

# Topographische Darstellungen digitaler Höhenmodelle mit Elementen nichtphotorealistischer Computergraphik

Roman Geisthövel

Diplomarbeit

Erstbetreuer: Dr. Werner Hansmann

Zweitbetreuerin: Prof. Dr. Leonie Dreschler-Fischer

Universität Hamburg  
Fachbereich Informatik

Oktober 2003

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Die Blockbild-Methode</b>	<b>3</b>
2.1	Überblick . . . . .	3
2.2	Die Linie als Gestaltungsmittel . . . . .	4
2.2.1	Definitionen und Terminologie . . . . .	4
2.2.2	Die Darstellung der Silhouette polygonaler Flächen . . . . .	5
2.2.3	Linien im Bildraum . . . . .	5
2.2.4	Linien im Objektraum . . . . .	7
2.2.5	Hybride Linienalgorithmen . . . . .	11
2.3	Das Gestaltungsmittel Schraffur . . . . .	13
2.3.1	Pixelbasierte Schraffuren . . . . .	13
2.3.2	Geometriebasierte Schraffuren . . . . .	15
2.3.3	Hybride Schraffuralgorithmen . . . . .	17
2.4	Die Erzeugung des Blockbilds . . . . .	19
<b>3</b>	<b>Die Relief-Kontur-Methode</b>	<b>31</b>
3.1	Überblick . . . . .	31
3.2	Die Erzeugung der Konturlinien . . . . .	32
3.3	Die Erzeugung der Reliefkonturen . . . . .	35
<b>4</b>	<b>Die Methode des orthogonalen Reliefs</b>	<b>38</b>
<b>5</b>	<b>Resümee und Ausblick</b>	<b>43</b>
<b>A</b>	<b>Technische Details der Arbeit</b>	<b>44</b>
<b>B</b>	<b>Weitere Beispiele für Blockbilder</b>	<b>44</b>
<b>C</b>	<b>Beispiele für Relief-Kontur-Darstellungen</b>	<b>47</b>
<b>D</b>	<b>Beispiele für orthogonale Reliefs</b>	<b>54</b>
<b>E</b>	<b>Symbolverzeichnis</b>	<b>58</b>

# 1 Einleitung

Topographische Darstellungen sind bildliche Abstraktionen von Geländeformen. Eine solche Darstellung besteht aus Gestaltungsmitteln, *Kartenzeichen* genannt, die einen Raumbezug der bezeichneten Objekte vermitteln [25]. Zu den elementaren Kartenzeichen gehören Punkt, Linie und Fläche. Zusammengesetzte Kartenzeichen sind neben Schrift und Diagrammen Schraffuren und Halbtöne. Die Ansprüche an die Darstellung [30] umfassen die Meßbarkeit bzw. Ablesbarkeit der räumlichen Dimensionen unter der Wahrung hoher Anschaulichkeit. Das graphische Gefüge soll dabei so einfach wie möglich gehalten werden, dem Maßstab entsprechend generalisiert unter Beibehaltung der Aussagekraft und Erfäßbarkeit der Kartenobjekte.

Das Grundproblem der topographischen Darstellung besteht darin, in einem zweidimensionalen Bild die geometrischen Eigenschaften der Lage und Form von dreidimensionalen Oberflächen zu vermitteln. Verschiedene *Tiefenhinweise* werden eingesetzt, um bei der bildlichen Interpretation die Vorstellung von Plastizität zu erzeugen [30, 36]. Sie basieren auf Erfahrungswerten im Umgang mit der physikalischen Welt. Zu ihnen gehören die Schattierung und luftperspektivische Tonwertabstufungen, die in grundrißlichen Kartendarstellungen zur Anwendung kommen. Schattierung bezeichnet die Variation der Flächentönung aufgrund schräg einfallender Lichtstrahlen einer angenommenen Lichtquelle. Durch die lagebedingte Helligkeit oder Verdunkelung lassen sich sehr detailreiche Formen anzeigen. Schlagschatten, Spiegelungen oder Glanzlichter bleiben dabei jedoch unberücksichtigt. Der Effekt der Luftperspektive ist, daß mit der Entfernung Farben weniger gesättigt, kontrastärmer und durch atmosphärische Streuung blauverschoben erscheinen. Eingesetzt wird dies z. B. bei farbigen Höhenstufen – blauhaltige Farben für Täler und Tiefebenen, Töne mit Rotgehalt für Gipfel. Besonders bei kartenverwandten Darstellungen [29] kommen weitere Tiefenhinweise hinzu wie die zentralperspektivisch bedingte scheinbare Verjüngung von Proportionen bei wachsender Entfernung, der Texturgradient oder die Überschneidung bzw. Überdeckung von Objekten.

Die Effekte der in der Kartographie verwendeten Tiefenhinweise müssen durch Auswahl und Kombination geeigneter Kartenzeichen in der Darstellung nachgebildet werden. Dies geschieht meist im Zusammenspiel von mittelbaren und unmittelbaren Gestaltungselementen. Unmittelbar in der Anschaulichkeit, in der Erzeugung eines plastischen Eindrucks, wirkt die Schattierung durch ihre implizite Vermittlung der Form, wobei sie jedoch keine dimensionsmäßige Erfassung geometrischer Sachverhalte erlaubt. Dem gegenüber stehen die symbolartigen, mittelbaren Darstellungsmittel wie die Höhenlinie oder Höhenstufenfarben, die Böschungsschraffe und -schummerung nach dem Prinzip ‚je steiler, desto dunkler‘. Sie besitzen weniger Anschaulichkeit, erleichtern aber wesentlich die Meßbarkeit bzw. das Herauslesen der Dimensionen.

Automatisierte Verfahren zur Erzeugung von Graphiken mittels Punkten, Linien und Halbtönung werden seit über einer Dekade verstärkt unter dem Schlagwort *nichtphotorealistische Computergraphik* (*non-photorealistic rendering*, kurz *NPR*) präsentiert [56]. Nichtphotorealistische Computergraphik ist ein Oberbegriff für eine Reihe von Ansätzen, welche, eben jenseits des photorealistischen Paradigmas, die Bedingungen und die Erzeugung abstrakterer Darstellungen wie Skizzen,

Illustrationen oder Schemazeichnungen, generalisierter graphischer Darstellungen [13] verfolgen. Andere Bezeichnungen spiegeln dabei den jeweiligen Schwerpunkt wieder: *comprehensible rendering* ermöglicht einem Betrachter die erleichterte Perzeption von Gegenständen [50], mit *illustrative rendering* produzierte Bilder stellen Mittel zur Auskunft wie anatomische Illustrationen oder technische Darstellungen [23] bereit, in der *nichtrealistischen Computergraphik* geht es um Visualisierung von Konzepten oder nichtexistenten Objekten. Viele Ansätze haben die Nachempfindung traditioneller künstlerischer oder handwerklicher Stile und Darstellungsmethoden (*stylistic rendering* [1]), oder die Simulation natürlicher Werkstoffe als Ausgangspunkt wie z. B. Tusche-, Kohle- und Bleistiftzeichnungen, Kupferstich und Aquarell [55, 11, 24, 51].

Gemeinsam ist den Ansätzen, für einen gegebenen Sachverhalt die beste Repräsentation anzustreben und somit bildliche Darstellungen, im Hinblick auf die Vermittlung des Sachverhalts, zu optimieren. Es sollen bedeutungsvolle Bilder (*significant images, expressive renderings* [35]) geschaffen werden, die effektive Ausdrucksmittel für den intendierten Verwendungszweck bereitstellen, durch Anpassung der Präsentation an den Kontext.

In dieser Arbeit geht es um den Versuch, durch den Einsatz von Elementen nichtphotorealistischer Computergraphik konstruktiv-zeichnerische topographische Darstellungsarten zu imitieren. Es soll der Frage nachgegangen werden, inwieweit topographische Sachverhalte – die Formung und Dimensionen des Geländes – sachgerecht durch automatische Verfahren vermittelt werden können. Das geschieht am Beispiel von *Blockbildern, Relief-Konturen* und *orthogonalen Reliefs*. Nach der Präsentation der methodenspezifischen Charakteristika werden relevante Algorithmen und Datenstrukturen sondiert, diskutiert und an die Anforderungen der topographischen Darstellungen angepaßt, um diese schließlich zu realisieren. Der Vergleich mit exemplarischen, manuell-erstellten topographischen Werken dient dabei jeweils als erster Gradmesser für den Erfolg der vorgeschlagenen computergraphischen Umsetzung.

Die topographischen Rohdaten werden in der Form *digitaler Höhenmodelle* zugrundegelegt. Ein digitales Höhenmodell beschreibt die Variation eines Geländereliefs im Raum durch die Angabe der Topologie und Wertigkeit von Höhendaten [7].

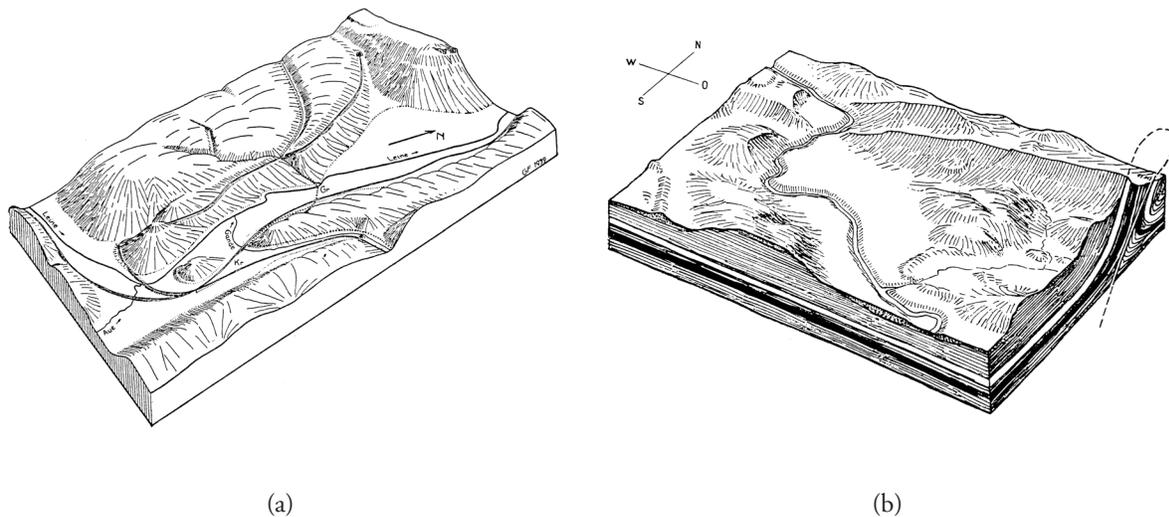


Abbildung 1: Blockbilder. (a) Leinetal bei Kreiensen [21], (b) Durchbruch der Limmat durch den Faltenjura bei Baden (Schweiz) [29].

## 2 Die Blockbild-Methode

### 2.1 Überblick

Das Blockbild ist eine zeichnerische Darstellung der Geländeformung eines räumlich eng begrenzten Ausschnitts der Erdkruste als Vogelschau-Schrägbild [21, 29]. Neben der Ansicht der Landschaftsoberfläche, gestattet das Blockbild die Wiedergabe tektonischer Schichtung auf seitlichen Profilflächen.

Raisz [46] empfiehlt bei der Konstruktion die Verwendung isometrischer Projektion. Andere Kartographen [28, 29] bevorzugen die Zentralprojektion. Der Einwand ist, daß aufgrund fehlender Verjüngung in der Tiefe eine Parallelprojektion einen verzerrten Eindruck der Geländeform hinterläßt, wodurch der Eindruck einer Verbreiterung des hinteren Landschaftsausschnittes entsteht.

Das Blockbild ist eine betont skizzenhafte Darstellung [41], deren Gestaltungsmittel sich typischerweise auf Punkt und Linie beschränken. Diese werden eingesetzt, um die wesentlichen morphologischen Eigenschaften der Landschaftssituation zu veranschaulichen. Die Illustration der Geländeformen erfolgt durch „gut gewählte“ [29] *Kanten-* und *Falllinien*. Kantenlinien als Kartenzeichen sind Indikatoren für Gefällewechsel im Gelände wie Grate (*positive* Kantenlinie) bzw. Schluchten und Rinnen (*negative* Kantenlinie). Falllinien laufen entlang der Richtung maximaler Böschung. Zur Herausarbeitung der Neigungsverhältnisse an Abhängen sowie für die Gestaltung schattenplastischer Eindrücke werden *Schraffuren* benutzt.

Anhand der Betrachtung exemplarischer Blockbilder (Abb. 1(a) und 1(b)) läßt sich ablesen,

was mit „gut gewählten“ Linien gemeint sein kann. Zunächst ist zu bemerken, daß das Blockbild sich mittels einer Umrandung vom Hintergrund abhebt. An zwei Seiten ist dies die Silhouette der Hügel- bzw. Berglandschaft, an den anderen Seiten die Begrenzung der Vertikalschnitte. Auch innerhalb des Landschaftsgefüges werden linienhafte Silhouetten benutzt, um Geländeformen herauszuarbeiten<sup>1</sup>. Desweiteren erkennt man, wie durch einzelne Striche Falten und scharfe Vorsprünge angedeutet werden. Dichtgescharte Falllinien und Schraffuren dienen in beiden Fällen als Formgeber: eng gedrängt zeigen sie eine steile Böschung oder einen lichtabgewandten Hang an, in loser Formation stellen sie eine entsprechend sanfte Steigung dar. Weiße Flächen repräsentieren ebenes Gelände.

Damit bietet sich auch als computergraphisches Gestaltungsmittel die Linie an: einzeln als Silhouette oder Kantenlinie, geschart als Schraffur.

## 2.2 Die Linie als Gestaltungsmittel

Die automatische Herstellung von Strichzeichnungen, insbesondere Silhouetten- und Schraffurzeichnungen, ist ein aktueller Forschungsgegenstand. Einen Überblick geben Isenberg et al. [32].

### 2.2.1 Definitionen und Terminologie

Sei  $S$  eine *reguläre Fläche* [15]. Die Silhouette  $\sigma$  ist die Menge aller Punkte  $\mathbf{P}_i$  auf  $S$ , in denen die Oberflächennormale  $\mathbf{n}_i$  orthogonal zur Blickrichtung eines Betrachters mit Augpunkt  $\mathbf{C}$  ist:  $\sigma = \{\mathbf{X}_i \mid \mathbf{n}_i \cdot (\mathbf{P}_i - \mathbf{C}) = 0\}$ . Dies gilt für den Fall der Zentralprojektion. Bei Parallelperspektive vereinfacht sich die Gleichung zu  $\mathbf{n}_i \cdot \mathbf{v} = 0$ , wobei  $\mathbf{v}$  die Projektionsrichtung ist.

Für *Polygonflächen* [18] ist obige Definition der Silhouette ungeeignet: die Normalen sind nur für die Polygone (ggf. nach Triangulation) wohldefiniert, nicht aber für beliebige Punkte (z. B. Kanten zwischen benachbarten Polygonen). Die Bestimmung der Silhouette eines Objekts, das durch Polygonflächen repräsentiert wird, geschieht wie folgt: Eine Kante wird zur Silhouette gezählt, wenn sie ein der Blickrichtung zugewandtes mit einem abgewandten Polygon (resp. Dreieck) verbindet. Die Menge der Kanten mit dieser Eigenschaft ergibt die Silhouette des Objekts. Für ein Polygon mit Normale  $\mathbf{n}$  gilt: Ist  $\mathbf{n} \cdot \mathbf{v} > 0$  ( $\mathbf{n} \cdot \mathbf{v} < 0$ ), so ist das Polygon der Blickrichtung zugewandt (abgewandt). Bei  $\mathbf{n} \cdot \mathbf{v} = 0$  liegt das Polygon parallel zur Blickrichtung.

Eine *Kontur* ist die Untermenge der Silhouette, die ein Objekt von der Umgebung abhebt. Die Silhouette, die nicht zur Kontur gehört, wird *innere* (engl. *internal*) Silhouette genannt.

*Formkanten* verbinden Polygone, deren Flächenwinkel einen bestimmten Schwellenwert über- bzw. unterschreitet.

---

<sup>1</sup>siehe z. B. den Bergkamm in west-östlicher Richtung in Abb. 1(b).

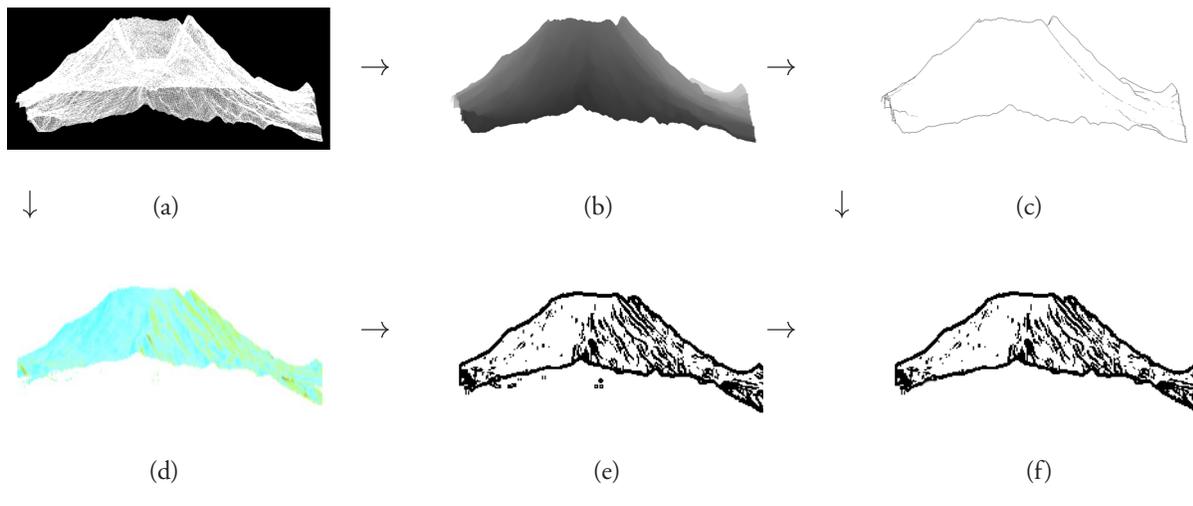


Abbildung 2: Liniendetektion im Bildraum. Ein Polygonmodell (a) wird gerendert, Unstetigkeitsstellen im Tiefenpuffer (b) korrespondieren zur Silhouette (c). Unstetigkeitsstellen im normal-buffer (d) entsprechen den Formkanten (e). (f) Überblendung von (c) und (e).

## 2.2.2 Die Darstellung der Silhouette polygonaler Flächen

Im Weiteren wird davon ausgegangen, daß Objekte, deren Silhouette ermittelt werden soll, als triangulierte Polygonflächen vorliegen, die *orientierbar* und *konsistent* [15] sind.

Zur Darstellung eines Objekts mittels seiner Silhouette sind drei wesentliche Schritte erforderlich:

- Identifikation der Menge der Silhouettenkanten
- Bestimmung der sichtbaren Teilmenge in Bezug auf einen Betrachter
- Präsentation der Ergebnisse

Hinsichtlich der Vorgehensweise bei der Identifikation lassen sich *Objektraum-* (engl. *object space*), *Bildraum-* (engl. *image space*) und *hybride* Algorithmen unterscheiden. Die Begriffe ‚object space‘ und ‚image space‘ beziehen sich auf Koordinatensysteme, wie sie in den Abschnitten einer üblichen Graphik-Pipeline zur Rasterung von Polygonen [20] verwendet werden. Bildraum-Algorithmen sind pixelorientiert und verarbeiten gerenderte Bilddaten im Gerätekoordinatensystem, Objektraum-Algorithmen dagegen analytisch-geometrische Modelle mit Welt- oder Kamerakoordinaten. Es folgt ein Überblick der einzelnen Verfahren.

## 2.2.3 Linien im Bildraum

Bildraum-Algorithmen zur Identifikation von Silhouetten und Formkanten bedienen sich Bildverarbeitungsmethoden: durch den Rasterungsprozess erhaltene Bildpuffer werden mittels Diffe-

renzenoperatoren auf ‚Unstetigkeitsstellen‘ (i. e. abrupte Wechsel in den Grauwerten benachbarter Pixel) überprüft, die mit den gesuchten Kanten korrespondieren. Diese Kanten werden dann durch Elemente einer Pixel-Matrix repräsentiert.

Die von Bildraum-Algorithmen benutzten Datenstrukturen – zweidimensionale Bildpuffer – lassen sich leicht mittels gebräuchlicher Graphik-Hardware und -Software generieren. Die algorithmische Komplexität dieser Verfahren ist eine Funktion der Anzahl der Pixel im Bildpuffer, unabhängig von der Anzahl der Polygone, die zur Repräsentation eines Objekts benutzt werden.

Zu den Vorzügen von Bildraum-Verfahren zählt die Allgemeinheit der Anwendbarkeit, d. h. von der Repräsentation der geometrischen Modelle (parametrisiert, implizit, polygonal etc.) kann abstrahiert werden, soweit diese eine Rasterung zulässt. Dieser Umstand bedingt auch die Robustheit gegenüber ‚Anomalien‘ in der Modellierung: Löcher oder Selbstüberschneidungen in den Modellen beeinträchtigen nicht den Verfahrensverlauf, da nur Pixel verarbeitet werden. Wenigstens teilweise lassen sich Bildraum-Algorithmen durch den Einsatz von Graphikhardware beschleunigen. Dazu zählen moderne Graphikadapter und insbesondere auch durch Schattierungsprogramme modifizierbare Graphikprozessoren.

Charakteristisch ist das Fehlen einer analytischen Repräsentation der Ausgabe von Bildraum-Algorithmen. Eine solche ist nur mittels zusätzlichen Aufwands zu erhalten, beispielsweise durch Raster-Vektor-Transformation. Desweiteren bedarf es der Behandlung von Alias-Effekten, hervorgerufen durch das beschränkte Auflösungs- und Quantisierungsvermögen der verwendeten Puffer.

Saito und Takahashi [50] benutzen den Tiefenpuffer, um die Silhouette eines gerasterten Objekts zu extrahieren; als Kantendetektor dient der Sobel-Operator. Dieser liefert Unstetigkeitsstellen nullter Ordnung ( $C^0$ -Unstetigkeitsstellen), die mit der Kontur des Objekts übereinstimmen. Mittels des Laplace-Operators werden Unstetigkeitsstellen erster Ordnung ( $C^1$ -Unstetigkeitsstellen) im Tiefenpuffer identifiziert. Diese zeigen Formkanten an.

