

2.1 Non-Photorealistic Rendering

Non-photorealistic rendering (NPR) has been inspired by artistic techniques and methods of representation [Gooch and Gooch 2001; Strothotte and Schlechtweg 2002]. NPR work often develops algorithms that generate images of a stylized nature. Usually, synthesized strokes of some kind are utilized, and they often play a defining role in the overall rendering style (e.g. pen-and-ink drawing, sketch, or oil painting).

Strokes need to follow some visually meaningful direction, and they can be applied in image space or object space. Image space approaches often use an existing image as input. The image is processed so that some meaningful quantities are produced and then used to guide the rendering, e.g. the normals of the image gradients are computed from the image and then used as input [Hertzmann 1998], or feature edges are detected via edge detection and then used to control stroke placement. Some researchers manually specify the necessary vector fields [Salisbury et al. 1997], or they 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 might not always be correctly computed or even available and that can cause problems in the rendering. If a

3D model of the scene is available, then geometric information from the model can be used to guide the stroke direction and placement, e.g. lines of curvature (the 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 (lines of constant brightness) [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.

2.2 Digital Terrain Rendering

LINE-DRAWN SKETCHES
Algorithmic landscape sketching of terrain was first introduced by Visvalingam and Dowson [Visvalingam and Dowson 1998]. Their goal was to create sketches that resemble artistic landscape drawings. Their algorithm computes a set of surface points that are part of the terrain profile curves perpendicular and orthogonal to the line of sight. A filtered set of the points 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 [Whelan and Visvalingam 2003]. These are silhouettes that would be brought into view if a higher viewpoint is to be selected. A different technique for terrain sketching was introduced by Lesage and Visvalingam [2002]. Here, the algorithm computes terrain luminance maps and then extracts sketch lines using four common edge detectors. Most recently, Buchin et al.[2004] proposed a line sketching technique incorporating slope lines and tonal variations.

LANDSCAPE PAINTINGS
Panorama map paintings, also called bird's-eye-view maps, are artistic cartographic renderings of extensive geographic areas (see Figure 3). Panorama maps are significantly different from images produced by GIS or by photographs. 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].

Saito and Takahashi [Saito and Takahashi 1990] developed methods for the visualization of topographical maps and provided an example of a bird's eye view map by combining 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.

An interactive tool for panorama map creation was proposed by Premože [Premoze 2002]. The editor 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 user to modify the geographical information in ways that improve its perceptual legibility, and is not automatic.

Panorama maps enhance the clarity of the spatial information via geometric manipulation, geometric feature enhancements, and perceptual and artistic textures. 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 land better. To our knowledge, there are no existing algorithms that allow us to produce such paintings

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

automatically.

2.3 Visual Perception and Pictorial Cues

Maps need to visually convey a sense of the spatial structure of the terrain being represented and the spatial context of natural and cultural features located on the terrain. In perceptual terms, this means conveying information about distances and surface orientations (e.g., see Gibson [Gibson 1950]).

When considering visual perception, distance information is often subdivided into *depth*, which is the distance between the viewer and a visible location in the environment, and *exocentric distance*, which 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. Panorama maps are intended to create a sense of directly viewing a three-dimensional terrain and so are dominated by perceptual information for depth 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 2 lists the pictorial cues most relevant to the perception of depth and surface orientation.

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 [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.

Fig. 3. Panoramas of Yellowstone by Heinrich Berann (left) and James Niehues (right). *(All Berann images used are in the public domain. All Niehues images used by permission.)*

3. PANORAMA ANALYSIS

In this secton we analyze the visual elements that affect the perception of panorama paintings. We believe the visual elements serve two roles: functional and stylistic. Functionally, the choice made in the expression of visual elements helps the visual system perceive surface shape and orientation, thus creating the illusion of three-dimensions. Stylistically, it allows the artist and the painting to be subjectively pleasing.

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 examine functionally meaningful pictorial cues, as suggested by cue theory. We support our analysis, when applicable, by providing examples from two Yellowstone National Park panoramas, painted by two stylistically very different, yet highly sophisticated panorama artists: Heinrich Berann and James Niehues (Figure 3). Second, we examine how these two maps differ stylistically.

3.1 Functional Analysis

Texture Gradients

Panorama maps 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 algorithms make assumptions about the 3D surface distribution of the texels comprising the textures. Homogeneity assumes nearly constant texture element surface density [Gibson 1950; Aloimonos 1988; Super and Bovik 1995]. Isotropy, a weaker assumption, presumes that the texture does not have a dominant direction or an orientation bias [Witkin 1981; Garding 1993]. Perceptual studies suggest that the visual system uses both assumptions [Rosenholtz and Malik 1997]. However, since isotropy cannot deal with directional textures (such as 3D forest textures), we use the stronger assumption of homogeneous texel density [Malik and Rosenholtz 1997], which is a fairly reasonable assumption when we deal with natural textures.

Shape-from-texture theory uses the texture distortions caused by projection to help recover surface shape and orientation. Such projective distortions will affect the *texel shape* (compression or foreshortening), the *texel size* (perspective scaling), and the *texel density*. In addition to these, 3D textures also introduce *texel occlusion*. We now examine these four cues and develop guidelines we believe are specifically applicable to the visualization of forested textures. Our goal is generate forest textures that provide as much functional 3D information in terms of perceptual cues as possible.

