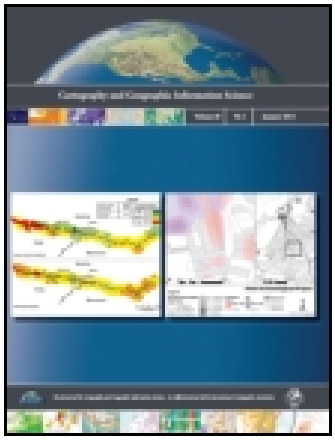

Cartography and Geographic Information Science
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tcag20

Supporting Automated Pen and Ink Style Surface Illustration with B-Spline Models

James E. Mower

Published online: 14 Mar 2013.

To cite this article: James E. Mower (2011) Supporting Automated Pen and Ink Style Surface Illustration with B-Spline Models, Cartography and Geographic Information Science, 38:2, 174-183, DOI: 10.1559/15230406382161

To link to this article: http://dx.doi.org/10.1559/15230406382161

Supporting Automated Pen and Ink Style Surface Illustration with B-Spline Models

James E. Mower

ABSTRACT: This paper introduces a methodology for constructing pen and ink style landscape illustrations using B-spline surface models. Following a brief discussion of manual methods and recent developments in automated non-photorealistic rendering (NPR) techniques, the paper discusses polynomial models for surface rendering and their implementation in the OpenGL graphics interface. It then discusses the generation of silhouettes and form lines for topographic surfaces from a set of triangulated vertices obtained from polygonal tessellations. An implementation of the methodology outlined in the paper is described and tested. The results are analyzed and suggestions for further research are presented.

Keywords: non-photorealistic rendering, terrain rendering, computer graphics

Introduction

In the 19th century, William Henry Holmes, Grove Karl Gilbert, and William Morris Davis perfected the construction of pen and ink style representations of physiographic forms and processes (Fernlund 2000). Rooted in the tradition of perspective landscape illustration expressed in the drawings and paintings of da Vinci and the later woodcuts of Murer and other 16th century artists, their illustrations are characterized by a restrained application of strokes, each contributing a critical aspect of landform structure or process. Examples of their art are characterized both by renderings of view-dependent features, such as the silhouette of a hill against the sky, or view-independent aspects of surface morphology, such as hydrologic ridges or valleys. Later artists such as Armin Lobeck (1939; 1958), Erwin Raisz (1948) and Eduard Imhof (2007), produced a number of excellent works detailing both the utility and construction of pen and ink style landscape representations. Lobeck argues that such images achieve an effective portrayal of natural surface forms primarily through the artist's selection of only those elements to which the viewer's attention should be directed (Lobeck 1958, pp. 1-2). Imhof (2007, p. 45) echoes Lobeck and provides a succinct comparison of landscape photographs (and, by extension, shaded surfaces) to line drawings, stating:

> *"Views of nature and their photographic reproduction contain no lines. The drawn line is a human invention, a useful abstraction or fiction. Forms and their spatial relationships can be clarified by lines, the essential can be emphasized and the nonessential can be subdued. A good sketch is simpler, more expressive and aesthetically more satisfying than a photograph…It should never be forgotten that sketching means leaving things out!"*

Lesage and Visvalingam (2002) note that cartographic sketches facilitate visual processing of static images by focusing attention on significant features. Santella and DeCarlo (2004) show this experimentally through eye tracking experiments, noting that subjects extract image information by looking at fewer locations in non-photorealistic rendering (NPR) abstractions than they do in literal photographic reproductions. Although significant progress in creating NPR landscape renderings has been made over the past decade, most attempts continue to fall short of the quality exhibited by hand-drawn examples. Some of this disparity can be attributed to the use of planar-faceted digital elevation models—built either from regular grid cell or triangulated irregular network structures—as the underlying basis for surface analysis. Planar surface tessellations derived from triangulated samples rarely provide sufficiently smooth analytical surfaces for landscape drawing. Imhof (2007, p. 41) notes that classically trained terrain illustrators use surfaces of ideal geometric solids (cylinders, cones, pyramids, and other forms) as models for landscape representations. By providing a surface of generalization, such forms help the artist to filter out local elements of surface roughness that would otherwise obscure dominant surface features. This paper presents a methodology, its implementation, and results for feature extraction and pen and ink style rendering from B-spline polynomial surface models. Polynomial surface representations are not, of course, new to automated cartography. Maxwell and Turpin (1968) propose the use of polynomial models for topographic

James E. Mower, Department of Geography, State University of New York at Albany, Albany, New York, 12222, USA, E-mail: <jmower@albany.edu>.
DOI: 10.1559/15230406382161

surface representation for civil engineering projects. Although attractive with respect to the minimal data storage capabilities of the 1960s, such models require untenable execution times when applied to the modeling and rendering of fine-resolution surface processes. Fortunately, as Imhof and Lobeck stress, fine surface detail is often unwanted in pen and ink style landscape representations. Furthermore, 3rd degree polynomial surface patches can be 'stitched' together while maintaining continuity at joins. Both characteristics make polynomial surface patches artistic and reasonably efficient tools for pen and ink style rendering of landscapes.

A Brief Review of Manual and Automated Pen and Ink Style Surface Rendering

Davis's Practical Exercises in Physical Geography (1908a) and its accompanying atlas (Davis 1908b) illustrate archetypal land forms through a set of interpretive exercises. The illustrations provide very clear examples of Davis's use of line density to depict variation in slope and directionality in illumination. His work is characterized by the application of innumerable short strokes (hachures) depicting drainage and surface forms. At first glance, much of his linework looks as though it is formed from longer strokes. On closer inspection, many individual 'lines' turn out to be composed of very short hachures. Even silhouettes and hydrologic ridge lines are sometimes drafted as a collection of short hachures pointing in the direction of steepest slope.