Decaudin [12] spricht sich gegen die Verwendung eines Differenzenoperators zweiter Ordnung aus, da dieser zu approximativ ist, um zuverlässig Formkanten zu lokalisieren, besonders bei einem Tiefenpuffer geringer Darstellungsgenauigkeit. Stattdessen benutzt er einen Differenzenoperator erster Ordnung, um  $C^0$ -Unstetigkeitsstellen in einem RGB-Bild (*normal buffer*) zu finden, dessen drei Farbkomponenten für jedes Pixel die  $(x, y, z)$ -Koordinaten der Oberflächennormalen des an der Stelle sichtbaren Objekts kodieren. Die entdeckten Kanten entsprechen Formkanten. Der normal-buffer kann auch zur Bestimmung der Silhouette verwendet werden [48]. Für jedes Pixel  $(R, G, B)$  des normal-buffers wird dazu das Skalarprodukt  $s = Rv_1 + Gv_2 + Bv_3$  gebildet, wobei  $(v_1, v_2, v_3)$  die Blickrichtung ist. Die Silhouette ist dann die Menge der Pixel mit  $s = 0$ .

Die genannten Techniken profitieren von dem Gebrauch spezieller Hardware und der Verwendung programmierbarer Graphikprozessoren. Mitchell et al. benutzen einen Vertex-Shader<sup>2</sup>, um eine RGBA-Textur mit  $(x, y, z)$ -Normalenkoordinaten und Tiefenwerten eines Objekts zu füllen

---

<sup>2</sup>auf ATI RADEON Graphikprozessor mit OpenGL-Erweiterungen `GL_ARB_vertex_program` und `GL_ARB_fragment_program`

[40]. Ein Kantendetektor, implementiert durch einen Pixel-Shader, lokalisiert Objektsilhouetten (in der Textur-RGB-Komponente) und Formkanten (in der Alpha-Komponente). Diese können noch im Rasterungsprozess mit dem Objekt kombiniert werden.

Ein wesentlicher Nachteil der Bildraum-Methode ist, daß die nachträgliche Modifikation lokalisierter Kanten, bedingt durch deren Repräsentation als Bildpunkte in einer Matrix, sich wiederum auf Bildverarbeitung einschränkt.

Mitchell et al. setzen z. B. einen Dilatationsoperator ein, um Kanten nachträglich zu verbreitern [40]. Weitergehende Attributierung jedoch, vor allem hinsichtlich künstlerischer Zeichnungstechniken, setzt typischerweise eine analytische Repräsentation der Kanten voraus.

Das Sichtbarkeitsproblem, d.h. Bestimmung der sichtbaren Kanten in Bezug auf einen Betrachter, wird bei der beschriebenen Art der Kantendetektion trivial gelöst: die verwendeten Bildpuffer enthalten Rasterdaten, die bereits Ergebnis von Sichtbarkeitstests vorangehender Abschnitte der Graphik-Pipeline sind. Die Menge der lokalisierten Kanten ist damit zugleich die Menge der sichtbaren Kanten.

## 2.2.4 Linien im Objektraum

Die Grundmethode zur Bestimmung der Silhouette im Objektraum besteht darin, alle Polygone hinsichtlich ihrer Ausrichtung in Bezug auf die Kamera zu klassifizieren, um dann die entsprechenden Kanten auszuwählen, gemäß der Definition der Silhouette polygonaler Flächen (siehe Abschnitt 2.2.1). Diese Schritte müssen nach jeder Änderung des Ansichtsvolumens oder des Objekts wiederholt werden, bei Animationen typischerweise für jeden Frame. Der Aufwand ist linear in Abhängigkeit von der Anzahl der Polygone und Kanten des Objekts; für Anwendungen mit Echtzeit-Anforderungen oder mit Objekten hoher Polygondichte kann dies unzureichend sein. Die Vorzüge der Methode liegen in der Einfachheit der Implementierung und der Eignung zum Einsatz bei sowohl Zentral- als auch Parallelprojektion. Die Methode ist vollständig: alle Silhouettenkanten werden gefunden. Wie bei den übrigen Objektraumalgorithmen ist das Ergebnis eine analytische Silhouettenrepräsentation, die für elaborierte Liniengestaltung bei der Darstellung grundlegend ist.

Vorschläge zur effizienten Realisierung der Grundmethode umfassen die Verwendung spezieller Datenstrukturen zur Kennzeichnung von Silhouettenkanten, sowie die Verlagerung der Schritte ‚Polygonklassifikation‘ und ‚Kantenauswahl‘ in den (programmierbaren) Graphikprozessor.

Buchanan und Sousa [6] benutzen eine *edge-buffer* genannte Datenstruktur, die eine kompakte Markierung von Polygonkanten hinsichtlich der Silhouettenzugehörigkeit ermöglicht. Dazu werden im *edge-buffer*, implementiert als Hashtabelle, für jede Kante zwei Bits ( $F^c$  und  $B^c$ ) gespeichert, die anzeigen, ob eine angrenzende Fläche der Kamera zu- ( $F = 1$ ) oder abgewandt ( $B = 1$ ) ist. Zu Beginn eines Frames haben beide Bits den Wert Null. Für jedes Polygon wird die Ausrichtung bestimmt und das entsprechende Bit seiner Kanten mittels XOR gesetzt ( $F \oplus 1$  bzw.

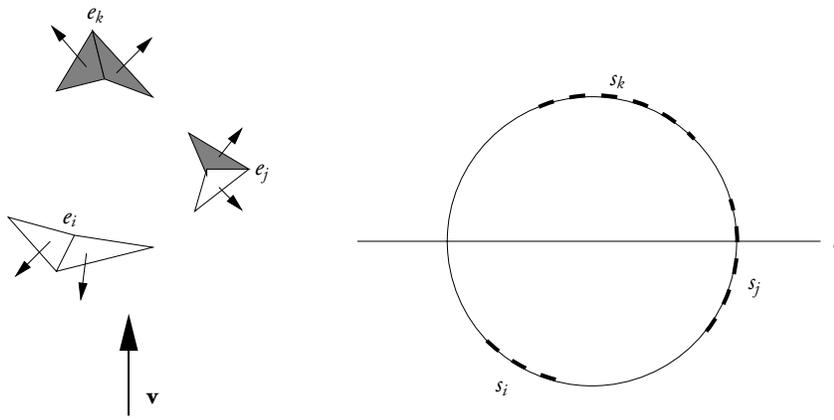


Abbildung 3: Urbilder und Bilder der Gauß-Abbildung im Querschnitt der Einheitssphäre. Links eine Szene mit drei Paaren von Polygonen, jeweils verbunden durch die Kante  $e_i$ ,  $e_j$  bzw.  $e_k$ . Der Pfeil  $\mathbf{v}$  zeigt die Blickrichtung an; dunkle Polygone sind abgewandt. Rechts sind die Bilder der Polygonnormalen unter der Gauß-Abbildung als durch die Geodäten  $s_i$ ,  $s_j$  bzw.  $s_k$  verbundene Punkte dargestellt. Die Schnittebene  $l$ , repräsentiert als horizontale Linie, korrespondiert zu einer Parallelprojektion der Szene. Die Kante  $e_j$  gehört zu einer Silhouette, da  $l \cap s_j \neq \emptyset$ .

$B \oplus 1$ ). Eine Kante wird genau dann zur Silhouette gezählt, wenn beide Bits den Wert Eins haben.

Card und Mitchell [9] konstruieren zu jeder potentiellen Silhouettenkante  $e_s = (v_{s1}, v_{s2})$  ein zusätzliches Viereck  $[(v_{s1}, v_{s2}), (v_{s2}, v_x), (v_x, v_y), (v_y, v_{s1})]$  längs der Kante, wobei  $(v_{s2}, v_x)$  und  $(v_x, v_y)$  die Länge Null haben und  $(v_x, v_y)$  kongruent zu  $e_s$  ist. Die zusätzlichen Knoten  $v_x$  und  $v_y$  dienen nur der Speicherung der Oberflächennormalen der zu  $e_s$  adjazenten Polygone. Beim Rendern der Szene überprüft ein Vertex-Shader-Programm, ob die Normalen von  $v_x$  und  $v_y$  der Blickrichtung zu- oder abgewandt sind. Wenn das Ergebnis für beide unterschiedlich ausfällt, ist  $e_s$  eine Silhouettenkante.

Mehrere Autoren benutzen das Konzept der *Raumpartitionierung* als Grundlage der Entwicklung effizienter Algorithmen zur Detektion von Silhouetten. Bei einigen Ansätzen wird dabei der projektive Raum als Erweiterung des Objektraums zugrunde gelegt. Gooch et al. [23] betrachten die Polygonkanten indirekt durch eine Gauß-Abbildung [15]. Die Gauß-Abbildung  $N : S \mapsto \mathbb{R}^3$  liefert für einen Punkt der Fläche  $S$  den Einheitsnormalenvektor an diesem Punkt mit Werten in der Einheitssphäre  $S^2 = \{(x, y, z) \in \mathbb{R}^3 \mid x^2 + y^2 + z^2 = 1\}$ . Die Normalen von zwei angrenzenden Polygonen werden durch Punkte auf der Einheitssphäre repräsentiert. Diese Punkte seien durch eine Geodäte verbunden. Einer Ebene durch den Sphärenmittelpunkt kann eine Parallelprojektion der Szene zugeordnet werden. Die Ebene trennt die Normalen der dem Betrachter zugewandten Polygone von denen der abgewandten Polygone. Jede Geodäte, die die Ebene schneidet, korrespondiert somit zu einer Silhouettenkante. Die Gauß-Abbildung und Geodäten werden

als singuläre Vorverarbeitungsschritte berechnet; nur die Ebene und ihre Schnittpunkte mit den Geodäten müssen bei wechselnden Kameraparametern ermittelt werden. Damit beschränkt sich der Aufwand zur Laufzeit praktisch nur auf den Schritt der Kantenauswahl. Die Methode ist bei Zentralprojektion jedoch nicht anwendbar.

Folgende Methode läßt sich sowohl bei Parallel- als auch bei Zentralprojektion verwenden. Hertzmann und Zorin [27] bedienen sich des Konzepts der *dualen Flächen* [45] zur schnellen Silhouettendetektion. Die Punkte der dualen Fläche  $S'$  zu einer Fläche  $S$  sind die Bilder der Tangentialebenen  $T_{\mathbf{P}}$  von  $S$  unter der *Dualitätsabbildung*  $f$ , die jede Ebene  $ax+by+cz+d=0$  auf den Punkt mit homogenen Koordinaten  $(a, b, c, d)$  im projektiven Raum abbildet. Sei  $\mathbf{n} = (n_1, n_2, n_3)$  die Einheitsnormale an der Stelle  $\mathbf{P} \in S$ , dann ist  $f(T_{\mathbf{P}}) = (n_1, n_2, n_3, -((\mathbf{P} - \mathbf{0}) \cdot \mathbf{n}))$ . Das Bild der Silhouette einer Fläche  $S$  unter der Dualitätsabbildung in Bezug auf einen Augpunkt  $\mathbf{C} = (c_1, c_2, c_3, c_4)$  ist der Durchschnitt der Ebene  $c_1x + c_2y + c_3z + c_4 = 0$  (der dualen Ebene des Augpunktes) mit der dualen Fläche  $S'$ . Damit reduziert sich das Problem der Silhouettendetektion auf die Bildung des Durchschnitts einer Ebene mit einer Fläche. Da die Dualitätsabbildung Fernpunkte enthalten kann, i. e. Punkte  $(x, y, z, 0)$ , wird der Durchschnitt im projektiven Raum gebildet.

Pop et al. [43] benutzen eine alternative Dualitätsfunktion, die ein planares Polygon in der Ebene  $ax + by + cz + d = 0$  auf den Punkt  $\mathbf{P} = (a/d, b/d, c/d)$  abbildet. Der Kante zwischen zwei angrenzenden Polygonen korrespondiert ihre *duale Kante*, die die dualen Punkte der beiden Polygone verbindet. Eine weitere Funktion  $f$  bildet einen Punkt auf seine duale Ebene ab. Das Bild des Augpunktes  $\mathbf{C} = (c_1, c_2, c_3)$  unter  $f$  ist die Ebene  $c_1x + c_2y + c_3z + 1 = 0$ . Grundlage des Verfahrens von Pop et al. ist die Beobachtung, daß eine Polygonkante genau dann zur Silhouette gehört, wenn ihre duale Kante die duale Ebene des Augpunktes schneidet.

Sander et al. [53] erstellen eine Hierarchisierung der zu überprüfenden Kanten eines Polygonmodells durch die Einordnen der Kanten in einen Suchbaum. Für einen Knoten  $n$  im Baum sei die Menge aller Polygone, die Kanten aus  $n$  sowie den Nachfolgern von  $n$  haben, der *face-cluster*  $F(n)$ . Die Menge aller Punkte auf der zugewandten Seite eines planaren Polygons  $f$  sei  $frontfacing(f) = \{\mathbf{P}_i | (\mathbf{P}_i - \mathbf{P}_f) \cdot \mathbf{n}(f) \geq 0\}$ , wobei  $\mathbf{P}_f \in f$  und  $\mathbf{n}(f)$  die Normale von  $f$  ist. Bei dem Verfahren von Sander et al. wird zur Laufzeit der Baum rekursiv nach Kanten durchsucht, die Teil der Silhouette sind. Gilt dabei das Prädikat  $P = \mathbf{V} \in frontfacing(F(n)) = \bigcap_{f \in F(n)} frontfacing(f)$  bzw.  $\neg P$  für einen Knoten  $n$ , wobei  $\mathbf{V}$  der Augpunkt ist, so brauchen die Kanten der Nachfolgeknoten von  $n$  nicht mehr auf Silhouettenzugehörigkeit geprüft zu werden. Zur effizienten Berechnung von  $P$  wird in jedem Knoten ein Paar von Kegeln gespeichert, die die Regionen  $frontfacing(F(n))$  bzw.  $\neg frontfacing(F(n))$  approximativ eingrenzen.

Eine weitere Möglichkeit zur Effizienzsteigerung liegt in der Verringerung der zu prüfenden Polygone bzw. Kanten zur Laufzeit.

Lake et al. [34] präsentieren in einem System zur Illustration von Polygonmodellen eine Kombination von graduelltem Detailniveau (*continuous level-of-detail*) mit der edge-buffer-Datenstruktur von Buchanan und Sousa, um interaktive Bildwiederholungsraten zu erreichen.

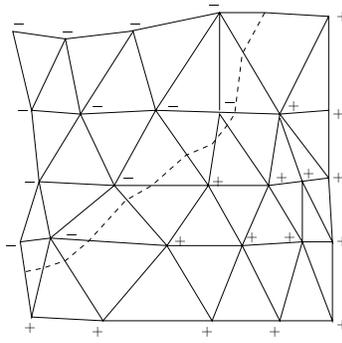


Abbildung 4: Subpolygonale Silhouette. Jeder Dreiecksknoten wird mit dem Vorzeichen des Skalarprodukts aus Blickrichtung und Normale versehen (Null gilt hierbei als positiv). Auf Kanten zwischen Knoten mit unterschiedlichen Vorzeichen wird durch lineare Interpolation der Punkt bestimmt, in dem das Skalarprodukt Null wird. Die Verbindung der Punkte über alle Dreiecke ergibt die Silhouette [26].

Typischerweise sind nur wenige Kanten eines Polygon-Modells auch Silhouettenkanten. Sander et al. sprechen von einer Größenordnung von  $O(\sqrt{n})$  bei  $n$  Polygonen [53]. Diesen Umstand nutzen Markosian et al. bei einer stochastischen Methode zur Silhouettendetektion [38]. Eine kleine Untermenge der Kanten im Modell wird zufällig ausgewählt. Es wird geprüft, ob darin Kanten enthalten sind, die zu der Silhouette gehören. Wird eine solche Kante entdeckt, erfolgt die weitere Suche entlang adjazenter Kanten. Die Wahrscheinlichkeit, einen zur Silhouette gehörigen Kantenzug zu entdecken, ist proportional zu dessen Länge. Das Verfahren lokalisiert einen Großteil der Silhouette mit ausreichender Geschwindigkeit für interaktive Anwendungen.

Wird ein Objekt durch Polygone approximiert, kann die Darstellung seiner Silhouette unerwünschte Artefakte enthalten wie z. B. kantige Übergänge. Zur Konstruktion einer möglichst objekt-treuen Silhouette wäre es sinnvoll, nicht nur Kanten zwischen, sondern auch solche innerhalb der Polygone in Betracht zu ziehen. Hertzmann verfolgt diesen Ansatz [26] unter Benutzung triangulierter Polygone: für den Knoten jedes Dreiecks wird das Skalarprodukt  $d$  aus Normale und Blickrichtung gebildet (die Normalen erhält man durch Mittelung der Normalen der angrenzenden Flächen). Für alle Kanten, deren Knoten unterschiedliche Vorzeichen des Skalarprodukts aufweisen (+, für  $d \geq 0$ , -, sonst), wird durch lineare Interpolation der Punkt bestimmt, an dem  $d$  gleich Null ist. Die Verbindung dieser Punkte ist ein Kantenzug, der weniger Artefakte enthält und eine bessere Annäherung an die Silhouette des Objekts als die der trivialen Methode ergibt (Abb. 4).

Eine alternative Herangehensweise bei der Silhouetten- und Formkantendetektion, die mehr Flexibilität in der Auswahl der darzustellenden Kanten als die bisher genannten Ansätze bietet, ist das *Virtual Lights*-Modell von Martín und Torres [39]. Sie definieren die Kontur und

Formkanten über Beleuchtungsverhältnisse: die Kontur ist die Grenze zwischen beleuchteten und unbeleuchteten Objektteilen. Befinden sich Lichtquelle und Beobachter an gleicher Position, so entspricht die Kontur, gleich der obigen Definition, dem Umriß des Objekts. Formkanten trennen Bereiche unterschiedlicher Beleuchtungsintensität. Die Identifikation der interessierenden Kanten geschieht in dem Modell nicht mittels Lichtquellen, sondern durch sogenannte virtual lights, die für jedes Objekt der Szene erstellt werden. Ein virtual light beeinflusst nicht die Schattierung eines Objekts, sondern dient nur zur Berechnung der Intensität des reflektierten Lichts von der Oberfläche. Durch Wahl eines Beleuchtungsmodells mit diffuser Reflexion erhält man Konturlinien und Formkanten, die von der Position des virtual light abhängen. Bei Hinzunahme spiegelnder Reflexion lassen sich Formkanten in Abhängigkeit von der Beobachterposition hervorheben.

Es existiert eine ganze Reihe von Algorithmen zur Lösung des Sichtbarkeitsproblems im Objektraum (vgl. Sutherland et al. [57]). Exemplarisch wird hier der Ansatz von Appel [2] vorgestellt. Der Algorithmus basiert auf dem Begriff der *quantitativen Unsichtbarkeit* (*quantitative invisibility*) ( $QU$ ), der Anzahl aller Polygone, die dem Betrachter zugewandt sind, und zwischen diesem und einem Punkt einer Kante liegen, die gerendert werden soll. Ein Kantensegment ist genau dann sichtbar, wenn  $QU = 0$ . Der  $QU$ -Wert auf einer Kante verändert sich, wenn diese eine Silhouette auf der Bildebene schneidet. Ein durch Strahlverfolgung bereits ermittelter  $QU$ -Wert von Null kann deshalb auf adjazente Kantensegmente übertragen werden, solange diese keine Bildraum-Silhouette schneiden.

### 2.2.5 Hybride Linienalgorithmen

Hybridalgorithmen vereinen Objektraum- und Bildraumtechniken; sie benötigen meist mehrere Rasterungsdurchläufe pro Frame. Exemplarisch wird hier das Verfahren von Raskar und Cohen [47] vorgestellt.

Zur Darstellung der Silhouette benutzen Raskar und Cohen Polygon-Culling und den Tiefenpuffer. Im Objektraum erfolgt die Bestimmung zweier Mengen  $P_1$  und  $P_2$  der dem Betrachter zugewandten bzw. abgewandten Polygone. Diese werden verwendet, um im Bildraum durch Bildung der Schnittmenge  $P_1 \cap P_2$  mittels Anpassung der Tiefenpufferfunktion  $f$  die Silhouette zu erhalten. Ein Pixel mit Tiefenwert  $z_1$  überschreibt ein  $z_2$  im Tiefenpuffer, wenn der Wert der Tiefenpufferfunktion  $f(z_1, z_2) = \text{wahr}$  ist. Das Verfahren zur Darstellung einer schwarzen Objektsilhouette auf weißem Hintergrund lautet in Pseudocode:

1. Tiefenpuffer mit  $z_i = \infty$  initialisieren
2. weißen Hintergrund erzeugen
3.  $f(z_1, z_2) := z_1 < z_2$
4. Polygone aus  $P_1$  in Weiß rastern
5.  $f(z_1, z_2) := z_1 = z_2$
6. Polygone aus  $P_2$  in Schwarz rastern

Die Nachteile der Methode resultieren aus den Mängeln der limitierten Darstellungsgenauigkeit von Tiefenpufferwerten; durch Rundungsfehler bei der Quantisierung der Tiefenwerte ist es möglich, daß die Schnittmenge  $P_1 \cap P_2$  unzusammenhängend oder sogar leer ist. Die Autoren schlagen deshalb vor, die  $P_2$ -Polygone vor der Rasterung in Richtung der Kamera zu verschieben und die Tiefenpufferfunktion ‚ $\leq$ ‘ zu verwenden. Die Translation kann um einen festen Betrag erfolgen oder proportional zur Entfernung von der Kamera ansteigen, um den Effekt des nichtlinearen Auflösungsvermögens<sup>3</sup> des Tiefenpuffers zu mindern. Durch diese Maßnahmen wird die darzustellende Schnittmenge vergrößert und somit die Auswirkung von Rundungsfehlern bei der Tiefenwertequantisierung abgeschwächt.