The effects of *texel shape* foreshortening for 3D textures has been mostly unexplored in the shape-from-texture literature. We believe it is very likely that, when it comes to trees, it does not serve a useful purpose, because the vertical directionality of the tree, as well as its amorphous form, make a foreshortening cue difficult to delineate.

Cue theory suggests that *texel size* should vary with distance, as would be dictated by true perspective projection. 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, and should guarantee that texture elements present even in the far distance are discernible in the final rendering. Both example panoramas support this (Figure 4).

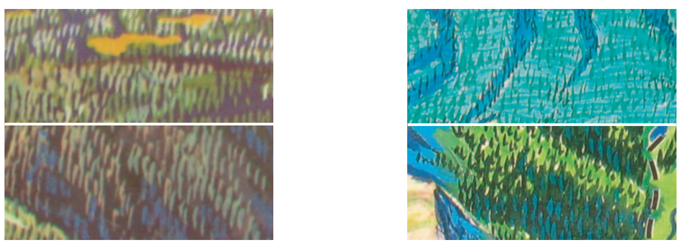

Fig. 4. Trees do not vary with true perspective: trees in the far distance (top), trees in the foreground (bottom) (from panoramas of Yellowstone by Heinrich Berann (left) and James Niehues (right)).

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 show 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 [Malik and Rosenholtz 1997]. If we are to benefit from visual information present due to the projection distortion, we should make sure 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] will occur. The occluded element is visually perceived as positioned behind the occluding element and that 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 do use occlusion in their depiction of forests (see Figure 5).

Fig. 5. Occlusion used in forest textures (from panoramas of Yellowstone by Heinrich Berann (left) and James Niehues (right)).

SURFACE CONTOURS

Surface contours were first introduced by Stevens [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. Stevens suggests surface contours help us perceive shape, because the visual system attributes a strict geometric relationship between the surface contours and the curvature of the 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].

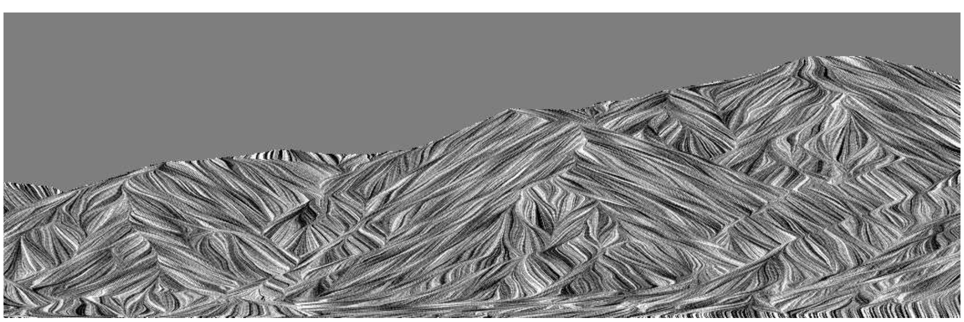

Fig. 6. Image with flow along fall lines.

Considering how vital fall lines are to relief, it is not surprising that panorama artists seem to consistently incorporate them in their paintings by using the direction of the fall lines as a guide for the stroke directions used to paint cliffs and other mountainous forms (see Figure 7). In addition, they also use them as imaginary lines along which to place tree strokes, spacing them according to slope (see Figure 7). We call these *meta-strokes*. We believe that in doing so, artists combine surface contours with texture cues and that provides as much perceptually relevant information as possible.

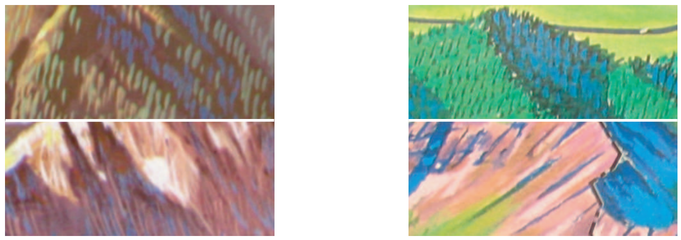

Fig. 7. Trees are spaced such that they follow the direction of imaginary fall lines (top). Cliffs and snow are painted with strokes that follow the direction of fall lines (bottom) (from panoramas of Yellowstone by Heinrich Berann (left) and James Niehues (right)).

Shading

Shading occurs as a result of illuminating surfaces that vary in their surface orientation. While it is known that shape-from-shading is an ill-posed problem, in that the same image can be produced by a family of surfaces [Brooks 1983], it is an effective and very useful depth cue for people [Palmer 1999]. And indeed, for centuries artists have used light and shade to create impressive pictorial representations of three-dimensional shapes.

Shading is vital in depicting terrain surface form [Horn 1981]. Imhof emphasizes the importance of using midtone to help the light and dark areas express the form of the terrain [Imhof 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 will help the viewer perceive the form better [Sullivan 1997; Da Vinci 2003].

Fig. 8. Use of lighting direction in panorama maps (from panoramas of Yellowstone by Heinrich Berann (left) and James Niehues (right)).

Lighting direction plays an important role for the successful landscape visualization. Imhof [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]. We can see that panorama artists follow these rules (see Figure 8).

Silhouettes

Silhouettes are the contours formed at the boundary between an object and its background.