Lobeck (1958) provides explicit instructions for interpreting geomorphic features with linework but leaves it to the student to develop techniques for varying line density based on slope and lighting characteristics. Imhof (2007, pp. 214-224), on the other hand, provides explicit rules for rendering slope and shadow hachures. Of those, the following are most relevant with respect to automated approaches:

- Hachures intersect contours at right angles;
- Slope is expressed through the width and spacing of hachures; and
- Slope values are modulated with directional illumination values to vary surface tone (as performed for analytical hill shading).

The application developed for this paper and described in the section "Creating and Rendering Geometry" applies Imhof's rules to the rendering of hachures with respect to local slope and illumination characteristics.

Previous Work in Automated Pen and Ink Style Rendering

Techniques for automating pen and ink style landscape rendering fall into the paradigm of non-photorealistic rendering. This project employs an NPR approach, referred to by Markosian and others (1997) as "an economy of line," that comes closest to the intent of Davis, Lobeck, and Imhof. This approach attempts to depict the essential form of an object using a minimal number of strokes, categorized as follows:

- Silhouettes—lines representing borders between visible and invisible regions from the point of view of the observer;
- Creases—lines representing important features that must be rendered without regard to the viewer's position; and
- Form lines—lines depicting overall surface curvature, independent of the point of view, but unrelated to particular surface features.

Most of the previous work in NPR techniques for landscape rendering share this aesthetic but approach it with different techniques. Lesage and Visvalingam (2002) implement a sketch-based approach to terrain rendering that interprets a shaded digital elevation model (DEM) as a luminance map. Although edge detection algorithms are applied to extract creases and silhouettes, form lines are not explicitly generated. Later work by Buchin and others (2004) describe a system for producing pen and ink representations through the application of scanned, hand-drawn slope lines from a look-up table. Linework is assembled as a seamless patchwork of raster scans with density dependent on surface slope and lighting conditions. Their surface analysis is based upon the calculation of gradients and surface curvature. Large patches are filled in first; smaller patches fill the remaining spaces. Conflict resolution procedures prevent linework collision.

Mower (2008; 2009) investigates the use of triangulated irregular network (TIN) surface models for pen and ink style rendering without resort to the application of scanned hand drawn samples. Using an adaptive resampling technique that composes a surface from relatively constant-area triangular facets in screen space independent of depth from the viewpoint, it extracts silhouettes, creases, and form lines from the DEM relative to the observer's point of view, and renders them as line segments on the surface. Although the implementation renders features similar to those of a comparable manual drawing, it tends to undergeneralize surfaces, rendering more, and sometimes irrelevant features due to noise in the underlying TIN and due to edge artifacts arising from discontinuities at the edges of relatively coarse planar boundaries (Figure 1). The application described in this paper addresses this issue by extracting linear features from a finely tessellated polynomial surface derived from a grid cell DEM (Figure 2).

Using Polynomial Surface Modeling as a Framework for NPR

Both Lobeck and Imhof suggest that students of geomorphology visualize landscapes as surfaces of ideal geometric solids. Rather than attempt a literal implementation of their directive, the application described in this paper generalizes landscape renderings to polynomial surfaces derived from grid

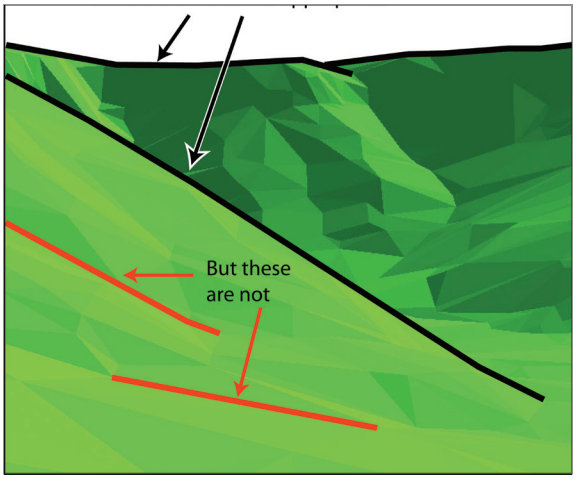

Figure 1. Simulation of a noisy surface model that produces inappropriate silhouettes. Although both sets of silhouettes represent edges bounding visible and invisible surfaces, those rendered in red are due to minor surface irregularities or noise in the DEM. An artist would consider such features irrelevant to the scene as a whole.

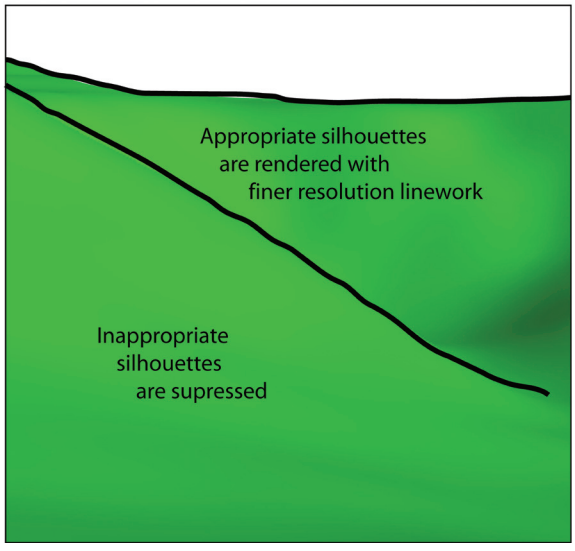

Figure 2. Simulation of the scene in Figure 1, but rendered with a B-spline surface. Silhouettes are rendered at the pixel resolution of the underlying polygonal tessellation. The edges causing the inappropriate silhouettes in the previous image have been smoothed.

cell DEMs. This has several important implications for pen and ink style representations:

- Given that an appropriate tessellation of the surface polynomial produces facets with relatively constant and sufficiently small screen space (approximately 1 facet per pixel), linework density will vary minimally with respect to perspective depth;
- Local surface variations will be smoothed with respect to the polynomial surface;
- Calculation of surface properties can be performed easily through the extraction of normals for the tessellated facets; and
- The approach is relatively simple to implement and allows for experimentation with a small number of variables.