---

<sup>3</sup>einem Tiefenintervall kleiner Entfernung stehen mehr Werte zur Verfügung als dem gleichen Intervall bei großer Entfernung

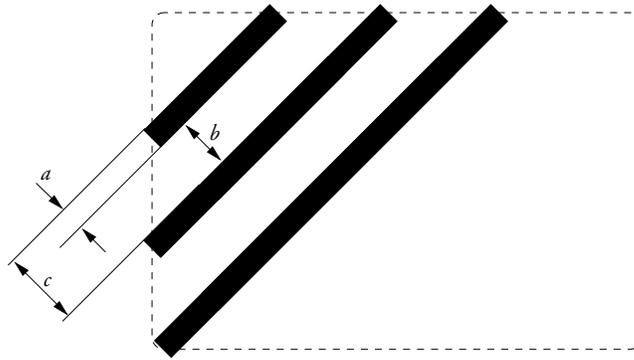


Abbildung 5: Elemente der Schraffur. Strichbreite  $a$ , Zwischenraum  $b$ , Schraffurweite  $c = a + b$  (nach Bollmann et al. [4])

## 2.3 Das Gestaltungsmittel Schraffur

Die Schraffur als graphisches Mittel zur Darstellung einer Fläche ist eine regelhafte Anordnung von Linien, bei der die einzelnen Striche noch sichtbar sind, deren Verbund aber einen Flächeneindruck erwirkt [4]. Die Schraffur vermittelt einen *Tonwert* durch das Verhältnis von Strichbreite zur Schraffurweite (siehe Abbildung). Der Tonwert ist ein Indikator der Beleuchtungsverhältnisse. Bei Rasterabbildungen ist er der prozentuale Anteil der geschwärtzten Fläche an der Gesamtfläche. Durch die Neigung der Linien wird eine *Ausrichtung* bzw. *Orientierung* der Vorlage angezeigt. Zusammen bewirken die Form und Lage der Striche den Eindruck einer *Texturierung*, die auf Materialeigenschaften der schraffierten Oberfläche schließen lässt, sowie Tiefenhinweise durch den Texturgradienten liefert.

Ähnlich der Klassifikation von Silhouettenverfahren erfolgt auch hier die Einteilung der rechnergestützten Erzeugung von Schraffurdarstellungen in pixelorientierte-, analytische oder geometriebasierte- und hybride Verfahren. Die Automatisierung umfasst die Erzeugung der Vielzahl von Schraffurstrichen in Abhängigkeit von Tonwertvorgaben.

Die Ausrichtung erfolgt anhand eines Richtungs- oder Vektorfelds, das bei den pixelorientierten Verfahren meist interaktiv spezifiziert werden muß, und bei den analytischen Methoden automatisch berechnet werden kann. Die Wahl fällt dabei oft auf die Orientierung entlang von Krümmungslinien. Es folgt ein Überblick der Arbeiten im Rahmen der genannten Einteilung.

### 2.3.1 Pixelbasierte Schraffuren

Eine einfache Methode der Schraffurerzeugung basiert auf Halbtönung mittels Ditherscreens [56]. Die binäre Approximation gradueller Grauwerte führt zu dem Auftreten visueller Artefakte im Halbtonbild. In klassischen Arbeiten zur Halbtönung wird das das Augenmerk auf die Minimierung dieser Artefakte gerichtet [19]. Veryovka und Buchanan [63, 62] präsentieren einen

Ansatz zur Erweiterung von Grauwertbildern um künstlerische Gestaltungselemente wie Schraffuren durch den gezielten Einsatz und Steuerung der Artefakte im Halbtönungsprozess. Grundlage ist eine Textur, die aus der gewünschten Schraffur besteht. Die Textur wird in eine Dithermatrix transformiert, indem sie in kleine Blöcke eingeteilt und für jeden Block eine adaptive Histogramm-Ebnung (engl. *adaptive histogram equalization*) durchgeführt wird. Durch die beiden Schritte wird sichergestellt, daß die wichtigen Eigenschaften räumlicher und quantitativer Gleichverteilung der Schwellenwerte der Dithermatrix erfüllt sind.

Die Textur, resp. Dithermatrix, wird durch eine Funktion  $M_{s,\vartheta}$  auf einen Ditherscreen abgebildet, der zur Halbtönung eines Grauwertbilds  $G$  verwendet werden kann. Die Funktion  $M_{s,\vartheta}$  rotiert und skaliert die Texturkoordinaten anhand zweier Funktionen  $s$  und  $\vartheta$ , die durch Referenzbilder definiert sind. Für  $s$  kann beispielsweise  $G$  verwendet werden: dunkle Grauwerte bewirken eine Skalierung zwischen Null und Eins, was zu einer Verminderung des Schraffurzwischenraums und korrespondierender Verdunkelung im Ditherscreen führt; entsprechend vergrößern helle Grauwerte den Schraffurzwischenraum. Zur Steuerung der Schraffurorientierung enthält  $\vartheta$  z. B. die Richtungen orthogonal zum Gradienten von  $G$ .

Ein weiteres pixelorientiertes Verfahren zur Herstellung von Strichzeichnungen präsentieren Saito und Takahashi [50]. Der *G-Buffer* ist die grundlegende Datenstruktur, ein zweidimensionales Feld von Rasterdaten, deren Werte geometrische Attribute eines Objekts kodieren. Ein G-Buffer entsteht durch Projektion einer Szene durch eine Betrachtungstransformation. Ein Beispiel für einen G-Buffer ist der Tiefenpuffer. Die Schraffierung einer parametrisierten Fläche basiert auf einem Verfahren<sup>4</sup> zur Erzeugung von Isolinien mittels Bildverarbeitungsoperationen. Folgende G-Buffer werden verwendet: ein Puffer für jede Koordinate der Oberflächennormalen  $(nx, ny, nz)$ , und je einer für die Kodierung der parametrischen Koordinaten der Oberflächenteile  $(ou, ov)$ . Durch Anwendung des Isolinienverfahrens auf  $ou$  und  $ov$  werden zwei neue Puffer  $cu$  und  $cv$  generiert, die jeweils Linien entlang der parametrischen Koordinaten enthalten. Aus  $nx, ny$  und  $nz$  generiert man bei Wahl eines Beleuchtungsmodells einen Puffer  $sh$  des schattierten Objekts. Durch Überblendung der Puffer lassen sich verschiedene Schraffierungen erzeugen: bei der Kombination von  $cu$  bzw.  $cv$  mit  $sh$  erhält man eine Schraffierung in einer Richtung mit Tonwerten in Abhängigkeit von der Schattierung,  $cu$  und  $cv$  und  $sh$  zusammen ergeben eine Kreuzschraffierung mit orthogonalen Linien.

Ein System zur interaktiven Erstellung von Schraffurdarstellungen präsentieren Salisbury et al. [51]. Die Interaktion besteht in der Festlegung der Schraffurparameter Tonwert und Textur, wahlweise ‚freihändig‘ oder mit Hilfe eines Referenz-Grauwertbilds des zu illustrierenden Objekts. Tonwert und Textur werden durch Auswahl einer eingebauten oder prozeduralen *Strichtextur* (engl. *stroke texture*) bestimmt, welche dann mit einem Pinselwerkzeug aufgebracht wird. Eine Strichtextur ist eine Schar von Linien. Die Erzeugung der einzelnen Linien wird vom System übernommen. Dabei werden die Linienparameter wie Länge oder Krümmung zufällig innerhalb

---

<sup>4</sup>eine ausführliche Beschreibung des Verfahrens findet sich in Kapitel 3.

kleiner Intervalle variiert, um einen artifiziellen Eindruck zu vermeiden. Eine Orientierung der Strichtextur ist möglich mittels Ausrichtung anhand des Gradienten des Referenzbildes. In einer folgenden Arbeit [52] beschreiben Salisbury et al. ein interaktives System mit erweiterter Möglichkeit zur Festlegung der Schraffurorientierung. Auf einem Referenzbild, dessen Grauwerte zur Bestimmung der gewünschten Tonwerte der Illustration herangezogen werden, generiert der Benutzer mit unterstützenden Editorfunktionen ein Richtungsfeld, das die Gestalt und Form des unterliegenden Objekts vermittelt. Ein einzelner Strich wird durch eine kubische B-Splinekurve repräsentiert. Die Orientierung geschieht durch Abbildung des Kontrollpolygons der Kurve in das Richtungsfeld. Zum Tonwertabgleich zwischen der Vorlage und der Illustration wird nach dem Zeichnen eines Strichs die Differenz, durch pixelweise Subtraktion der Grauwerte, zwischen dem Referenzbild und einem durch Filterung der Illustration gewonnenem Unschärfbild berechnet. Die Filterungsoperation ist eine Mittelung durch einen Faltungskern mit einer Ausdehnung entsprechend dem durchschnittlichen Strichzwischenraum. Es werden solange Striche hinzugefügt, bis die Differenz gleich oder nahe Null ist.

Ein alternativer Ansatz zur interaktiven Illustration dreidimensionaler Modelle findet sich bei Rössl und Kobbelt [48]. Die Verarbeitung eines Modells, einer triangulierten Polygonfläche, erfolgt in einem dreistufigen Prozess. Zuerst wird eine 2.5-D-Repräsentation des Modells durch Projektion in einen *erweiterten Rahmenpuffer* (engl. *enhanced framebuffer*) erstellt. Dieser Puffer umfasst den normal-buffer (siehe Abschnitt 2.2.3), für jedes Pixel die Hauptkrümmungsrichtungen des Modells, sowie die Schattierung als Grauwert. Danach wird durch den Benutzer die Projektion hinsichtlich homogener Bereiche der Krümmungsrichtungen segmentiert. Für jede Region der Segmentierung wird dann entlang der Richtung maximaler Krümmung mit äquidistanten Linien schraffiert.

### 2.3.2 Geometriebasierte Schraffuren

Elber [17] präsentiert einen Ansatz zur Schraffierung parametrisierter Modelle  $S := S(u, v)$ , bei denen die Linien sich an einem beliebigen auf der Oberfläche definierten Vektorfeld  $V := V(u, v)$  orientieren. Die Platzierung der Striche erfolgt durch eine uniforme Punktüberdeckung der Fläche mittels einer gleichverteilten Zufallsvariable. Von jedem Punkt aus wird eine Schraffurlinie gezogen. Dies geschieht durch Konstruktion eines Linienzugs im Parameterraum von  $S(u, v)$ . Gegeben einen Kontrollpunkt  $(u_i, v_i)$  des Linienzugs, wird die Richtung zum nächsten Kontrollpunkt  $(u_{i+1}, v_{i+1})$  in Eintracht mit dem Vektor  $\mathbf{v} = V(u_i, v_i)$  folgendermaßen bestimmt: seien  $\mathbf{n} = \frac{S_u(u_i, v_i) \times S_v(u_i, v_i)}{\|S_u(u_i, v_i) \times S_v(u_i, v_i)\|}$  die Einheitsnormale und  $\mathbf{v}_p = \mathbf{v} - (\mathbf{v} \cdot \mathbf{n})\mathbf{n}$  die Projektion von  $\mathbf{v}$  auf die Tangentialebene von  $S$  im Punkt  $S(u_i, v_i)$ . Der Vektor  $\mathbf{v}_p$  läßt sich als Linearkombination  $\alpha S_u(u_i, v_i) + \beta S_v(u_i, v_i)$  schreiben (siehe Abb. 6). Die Koeffizienten  $\alpha$  und  $\beta$  zeigen die Richtung zum gesuchten Kontrollpunkt an:  $(u_{i+1}, v_{i+1}) = (u_i, v_i) + w \frac{(\alpha, \beta)}{\|(\alpha, \beta)\|}$ , wobei  $w$  die Schrittweite bestimmt. Die Abbildung des Linienzugs vom Parameterraum nach  $S$  ergibt die Schraffurlinie auf dem Modell.

Rössl et al. [49] übertragen Elbers Ansatz auf triangulierte Polygonmodelle. Sie verwenden

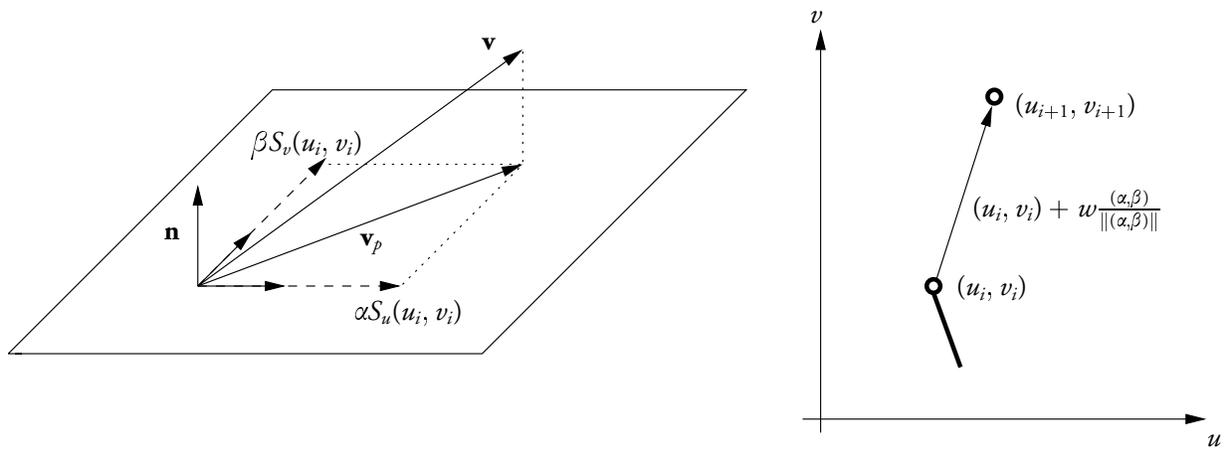


Abbildung 6: Konstruktion eines Linienzugs im Parameterraum. (links) Der auf die Tangentialebene projizierte Richtungsvektor ist eine Linearkombination  $\alpha S_u(u_i, v_i) + \beta S_v(u_i, v_i)$  der Ableitungen von  $S$  nach  $u$  und  $v$ . (rechts) Die Koeffizienten der Linearkombination bestimmen die Richtung zum nächsten Kontrollpunkt im Parameterraum.

ebenso eine Punktüberdeckung zur gleichmäßigen Verteilung der von den Punkten ausgehenden Linien. Die Richtung der Linien orientiert sich an dem Vektorfeld der approximierten Hauptkrümmungsrichtungen. Dieses wird erzeugt, indem zuerst an jeden Knoten eines Dreiecks eine lokale parametrisierte Fläche angepasst wird, für die sich die Hauptkrümmungsrichtungen bestimmen lassen. Diese gelten dann als geschätzte Hauptkrümmungsrichtungen des Polygonmodells in den Knoten. Ein kontinuierliches Vektorfeld für jedes Dreieck wird durch lineare Interpolation der drei Richtungsvektoren mittels baryzentrischer Kombination bestimmt. Die Schraffurlinien werden als Linienzüge auf den Dreiecken gezeichnet. Seien  $\mathbf{P}_n$  ein Kontrollpunkt einer Linie auf einem Dreieck  $\Delta$  und  $\mathbf{c}_{\max}(\mathbf{P}_n)$  die Hauptkrümmungsrichtung maximaler Krümmung in  $\mathbf{P}_n$ . Mit einer gegebenen Schrittweite  $w$  berechnet sich der nächste Kontrollpunkt der Linie durch  $\mathbf{P}_{n+1} = \mathbf{P}_n + w \frac{\mathbf{c}_{\max}^p}{\|\mathbf{c}_{\max}^p\|}$ , wobei  $\mathbf{c}_{\max}^p$  die Projektion von  $\mathbf{c}_{\max}$  auf die Ebene ist, in der  $\Delta$  liegt. Anhand der baryzentrischen Koordinaten von  $\mathbf{P}_{n+1}$  bezüglich  $\Delta$  wird geprüft, ob  $\mathbf{P}_{n+1}$  noch innerhalb von  $\Delta$  liegt (alle Koordinaten sind größer Null).

Praun et al. präsentieren eine Methode zur Generierung von Schraffierungen mit Echtzeitverarbeitung [44]. Damit sollen die Anforderung nach Frame-Kohärenz (frame coherence) sowie Tonwertkonstanz bei dynamischen Objekten erfüllt werden. Frame-Kohärenz bedeutet, daß die Topologie der Schraffurstriche einer Region bei aufeinanderfolgenden Frames erhalten bleiben muß, um den Eindruck eines flackernden oder zufällig variierten Bildes zu vermeiden. Tonwertkonstanz betrifft die Regulierung von Strichbreite und -dichte in Abhängigkeit von den Betrachtungsparameten. Die Herangehensweise der Autoren basiert auf dem Einsatz von Texturenabbildungshardware. Schraffuren mit abgestuften Tonwerten werden vor dem Gebrauch als eine Menge von

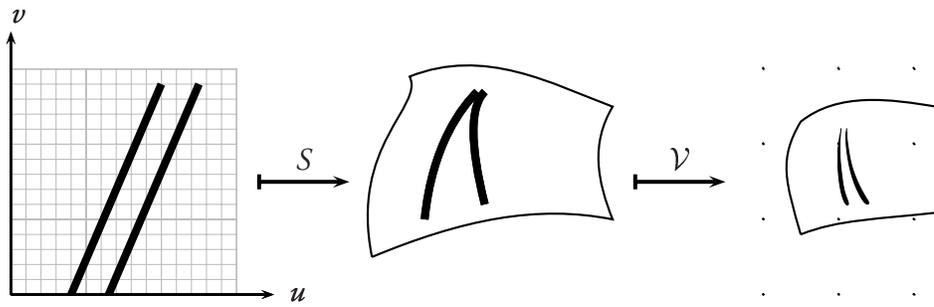


Abbildung 7: Die Abbildung paralleler Linien im Parameterraum durch die Betrachtungstransformation  $M = \mathcal{V} \circ S$ .  $S$  bildet die Linien ins Weltkoordinatensystem ab,  $\mathcal{V}$  bewirkt die Projektion in den Bildraum. Die Verjüngung der Linien auf der Fläche  $S$  hat eine lokale Tonwertänderung zur Folge. Der Ausgleich erfolgt im Bildraum durch die Anpassung der Linienbreite.

Mip-Mapping-Bildern erzeugt, die zur Laufzeit bei der Texturierung von Oberflächen verwendet werden. Die zu einer Schraffur gehörigen Bilder werden zusammen als *tonal-art map* bezeichnet. Die einzelnen Bilder sind in einer Matrix angeordnet, deren Zeilenindex das Mip-Map-Niveau und Spaltenindex den Tonwert angibt. Zwischen den ansteigenden Detail- und Tonwertniveaus besteht eine echte Teilmengenbeziehung. Die Schraffierung einer Fläche durch tonal-art maps entspricht Gouraud-Schattierung mittels Texturen. Dabei wird zuerst für jedes Dreieck der Tonwert der Knoten nach einem Beleuchtungsmodell berechnet. Dieser bestimmt die Auswahl zweier Texturen einer Zeile der tonal-art map, deren Überblendung dem Tonwert eines Knotens entspricht. Die so erhaltenen drei Texturen werden trilinear über dem Dreieck interpoliert. Die Kombination der Texturen kann in einem Rendering-Durchlauf mittels Multitexturierung erfolgen. Die Auswahl des Detailniveaus (innerhalb einer tonal-art map-Spalte) geschieht durch konventionelles Hardware-Mip-Mapping.

### 2.3.3 Hybride Schraffuralgorithmen

Ein Problem bei der Schraffierung parametrisierter Flächen ist die Wahrung eines konstanten Tonwertes der Gesamtfläche, wenn die Konstruktion der Linien im Parameterraum erfolgt (siehe Abbildung). Winkenbach und Salesin schlagen zur Lösung eine *controlled-density hatching* genannte Methode vor [64], ein Verfahren, bei dem die Strichbreite angepasst wird, um den wahrgenommenen Tonwert im Bildraum nach der Betrachtungstransformation konstant zu halten. Genauer: die Autoren spezifizieren eine Schraffur als Menge paralleler Linien im Parameterraum  $(u, v)$ . Ein einzelner Strich wird als Paar  $(\lambda(t), \vartheta(t))$  einer Linie  $\lambda(t)$  und einer Breite-Funktion  $\vartheta(t)$  definiert. Die Funktion  $\vartheta$  beschreibt die benutzte Breite bei der Darstellung des Strichs für jeden Parameterwert  $t$ . Gegeben eine parametrisierte Fläche  $S : (u, v) \mapsto (x_w, y_w, z_w)$ , eine Zentralprojektion  $\mathcal{V} : (x_w, y_w, z_w) \mapsto (x, y)$  und eine Schraffurrichtung  $\mathbf{h} = (h_u, h_v)$  im Parameterraum, lautet das controlled-density hatching-Problem, eine Schraffur  $\gamma_i = (\lambda_i, \vartheta_i)$  mit  $\lambda_i \parallel \mathbf{h}$  zu finden, so daß der im Bildraum wahrgenommene Tonwert der Schraffur nach der Transformation  $\mathcal{M} = \mathcal{V} \circ S$

konstant ist. Eine Lösung erfordert die Anpassung der Breite-Funktion  $\vartheta_i$  in Abhängigkeit von der Abstandsfunktion  $d'$ , welche den Zwischenraum der Bilder von  $\lambda_i$  unter  $\mathcal{M}$  angibt.

Das Verfahren von Deussen et al. verwendet zur Schraffierung Durchschnitte einer Ebene mit einem geometrischen Modell als Schraffurlinien [14]. Die Erzeugung der Linien in Pseudocode:

1. Modell rendern
2. Kontur  $K_M$  des Modells extrahieren
3. Modell mit zusätzlicher Schnittebene rendern
4. Kontur  $K_{MS}$  extrahieren
5. Mengendifferenz  $L := K_{MS} \setminus K_M$  bilden
6. aus  $L$  eine analytische Linienrepräsentation  $l_i$  generieren
7. Schritte 3 bis 6 für alle weiteren Schnittebenen wiederholen

Die Schnittebene dient dazu, das Modell auf einer Seite der Ebene zu kappen. Dies geschieht im dreidimensionalen Objektraum. Die Auswahl der Schnittebenen ist maßgeblich für eine gelungene Darstellung. Die Autoren beschreiben zwei alternative Möglichkeiten. Bei der interaktiven Spezifikation der Schnittebenen positioniert der Benutzer durch Angabe von Kontrollpunkten eine Spline-Kurve  $S$  im Objektraum und bestimmt eine Anzahl von Punkten  $\mathbf{P}_i$ , die äquidistant auf der Kurve verteilt werden. Die Schnittebenen werden durch die Punkte  $\mathbf{P}$  mit  $\nabla S(\mathbf{P}_i) \cdot (\mathbf{P} - \mathbf{P}_i) = 0$  definiert, wobei  $\nabla S(\mathbf{P}_i)$  der Gradient der Kurve im Punkt  $\mathbf{P}_i$  ist.

Die zweite Möglichkeit besteht in der automatischen Berechnung der Spline-Kurve. Dazu wird erst das Skelett des Modells konstruiert. Die Definition des Skeletts basiert auf dem Begriff des Balls [54]. Ein Ball  $B(\mathbf{P}, r)$  mit Mittelpunkt  $\mathbf{P}$  und Radius  $r \geq 0$  ist die Menge aller Punkte mit Abstand  $d$  zum Mittelpunkt kleiner oder gleich  $r$ . Ein Ball  $B \subset X$  ist maximal genau dann, wenn es keinen größeren Ball in  $X$  gibt, der  $B$  enthält. Das Skelett einer Menge  $X \subset \mathbb{R}^3$  ist die Menge der Mittelpunkte  $\mathbf{P}$  maximaler Bälle in  $X$ . Die Autoren approximieren das Skelett als ungerichteten Graphen durch die Reduktion des ursprünglichen Polygonmodells mittels der Verschmelzung von Kanten. Die Knoten des Graphen bilden die Kontrollpunkte der Spline-Kurve, entlang derer die Schnittebenen wie oben beschrieben platziert werden.