They are perceptually useful, because they indicate depth edges. Such edges result from an image-based spatial discontinuity and signal distance between the surfaces separated by the silhouette curve [Palmer 1999]. Silhouette edges are especially appropriate for expansive mountainous terrain, where the landscape is such that there is an overlap of rolling hills as seen in the distance. The silhouette edges of the overlapping hills imply a varying degree of depth. It is therefore not surprising, that artists consistently use that cue to increase the portrayed depth of their paintings (see Figure 9).

Fig. 9. Silhouetted curves used in panorama maps (from panoramas of Yellowstone by Heinrich Berann (left) and James Niehues (right)).

3.2 Stylistic Exploration.

Here we explore some basic stylistic variations present in two different panorama maps visualizing the same landscape - Yellowstone National Park. The physical act of painting the landscapes is achieved by layering color shapes and brush elements that vary in color and size depending on the natural elements portrayed [Patterson 2000]. Therefore, we believe the most basic parameters for our stylistic analysis should be variations in the *base color*, *brushstroke color*, and *brushstroke shape*. The base color provides the basic shading of the terrain, and varies with classification type. The brushstroke color and brushstroke shape make up the texture elements. The interaction of the color of these basic elements affects our visual perception due to the effect of simultaneous contrast.

TREE TEXTURES

The brushstroke color of the tree textures in both paintings is very important (see Figure 11 (a-d)), as it changes with respect to the orientation of the surface towards the light. In Niehues case the change is mostly in value and is not as strong as Berann's, who also adds a variation in color. By providing a good distribution between light, mid, and dark tones in the trees, Berann achieves a strong articulation of the form, especially due to the appropriately chosen base color for the trees background. Some artists like Niehues paint evergreen trees in a distinctly different style than deciduous trees. Others like Berann do not seem to make such a distinction. For the most part, the brushstroke shapes of the trees are very simple, elongated, perhaps a little bit fatter at the bottom, and orientated in the vertical direction (see Figure 10).

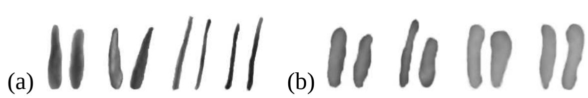

Fig. 10. Sample of the shapes of tree strokes used by (a) Niehues and (b) Berann (from panoramas of Yellowstone by Heinrich Berann (left) and James Niehues (right)).

CLIFF AND SNOW TEXTURES

The base color and brushstroke color of the *cliff and snow textures* also corresponds to surface orientation. Berann uses more natural tones than does Niehues (see Figure 11 (e-h)). Niehues uses brighter and less natural colors, and he paints the shade so as to indicate a strong blue shift, both in the foreground and the background. Berann's choice of colors for texture curves is more natural and diverse than the one used by Niehues. His cliff lines are portrayed in warm brown and the highlights in peach and yellow. Snow is mostly painted by the use of strokes, and the base color does not come through much. It is pure white only in light and for higher mountain ranges. In lower areas, it often has a yellow or a pink cast. In the shade, it is painted with a bluish cast.

In Berann's work, the shapes of the cliff strokes use various scales - small, long and distinct, or applied sweepingly and semi-transparently over large areas so as to to unify them. Niehues uses much simpler, uniform brushstroke lines. For the most part they seem to alternate between chosen colors, following the direction of fall lines.

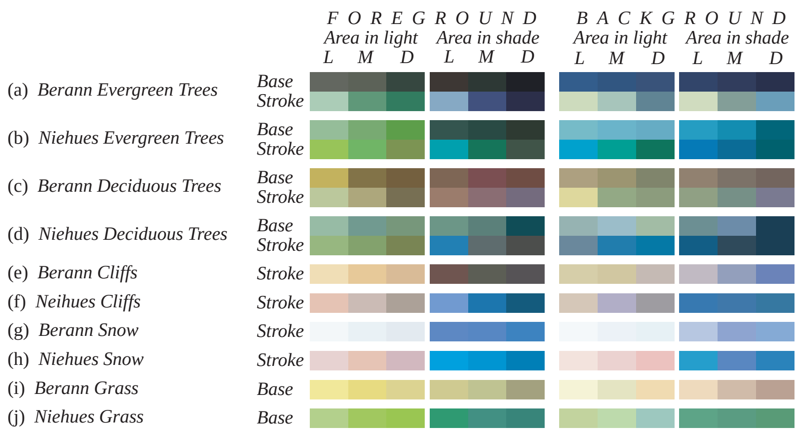

Fig. 11. Sample of light (L), medium (M) and dark (D) colors used by Niehues and Berann in their Yellowstone painting for areas in the foreground and the background.

GRASS AND WATER TEXTURES

For the most part, *grass textures* do not have strokes, just base color (see Figure 11 (i-j)), and the shading corresponds to surface orientation.

In Berann's work *water textures* are achieved by stokes that are horizontal to the image plane. The base color varies with viewing angle, and the brushstroke color even further emphasizes the viewing angle. In Niehues' work the base color varies little with viewing angle and the painted colors mimic the effect of physical water reflection.

3.3 Principles and Heuristics

While traditional cartography has developed various cartographic principles 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 bird's-eye-view maps are by necessity self-thought [Patterson 2005].