A graphics processor must tessellate a polynomial surface before it can render it. We achieve the illusion of surface continuity by forcing the tessellation procedure to maintain facet sizes in screen space of approximately 1 facet per pixel. Furthermore, we use the set of tessellated vertices to provide a more finely grained model for surface analysis than would be practical with a comparably sampled grid cell DEM. Not only does the resulting linework appear smooth, many analytical problems related to grid cell sampling artifacts (such as false pit identification in drainage accumulation analysis) are greatly reduced. Although the tessellated polynomial surface is not particularly valid for numerical modeling of geomorphologic processes, it provides a very useful model for generalized surface curvature.

Polynomial Surface Models

Much of the early work in DEM data structures and interpolation involving the use of polynomial equations for surface modeling is summarized by Mark and Smith (2004). Momentarily limiting our discussion to 2-dimensional curves for simplicity, polynomial representations are sensitive to the number of control points that influence the shape of the curve in their vicinity. A Bezier curve described by $n+1$ control points (a polynomial curve of degree n or order $n+1$) is shown in Equation 1 (Piegl and Tiller 1997, pp. 9-10):

$$C(u) = \sum_{i=0}^{n} B_{i,n}(u) P_i \qquad (1)$$

where:

$$0 \le u \le 1,\ B_{i,n}(u) = \frac{n!}{i!(n-i)!} u^i (1-u)^{n-i}$$

and P_i is a control point.

In Equation 1, $B_{i,n}(u)$ is referred to as a blending or basis function, u is a normalized parameter with a value of 0 at the start of the curve and 1 at its end, i is the index of the current control point, and n is the degree of the curve (the number of control points minus 1). Unfortunately, Bezier basis functions become numerically unstable as n becomes large and the dependant values of the factorial computations exceed those that can be represented in standard integer or floating point storage types. Fortunately, low order Bezier curves can be appended or 'stitched together' to form larger piecewise polynomial curves under specified constraints of continuity at the joins. For rendering purposes, the most popular of these forms is the B-spline (or basis spline) curve. Like Bezier curves, B-splines extend easily to 2-dimensional

surface representations through the addition of a second parameter v. The influence that a particular basis function exerts over a portion of the surface is controlled by 2 arrays of 'knots,' values that specify the range in u and v over which the value of a basis function is non-zero. A piecewise surface is infinitely differentiable in its interior but not so at its joins; this project utilizes piecewise surfaces of degree 3 with continuity up to the 2nd derivative at joins, sufficient for rendering purposes without producing noticeable edge artifacts.

It is important to note that the resulting surface is not a surface of interpolation; it is only guaranteed to pass through samples at the edge of a region (Figure 3). Although it is certainly possible to define a polynomial surface that passes through each of its sample points, Piegl and Tiller (1997, p. 361) note that such surfaces tend to "wiggle" through them, generating surface noise as problematic for feature extraction as that produced by coarser planar faceted models.

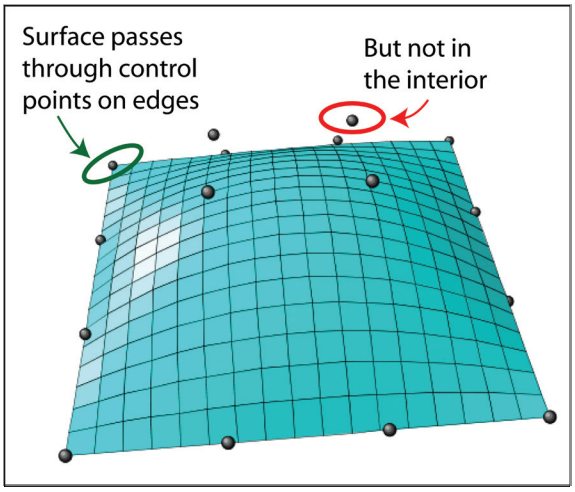

Figure 3. A Bezier patch is not a surface of interpolation. Interior control points influence curvature in their vicinity but do not necessarily intersect the surface. The surface in this image was created by a Java applet written by Little (2006).

Creating and Rendering Geometry

The author wrote the pen and ink rendering system described in this paper in C++ with links to the OpenGL v. 2.1 software interface for graphics rendering hardware. OpenGL provides a high-level interface for non-uniform, rational, B-spline (NURBS) curves and surfaces (Shreiner et al. 2007) that can also be used to construct the uniform, non-rational, B-spline surface renderings of the type implemented in this project. In either case, the surface is specified by a regular (but not necessarily square) grid of 3-dimensional control points. The NURBS rendering engine creates a polygonal tessellation over the region described by the control points such that the screen space occupied by a polygonal facet is independent of its distance from the viewpoint in world space. Figure 4 illustrates an enlarged wireframe view of a portion of a tessellation.

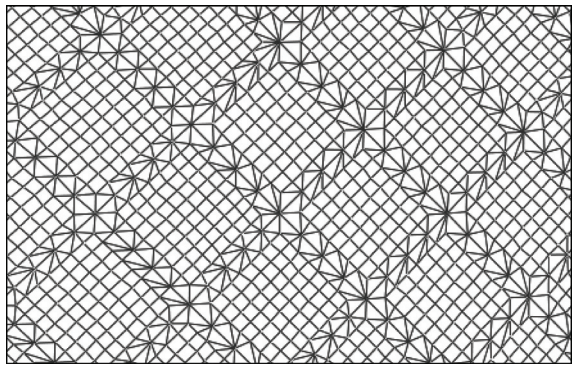

Figure 4. An enlargement of a tessellated B-spline surface produced by the OpenGL NURBS evaluator. Note that facets of OpenGL tessellations are polygonal, not necessarily triangular.