Die Erzeugung der Liniensegmente  $l_i$  erfolgt durch Raster-Vektor-Transformation mittels der Methode der kleinsten Quadrate. Die analytische Beschreibung ermöglicht eine weitergehende Manipulation der Linien vor der endgültigen Darstellung als Schraffurstriche. Die Autoren modulieren z. B. die Strichbreite anhand der Beleuchtungsintensität entlang der Linie.

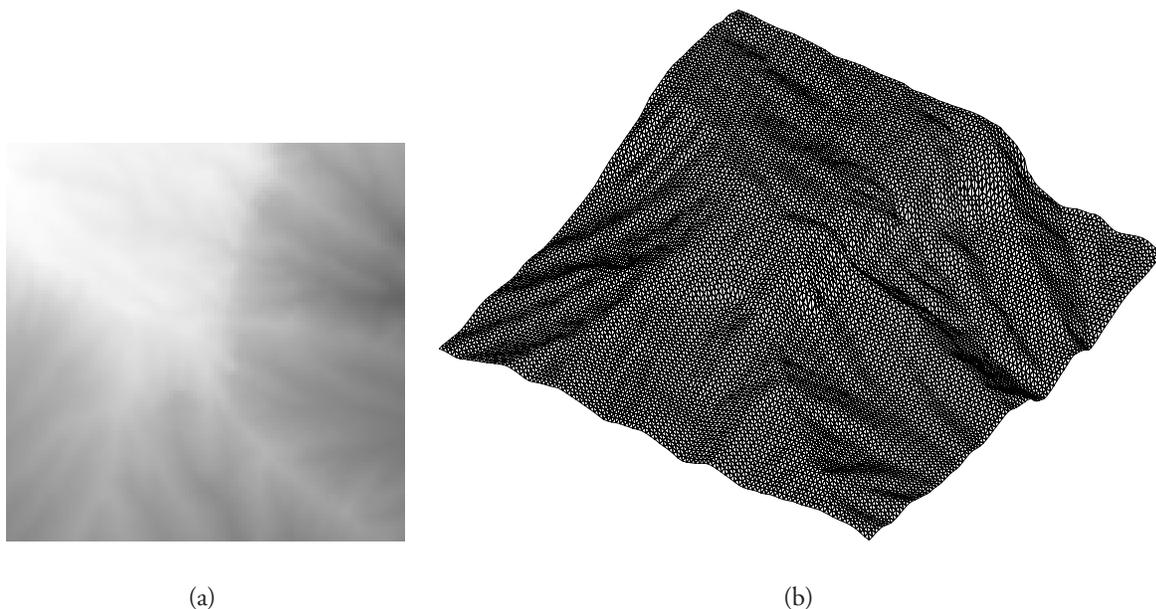


Abbildung 8: Höhenraster und Dreieckmodell. (a)  $H_{m \times n}$ , Höhen als Grauwerte kodiert. Weiß entspricht der maximalen, Schwarz der minimalen Höhe. (b) aus den Höhenwerten generiertes Dreieckmodell in Zentralperspektive

## 2.4 Die Erzeugung des Blockbilds

Gegeben ein Raster  $H_{m \times n}$  aus Höhenwerten, sei die Aufgabe, aus  $H_{m \times n}$  ein Blockbild zu erstellen, unter Benutzung von Techniken beschrieben in den Abschnitten 2.2.3 bis 2.3.3. Der erste Verarbeitungsschritt besteht in dem Aufbau eines triangulierten Polygonmodells aus den Höhenwerten, im Folgenden als Dreieck- oder Geländemodell bezeichnet, das als grundlegendes Datenobjekt fungieren soll. Ein Dreieckmodell ist einfach zu erstellen und im Kontext gängiger Software-Schnittstellen für Graphikprozessoren, die im Wesentlichen die Rasterung von Dreiecken durchführen, die Datenstruktur der Wahl. Bei Bildraum-Algorithmen ist die Art der Modellierung geometrischer Objekte sekundär, von ihr wird durch den Rasterungsprozess abstrahiert. Für Verfahren, die im Objektraum arbeiten, ist die Repräsentation als Dreieckmodell unter Umständen unzureichend, beispielsweise bei der Bestimmung einer differentialgeometrischen Eigenschaft wie der Krümmung. In einem solchen Fall läßt sich die Repräsentation Dreieckmodell ‚anreichern‘, etwa durch die lokale Anpassung einer parametrisierten Fläche und Approximation der gesuchten Eigenschaft.

Die Darstellung eines Blockbilds beginnt mit der Hervorhebung von Silhouetten und Formkanten. Saito und Takahashi [50] gehen folgendermaßen vor:

1. Kameraparameter wählen und Dreieckmodell rendern.

$A$	$B$	$C$
$D$	$g_{ij}$	$E$
$F$	$G$	$H$

Abbildung 9: Achter-Nachbarschaft eines Pixels  $g_{ij}$ .

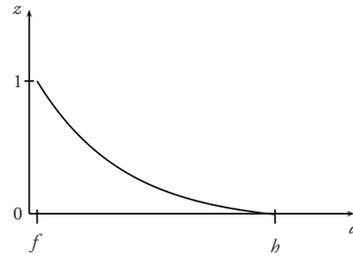


Abbildung 10: Das Auflösungsvermögen des Tiefenpuffers  $z$  variiert nicht-linear mit der Distanz  $d$  (nach Buss [8])

2. Unstetigkeitsstellen im Tiefenpuffer  $z$  bestimmen und diese als Linien extrahieren.

Die Unstetigkeitsstellen können mit einem Differenzenoperator erster Ordnung bestimmt werden. Die Autoren empfehlen den Sobel-Operator zur Schätzung des Gradienten in horizontaler und vertikaler Richtung [54]:

$$s_1 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}, \quad s_2 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Der Gradientenbetrag wird durch  $[g_{ij}] = |z \star s_1| + |z \star s_2|$  approximiert. Für ein Pixel  $g_{ij} \in [g_{ij}]$  ist der Gradientenbetrag angenähert

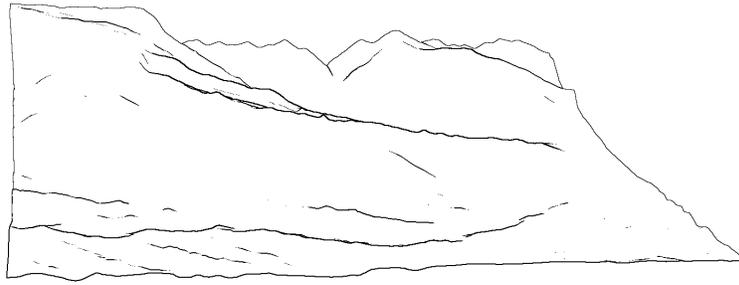
$$g_{ij} = (|-A - 2B - C + F + 2G + H| + |-C - 2E - H + A + 2D + F|)/8$$

wobei  $A, \dots, H$  die Pixel der Achter-Nachbarschaft von  $g_{ij}$  sind (vgl. Abb. 9) und das Ergebnis zusätzlich normalisiert wird.

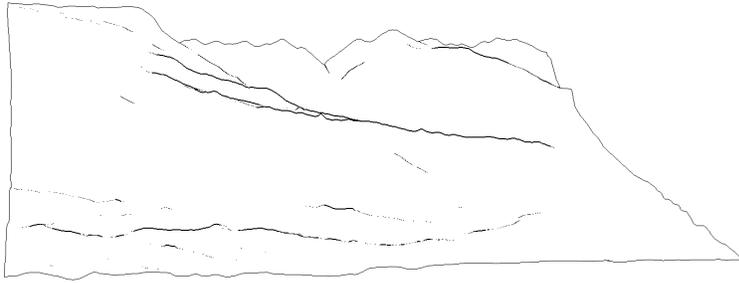
Die Matrix  $[g_{ij}]$  enthält Artefakte, die nicht Unstetigkeitsstellen entsprechen, und deshalb bei der Linienextraktion unberücksichtigt bleiben sollen. Die Artefakte sind bedingt durch die nicht-lineare Abhängigkeit der Tiefenpufferwerte von der euklidischen Distanz zwischen Augpunkt und Objekt (siehe dazu Abb. 10). Tiefenpufferwerte variieren schneller bei kleinen Distanzen als bei großen. Mit steigenden Tiefenwerten wird die Unterscheidung einer Unstetigkeitsstelle von einer ‚graduellen‘ Variation schwierig. Der Effekt kann vermindert werden, indem die Distanz zwischen der nahen und fernen Schnittebene so gering wie möglich gewählt wird.

Die Autoren wenden zudem vor der Linienextraktion einen Korrekturterm auf den Gradientenbetrag an, als Ausgleich des nichtlinearen Auflösungsvermögens des Tiefenpuffers.

$$\hat{g}_{ij} = \begin{cases} \frac{g_{\min} - g_{ij}}{g_{\max} - g_{\min}} & \text{für } g_{\max} - g_{\min} > k_g \\ \frac{g_{\min} - g_{ij}}{k_g} & \text{für } g_{\max} - g_{\min} \leq k_g \end{cases}$$



(a)



(b)

---

Abbildung 11: Bestimmung der Silhouette im Bildraum durch (a) Sobel-Operator, (b) Laplace-Operator.

Die Korrektur erfolgt durch Benutzung der minimalen und maximalen Gradientenbeträge  $g_{\min}$  und  $g_{\max}$  in der Achter-Nachbarschaft von  $g_{ij}$ . Die Konstante  $k_g$  dient zur Differenzierung zwischen Unstetigkeitsstellen und graduellen Veränderungen und ist ein Eingabeparameter. Sie hängt im Wesentlichen von der Präzision des Tiefenpuffers ab.

Eine Schwellenwertfunktion wird benutzt, um die Linien als Sequenzen hoher Gradientenbetragswerte zu extrahieren:

$$s(\hat{g}_{ij}) = \begin{cases} 1 & \hat{g}_{ij} > \vartheta \\ 0 & \text{sonst} \end{cases}$$

Ein weiterer Operator, der zur Bestimmung der Unstetigkeitsstellen verwendet werden kann, ist der Laplace-Filter[54]. Dieser approximiert rotationsinvariant die zweite Ableitung.

$$l = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

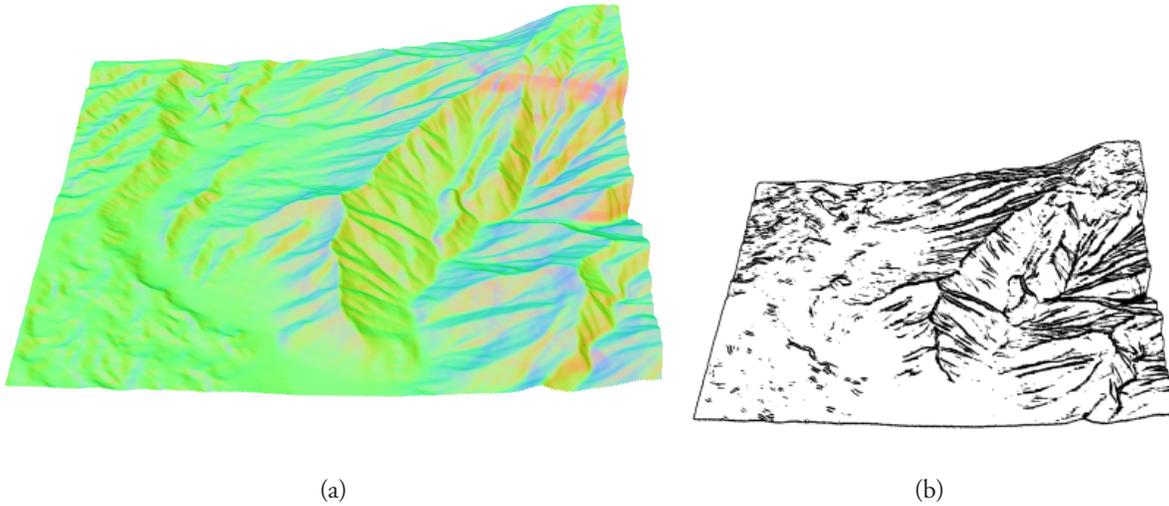


Abbildung 12: Detektion von Formkanten im normal-buffer. (a) normal-buffer, (b) Extraktion der Kanten.

Die Silhouette entspricht den Nulldurchgängen von  $z \star l$ .

Die extrahierten Linien sind noch zu breit und durch Anwendung der abrupten Schwellenwertfunktion ungeglättet. Die Strichbreite ist abhängig von der Größe des Faltungskerns und der Wahl von  $\vartheta$  in  $s$ . Die Breite kann dadurch angepasst werden, daß die Linien aus einem vergrößerten Bild extrahiert werden, und das Ergebnis dann durch Abtastung auf die endgültigen (kleineren) Abmessungen gebracht wird. Bei der Abtastung eignet sich zur Interpolation der Grauwerte der Faltungskern  $k(x, y) := s(x)s(y)$  [54], wobei

$$s(t) = \begin{cases} 1 - 2|t|^2 + |t|^3 & 0 \leq |t| < 1 \\ 4 - 8|t| + 5|t|^2 - |t|^3 & 1 \leq |t| < 2 \\ 0 & 2 \leq |t| \end{cases}$$

ist. Die Linien des verkleinerten Bildes sind schmäler und aufgrund der Mittelung durch den Faltungskern zudem geglättet. Schmalere und glattere Linien erhält man durch entsprechend größere Dimensionen des Ausgangsbildes. Jedoch ist zu beachten, daß bei  $n$ -facher Skalierung die  $n^2$ -fache Anzahl von Pixeln verarbeitet werden muß.

Die Bestimmung der Formkanten verläuft in ähnlicher Weise wie die Silhouettendetektion, jedoch ist die Grundlage in diesem Fall der normal-buffer:

1. Kameraparameter wählen und Dreieckmodell rendern.
2. Unstetigkeitsstellen im normal-buffer  $n$  bestimmen und diese als Linien extrahieren.

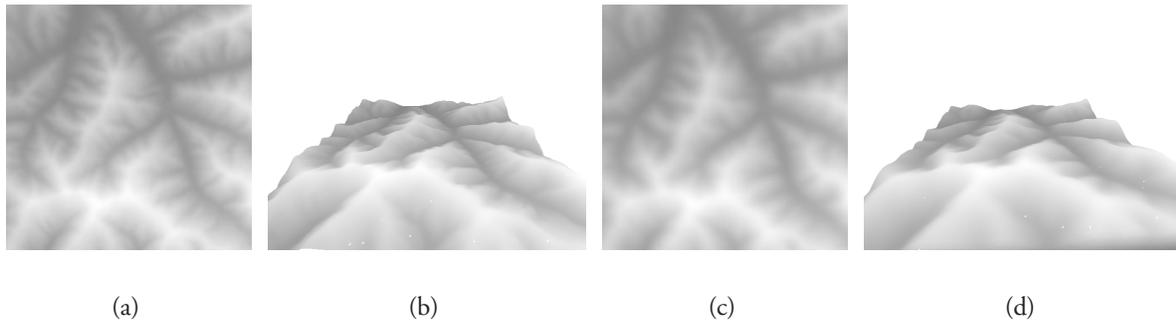


Abbildung 13: Glättung des Geländemodells. (a) Ursprüngliches Höhenraster und (b) Dreieckmodell, (c) durch einen Gauss-Filter geglättetes Höhenraster mit (d) Dreieckmodell.

Der normal-buffer ist ein RGB-Bild, dessen Kanäle für jedes Pixel die Koordinaten des Einheitsnormalenvektors des an der Stelle sichtbaren Objekts kodieren. Die Erzeugung dieses Bildes kann einfach durch Einsatz der Gouraud-Schattierungsfunktion der Rasterungs-Hardware geschehen [48]. Dazu wird jedem Knoten  $v_i$  des Dreieckmodells ein Einheitsnormalenvektor  $\mathbf{n}(v_i)$  zugeordnet, durch Mittelung der Normalenvektoren der Dreiecke, deren Kanten zu  $v_i$  inzident sind, jeweils mit dem Flächeninhalt  $\alpha$  des Dreiecks gewichtet:  $\mathbf{n}(v_i) = \frac{\sum_j \alpha(\Delta_j) \mathbf{n}(\Delta_j)}{\|\sum_j \alpha(\Delta_j) \mathbf{n}(\Delta_j)\|}$ . Der Normalenvektor eines Knotens wird dann beim Rendern als Farbvektor dieses Knotens angegeben. Da die Komponenten des Vektors im Intervall  $[-1, 1]$  liegen, müssen sie noch vor der Angabe als RGB-Anteile auf den Bereich  $[0, b]$  normalisiert werden, wobei  $b$  abhängig von der Anzahl der verfügbaren Bits pro Farbkanal ist. Dies erfolgt durch eine einfache Transformation  $t : (n_x, n_y, n_z) \mapsto \frac{b}{2}(n_x + 1, n_y + 1, n_z + 1)$ .

Die Farben, resp. Normalenvektoren werden bei Verwendung von Gouraud-Schattierung trilinear über jedem Dreieck interpoliert, wobei Lichtquellen und Texturierung ausgeschaltet sein müssen. Das Ergebnis ist der farbkodierte normal-buffer (siehe Abb. 12(a)). Die Unstetigkeitsstellen des normal-buffers zeigen Formkanten an.

Zur Verminderung einer großen Anzahl von Formkanten, können im zugrundeliegende Höhenraster  $H_{m \times n}$  hochfrequente Anteile durch einen Gauss-Filter reduziert werden. Die durch den Filterkern

$$s = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

bewirkte Glättung des Höhenrasters verursacht im Dreieckmodell eine entsprechende Verringerung der diedralen Winkel angrenzender Polygone (Abb. 13).

Die Schraffierung erfordert die Spezifikation der Parameter Tonwert und Orientierung.

Die Orientierung der Striche erfolgt, wie in der Einleitung gezeigt, anhand von Falllinien. Ein kontinuierliches Vektorfeld der Fallrichtungen soll dabei als ‚Unterlage‘ zur Steuerung der Linien-

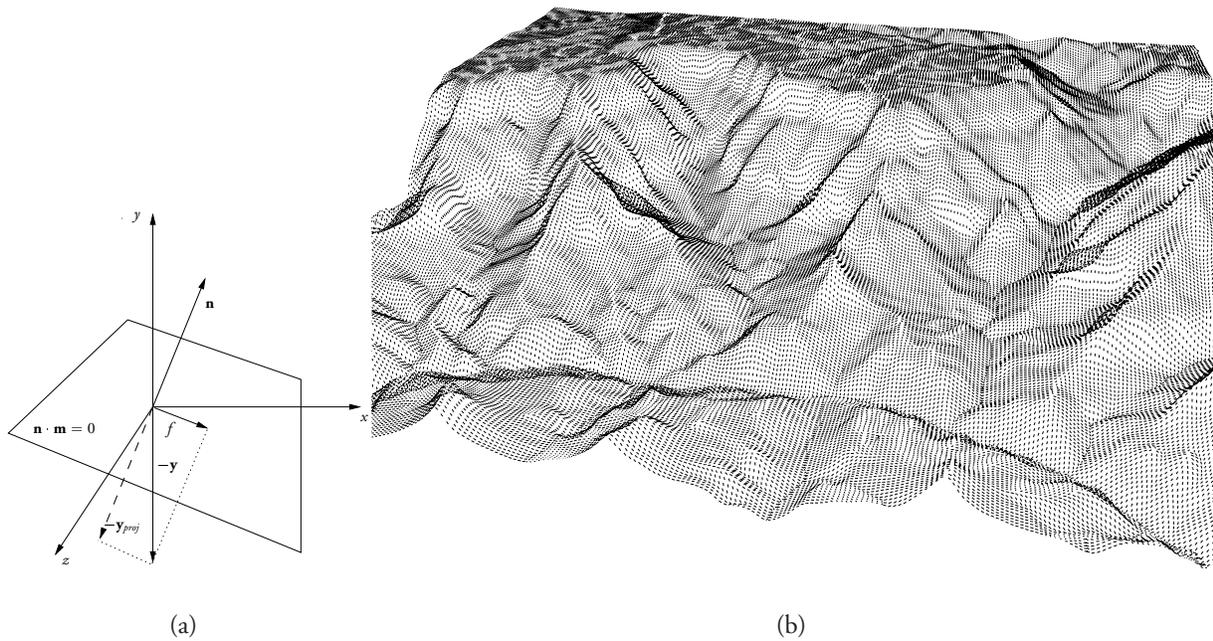


Abbildung 14: Fallrichtungen. (a) Bestimmung der Fallrichtung durch Projektion von  $-\mathbf{y}$  auf die Ebene  $\mathbf{n}(v) \cdot \mathbf{m} = 0$ . (b) Visualisierung der Fallrichtungsvektoren in den Knoten des Dreieckmodells.

richtungen dienen. Zur Konstruktion des benötigten Vektorfelds wird zuerst in jedem Knoten des Dreieckmodells die Fallrichtung, d. h. die Richtung der größten Neigung bestimmt. Man erhält die Fallrichtung  $\mathbf{f}(v)$  in einem Knoten  $v$ , indem die Projektion des Vektors  $-\mathbf{y} = (0, -1, 0)$  auf die Ebene  $\mathbf{n}(v) \cdot \mathbf{m} = 0$  bestimmt wird, in einem Koordinatensystem mit Achsenbezeichnungen wie in Abb. 14(a). Der Vektor  $-\mathbf{y}$  ist der Einheitsvektor in Richtung der Erdanziehung. Zur Bestimmung von  $\mathbf{f}(v)$  wird erst  $-\mathbf{y}$  auf  $\mathbf{n}(v)$  projiziert:  $-\mathbf{y}_{proj} = (-\mathbf{y} \cdot \mathbf{n}(v))\mathbf{n}(v)$ . Die Fallrichtung ist dann  $\mathbf{f}(v) = -\mathbf{y} - (-\mathbf{y}_{proj}) = \mathbf{y}_{proj} - \mathbf{y}$ .

Ein kontinuierliches Vektorfeld der Fallrichtungen im Bildraum wird, wie bei der Erzeugung des normal-buffers, mit Hilfe der Rasterungs-Hardware generiert. Das Ergebnis ist ein zweidimensionaler Bildpuffer („*F-Puffer*“), der die *Fallrichtungen im Objektraum* als RGB-Farbvektoren  $\mathbf{f}$  kodiert (Abb. 15(a)).