In the hope of reverse-engineering certain conventions used by panorama artists, we analyzed the visual elements of panoramic landscape paintings. Based on our functional and stylistic analysis of existing panorama map paintings, we have developed the following heuristics and principles:

(1) The image texel size of surface textures that represent 3D elements (e.g. forest) should vary with distance, but should not match true perspective.
(2) The image space distribution of texel elements of 3D textures (e.g. forest) should try to mimic one that would result from the projection of homogeneously distributed surface elements.
(3) Care needs to be taken when determining image space texel spacing of 3D textures, to ensure that texels overlap, especially in steep areas.
(4) For extended terrain areas, it might useful to indicate silhouettes, especially between occluding hills.
(5) Fall lines illustrate an essential structure of terrain. They act as surface contours, and are used by panorama artists to paint cliff and snow textures.
(6) Fall lines are used as imaginary lines along which tree strokes are placed, acting as texture *meta-strokes*.
(7) Light position should be chosen such that the image of the terrain exhibits a good balance of light and shade as seen from the selected viewpoint.
(8) Shading tone should have a good distribution of value - light, medium, and dark.

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 examine panorama map texture generation methods. This work does not address geometry manipulation or geometric features enhancement, though such techniques are very valuable, and are used by panorama artists [Patterson 2000]. Our algorithm proceeds in three stages: preliminary, base shading, and surface textures. See Figure 12 for an overview.

4.1 Preliminary Stage

Terrain Geometry and Classification

The geometry of the terrain is specified by a digital elevation model (DEM). The data is stored in a floating point format and is a height-field. We also read-in a classification image that corresponds to the terrain data and 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).

Geometry Exaggeration

Since panoramas depict terrain that covers large areas, locally significant features are often lost in the vast extent of the geometry. To deal with that problem, cartographers often exaggerate the terrain vertically. Exaggeration scales often vary with the type of depicted terrain, but scales typically range from 1:1.5 to 1:5.

We control vertical exaggeration by using two scales - one for the lowest terrain, and another for the highest. We linearly blend for the values in between. That allows us to exaggerate and accentuate the smaller geometric features (if high mountain ranges are prevalent), or the other way around.

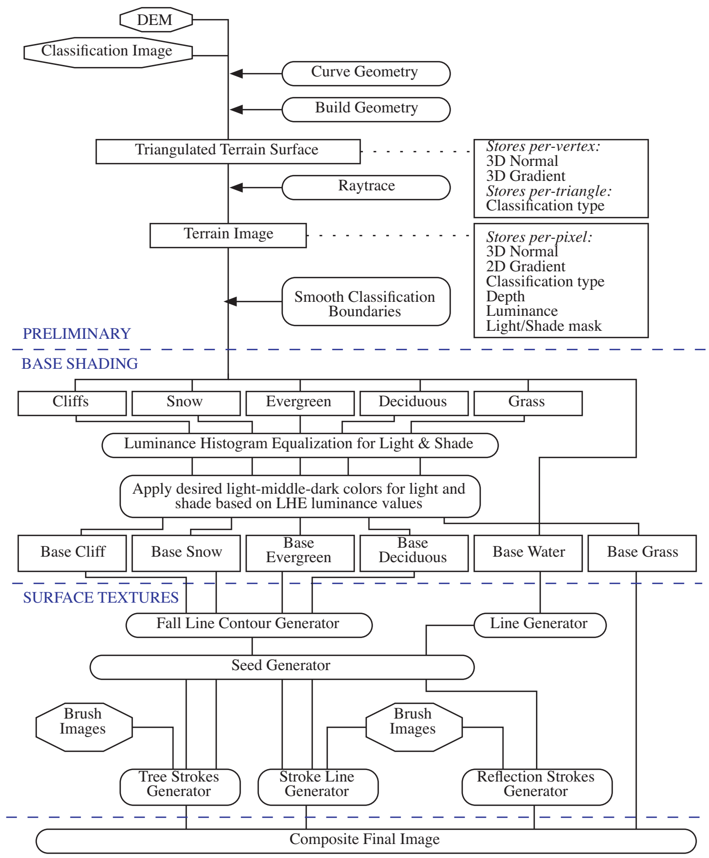

Fig. 12. An overview of the steps taken by our algorithm.

FIRST-ORDER TERRAIN DERIVATIVES

Our algorithm relies on per-vertex surface normals and gradients. We approximate the local surface at each grid point of the height-field by eight triangles (since the grid point has 8 direct neighbors). We compute the normals of each of these triangles, average the result, and store it as the normal corresponding to the current grid point.

At each grid point of the height-field we also compute the three components of the gradient vector. The 3D gradient vector of a terrain surface f at location (x,y,z) points in the direction of maximum rate of change of f. We approximate $\frac{\partial f}{\partial x}$ and $\frac{\partial f}{\partial y}$ for each grid point using central differences. Then we can compute the $\frac{\partial f}{\partial z}$ component [Koenderink and van Doorn 1998; Horn 1981] as follows:

$$\frac{\partial f}{\partial z} = \left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2$$

GEOMETRY CURVING

When rendering 3D terrain, if we choose a low-elevation viewpoint, we will be able to see the horizon. However, features in the foreground will occlude features in the background.

If, instead, we choose a high-elevation viewpoint, we will see larger parts of the terrain, but the view would resemble the traditional two-dimensional cartographic map and would lose its 3D spatial quality.