OpenGL provides callback functions that allow the programmer to retrieve the vertices of a tessellated surface. The vertices can then be triangulated through functions provided by the Computational Geometry Algorithms Library (CGAL Editorial Board 2007) and then incorporated into a TIN data structure. CGAL provides an interface for building a 2D Delaunay triangulation from a set of 3D vertices using easting and northing for planar coordinates and elevation as a triangulated vertex attribute. Although CGAL provides an n-dimensional triangulation interface, 2D triangulation is generally most appropriate for non-wrapping sections of topographic surfaces.

Building a B-Spline Surface from Elevation Data in OpenGL

Figure 5 illustrates the project data stream. Elevation samples are extracted (at a user specified resampling rate) from 10m horizontal resolution grid cell DEM source files in USGS 1:24000 native format and stored in an internal rectangular grid (referred to here as an arbitrary grid). To reduce unnecessary processing, tessellated vertices are projected to viewport (screen) coordinates and rejected for triangulation if they are located outside the bounds of the window.

The samples in the arbitrary grid are passed to the NURBS renderer that interprets them as control points for a degree 3 B-spline surface. The rendering engine produces 2 products: 1) an image held in the graphics hardware z-buffer (the surface mask), and 2) a set of 3D points representing the vertices of the tessellated B-spline surface. The mask is rendered in the background color as an emissive surface. By doing so, the shaded facets appear uniform and remain

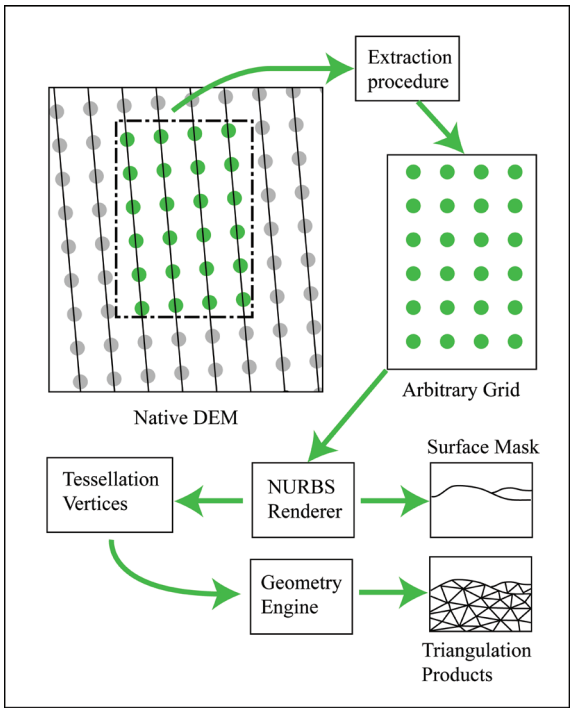

Figure 5. The data stream. The NURBS rendering engine uses a rectangular grid of extracted samples (the arbitrary grid) to create a rendered image (the surface mask) and a set of polygon vertices from the dense surface tessellation. The geometry engine assembles the tessellation vertices into a Delaunay triangulation with topological relationships among the triangle facets, edges, and vertices (triangulation products).

indistinguishable from the background. Yet as an opaque surface, each facet hides linework rendered on any surface behind it, relative to the observer's point of view (Figure 6).

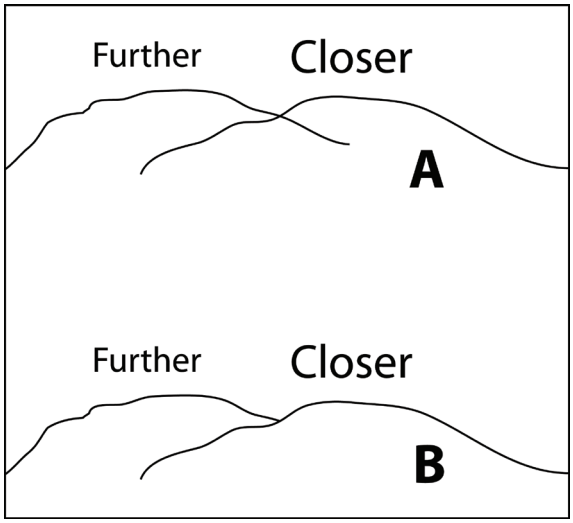

Figure 6. The silhouettes of the 2 hills in A are rendered without use of a surface mask, allowing part of the silhouette of the further hill to inappropriately intersect that of the closer hill. In B, surface masking resolves the intersection problem.

Figure 7 shows an example of a rendered B-spline surface shaded with a non-emissive color. The vertices are passed to the CGAL Delaunay triangulation interface where they are assembled into topologically-related lists of triangle facets, their edges, and their vertices. Form lines and silhouettes are rendered from edges as appropriate.

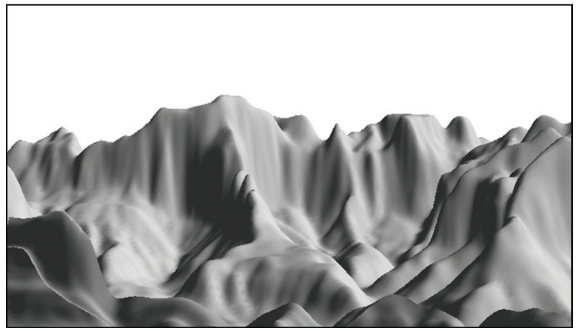

Figure 7. An arbitrary grid rendered as a B-spline surface with non-emissive gray shading. Although a surface mask requires emissive shading in the background color, this figure uses non-emissive shading for purposes of illustration.

Finding and Rendering Silhouettes

For the observer, a silhouette is the line marking the boundary between surface and sky or the outline of a closer object against a further one. For rendering, however, it is more convenient to identify silhouettes in the triangulation by checking the visibility of pairs of adjacent triangular facets (Figure 8).