Die Farbkomponenten haben Werte im Bereich  $[0, b]$  und müssen zur Verwendung als Komponenten der Richtungswerte im Bildraum in drei Schritten transformiert werden:

1. Abbildung durch  $t : (x, y, z) \mapsto \frac{2}{b}(x, y, z) - (1, 1, 1)$  in das Intervall  $[-1, 1]$ .
2. Anwendung der Rotation  $R$  der Betrachtungstransformation.
3. Projektion  $p : (x, y, z) \mapsto (x, y)$  auf die Bildebene.

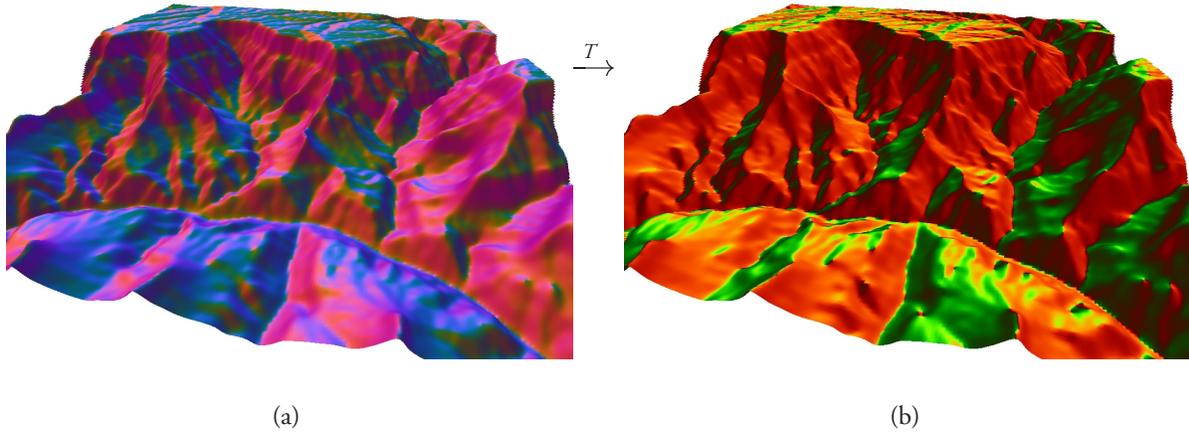


Abbildung 15: F-Puffer und davon abgeleitete Bildraum-Fallrichtungen. Jedes Pixel in (a) kodiert die Fallrichtung im Objektraum entlang der sichtbaren Oberflächen. In (b) sind die Bildraum-Fallrichtungen dargestellt. Die Transformation  $T = p \circ R \circ t$  bedingt die verschwindende Intensität auf dem Blaukanal.

Damit ist an der Position  $(i, j)$  im Bildraum die Schraffurlinienrichtung durch  $\mathbf{f}_{ij} = (p \circ R \circ t)(\mathbf{f}'_{ij})$  vorgegeben.

Ein Schraffurstrich wird nach dem numerischen Integrationsverfahren von Euler [5], gegeben einen Anfangspunkt  $(i_0, j_0)$ , als Linienzug mit den Kontrollpunkten  $((i_{-m}, j_{-m}), \dots, (i_{-1}, j_{-1}), (i_0, j_0), (i_1, j_1), \dots, (i_n, j_n))$  konstruiert, wobei

$$(i_k, j_k) = \begin{cases} (i_{k-1}, j_{k-1}) + w \frac{\mathbf{f}_{i_{k-1}j_{k-1}}}{\|\mathbf{f}_{i_{k-1}j_{k-1}}\|} & 1 \leq k \leq n \quad (\text{Vorwärtsintegration}) \\ (i_{k+1}, j_{k+1}) - w \frac{\mathbf{f}_{i_{k+1}j_{k+1}}}{\|\mathbf{f}_{i_{k+1}j_{k+1}}\|} & -m \leq k \leq -1 \quad (\text{Rückwärtsintegration}) \end{cases}$$

und  $w$  der vorgegebene Abstand zwischen den Kontrollpunkten ist. Die Integration wird bei Überschreitung der Schraffurfläche und in singulären Punkten ( $\mathbf{f}_{ij} = \mathbf{0}$ ) beendet. Die Festlegung einer Obergrenze der Anzahl der Integrationsschritte verleiht dem Verfahren zusätzliche Stabilität. Damit lässt sich beispielsweise eine nicht-terminierende Oszillation zwischen zwei Kontrollpunkten in einer Talsohle vermeiden. Sinnvoll ist eine Obergrenze invers proportional zu  $w$ .

Die Platzierung der Linien erfolgt nach einem Verfahren aus dem Bereich der Stromlinienvisualisierung in zweidimensionalen Vektorfeldern von Jobard und Lefer [33]. Die Methode ist zur Erzeugung von orientierten Schraffuren mit äquidistanten Linien geeignet. Der Zwischenraum in Pixeln  $d_{sep}$  benachbarter Striche ist eine vorgegebene Konstante. Durch Multiplikation von  $d_{sep}$  mit  $d_{rest} \in [0, 1]$  kann der Zwischenraum unter bestimmten Bedingungen variiert werden (s.u.). Ausgehend von einem Schraffurstrich  $s$  wird ein Anfangspunkt im Abstand  $d_{sep}$  von  $s$  gesucht, von

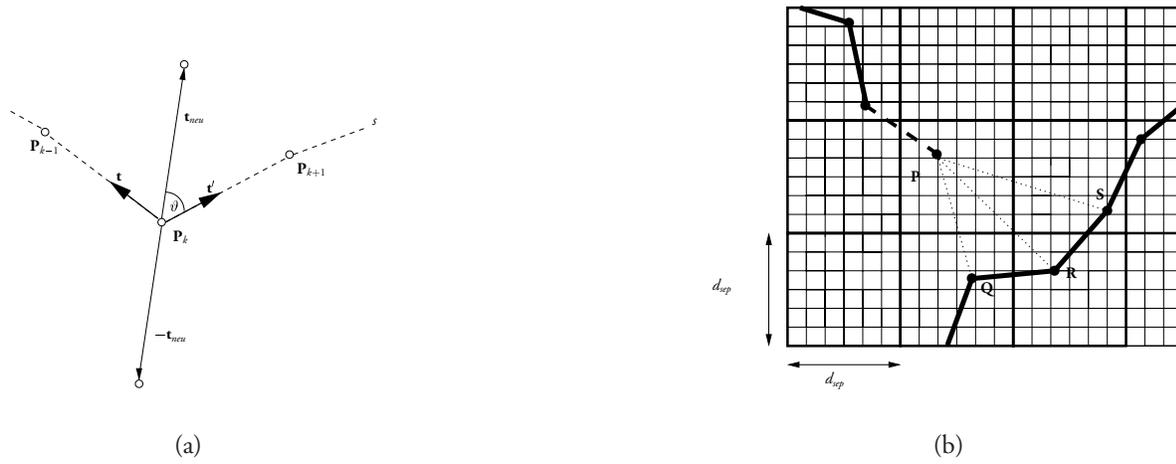


Abbildung 16: Abstandsmessungen bei der Erzeugung von Linienzügen. (a) Konstruktion neuer Anfangspunkte im Abstand  $d_{sep}$  von den Kontrollpunkten bestehender Linien aus. Die Richtung ist durch den halben Winkel zwischen den jeweiligen Liniensegmenten der Kontrollpunkte gegeben. (b) zeigt ein Abstandsraaster und die Integration einer neuen Linie (gestrichelt).  $\mathbf{P}$  ist ein gültiger Punkt, wenn die Abstände zu  $\mathbf{Q}$ ,  $\mathbf{R}$  und  $\mathbf{S}$  größer oder gleich  $d_{test} \cdot d_{sep}$  sind.

dem aus ein weiterer Strich  $s'$  konstruiert wird, der den Mindestabstand  $d_{test} \cdot d_{sep}$  zu  $s$  einhält. Dies wird solange wiederholt, bis keine entsprechenden Punkte mehr in der Umgebung von  $s$  gefunden werden können. Ebenso wird mit den übrigen Linien verfahren, in der Reihenfolge, in der sie erzeugt wurden. Zur Verwaltung dient eine Warteschlange, die anfänglich  $s$  enthält.

Die Anfangspunkte neuer Linien werden wie folgt ausgewählt (Abb. 16(a)): gegeben einen Linienzug  $s$ , aufeinanderfolgende Kontrollpunkte  $\mathbf{P}_{k-1}, \mathbf{P}_k$  und  $\mathbf{P}_{k+1}$  in  $s$ , seien  $\mathbf{t} = \frac{\mathbf{P}_{k-1} - \mathbf{P}_k}{\|\mathbf{P}_{k-1} - \mathbf{P}_k\|}$  und  $\mathbf{t}' = \frac{\mathbf{P}_{k+1} - \mathbf{P}_k}{\|\mathbf{P}_{k+1} - \mathbf{P}_k\|}$  die Einheitsvektoren von  $\mathbf{P}_k$  in Richtung  $\mathbf{P}_{k-1}$ , resp.  $\mathbf{P}_{k+1}$ . Der halbe Winkel zwischen  $\mathbf{t}$  und  $\mathbf{t}'$  ist  $\vartheta = \frac{\arccos \mathbf{t} \cdot \mathbf{t}'}{2}$ . Sei  $\mathbf{t}_{neu} = d_{sep} R_\vartheta \mathbf{t}$  nach Rotation von  $\mathbf{t}$  mittels  $R_\vartheta = \begin{bmatrix} \cos \vartheta & -\sin \vartheta \\ \sin \vartheta & \cos \vartheta \end{bmatrix}$  und Skalierung mit  $d_{sep}$ . Dann erhält man zwei neue Punkte durch  $\mathbf{P}_k + \mathbf{t}_{neu}$  und  $\mathbf{P}_k - \mathbf{t}_{neu}$ . Mindestens einer davon hat den Abstand  $d_{sep}$  von  $s$ .

Ein kartesisches *Abstandsraaster* mit Zellenbreite und -höhe  $d_{sep}$  wird über dem Vektorfeld konstruiert, um die Einhaltung der Abstände zwischen den Strichen effizient überprüfen zu können. Eine Zelle enthält Zeiger auf die in ihr enthaltenen Kontrollpunkte. Führt die Integration eines Linienzugs zur Platzierung eines Kontrollpunkts  $\mathbf{P}$  in eine Zelle, und befindet sich in dieser Zelle oder ihrer Achter-Nachbarschaft bereits ein Kontrollpunkt  $\mathbf{Q}$  einer *anderen* Linie mit  $\|\mathbf{P} - \mathbf{Q}\| < d_{test} \cdot d_{sep}$ , dann ist  $\mathbf{P}$  *ungültig* und die Konstruktion des Linienzugs wird beendet (Abb. 16(b)).

Wichtig ist, die Schrittweite  $w$  des Euler-Integrators kleiner als  $d_{sep}$  zu wählen, da andernfalls

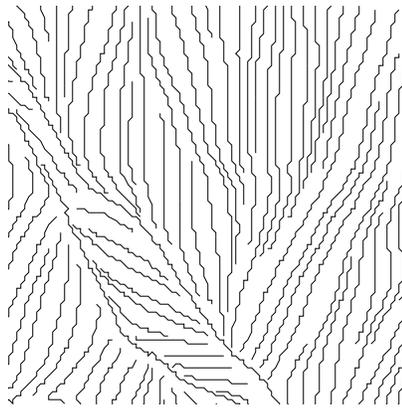


Abbildung 17: Vergrößerte Ansicht einer schraffierten Fläche nach dem Verfahren von Jobard und Lefer. Die Darstellungsparameter sind  $d_{sep} = 3$ ,  $d_{test} = 0.5$  und Schrittweite  $w = 1$ .

Überschneidungen von Liniensegmenten eintreten können. Der Fall der Selbstüberschneidung einer Linie wird von dem beschriebenen Verfahren nicht verhindert, da nur die Abstände der Kontrollpunkte einer Linie zu anderen Linien überprüft werden. Dieser Umstand hat sich bei dem Einsatz der Methode jedoch als unproblematisch erwiesen. In Abb. 17 sind Schraffurstriche nach der Methode von Jobard und Lefer in fünfzehnfacher Vergrößerung dargestellt.

Verschiedene Abbruchkriterien des Integrationsverfahrens beeinflussen die Länge der Linien. Bereits genannt wurde das Unterschreiten der minimalen Zwischenbreite.

Ein Blick zurück auf Abb. 1 zeigt, daß ebene Regionen unschraffiert bleiben. Als weiteres Kriterium bietet sich also das Überschreiten der Grenze zwischen Gefälle und horizontalem Gelände an.

Texturierung kann dazu verwendet werden, die Schraffuren auf Hanglagen zu beschränken. Die Textur teilt dabei das Geländemodell in homogene Bereiche geringer und starker Neigungsverhältnisse ein. Diese Bereiche korrespondieren zu Regionen niedriger und hoher Gradientenbeträge im Höhenraster  $H_{m \times n}$ . Zur Erzeugung der Textur wird  $[g_{ij}] = |\nabla(H_{m \times n})|$  durch eine Schwellenwertfunktion  $s$  partitioniert:

$$s(g_{ij}) = \begin{cases} 1 & (g_{\max} - g_{\min})^2 < t \\ 0 & \text{sonst} \end{cases}$$

wobei  $g_{\max}$  und  $g_{\min}$  die maximalen bzw. minimalen Werte in der  $3 \times 3$ -Umgebung von  $g_{ij}$  sind und  $t$  eine vorgebene Schranke ist. Ein hohes  $t$  klassifiziert mehr Bereiche als ebenes Gelände. Eine vorangehende Glättung von  $H_{m \times n}$  mit einem Gauss-Filter schafft zusammenhängende Bereiche von Hängen bzw. Plateaus. Die Textur wird nach Abbildung von  $s$  auf Schwarz und Weiß erstellt (Abb. 18).

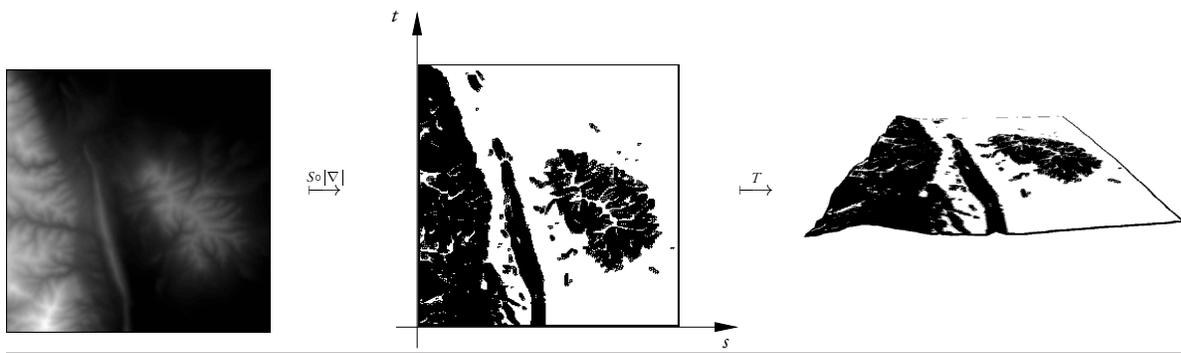


Abbildung 18: Erzeugung der Schraffurmaske. Der Gradientenbetrag eines geglätteten Höhenraster wird nach Anwendung einer Schwellenwertfunktion  $s$  als Textur verwendet. Diese beschreibt die Neigungsverhältnisse des Geländes. Weiße Bereiche entsprechen ebenen Flächen, schwarze Regionen bezeichnen Hänge. Durch die Abbildung auf das Dreieckmodell kann die Textur zur Ausblendung von Strichen in horizontalem Gelände verwendet werden.

Die gerenderte Szene des mit der Textur versehenen Geländemodells wird im Folgenden als *Schraffurmaske* bezeichnet. Diese wird bei der Erzeugung der Schraffurlinien verwendet, um die Integration beim Übergang von schwarzen in weiße Bereiche abzurechnen. Abb. 19 zeigt ein Blockbild in Kombination mit einer Schraffurmaske.

Die Schraffurmaske erlaubt weitere Effekte. Durch das Eintragen der Silhouette in die Maske als weiße Punkte wird verhindert, daß die Schraffurstriche über Silhouettenlinien hinweg gezogen werden.

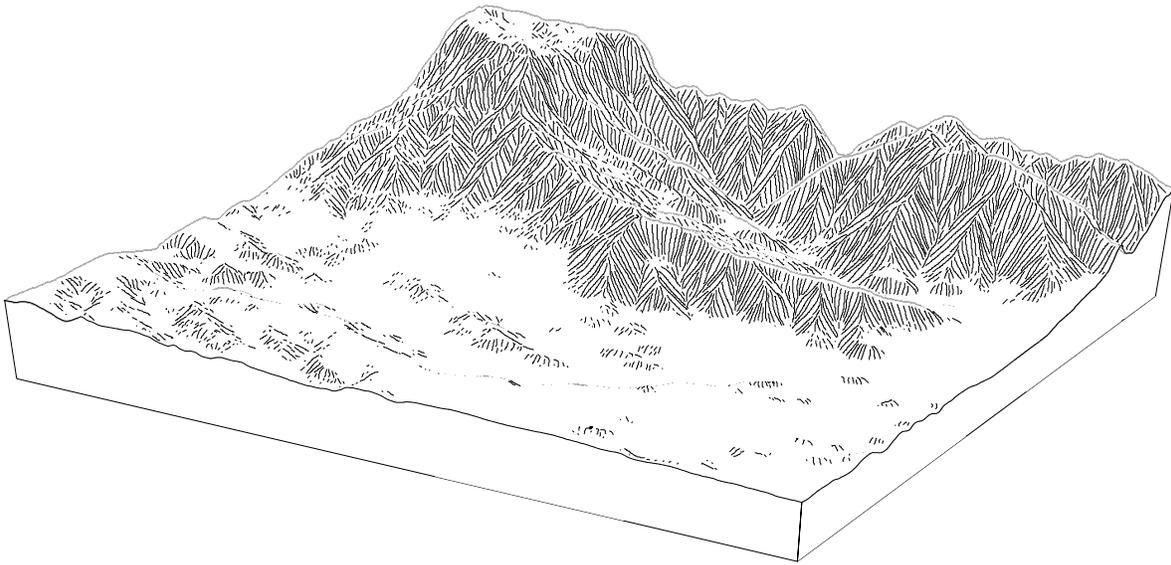
Zur Betonung der Silhouette kann diese mit einem *Halo*<sup>5</sup> versehen werden, einem zusätzlichen Zwischenraum, der die Silhouette von anderen Bildelementen abtrennt. Dazu wird die Silhouette in der Schraffurmaske zum oberen und unteren Bildrand hin um konstante Beträge verbreitert. Ein Blockbild mit Silhouettenhalo ist in Abb. 20 dargestellt.

Durch Hinzunahme einer Lichtquelle können Schraffuren auch in Abhängigkeit von den Beleuchtungsverhältnisse erzeugt bzw. manipuliert werden. Bei der Verwendung einer weißen Lichtquelle entstehen Schraffurmasken mit mehr als zwei Graustufen. Vergleichbar dem Prinzip der Schattenschraffen [30], lassen sich die Linienfarbe oder -breite anhand der Graustufen modulieren (siehe Beispiele im Anhang). Alternativ läßt sich in dunklen Bereichen der Linienzwischenraum verringern, so daß die Striche dort dichter geschart sind.

Mehrwertige Schraffurmasken erlauben auch Schraffierung nach dem Prinzip der Böschungsschraffen ‚je steiler, desto dunkler‘ [30], entweder durch Verbreiterung der Linien oder Verkürzung der Abstände.

---

<sup>5</sup>in Anlehnung an Appel et al. [3]



---

Abbildung 19: Blockbild mit Schraffurmaske und Hervorhebung der Silhouettenlinie. Darstellungsparemeter:  $d_{sep} = 3$ ,  $d_{rest} = 0.5$ , Schrittweite  $w = 1$ .

Durch weitere Unterteilung der Linienzüge können bei Bedarf abgerundete Striche erzeugt werden. Hier wird das einfache Unterteilungsverfahren von Chaikin [10] benutzt.

Zusammenfassend folgt hier eine kurze Rückschau auf die Abfolge der Verfahrensschritte. Die Erzeugung der Darstellung erfolgt im Bildraum. Aus einem Höhenraster wird das Dreieckmodell erstellt. Das Dreieckmodell wird in einen erweiterten Framebuffer gerendert. Dieser umfaßt den Tiefenpuffer, normal buffer, F-Puffer und eine oder mehrere Schraffurmasken. Die Silhouette erhält man durch Kantenextraktion im Tiefenpuffer, Formkanten durch Kantenextraktion im normal buffer. Nach der Transformation der Fallrichtungen des F-Puffers in die Bildebene wird über dieser das Abstandsraaster aufgebaut. Die Koordinaten von Anfangspunkten für Schraffurlinien werden zufällig ermittelt. Von einem gültigen Anfangspunkt wird ein Schraffurstrich gezogen und ausgehend von diesem alle in seiner Umgebung möglichen Striche. Die Striche liegen analytisch als Linienzüge vor, wodurch sie leicht unter Einsatz der verwendeten Schraffurmasken attribuiert werden können. Die Darstellung wird durch Anbringung der sichtbaren Profilflächen vervollständigt.

Die Beschränkung auf den Bildraum bedingt eine Abhängigkeit der Qualität der Darstellung von der Geräteauflösung. Vorteilhaft erweist sich die durchgehend homogene Schraffurweite in allen Bildteilen, was zu einer gleichmäßigen Tonwertverteilung führt.