To address that problem, some panoramists like Heinrich Berann emulate the view from a high-elevation viewpoint. By curving the base of the terrain [Patterson 2000], it is possible to see both non-obscured features in the foreground as well as the horizon and the sky in the distance.

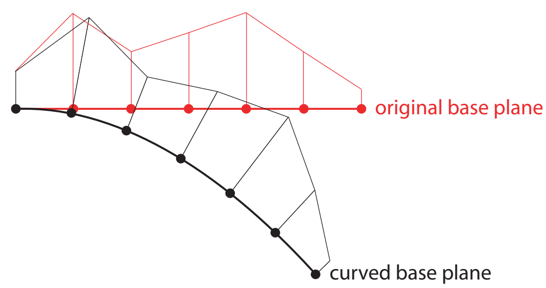

Fig. 13. Curving the base of the terrain.

In our system, we implement this by curving the hypothetical base plane of the terrain at sea level (elevation 0) in the direction of view, via a power function $y = a * x^2$ (see Figure 13). The arc length of the curved plane is the same as the length of the original base plane in the direction of view. To find the new 3D position for each height-field point, we first find its corresponding base position on the curved base plane (using only the height-field offsets, and not the elevation value itself). We then find the curved 3D surface position by using the original elevation as an offset that tells us how long we should go along the normal at the curved base plane position.

We also need to update the 3D gradient to reflect the properties of the new surface. We do so by rotating the stored gradient vector computed from the original base plane by the angle between the vector *up = (0, 1, 0)* and the vector normal to the surface at the new 3D position.

Terrain Surface

We use the new curved 3D surface positions to build a triangulated surface model of the geometry. We associate a classification type with each triangle, based on the original position of the surface point in the grid of the height-field.

Preliminary Terrain Image

We use a raytracer to render an image of the triangulated surface model for a specified light position and viewing direction. We try to choose our light position so that we balance light and shade appropriately.

For each pixel in the rendered image, we compute and store six variables - its *luminance value*, *3D surface normal at hit point*, *image plane projected negative 2D gradient*, *depth from the hit surface point to the image plane*, *classification type*, and *light or shade mask*.

We treat the terrain surface as perfectly diffuse. For each pixel in the image, we compute the luminance value as a function of the cosine of the angle between the surface normal at the hit point and the vector pointing towards the light. If the pixel is in shade, then we use ambient occlusion to approximate the luminance value. The “light or shade mask” simply indicates whether the pixel is in light or in shade, as computed by the raytracer.

If, while raytracing, we hit a triangle that is classified as water, we then send secondary reflection rays in order 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 so that the reflection values match our visual expectations.

CLASSIFICATION BOUNDARIES

As a result of the rendering projection, the boundaries of the classification types in image space will likely not be 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 do the smoothing 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.

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 do retain important geometric and shape information, and we also emphasize three-dimensionality. Based on this convention, we now describe a method for terrain base shading.

For this procedure we rely on the stored luminance information, on the classification types, as well as on the light or shade mask. We do this step for the pixels in each classification type independently. We go through all pixels of a certain type, marked as being in light, and we use their luminance values to compute a three tonal, *light, medium, and dark* segmentation. We then do the same for the pixels marked in shade. The three color segmentation is accomplished by performing three-bin histogram equalization on each specified set of luminance values.

For each classification type, the user specifies the3 colors that should be used in light, and the 3 colors that should be used in shade, in the foreground, as well as the background (for a total of 12 colors per classification). Based on the tonal segmentation (light, medium, or dark), and based on the 3D distance of the point from our selected viewpoint, we blend and apply the appropriate color for each pixel. The lightest valued pixels are assigned a linearly weighted blend between the light foreground and background colors for that type, the middle value pixels, a blend of the medium toned, and the darkest pixels, a blend of the dark tones. When the color is blended, based on terrain distance, we also add a bluish cast and we also de-saturate the color slightly, to match the effect of Aerial Perspective.

In addition, we read-in a sky image, and composite it underneath our base shading. All classification types, but grass, will have additional surface textures applied on top of this base shading.

4.3 Surface Textures Stage

We now demonstrate how to build the 2D fall lines so that we can use them later to guide the surface texture rendering. Since not all surface textures require fall lines (e.g. grass), we build the fall line paths for each classification type independently. Our goal is to find the unique set of fall lines that go through each pixel for each of our classification types. Each pixel belonging to a classification type of interest serves as a starting point for a stroke path (see Algorithm 1). The length of this path is user-specified, but a length that seems to work well for our images is from 40 to about 60 pixels.

Algorithm 1 *ComputeFallLines (paths, pathLengths, negGradients)*

for all *clsfyType* ∈ Classification Types **do**
  for all *pixel* ∈ *clsfyType* **do**
    pathPixel = *pixel*
    for *length* = 0 to *pathLengths[clsfyType]* **do**
      RungeKutta (&pathPixel, negGradients[pathPixel])
      paths[clsfyType].path[length] = pathPixel
    end for
  end for
end for

We want to create a stroke path that visually follows the direction of maximum descent. To compute that path, we numerically approximate the line derivative ODE of the 2D fall line vector by using Runge-Kutta order 4 [Ralston and Rabinowitz 2001]. In each step along the stroke path, Runge-Kutta approximates the (x,y) position for the next point along the path, based on the 2D projected and negated gradient vector direction referred by the current pixel.