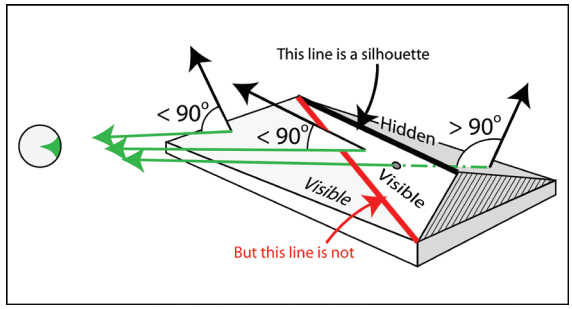

Figure 8. The silhouette model. A surface is visible from the viewpoint (disregarding obscuring surfaces) if the angle between the surface normal and the viewpoint vector is less than 90°. A silhouette is drawn on the shared edge of 2 facets with contrasting visibility.

If one of the facets faces the viewer and the other faces away, the boundary between them is a silhouette. Visibility is determined by the angle made between the facet normal and the vector from the observer to the facet. If the angle is between 0 and 90 degrees, the facet is visible; otherwise it is not. If the visibility attribute of the adjacent facets differ, the edge is

rendered as a silhouette. Since the surface tessellation produces a very dense triangulation, edges are sufficiently short to suggest curvature when rendered with other connected segments. Figure 9 shows the silhouettes that were generated for the scene in Figure 7.

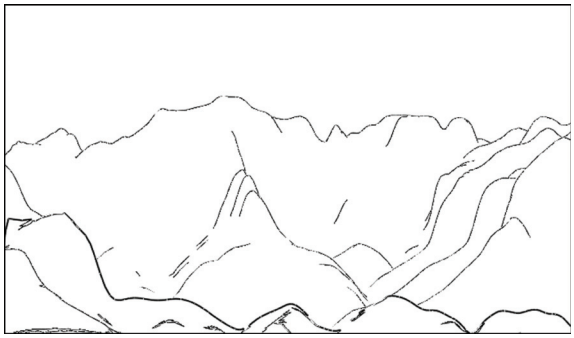

Figure 9. Silhouettes generated for the scene in Figure 7. The surface is rendered in emissive white and now serves as a depth or surface mask, hiding silhouettes drawn behind the surfaces nearest to the viewpoint.

Drawing Form Lines with a Dense Drainage Direction Model

Form lines are considerably more difficult to define than are silhouettes. An inspection of many of the illustrations produced by Davis (1908b, p. 4) suggests that most of the drafted form lines can be interpreted as drainage direction vectors (Figure 10).

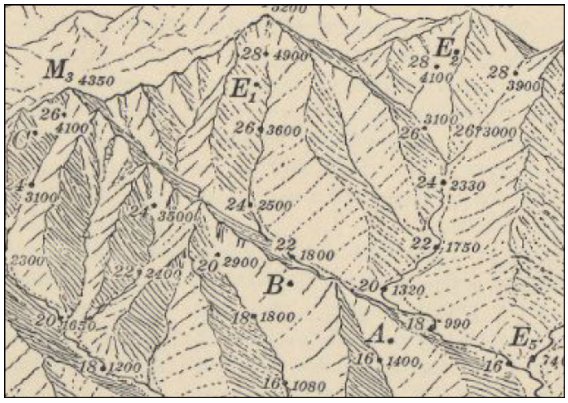

Figure 10. A section of an illustration by W.M. Davis. Form lines perpendicular to ridges mimic drainage direction vectors. Differential density of form lines for apparently equivalent slopes suggests an illumination source from the left.

The program developed for this project (PenAndInk) implements this approach through a 2-step procedure incorporating an illumination model (Figure 11). First, a drainage direction map is created. For every vertex in the triangulation, the connected vertex with the steepest downhill slope is identified and marked as its sink. The edge containing the starting vertex (the source) and sink vertices represents the drainage direction vector for the source. Second, the normal of the source's parent facet is evaluated to determine the slope of the facet and its illumination value with respect to the user-specified light source vector. Edges that belong to facets with steep slopes and aspects facing away from the incoming light source are rendered with lines of maximum thickness. Those belonging to facets on gentle slopes facing the light source are rendered with the thinnest form lines or none at all. The width of a given stroke is calculated using Equations 2 through 4:

$$SlopeC = SR \times MW \times SW \quad (2)$$

$$ShadeC = (1\text{-}v) \times MW \times (1\text{-}SW) \quad (3)$$

$$W = SlopeC + ShadeC \quad (4)$$

where SR is the slope of the facet, stated as a ratio from 0 to 1 (where an SR of 1 is equivalent to a vertical slope), MW is the maximum line width in points, SW is a ratio from 0 to 1 expressing the weighting of slope over lighting (where a slope weight of .4 implies a lighting weight of .6), v is the cosine of the angle between the light source vector and the surface normal, and W is the computed weight of the rendered line in points. Slope weighting is controlled by a user interface parameter.

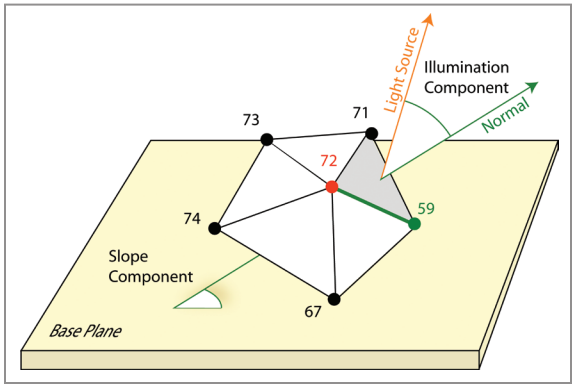

Figure 11. The form line attenuation model. For the source vertex at 72 m elevation, the vertex at 59 m is found to be the sink. The green edge represents the drainage direction. The angle between the parent facet normal and the light source vector provides the shading component. The angle between the normal and the base plane provides the slope component.