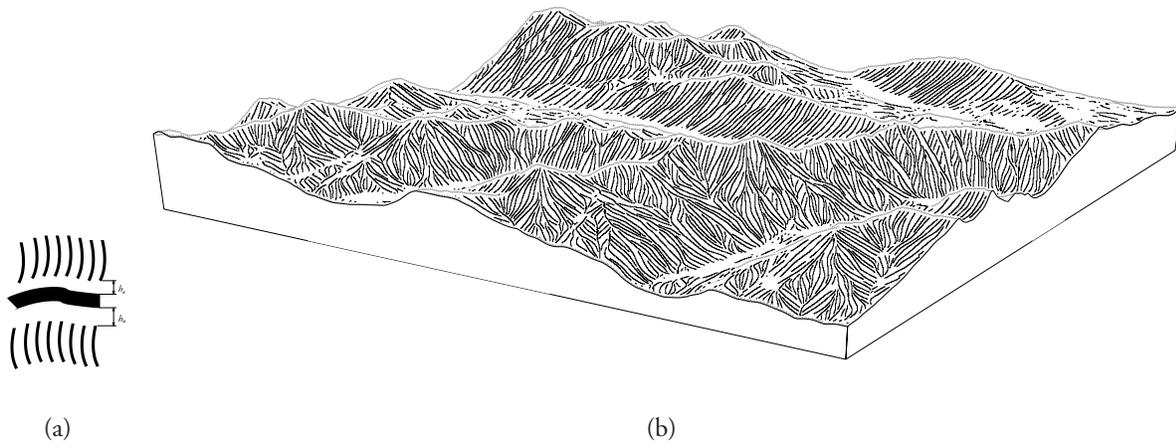


Abbildung 20: Silhouetten-Halos. (a) Silhouettensegment mit oberer Halobreite  $h_o$  und unterer Breite  $h_u$ .  
 (b) Blockbild mit Silhouettenhalo,  $h_o = \frac{3}{4}h_u$ .

Als unbefriedigend hat sich die Erzeugung der Formkanten herausgestellt. Die Kantendetektion im normal buffer liefert bei großen Dreieckmodellen zu viele doppelte Linien zu beiden Seiten der Formkanten anstatt der Formkanten selbst. Auch Versuche, die Kanten über die Überprüfung der diedralen Winkel angrenzender Dreiecke im Objektraum zu lokalisieren, haben nicht das gewünschte Ergebnis gezeigt. Hier könnte ein elaborierterer Ansatz wie z. B. von Interrante et al. [31] zu besseren Resultaten führen.

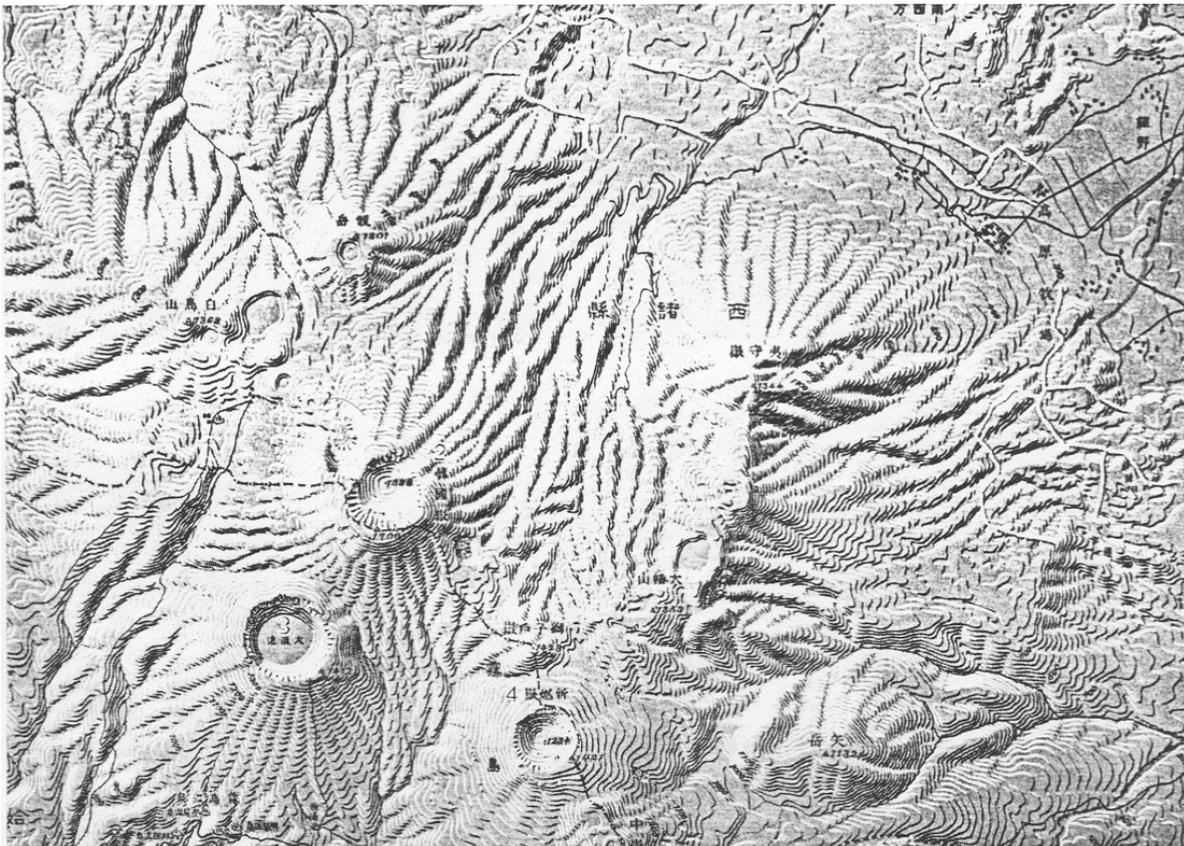


Abbildung 21: Topographische Darstellung einer vulkanischen Landschaft bei Kagoshima durch die Relief-Kontur-Methode [59].

### 3 Die Relief-Kontur-Methode

#### 3.1 Überblick

In diesem Kapitel soll die Erzeugung von Kartendarstellungen nach der *Relief-Kontur-Methode* von Tanaka [59] behandelt werden. Die Methode ist eine Alternative zur Schummerungsdarstellung [30] topographischer Karten mit dem Anspruch zur detaillierten Reliefdarstellung bei gleichzeitiger Visualisierung von Konturlinien<sup>6</sup>.

Die Vorgehensweise basiert auf der Konstruktion von Konturlinien, bei denen die Breite und Helligkeit in Abhängigkeit von schräg einfallenden Lichtstrahlen einer angenommenen Lichtquelle variiert werden. Die Methodik der Variation beruht auf der Theorie Lambertscher Reflektierer, die die diffuse Reflexion von Oberflächen beschreibt [20]. Diese soll kurz erläutert werden:

Lambert-Reflektierer sind Oberflächen, die einfallendes Licht gleichmäßig (i. e. diffus) in alle

---

<sup>6</sup>Konturlinie: in diesem Kapitel die Bezeichnung für Höhenlinie, oder Isohypse.

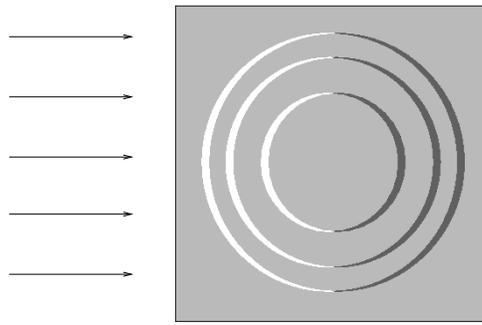
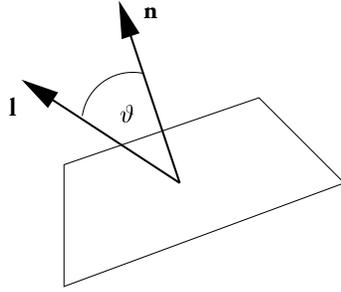


Abbildung 22: Lambert-Reflektierer.  $\mathbf{n}$  Oberflächennormale,  $\mathbf{l}$  Vektor in Richtung der Lichtquelle

Abbildung 23: Relief-Konturen einer Kuppe. Parallele Lichtstrahlen und die Neigungsrichtungen der Kuppe bestimmen die Konturbreite und -helligkeit.

Richtungen streuen. Die Intensität  $I$  des reflektierten Lichts an einem Punkt  $\mathbf{P}$  des Reflektierers, in Abhängigkeit von der Intensität  $I_L$  des einfallenden Lichts, wird durch das Kosinusetz von Lambert beschrieben:

$$I = I_L k_d \max(\cos \vartheta, 0) = I_L k_d \max(\mathbf{n} \cdot \mathbf{l}, 0)$$

Dabei sind  $k_d \in [0, 1]$  der diffuse Reflexionskoeffizient, eine materialabhängige Konstante, und  $\vartheta$  der Winkel zwischen der Einheitsnormalen  $\mathbf{n}$  in  $\mathbf{P}$  und dem Einheitsvektor  $\mathbf{l}$  in Richtung der Lichtquelle.

Bei Tanakas Methode wird eine topographische Oberfläche als *Lambertscher Reflektierer* unter schräger Beleuchtung mit parallelen Lichtstrahlen angenommen. Die Strichbreite  $b$  einer Konturlinie wird mit dem Kosinus von  $\vartheta$  variiert. Sei  $b_0$  die größte darzustellende Strichbreite, dann ist  $b = b_0 \cos \vartheta$ , mit Maxima bei  $\vartheta \in \{0, \pi/2\}$ .

Zusätzlich erhält die Linie einen hellen Farbton oder Grauwert, wenn  $\cos \vartheta$  größer als Null ist, sonst einen dunklen. Tanaka nennt Konturlinien dieser Art *Relief-Konturen*.

Zur Erzeugung der Konturlinien wird eine Methode von Saito und Takahashi gewählt [50], die im Folgenden beschrieben wird. Danach folgt die Anpassung dieser Methode, um Tanakas Relief-Konturen zu erzeugen.

### 3.2 Die Erzeugung der Konturlinien

Saito und Takahashi präsentieren eine Methode, die die robuste Visualisierung von geglätteten Konturlinien ermöglicht. Es lassen sich beliebige Äquidistanzintervalle angeben. Sie gestattet die Variation von Linienparametern in Abhängigkeit von skalaren oder vektoriellen Feldern. Diese Eigenschaften qualifizieren die Methode zur Konstruktion von Relief-Konturen. Das

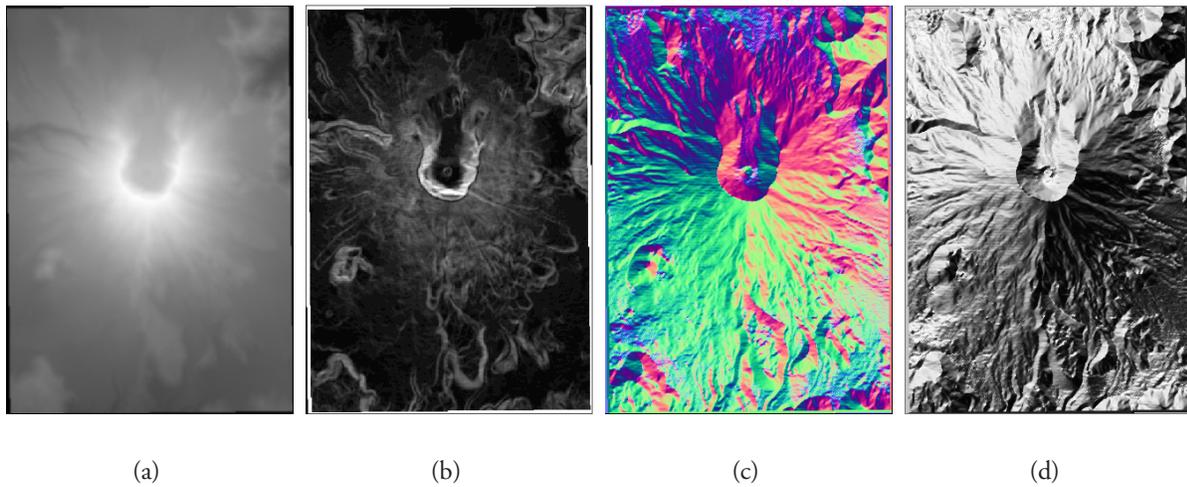


Abbildung 24: Die Datenquellen der Relief-Kontur-Methode als Grauwerte- und RGB-Bilder. (a) Raster der Höhenwerte, (b) Faltung des Höhenwerterasters mit dem ‚Saito-Operator‘, (c) die Oberflächennormalen als RGB-Werte, (d) Abbildung von  $\cos \vartheta$  auf eine Grauwerteskala:  $-1 \mapsto$  Schwarz,  $1 \mapsto$  Weiß. Lichtquelle in Richtung Azimut  $315^\circ$  und Zenit  $45^\circ$ .

Verfahren ist pixelorientiert, die Konturen werden durch Bildverarbeitungsmethoden generiert. Als Eingabebild dient ein Raster  $H$  aus Höhendaten (in Abb. 24(a) als Grauwertebild dargestellt<sup>7</sup>).

Seien  $K$  das Konturlinienbild,  $h_{ij} \in H$  und  $f_b, f_l$  die Hintergrund- bzw. Linienfarbintensität (oder Helligkeit bei Grauwertebildern) in  $K$ . Die Berechnung der Konturlinien der Höhe  $h_0$  erfolgt durch

$$\forall i, j : k_{ij} = f_b + s \left( \frac{|h_{ij} - h_0|}{|\nabla h_{ij}|} \right) \cdot (f_l - f_b)$$

Die Funktion  $s$  dient der Auswahl der Farbintensität in Abhängigkeit vom Abstand der Höhe  $h_{ij}$  von der Konturhöhe, gewichtet mit dem inversen Gradientenbetrag von  $H$  an der Stelle  $(i, j)$ .

Im einfachsten Fall ist  $s$  eine Stufenfunktion

$$s(x) = \begin{cases} 1 & 0 \leq x \leq \frac{b}{2} \\ 0 & \text{sonst} \end{cases}$$

wobei  $b$  die Konturbreite in Pixeln ist. Die Farbintensitäten des Konturbildes beschränken sich bei dieser Funktion  $s$  auf  $f_b$  und  $f_l$ .

<sup>7</sup>Datensatz *Mount St. Helens, Wash. (Post-Eruption)*.

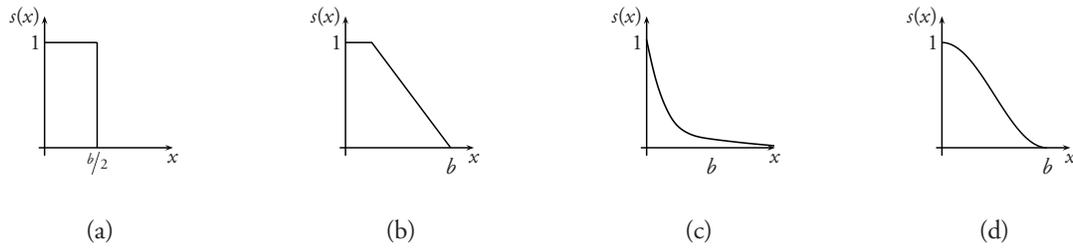
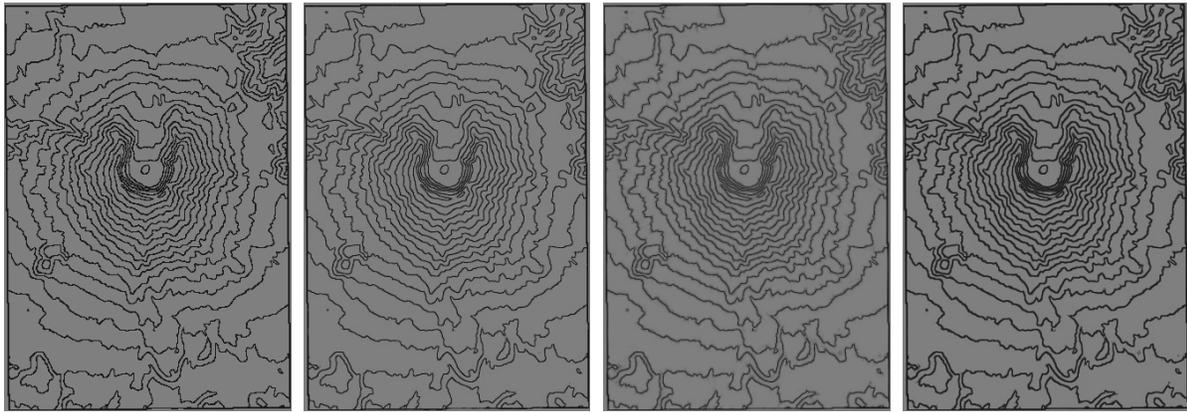


Abbildung 25: Konturlinien mit verschiedenen Funktionen  $s$ , Äquidistanz 100 Meter. (a) Stufenfunktion (b) stückweise linear (c) exponentiell (d) Kosinus (vergrößerte Darstellungen finden sich im Anhang)

Zur Approximation des Gradientenbetrags  $g_{ij} = |\nabla h_{ij}|$  empfehlen die Autoren folgenden Operator:

$$g_{ij} = (|A - h_{ij}| + 2|B - h_{ij}| + |C - h_{ij}| + 2|D - h_{ij}| + 2|E - h_{ij}| + |F - h_{ij}| + 2|G - h_{ij}| + |H - h_{ij}|) / 8$$

Die Symbole  $A, B, \dots, H$  bezeichnen dabei die Pixel in der Achter-Nachbarschaft von  $h_{ij}$  (vgl. Abb. 9).

Wichtig ist die Behandlung des Spezialfalls  $g_{ij} = 0$ . Eine einfache Lösung sieht vor, in diesem Fall  $k_{ij}$  die Hintergrundintensität zuzuweisen, aufgrund der Annahme, daß ein verschwindender Gradient immer zu einem Plateau in  $H$  korrespondiert und ebene Flächen keine Konturierung erhalten.

Äquidistante Konturlinien der Höhe  $h_k = m\Delta h$  mit dem Distanzintervall  $\Delta h$  können durch

iterative Anwendung der folgenden Gleichung bestimmt werden:

$$k_{ij,m} = \min(\forall i,j : f_b + s \left( \frac{|h_{ij} - m\Delta h|}{|\nabla h_{ij}|} \right) \cdot (f_i - f_b), k_{ij,m-1}), 1 \leq m \leq n$$

Dabei wird vorausgesetzt, daß die Helligkeit des Hintergrunds größer als die der Konturlinien ist und daß  $k_{ij,0}$  die höchste Intensität besitzt.

Durch Anpassung der Funktion  $s$  können geglättete Linien produziert werden. In Abb. 25 ist die Konturierung nach Saito und Takahashis Methode für verschiedene Funktionen  $s$  dargestellt. Im linken Bild wird die Stufenfunktion benutzt. Diese führt zu sichtbaren Alias-Effekten. In den folgenden Bildern werden der Reihe nach die (empirisch begründeten) Funktionen

$$s(x) = \begin{cases} 1 & 0 \leq x < \frac{b}{4} \\ 1 - \frac{2}{b}x & \frac{b}{4} \leq x < \frac{3b}{4} \\ 0 & \text{sonst} \end{cases}, \quad s(x) = \exp\left(-\frac{2x}{b}\right) \quad \text{und} \quad s(x) = \begin{cases} \cos\left(\frac{x\pi}{2b}\right) & 0 \leq x < b \\ 0 & \text{sonst} \end{cases}$$

benutzt, wodurch jeweils eine Qualitätssteigerung durch verbesserte Linienglättung erreicht wird. Die Exponentialfunktion führt je nach Wahl des Parameters  $b$  zu einem leichten bis schwerwiegenden Verwischungseffekt.

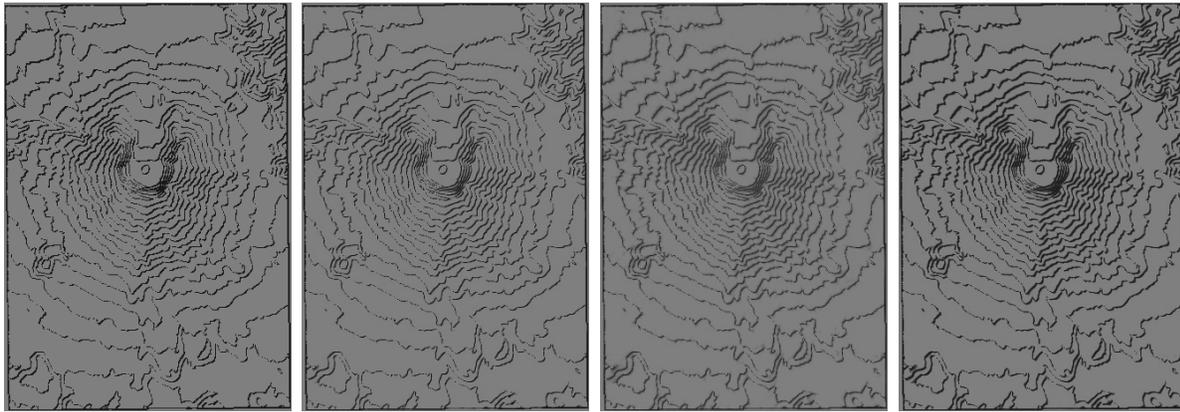
### 3.3 Die Erzeugung der Reliefkonturen

Der Einsatz von Saito und Takahashis Konturlinienmethode zur Erzeugung von Reliefkonturen erfordert die Möglichkeit zur Anpassung der Linienbreite und -helligkeit. Die Variation ist abhängig von der angenommenen Lichtquelle. Tanaka setzt eine unendlich weit entfernte Lichtquelle voraus. Die Beleuchtung erfolgt also mit parallelen Strahlen und der Vektor  $\mathbf{l}$  in Richtung der Lichtquelle ist für jeden Punkt der topographischen Oberfläche gleich. Zur Berechnung des Einfallswinkels der Lichtstrahlen muß für jedes Rasterelement die Oberflächennormale bestimmt werden. Dazu approximiert man erst die Tangentenvektoren des Höhenwertebildes durch Differenzbildung in orthogonalen Richtungen:

$$\Delta_i h_{ij} = \begin{bmatrix} 1 \\ 0 \\ h_{(i+1)j} - h_{ij} \end{bmatrix} \quad \text{und} \quad \Delta_j h_{ij} = \begin{bmatrix} 0 \\ 1 \\ h_{i(j+1)} - h_{ij} \end{bmatrix}$$

Die Oberflächennormale  $\mathbf{n}_{ij}$  erhält man dann durch Bildung des Kreuzprodukts der beiden Tangentenvektoren:

$$\mathbf{n}_{ij} = \Delta_i h_{ij} \times \Delta_j h_{ij} = \begin{bmatrix} h_{ij} - h_{(i+1)j} \\ h_{ij} - h_{i(j+1)} \\ 1 \end{bmatrix}$$



(a)

(b)

(c)

(d)

Abbildung 26: Konturlinien mit variabler Breite bei verschiedenen Funktionen  $s$ . (a) Stufenfunktion (b) stückweise linear (c) exponentiell (d) Kosinus

Bei der Vorgabe eines Lichtvektors  $\mathbf{l}$ , kann der Variationsparameter der Konturlinien  $\cos \vartheta_{ij} = \frac{\mathbf{n}_{ij}}{\|\mathbf{n}_{ij}\|} \cdot \frac{\mathbf{l}}{\|\mathbf{l}\|}$  für alle  $i, j$  bestimmt werden. In Abb. 24(d) ist die Darstellung von  $\cos \vartheta_{ij}$  als Grauwerteskala zu sehen.