After we build a path that follows the fall line for each pixel per classification type, we are likely to end up having too many paths spanning the image. The paths will converge and diverge, and we will have many paths running through individual pixels. What we need, instead, is a a set of paths that do not overlap - i.e. a unique set of paths.

To accomplish this, we prune the set of paths paths as follows. We go through each pixel of the image and index the set of paths that run through it. We compute the length of each path at that pixel, and keep the one that has the most elements in the path up to this point. We prune the lengths of all other paths, so that they no longer reach this pixel (see Figure 14). By pruning, we are only removing the excess, and not losing any information. We choose to keep the longest path since we want to produce the longest running image paths.

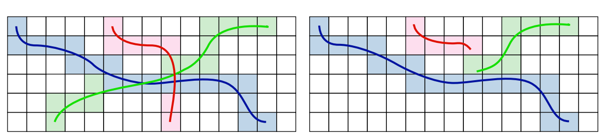

Fig. 14. Paths before (left) and after (right) pruning.

Next we want to make the pixels indexed in our paths unique. As a result of the Runge-Kutta approximation and the integer rounding due to the discretization of the image, some paths index the same pixel twice. We remove such points, and shorten the length of the paths appropriately. Now we have unique paths that run through every pixel in each classification region.

Finally, we sort the paths so that they are indexed by length, with the longest ones being first. This is necessary in the application of strokes to the image, as we want to give priority to the longer strokes since they are visually more important. By sorting the strokes by length and then placing them in sorted order, we achieve exactly that.

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. We make their effects even stronger, by combining the texture cues with surface contour directions, utilizing forest meta-strokes.

To handle the texel size issues, we allow the user to specify a texel size perspective factor, i.e. how much should texels in the farthest distance be scaled down by. To estimate the brush size scaling weight at each pixel from the maximal perspective factor, we use the distance to that pixel from the image plane and the maximum distance observed in the image (we have pre-computed those distances for each pixel in the image in the preliminary stage of the algorithm).

We now need to determine where to place the trees in the image. We find it easiest to compute the image space "seed" positions (i.e., the positions where we want to draw a tree stroke) for the forest classification region first, store these positions, and later draw the actual trees.

We determine the positions for our tree seeds by having the fall line paths serve as imaginary guiding directions (see Algorithm 2). We use the longest paths first, and end the algorithm when all paths have been tested. In the beginning, we mark all pixels within the forest classification region as potentially available for seed placement.

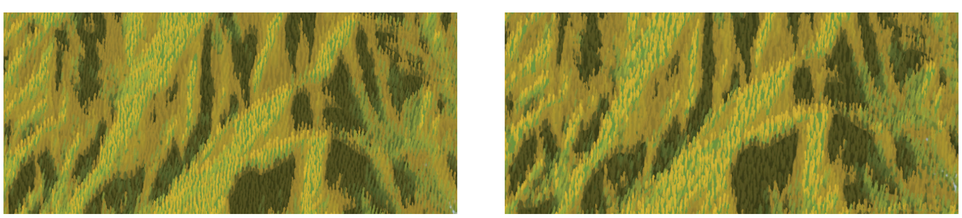

Fig. 15. Here we compare our method of tree distribution (left) to that of jittered sampling tree distribution (right).

We walk along the (x,y) points for each fall line path, starting from its origin, using stride lengths determined by the slope of the surface at each stopping point along the path. This way we approximate necessary texel spacing foreshortening effects, which helps us address our requirement for homogeneously distributed object-based tree texels. If the circular region (the radius of the region is set based on the user's desire for texel occlusion) is still marked as available, we now mark it as used, mark the center pixel as a seed point, and continue walking the path and testing for availability, until we reach the path's end.

We then continue walking the next longest path, until we exhaust all paths within the classification region.