The dense triangulation produced from the tessellated vertices, combined with an appropriate initial form line width, provides a smooth shading gradient from steep and shadowed regions to those that are flat and directly illuminated. Since edges

in a Delaunay triangulation are guaranteed not to cross one another, conflict resolution is not required and no additional geometry need be created. Form lines are generated implicitly as connected edges along rendered drainage paths. Figure 12 describes a portion of a scene rendered by PenAndInk.

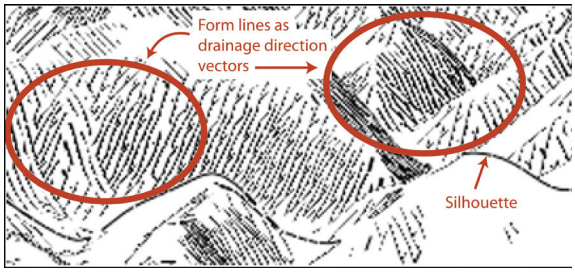

Figure 12. Linework rendered by PenAndInk shown at.1.5x enlargement over the original image.

Results

A series of images in the vicinity of the West Temple feature in Zion National Park, Utah, were rendered from an arbitrary grid containing 120 rows and 140 columns of elevation samples with a horizontal resolution of 50 meters. It was found through experimentation that a 50 meter resampling of a 10 meter DEM grid produced a sufficiently generalized surface model for the images in this sequence.

Each of the images shows the entire scene generated from a viewpoint at UTM coordinates 330249m E, 4116000m N, Zone 12N, with an elevation of 2835m relative to a vertical exaggeration factor of 1.5. The viewing azimuth and altitude are 295° and 0° respectively with a field of view of 18°. Figures 13 through 17 represent images that vary in maximum form line width, the weighting of slope over aspect for line density calculations, and the direction of the light source. The lighting, line width, and slope weighting parameters for these figures are listed in Table 1. Images for each figure require approximately 3 minutes of processing time with an Intel dual core processor and an NVIDIA Quadro graphics processor (GPU). A large portion of the processing time is spent in the construction of the B-spline surface; unlike the OpenGL primitives, the NURBS package runs on the CPU rather than the GPU. Substantial time reductions may be achieved by moving to a simpler, but equally effective B-spline rendering method on the GPU.

Figures 13 and 14 depict the same scene but differ in the maximum width permitted for form lines. Both images use a weighting of 40% for absolute slope (slope component) and 60% for lighting (shading component). Any line with an attenuated width less than 1 point is not rendered. Flatter regions are highlighted in Figure 14 by their relative lack of linework.

Figures 15 and 16 show the effect of changing the bias between the absolute slope and shading parameters. Both images use a maximum form line width of 3 points. With a weighting of absolute slope of 90%, the flat surface at A on Figure 15 is rendered with few strokes. Figure 16 weights absolute slope by 10%, attenuating line width primarily by light source direction. Now the surface at A, with the light source vector inclined approximately 45° with respect to the

Figure	Light Az. (degress)	Light Alt. (degrees)	Max form line width (pts)	Slope weighting
13	250	45	3	40%
14	250	45	2	40%
15	250	45	3	90%
16	250	45	3	10%
17	70	45	3	40%

Table 1. Lighting and line width parameters for Figures 13 through 17.

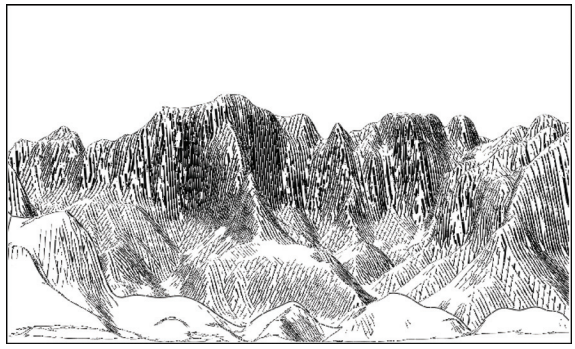

Figure 13. The West Temple feature of Zion National Park, with a viewing azimuth of 295°. Lighting is from 250° azimuth (upper left corner), 45° altitude, with a maximum linewidth of 3 points and slope weighting of 40%.

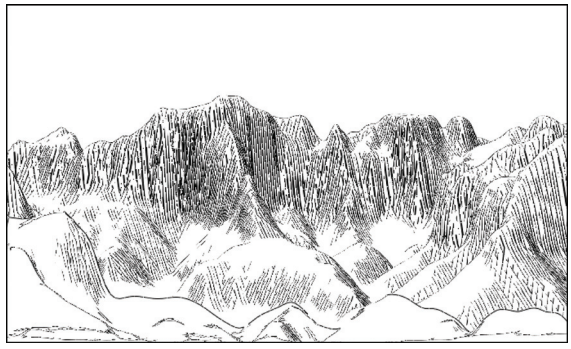

Figure 14. The same scene and viewing parameters depicted in Figure 13 but with a maximum form line width of 2 points. Starting with a lower maximum width, more lines in flatter areas are attenuated to below 1 point in width and are left unrendered, highlighting those regions.

surface normal, appears relatively dark, regardless of its minimal slope value.

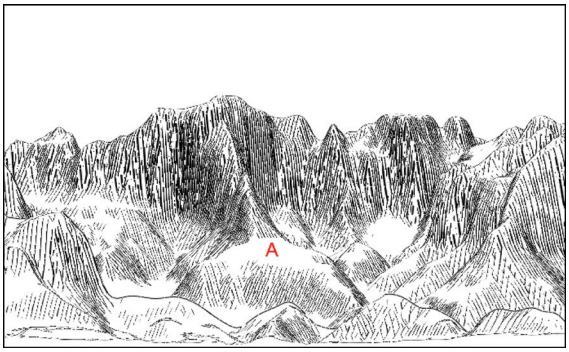

Figure 15. Maximum line width of 3 points. Line width attenuation is weighted by 90% for absolute slope, 10% for lighting. Note the flat region at A is bright.