Die Variation der Linienbreite in einem Intervall  $[0, b_0]$  erhält man nach Substitution der Breite  $b$  in der Funktion  $s$  durch  $b_0 \cos \vartheta_{ij}$ . Der Effekt ist in Abb. 26 dargestellt.

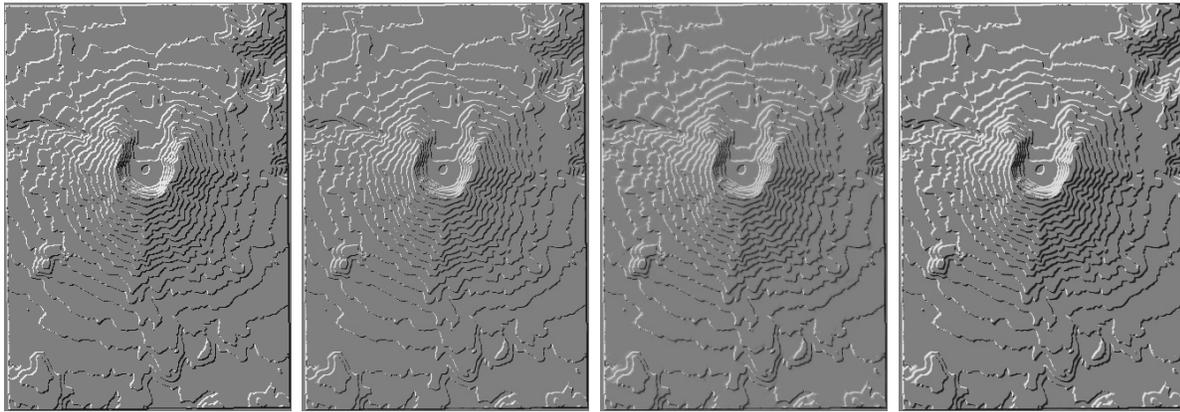
Der letzte Schritt besteht in der Anpassung der Helligkeit der Konturlinien unter Lichteinfall. Für  $k_{ij}$  lautet die Vorgehensweise in Pseudocode, wieder unter der Voraussetzung von  $f_b > f_l$ :

1. wenn  $k_{ij} \geq f_b$ , dann beenden
2. wenn  $\cos \vartheta_{ij} < 0$ , dann beenden
3.  $k_{ij} := f_l - (k_{ij} - f_l)$

Zuerst wird dabei geprüft, ob sich an der Stelle überhaupt eine Konturlinie befindet, gefolgt von der Entscheidung, ob Lichteinfall vorliegt. Im Erfolgsfall wird  $k_{ij}$  die helle Konturlinienintensität  $f_l$  zugewiesen, modifiziert durch einen Korrekturterm zur Glättung.

Einige nützliche Werte für  $f_b, f_l$  und  $f_l'$  sind [59]:

	HELLIGKEIT $\in [0, 1]$		
TONWERT	$f_l'$	$f_b$	$f_l$
dunkel	0.8	0.4	0.02
mittel	0.85	0.45	0.05
hell	0.9	0.5	0.1



(a)

(b)

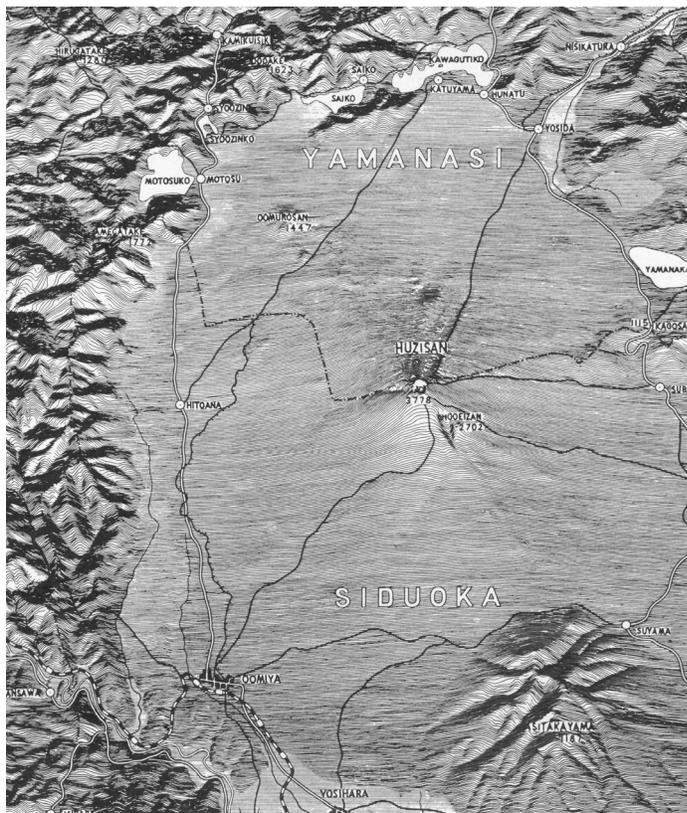
(c)

(d)

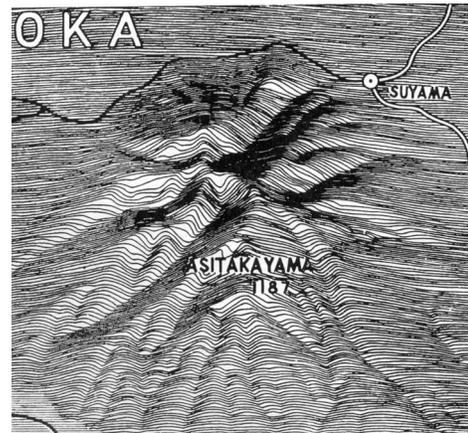
---

Abbildung 27: Relief-Konturlinien mit verschiedenen Funktionen  $s$ . (a) Stufenfunktion (b) stückweise linear (c) exponentiell (d) Kosinus

Das Ergebnis der ganzen Prozedur zur Erzeugung von Relief-Konturen ist in Abb. 27 dargestellt. Man beachte den qualitativen Sprung bei der Wahrnehmung der Reliefformen im Vergleich der Konturliniendarstellungen mit und ohne Helligkeitsvariation.



(a)



(b)

Abbildung 28: Methode des orthogonalen Reliefs. (a) Region des Huzisan (Fujisan), (b) vergrößerter Ausschnitt des süd-östlichen Quadranten. ([58])

## 4 Die Methode des orthogonalen Reliefs

Der Kartograph Kitirô Tanaka ist Urheber einer weiteren Art der Darstellung von Geländereiefs. Die erzeugende Methode (*orthographical relief method*) [58] ist Gegenstand dieses Kapitels. In der obigen Abbildung sind eine Kartendarstellung nach Tanakas Methode und ein vergrößerter Ausschnitt zu sehen. Das Bild ähnelt der Daraufrsicht eines geschummerten Reliefs mit Südbeleuchtung. Die Flächentönung ist jedoch nur scheinbar, da eine reine Liniendarstellung vorliegt: unterschiedliche Tonwerte entstehen durch die Variation der Zwischenräume. Man kann das Prinzip als ein Halbtönungsverfahren mit Linien ansehen.

Tanaka nennt die einzelne Linie einen *geneigten Umriß* (engl. *inclined contour*), definiert als der

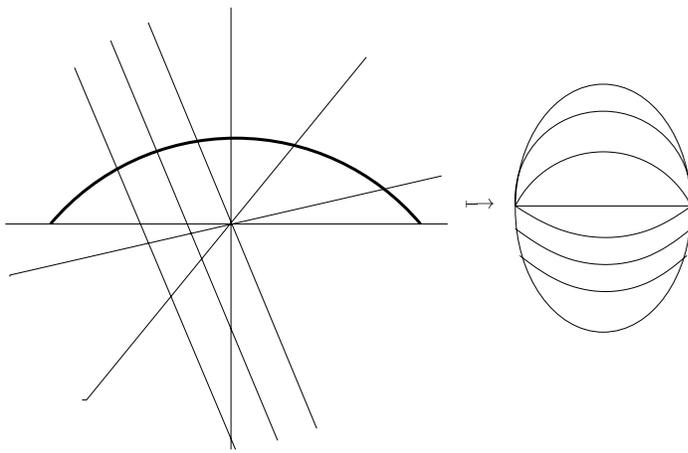


Abbildung 29: Grundriß von Schnitten eines elliptischen Körpers mit verschiedenen geneigten Ebenen.

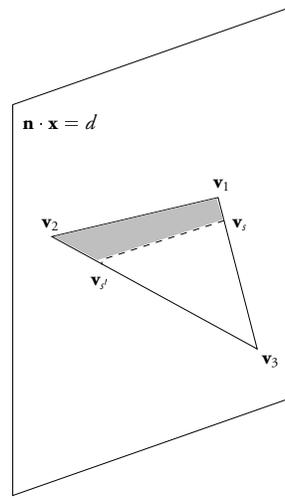


Abbildung 30: Schnitt eines Dreiecks mit einer Ebene.

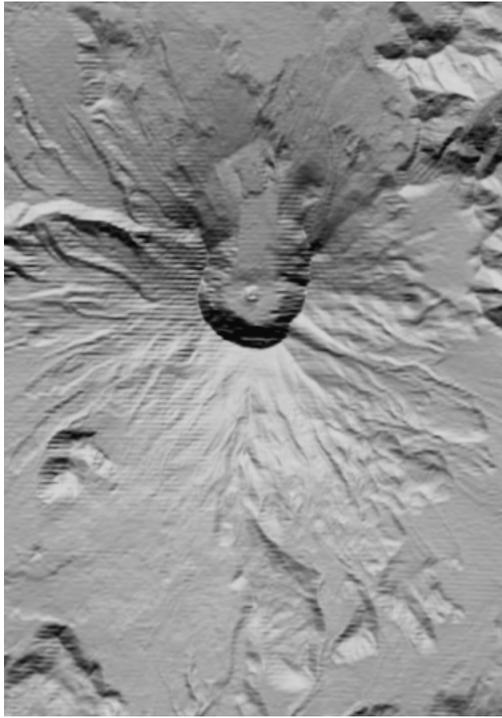
Grundriß eines Schnitts einer schiefen Ebene mit der Geländeoberfläche. Die Neigung der Ebene wird dabei so gewählt, daß der Umriss weder einer üblichen Höhenlinie (Grundriß horizontaler Schnittebene), noch einer Profillinie (Grundriß vertikaler Schnittebene) entspricht (vgl. Abb. 30). In dem Beispiel sind die Ebenen gegen den oberen Kartenrand hin geneigt, bilden dabei mit der Horizontalebene einen Winkel von  $45^\circ$  und schneiden diese in West-Ost-Geraden.

Die geneigten Umrisse bei dicht gescharten äquidistanten Parallelebenen lassen den Eindruck eines geschummerten Reliefs entstehen. Dabei variiert der Tonwert, resp. der Zwischenraum benachbarter Linien, mit dem Kosinus des Flächenwinkels zwischen der Schnittebene und den Tangentialebenen des Geländes. Man stellt leicht den Bezug zum Gesetz Lambertscher Reflexion her: Der Tonwert ist abhängig von dem Kosinus des Winkels zwischen dem Lichtvektor, i. e. dem Normalenvektor der Schnittebene und dem Normalenvektor der Tangentialebenen.

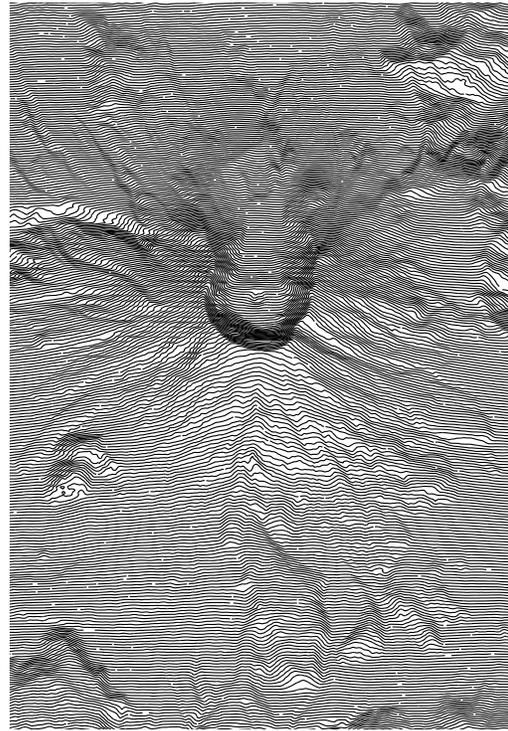
Die Herstellung der Umrisslinien erfolgt in Tanakas Arbeit konstruktiv-zeichnerisch unter Benutzung von Höhenlinien und einer überlagerten Schar paralleler Geraden.

Die Verfügbarkeit eines digitalen Geländemodells ermöglicht die Erzeugung von Karten nach Tanakas Methode durch direkte Umsetzung der Definition der geneigten Umrisse, indem erst die Schnitte der Parallelebenen mit dem Dreieckmodell ermittelt und diese dann durch Orthogonalprojektion grundrißlich dargestellt werden.

Dieses Vorgehen ähnelt der Schraffurmethode von Deussen et al. [14] (vgl. Abschnitt 2.3.3), mit dem Unterschied, daß die Linien im Objektraum bestimmt werden. Dies geschieht durch Anwendung eines Polygon-Clipping-Algorithmus' auf das Dreieckmodell  $M$  [16]. Seien  $\mathbf{n} \cdot \mathbf{x} = d$  die Schnittebene und  $\mathbf{v}_i$ ,  $i \in \{1, 2, 3\}$ , die Knoten eines Dreiecks  $\Delta \in M$ . Eine Kante  $(\mathbf{v}_j, \mathbf{v}_k)$ ,  $j \neq k$ , von  $\Delta$ , schneidet genau dann die Ebene, wenn  $p_j \cdot p_k < 0$  ist, wobei  $p_j = \mathbf{n} \cdot \mathbf{v}_j - d$  und  $p_k =$



(a)



(b)

Abbildung 31: Vergleich der Schummerung und Methode des orthogonalen Reliefs zur Darstellung eines Geländemodells. (a) Schummerung durch gerichtetes Licht, Südbeleuchtung, Lichtquelle mit Zenit  $45^\circ$ , (b) geneigte Umriss, Schnittebenen im Winkel von  $45^\circ$  zur Horizontalebene.

$\mathbf{n} \cdot \mathbf{v}_k - d$ . Die Ungleichung drückt aus, daß ein Knoten der Kante in dem von der Ebene erzeugten positiven Halbraum liegt und der andere Knoten in dem negativen Halbraum; Dreiecke, die in der Schnittebene liegen, enthalten somit keine schneidenden Kanten. Der Schnittpunkt  $\mathbf{v}_s$  auf der Kante ist

$$\mathbf{v}_s = \mathbf{v}_j + \frac{p_j}{p_j - p_k} (\mathbf{v}_k - \mathbf{v}_j)$$

Die Linie zwischen den Schnittpunkten der Ebene mit den Kanten von  $\triangle$ , für die die obige Ungleichung zutrifft, ist die Schnittmenge von  $\triangle$  mit  $\mathbf{n} \cdot \mathbf{x} = d$ . Zu einem geneigten Umriss gehören alle Linien, die durch den Ebenenschnitt mit dem Geländemodell erhalten werden. Eine Rendering-Bibliothek liefert die Grundrißperspektive. In Abb. 31(b) ist eine Darstellung mit geneigten Umrissen zu sehen.

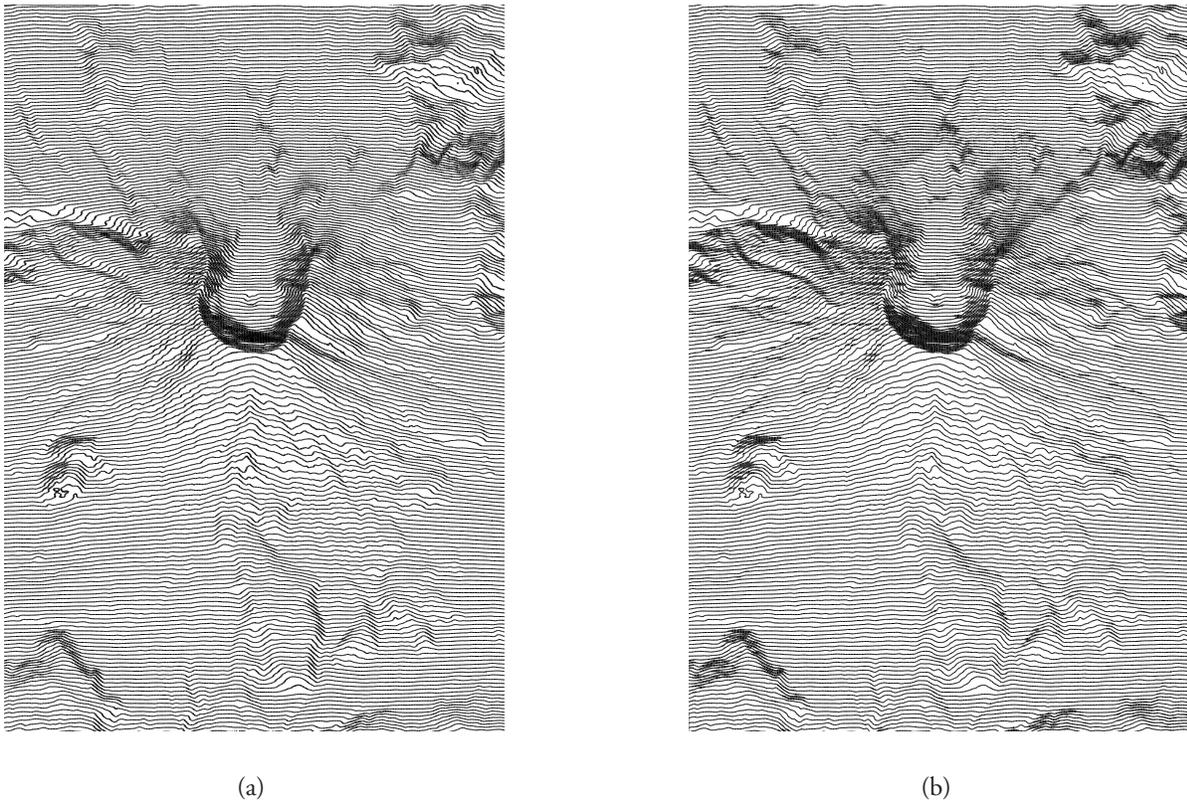


Abbildung 32: Variation der Linienbreite der geneigten Umrisse. (a) nach dem Prinzip der Böschungsschraffe, (b) nach dem Prinzip der Schattenschraffe.

Ist die Geländeböschung größer als die Neigung der Schnittebene, kann es zur Darstellung von geneigten Umrissen kommen, die eigentlich durch das Relief verdeckt wären. Diese können mit einer der üblichen Methoden zur Entfernung verdeckter Linien ausgeblendet werden wie beispielsweise der Methode des gleitenden Horizonts [18].

Durch Variation der Linien- und Schnittebenenparameter lassen sich unterschiedliche Effekte erzielen. Tanaka berichtet von Versuchen, die Breite der geneigten Umrisse in Abhängigkeit der Geländeneigung anzupassen, beschränkt sich jedoch aus Gründen der Einfachheit auf eine konstante Linienbreite<sup>8</sup>. Er empfiehlt eine mittlere Anzahl von dreißig bis siebzig Linien pro Zentimeter, mit einer Liniendicke von  $0.36d$ , wobei  $d$  der Zwischenraum geneigter Umrisse auf flachem

<sup>8</sup>„An attempt was made to represent the thickness of these inclined contours by a mathematical formula [...]. To apply this refinement to the actual drawing however would be impracticable, if not impossible, since it would be necessary to change the thickness of the inclined contours for every inclination and every direction of the inclination of the ground surface.“[58]

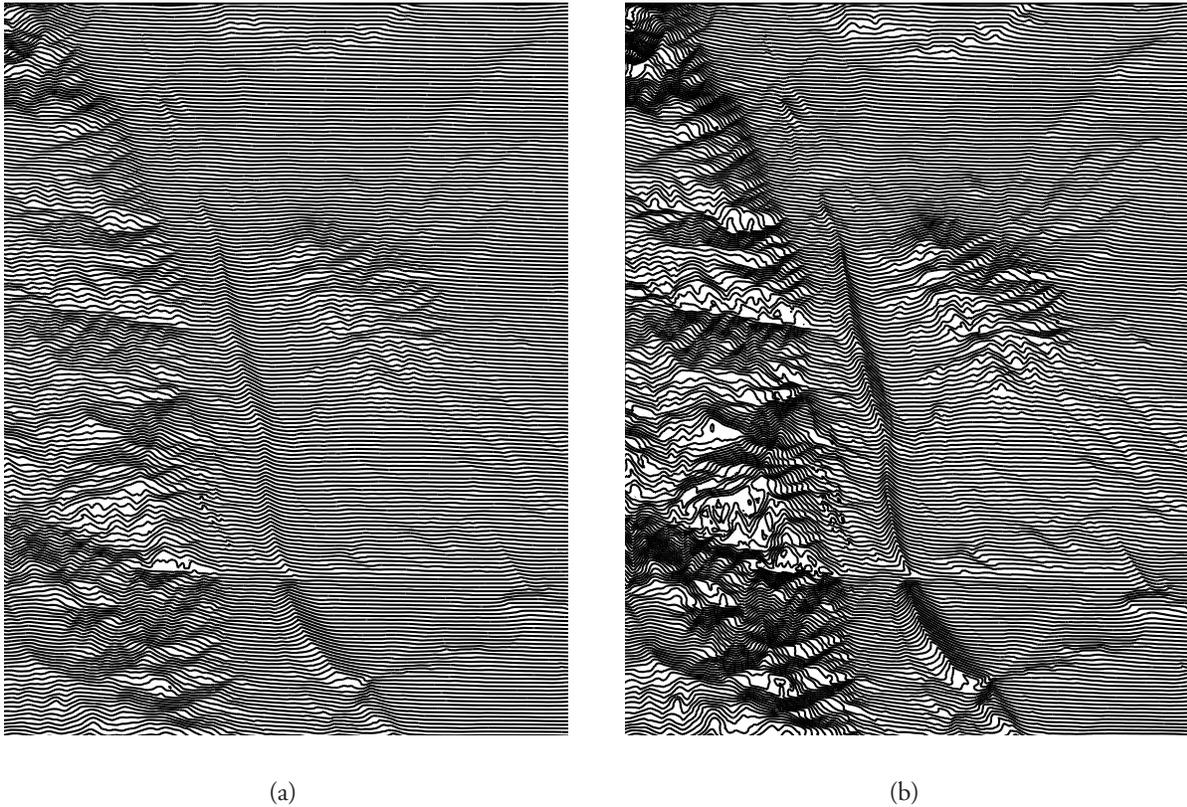


Abbildung 33: Effekt der scheinbaren vertikalen Überhöhung. In (b) werden die Schnittebenen aus (a) mit halbiertem Neigung verwendet, wodurch die Reliefformen in (b) vertikal überhöht erscheinen.