Algorithm 2 *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
      end if
      stride = treeSpacing
      stride *= Dot (-viewDir, surfNormal[pixel])
      length += stride
    end while
  end for
end for
```

Once we are done selecting the positions of the tree seeds, we paint the actual desired tree brush images in the seed positions, in a back-to-front order. 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.

The user provides 12 colors for each brushstroke, 6 for trees in the foreground - 3 for light, medium, and dark shading for trees positioned in the light, and another 3 for light, medium, and dark shading for trees positioned in the shade, and similarly another 6 for trees in the background. We use the position of the seed point in image space to choose the appropriate blended color, based on the three-tonal luminance segmentation step we performed earlier (into light, medium, and dark).

To add some variety to the trees, we stochastically select a weight factor (between 0% and 30%), and use it to linearly interpolate between the chosen color, and the closest darker tree color. That forms the final color for each tree stroke.

CLIFF AND SNOW TEXTURES

Cliff and snow texture rely on the fall line directions to become more structurally expressive. Four basic parameters control the application of strokes - *stroke color*, *stroke width*, *stroke overlap*, and *stroke image definition*. When we refer to cliff and snow strokes, we mean the fall line strokes we have computed within the appropriate cliff and snow classification regions.

Similarly to forest strokes, for cliff and snow textures, the user provides 12 *stroke colors* for each brushstroke type. Here, again, the luminance shading segmentation type specifies the appropriate color to be selected. And again, to add some color variety, we use a weight factor (0% to 50%) and linearly interpolate between selected stroke color and its closest darker color. The difference 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.

Stroke density relates to how many strokes we see, and is controlled by *stroke overlap* and *stroke width*. The *stoke width* is randomly selected from the ones provided by the user, so as to introduce some irregularity in the forms. Stroke overlap is fixed and is also specified by the user. We use the sorted set of strokes and start processing from the longest. We use the width and overlap settings to resize the stroke footprint specified at each path point along the stroke. If the current stroke footprint is still unoccupied, we mark it as occupied, and indicate that we should draw part of the brush stroke there. If it is already occupied, we start testing the next path. We continue testing each stroke footprint pixels along the path, until we reach its end. We then test the second longest path, and repeat until we exhaust all strokes. To examine the effect of stroke textures, see Figure 16, where we only have base shading, and compare it to Figure 1.

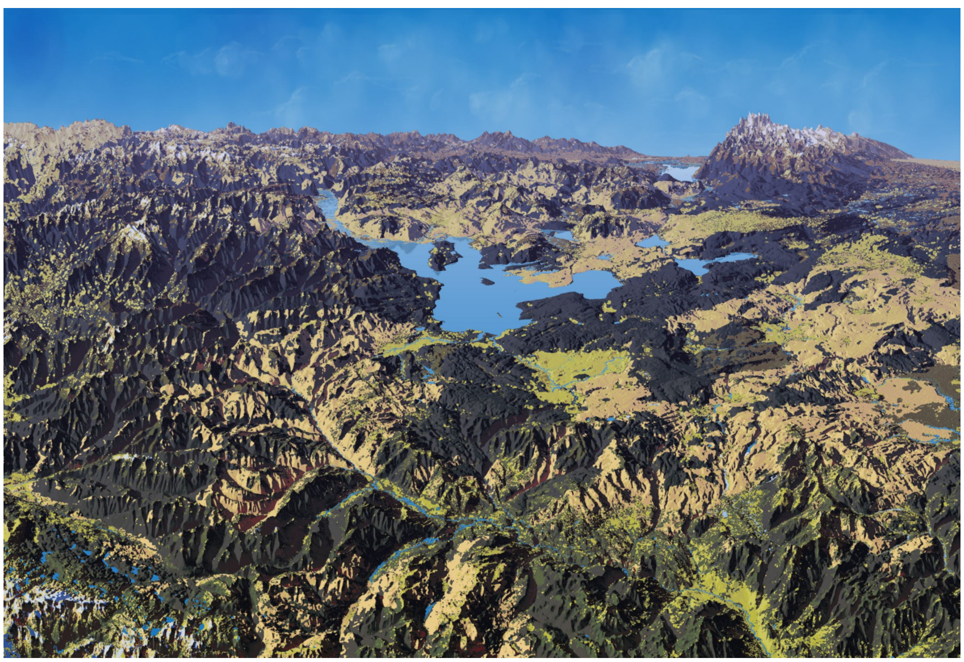

Fig. 16. Here textures are using only base color.

Our cliff brush *stroke image definitions* are defined similarly to skeletal strokes [Hsu et al. 1993]. We read in the brush stroke image. For each cliff stroke, we resample the brush stroke image to match our desired stroke width, and the length of our currently selected path. We then walk along each pixel that makes the path, and if 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
To render textures for water surfaces, we generate strokes of varying lengths in the horizontal direction for all areas in the image that are classified as water. 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. We then, just as we did for the other textures, alpha-blend the strokes with the background.

4.4 Compositing

Finally, all the layers are composited together to form the final panorama map rendering.

5. RESULTS

We render our images on a PowerPC G5, 2.74 GH. For our chosen image size, rendering consumes from 1.5 to 2.4 GB RAM. We use two different mountainous terrain datasets - one for Yellowstone National Park, and another for Rocky Mountain National Park. We view them from a far distance, so as to reduce the perspective effects. Our tree strokes are based on the shapes and dimensions of the strokes we sampled from the examined Yellowstone maps (Figure 10), and our cliff strokes are shown in Figure 17.

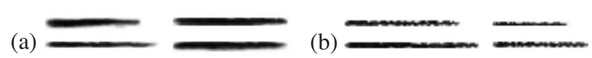

Fig. 17. Cliff strokes used by our system for (a) Yellowstone (b) Rocky Mountain National Park.