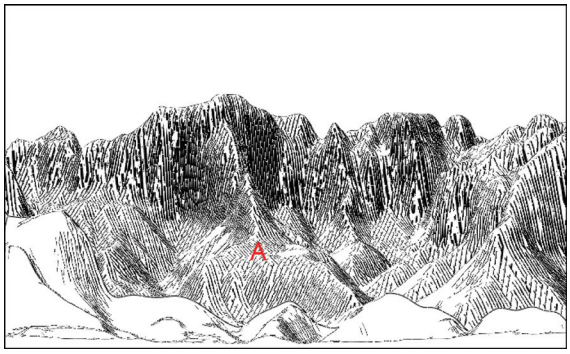

Figure 16. Maximum line width of 3 points but line width attenuation is weighted by 10% for absolute slope and 90% for lighting. The flat region at A is now dark.

In Figure 17 the light source has been rotated 180° in azimuth from that of the other figures. Surfaces facing the lower right (the direction toward the light source) now receive high shading attenuation.

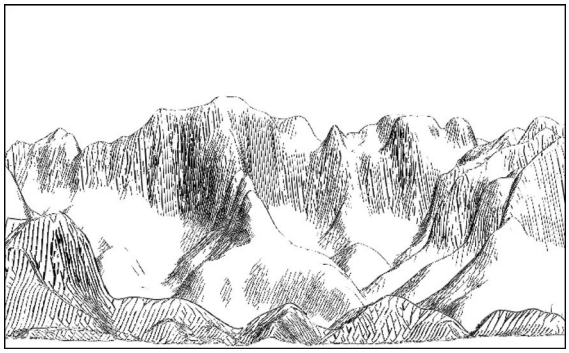

Figure 17. This figure uses the same line width parameters as Figure 13 but with a light source at an azimuth of 70° (lower right). Surfaces facing the light source are highlighted.

Discussion

The images in figures 13 through 17 demonstrate some of the graphic qualities inherent in hand drawn pen and ink style landscape representations, especially with regard to the rendering of attenuated lines of drainage direction. Unfortunately, not all types of landscapes render suitably with this approach. Some landscapes, such as the one depicted by Davis (1908b, p. 28) in Figure 18, would be 'over rendered,' placing far too many lines over a surface best characterized by few. To achieve the intended simplicity demonstrated by this illustration, an automated procedure could follow 2 approaches:

- Define regions over which line density parameters could be imposed and either
- Establish parameters for ranking the importance of individual strokes or
- Create a medial axis from which 2D rays extend perpendicularly out to the region boundary.

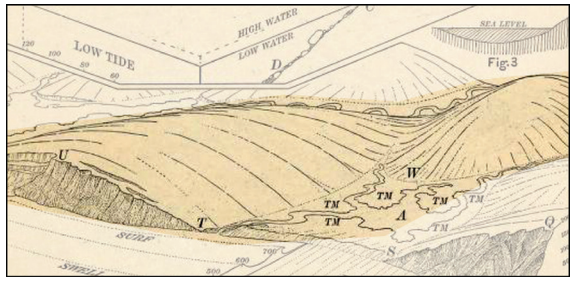

Figure 18. A portion of a W.M. Davis drawing depicting shoreline structures. The features at the edges of the image have been partially masked to highlight the valley and rolling hills at the center.

Both of these approaches could be implemented through the application of a drainage accumulation model. Mower (2009) uses this approach to render drainage lines from an adaptively resampled DEM. Although it successfully models stream segments by flow and renders segments with flows above a user-defined threshold, it operates on a relatively coarse elevation model that requires extensive preprocessing operations to remove false drainage pits and postprocessing operations to smooth vertices of the extracted line segments. Since PenAndInk conducts drainage analyses on the very fine set of vertices retrieved from the smooth polynomial surface tessellation, an accumulation analysis based on those vertices should encounter relatively few false pits and the length of rendered drainage paths should be near the resolution of the frame buffer. Furthermore, the drainage accumulation model can be extended to determine drainage basin boundaries. Once discovered, drainage basins could serve as regions

over which specific surface characteristics, such as optimal linework density, could be calculated.

Imhof (2007, p. 170) suggests that landscape artists lay out 'skeletal lines' composed of hydrological features and slope breaks both in sketches and in completed works. It would be feasible to automate his suggestion by first constructing a medial axis of a drainage basin derived as a B-spline curve of the primary drainage channel (Figure 19). Form lines could then be based on 2D equations of curve normals at fixed or variable intervals along the primary drainage channel, each of which intersects numerous tessellated facets in 2D out to the drainage basin boundary. Renderings could then be constructed for each segment by evaluating the 2D intersections to the B-spline surface and connecting them with straight line segments.

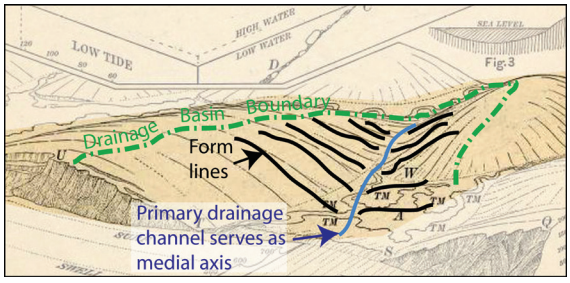

Figure 19. By defining drainage basin regions, a primary drainage channel derived from an accumulation analysis could serve as the medial axis for the region. Form lines could be drawn as perpendicular lines from the medial axis at constant or variable ground distance intervals and evaluated to the polynomial surface once for each intersected tessellation facet.