Each stroke is assigned randomly one of the available shapes. Our colors are based on the light, medium, and dark color samples presented in Figure 11, and the colors have been alpha-blended between the near and far color samples, using distance as a weighting factor. Stroke length is set to 50 for tree strokes, and 70 for cliff strokes. Stroke width for snow and cliff strokes for all images varies between 3 and 6 pixels. In the application of the tree strokes, we pad the shape by 3 pixels for the Yellowstone renderings and 4 for the Rocky Mountain one, to control spacing.

For the Berann-style rendering of Yellowstone (see Figure 1), we used Berann-style tree strokes and samples of his colors for near and far distance that we presented in Figure 11. For the Niehues-style rendering of Yellowstone (see Figure 18), we used Niehues-style tree strokes and again samples of his colors. For rendering statistics, refer to Figure 19.

We find that assigning colors that work well for novel terrain datasets is a fairly difficult and involved task. See Figure 20 for an example rendering of Rocky Mountain National Park whose colors were based on samples from images of the area.

Despite the fact that our colors were sampled from real colored images, corresponding to the same terrain, the colors of the rendered image are not satisfactory. To address this problem, we did a color transfer based on Reinhard et al. [Reinhard et al. 2001], where image statistics were matched in RGB space. 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 21.

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

	image dimension	*terrain dataset*	*rendering (AmbOccl)*	*texture generation*	*total rendering*
Yellowstone NP (Berann Style)	*5284 x 3587*	*1889 x 1880*	*30.34 min*	*25.02 min*	*58.47 min*
Yellowstone NP (Niehues Style)	*5284 x 3587*	*1889 x 1880*	*29.09 min*	*25.00 min*	*57.96 min*
Rocky Mountain NP	*5284 x 3699*	*927 x 767*	*16.58 min*	*15.02 min*	*32.31 min*

Fig. 19. Rendering statistics.

6. CONCLUSION

We have demonstrated an automatic technique to generate images whose visual style is reminiscent of those in panorama maps. We have chosen to operate mainly in image space because that is the natural space in which to generate painterly strokes. The downside of using image space is that an animated change of viewpoint is then 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 geometry manipulation (and mostly for visibility purposes). 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. Our work is also limited in visual quality. But that comes as no surprise; NPR renderings are used to improve the speed and availability of imagery rather than the quality readily achievable when a good artist is available.

Fig. 20. Rocky Mountain National Park terrain, rendered with rock, snow, water, and two kinds of tree textures, rendered with default colors.

Acknowledgements

This work was supported by NSF grant 03-12479. 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 providing 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.

BROOKS, M. 1983. Two results concerning ambiguity in shape from shading. In *AAAI*. 36–39.

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.

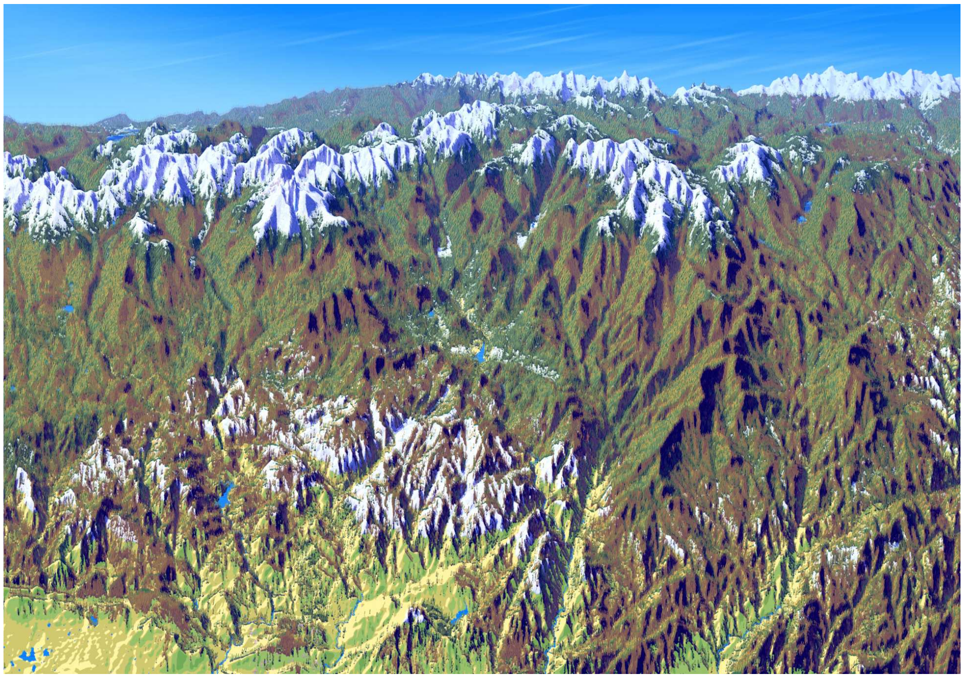

Fig. 21. Color transfer of the Rocky Mountain National Park terrain, rendered with rock, snow, water, and two kinds of tree textures colors.

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, B. AND GOOCH, A. 2001. *Non-Photorealistic Rendering*. AK Peters Ltd.

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.

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*.

RALSTON, A. AND RABINOWITZ, P. 2001. *A first Course in Numerical Analysis*. Dover.

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.

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.

SANTELLA, A. AND DECARLO, D. 2004. Visual interest and npr: an evaluation and manifesto. In *NPAR '04: Proceedings of the 3rd International Symposium on Non-Photorealistic Animation and Rendering*. ACM Press, 71–150.

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.

STROTHOTTE, T. AND SCHLECHTWEG, S. 2002. *Non-photorealistic computer graphics Modeling, rendering, and animation.* Morgan Kaufmann Publishers.

SULLIVAN, C. 1997. *Drawing the Landscape*. John Wiley and Sons, Inc.

SUPER, B. AND BOVIK, A. 1995. Shape from texture using local spectral moments. *IEEE Trans. on PAMI 17,* 4, 333–343.

TUFTE, E. 1990. *Envisioning Information.* Graphics Press.

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.

ZEKI, S. 1999. Art and the brain. *Journal of Consciousness Studies 6,* 6-7, 76–96.