Once defined, drainage basin parameters could be further used to improve the shading quality of specific landscape features. Imhof (2007, p. 174) notes that pen and ink landscape illustrators continually adjust lighting parameters to compensate for unflattering alignments of features (such as long ridges) with the default light source direction. To address this issue, PenAndInk could apply Brassel's (1974) automated methods for adjusting lighting direction at points, based upon direction coefficients computed over enclosing basins. The maximum amount of deflection of local light source variation from the default could be set from a user-specified value.

All of the major landscape illustrators of the 20th century were aware of the need to match specific drawing techniques with landscape types. How would an expert system make the same determination? Given that the region framed by an illustration could be broken down into discrete drainage basins, measures of roughness (through calculation of fractal dimensions or other measures) could be applied as a 'technique' heuristic to identify rock faces, dunes, or other features (Pentland 1984). In this way, multiple drawing strategies could be applied to a single image of varying landscape types.

Conclusion

This paper has demonstrated a non-photorealistic rendering methodology for topographic surfaces using pen and ink style strokes. It has provided a new technique for rendering form lines from a dense set of drainage direction vectors derived from B-spline surfaces and attenuated by local slope and shading characteristics. Future work in this area will explore automated light source positioning to improve overall feature representation and the addition of drainage accumulation modeling for rendering scenes of low surface roughness.

The author hopes that this work contributes to an understanding of the construction and importance of pen and ink style landscape illustration while renewing interest in the artistry of the work of the 19th and 20th century masters of the discipline.

REFERENCES

Buchin, K., M.C. Sousa, J. Dollner, F. Samavati and M. Walther. 2004. *Illustrating Terrains Using Direction of Slope and Lighting.* 4th ICA Mountain Cartography Workshop in Vall de Núria, Catalunya, Spain, 30 September – 2 October, Online at http://www.mountaincartography.org/ publications/papers/papers_nuria_04/buchin.pdf.

Brassel, K. 1974. A Model for Automatic Hill-Shading. *The American Cartographer* 1(1): 15-27.

CGAL Editorial Board. 2007. CGAL *User and Reference Manual*, ed. 3.3. Online at http://www.cgal.org/Manual/3.3/doc_html/cgal_manual/packages.html.

Davis, W.M. 1908a. *Practical Exercises in Physical Geography*, Ginn and Co., Boston. Online at http://books.google.com/books?id=UpgtAAAAYAAJ&printsec=frontcover&dq=William+Morris+Davis+"practical+exercises+in+physical"&source=bl&ots=jF3oV6BVZu&sig=iWZO5205zXFWNEG_LUo4_ewbfzg&hl=en#v=onepage&q&f=false.

Davis, W.M. 1908b. *Atlas for Practical Exercises in Physical Geography*, Ginn and Co., Boston. Online at http://pds.lib.harvard.edu/pds/view/8916580.

Fernlund, K.J. 2000. *William Henry Holmes and the Rediscovery of the American West*, University of New Mexico Press, Albuquerque.

Imhof, E. 2007. *Cartographic Relief Presentation*, ESRI Press, Redlands, CA.

Lesage, P.L. and M. Visvalingam. 2002. Towards Sketch-Based Exploration of Terrain. *Computers and*

Graphics 26: 309-328.

Little, D. 2006. *Bezier Surfaces. Java applet*, Online at http://www.math.psu.edu/dlittle/java/parametricequations/beziersurfaces/index.html.

Lobeck, A.K. 1939. *Geomorphology: An Introduction to the Study of Landscapes*, McGraw-Hill Book Co., Inc, New York.

Lobeck, A.K. 1958. *Block Diagrams and Other Graphic Methods Used in Geology and Geography*, 2nd ed., Emerson-Trussell Book Company, Amherst, MA.

Mark, D.M. and B. Smith. 2004. A Science of Topography: Bridging the Qualitative-Quantitative Divide In: M. P. Bishop and J Shroder, Chichester (eds.), *Geographic Information Science and Mountain Geomorphology*, England, Springer-Praxis, pp. 75-100.

Markosian, L., M.A. Kowalski, D. Goldstein, S.J. Trychin, J.F. Hughes and L.D. Bourdev. 1997. Real-time Nonphotorealistic Rendering. In *Proceedings of the 24th Annual Conference on Computer Graphics and interactive Techniques*, ACM Press/Addison-Wesley Publishing Co., New York, pp. 415-420.

Maxwell, D.A. and R.D. Turpin. 1968. *Numeric Ground Image Systems Design, Research Report 120-1*, Numerical Ground Image, Research Study Number 2-19-68-120, Texas Transportation Institute, Texas A&M University, College Station, TX.

Mower, J.E. 2008. Non-Photorealistic Rendering of Stylized Block Diagrams, *Proceedings, AutoCarto2008*. The 17th International Research Symposium on Computer-based Cartography, Shepherdstown, West Virginia, USA. September 8-11, 2008. CD-ROM, Cartography and Geographic Information Society, with assistance by ESRI. 10 pp.

Mower, J.E. 2009. Automating Landscape Illustration with Pen and Ink Style Rendering. *Cartography and Geographic Information Science* 36(1): 117-128.

Pentland, A.P. 1984. Fractal-Based Description of Natural Scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 6(6): 661-674.

Piegl, L. and W. Tiller. 1997. *The NURBS Book*, 2nd ed., Springer-Verlag Berlin Heidelberg, Berlin.

Raisz, E. 1948. *General Cartography,* McGraw-Hill Book Co. Inc., New York.

Santella, A. and D. DeCarlo. 2004. Visual Interest and NPR: an Evaluation and Manifesto. NPAR '04, *Proceedings of the 3rd International Symposium on Non-Photorealistic Animation and Rendering*. June 7-9, 2004. ACM, New York. 9 pp.

Shreiner, D., M. Woo, J. Neider and T. Davis. 2007. *OpenGL Programming Guide*, 6th ed., Addison-Wesley Professional, Upper Saddle River, NJ.