Terrain ist. In Abb. 32(a) und 32(b) wird die Breite der geneigten Umriss nach dem Prinzip der Böschungsschraffe, ‚je steiler, desto breiter (i. e. dunkler)‘, und, unter Voraussetzung eines Beleuchtungsmodells mit diffuser Reflexion, nach dem Prinzip der Schattenschraffe, ‚je heller, desto schmaler‘, variiert.

Thrower [60] weist auf den Effekt der *scheinbaren vertikalen Überhöhung* hin, der bei der Verringerung des Winkels zwischen der Schnittebene und der Horizontalebene auftritt. Scheinbare vertikale Überhöhung kann verwendet werden, um bei Darstellungen in kleinem Maßstab die Reliefformen prägnanter in Erscheinung treten zu lassen, ohne eine tatsächliche Maßstabsänderung in der Vertikalen vorzunehmen. In Abb. 33 wird der Effekt beim Vergleich zweier Darstellungen einer Geländesituation mit unterschiedlichen Neigungen der Schnittebenen deutlich.

## 5 Resümee und Ausblick

In dieser Arbeit wurde die Anwendbarkeit von Elementen nichtphotorealistischer Computergraphik in Bezug auf die Erzeugung topographischer Darstellungen anhand dreier konstruktiv-zeichnerischer Darstellungsarten, Blockbilder, Relief-Konturen und orthogonale Reliefs, untersucht. Es ist gezeigt worden, wie traditionelle, manuelle kartographische Techniken sich mittels Methoden der digitalen Datenverarbeitung erschließen lassen.

Dabei hat sich herausgestellt, daß sich bestimmte Aspekte im Herstellungsprozess topographischer Darstellungen sachgerecht automatisieren lassen. Dazu gehören etwa die Platzierung von äquidistanten Schraffurlinien oder die Variation von Konturbreiten nach Beleuchtungsverhältnissen.

Ohne die umfangreiche Anzahl von Ansätzen nichtphotorealistischer Darstellungstechniken ausgeschöpft zu haben, lassen sich schon aussagekräftige Bilder erzeugen, im Sinne der Vermittlung von Reliefformen und Geländesituationen.

Insbesondere erlauben die beiden von Tanaka präsentierten Verfahren, bedingt durch ihre stringente, formale Begründung basierend auf dem Gesetz Lambertscher Reflexion, den unmittelbaren Einsatz computergraphischer Darstellung. Im Fall der Methode des orthogonalen Reliefs steht die automatisierte Umsetzung der Definition geneigter Umrisse sogar näher als die manuelle Technik von Tanaka selbst, wodurch die von Tanaka geforderten aber nicht durchgeführten Variationen auf einfache Weise erfüllt werden konnten.

Andere Aspekte der Darstellung bleiben in den präsentierten Anwendungen unberücksichtigt. Diese sind enger mit der Semantik kartographischer Darstellungen verknüpft und nicht so leicht zu realisieren wie die technische Ausführung der Bilderzeugung. Diese Aspekte umfassen z. B. den komplexen Bereich der kartographischen Generalisierung. Dazu gehören Themen wie die Auswahl der darzustellenden Objekte, ihre Gruppierung und Zusammenfassung aufgrund gemeinsamer Merkmale. Solche Prozesse basieren auf klassifikatorischen Leistungen. Vereinfachung sowie die Betonung von Kartenzeichen, je nach Bedeutung der dargestellten Objekte, sind weitere Beispiele für Generalisierung. Diese Formen der Typisierung und Abstraktion setzen kartographisches Expertenwissen voraus.

Interessant wäre deshalb die Einbettung der präsentierten Techniken in ein Rahmenwerk zur topographischen Kartenerstellung in Kombination mit einem regelbasierten System, das kartographisches Wissen speichert. Solches Wissen würde nicht nur *Prozeduren* umfassen, wie sie in dieser Arbeit zum Einsatz gekommen sind, sondern auch *konzeptuelle* Einsichten, die zur Abstraktion befähigen. Damit ließen sich fortgeschrittene Darstellungsaufgaben lösen und neben einer zeichnerischen Rationalisierung vor allem die Abstimmung der verschiedenen Bildelemente aufeinander erreichen, zur Erzeugung von kartographischen Bildern, die über eine inhaltliche und graphische Einheit verfügen. Dies liegt im Kern der nichtphotorealistischen Computergraphik.

## A Technische Details der Arbeit

Die der Arbeit zugrundeliegenden digitalen Höhenmodelle tragen die Bezeichnung *USGS ASCII DEM* und werden vom U.S. Geological Survey herausgegeben. Sie verfügen über ein Format mit gut dokumentierter Struktur [61].

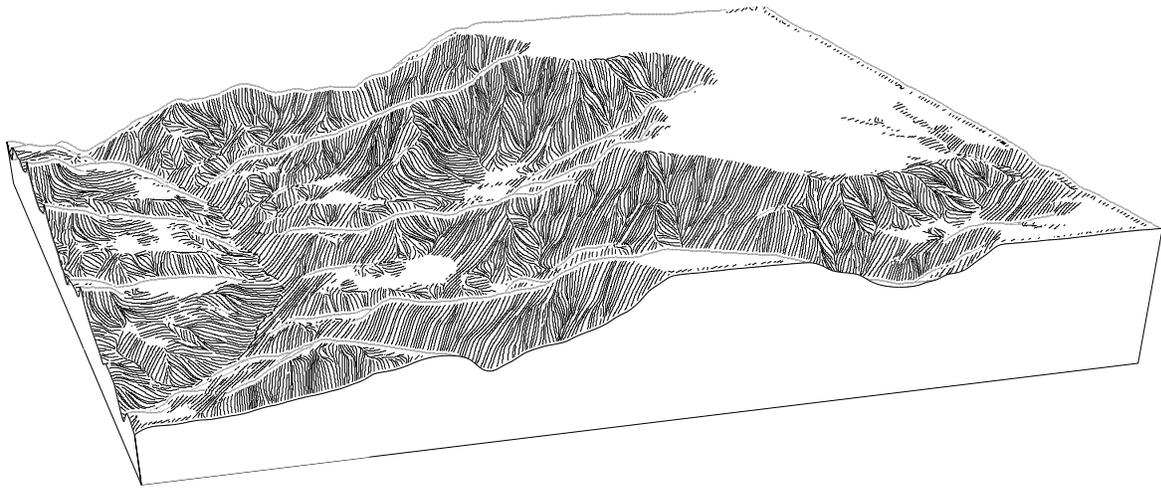
Die gesamte Programmierarbeit erfolgte unter Benutzung der Sprachen C++ und PostScript. Zum Aufbau der Dreieckmodelle wurde eine für diese Arbeit modifizierte Version des *GNU Triangulated Surface*-Programmpakets [42] eingesetzt. Die PostScript-Ausgabe von OpenGL-Graphiken wurde mit dem Programm *GL2PS* erzeugt [22].

Soweit möglich wurden englische Fachtermini mit [37] übersetzt.

## B Weitere Beispiele für Blockbilder

Seite 45: *Grand Canyon, Arizona*. Blockbild mit Schraffurmaske, Silhouettenhalo und Variation des Schraffurstrich-Zwischenraums. (a)  $d_{sep} = 3$ , (b)  $d_{sep} = 6$ , (c)  $d_{sep} = 9$ .

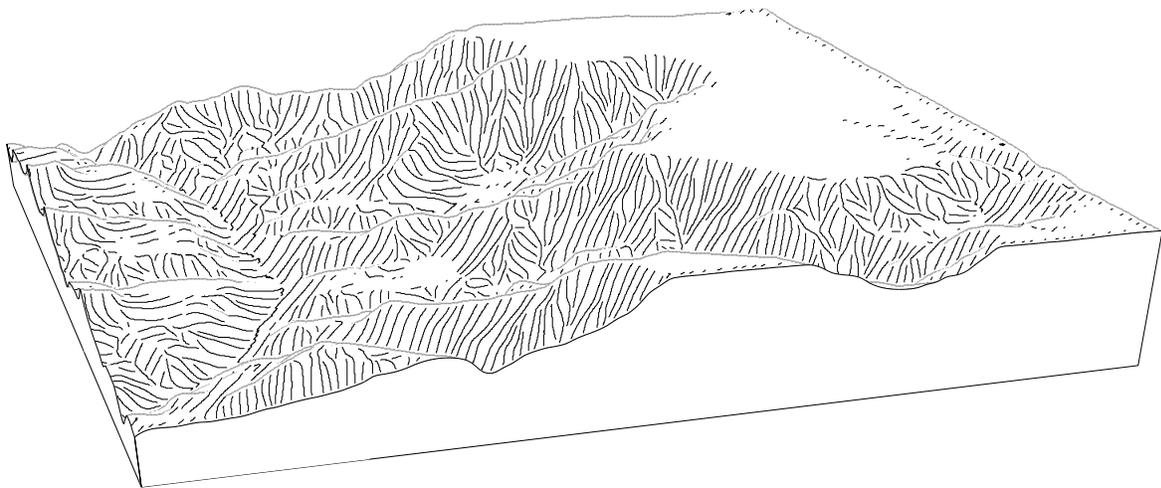
Seite 46: Schraffurstrichbreite variiert nach Beleuchtungsverhältnissen. Die Lichtquelle liegt jenseits des linken Bildrands.



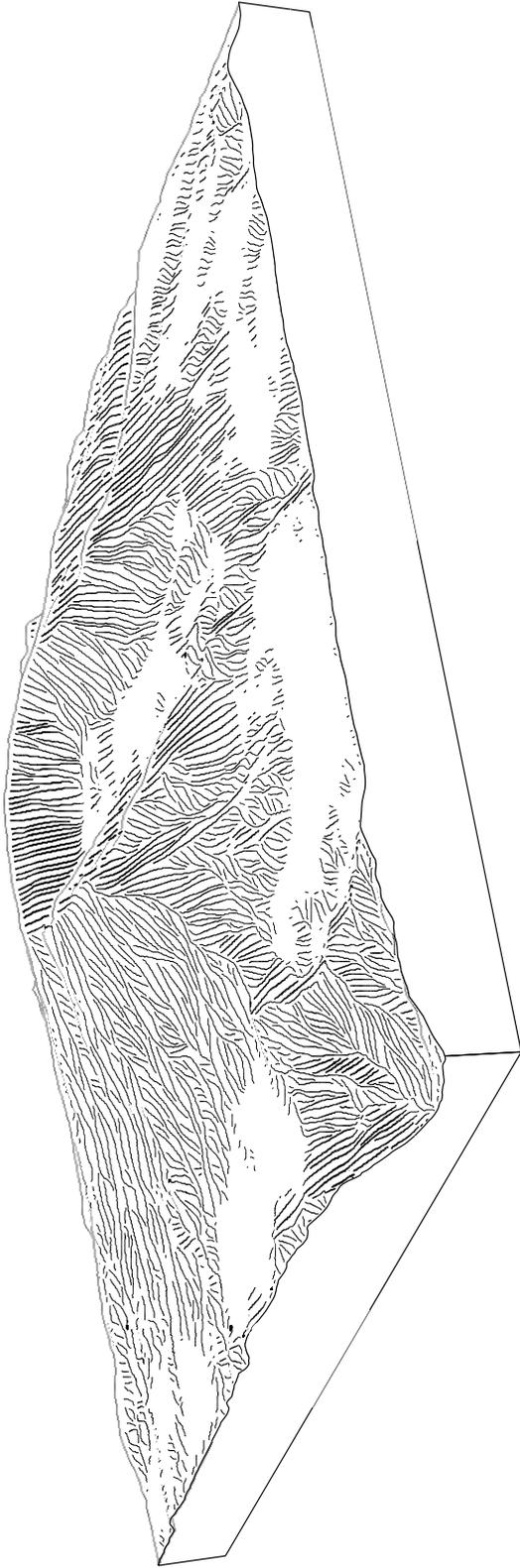
(a)



(b)



(c)



## C Beispiele für Relief-Kontur-Darstellungen

Die folgenden Darstellungen zeigen vergrößerte Abbildungen aus dem Kapitel 3 sowie die Variation von Äquidistanz und Tonwerten.

Seite 48: Konturen nach der Methode von Saito und Takahashi unter Benutzung von verschiedenen Schwellenwertfunktionen. (a) Stufenfunktion  $s_{bil}$ , (b) stückweise linear  $s_{lin}$ , (c) exponentiell  $s_{exp}$ , (d) Kosinus  $s_{cos}$ .

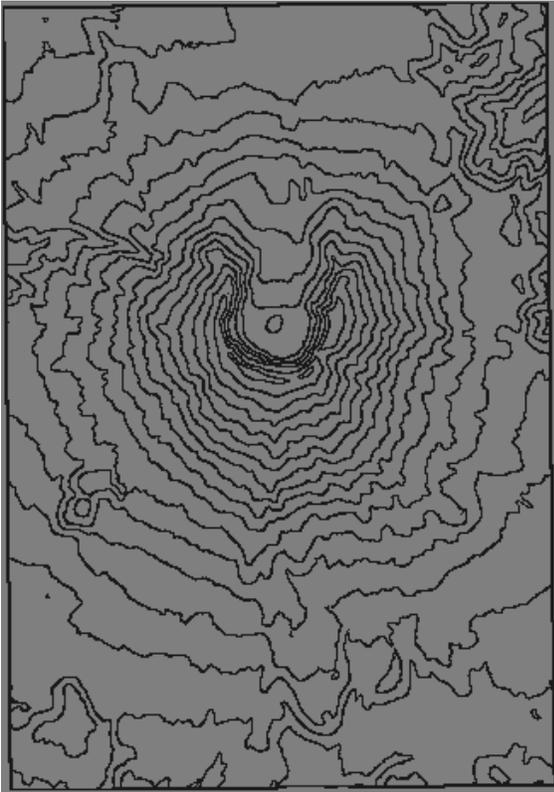
Seite 49: Variation der Konturbreite nach Beleuchtungsverhältnissen. (a)  $s_{bil}$ , (b)  $s_{lin}$ , (c)  $s_{exp}$ , (d)  $s_{cos}$ .

Seite 51: Relief-Konturen. (a)  $s_{bil}$ , (b)  $s_{lin}$ , (c)  $s_{exp}$ , (d)  $s_{cos}$ .

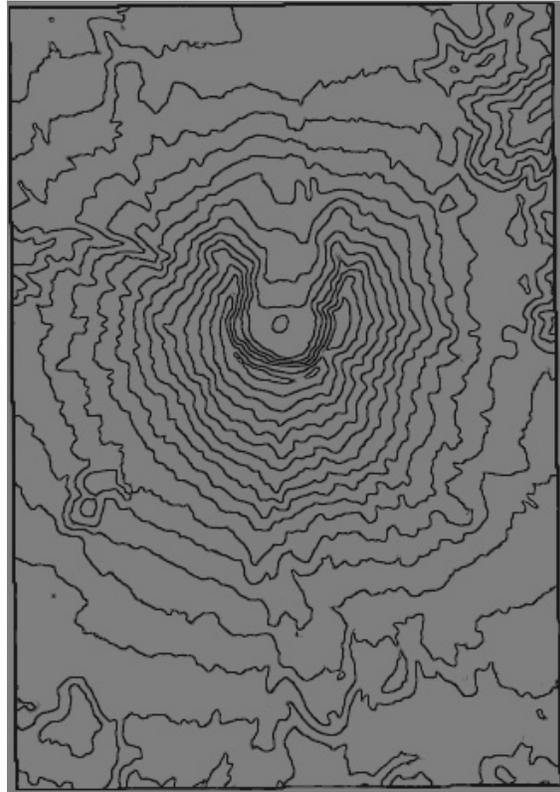
Seite 52: Variation der Konturbreite, Äquidistanz 30 Meter,  $s = s_{cos}$ . (a)  $b = 0.5$ , (b)  $b = 1$ , (c)  $b = 1.5$ , (d)  $b = 2$ .

Seite 50: Variation der Äquidistanz. (a) 25 Meter, (b) 50 Meter, (c) 75 Meter, (d) 100 Meter. Bei Verringerung des Distanzintervalls nähert sich die Darstellung einem schattierten Schummerungsbild an. Durch die gedrängte Linienscharung treten in (a) und (b) Moiré-Effekte auf.

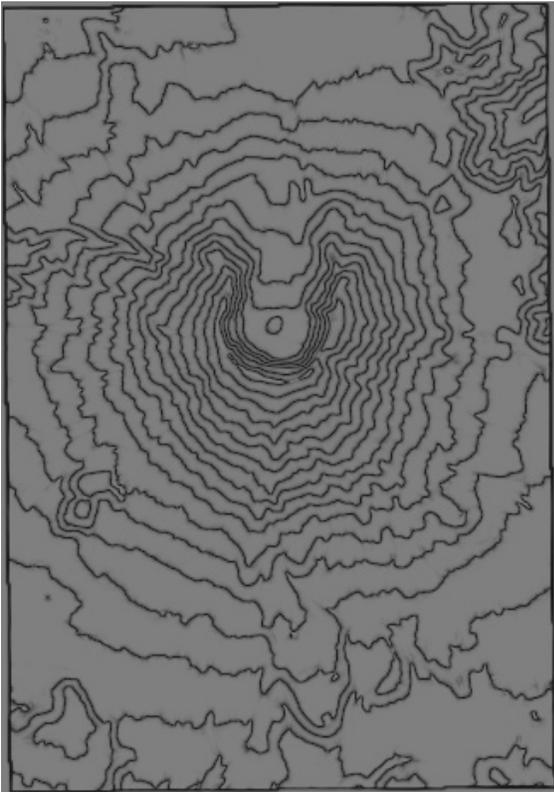
Seite 53: *The Cradle, Washington*. Maßstab ca. 1 : 65000, Äquidistanz 45 Meter, Tonwerte:  $f_v = 1$ ,  $f_b = 0.7$ ,  $f_l = 0.25$ ,  $s = s_{cos}$ .



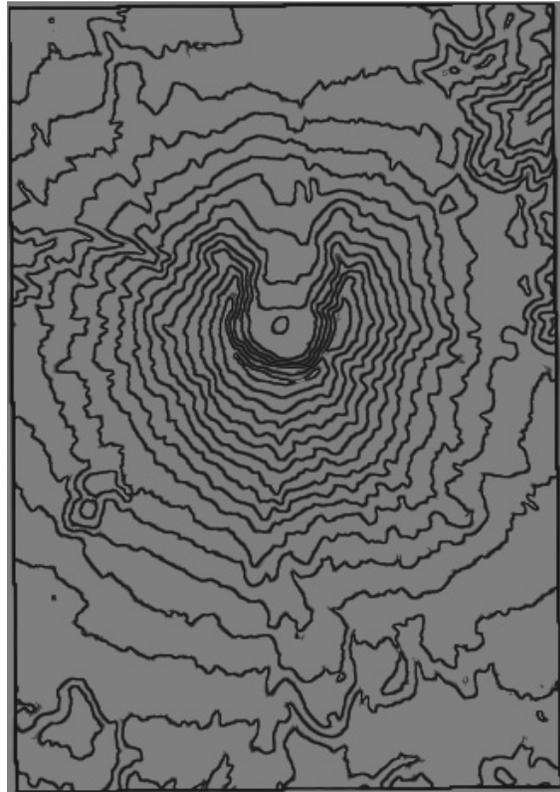
(a)



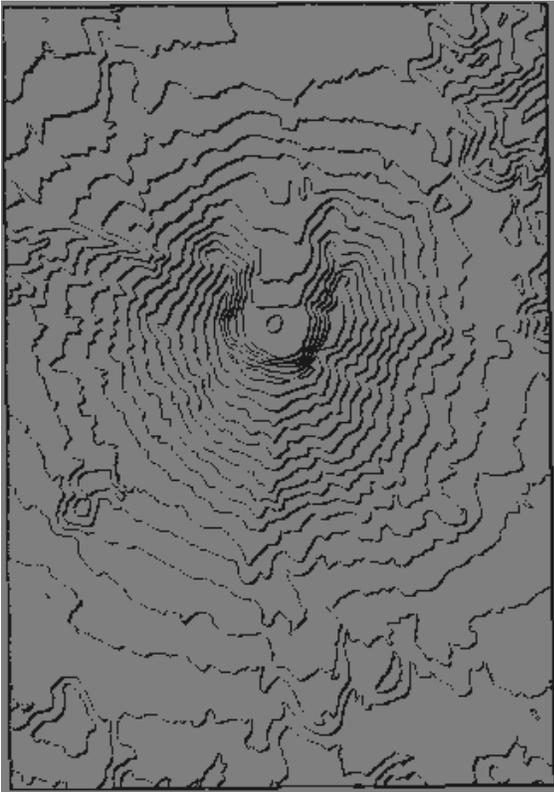
(b)



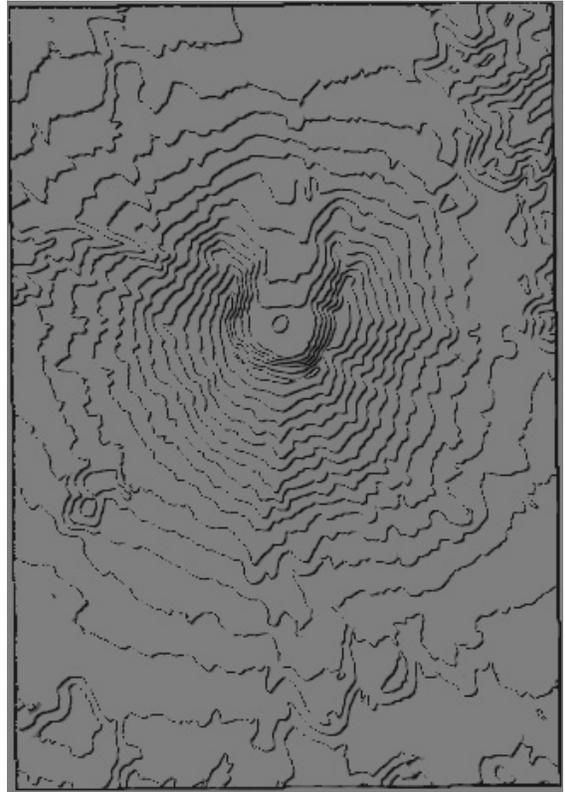
(c)



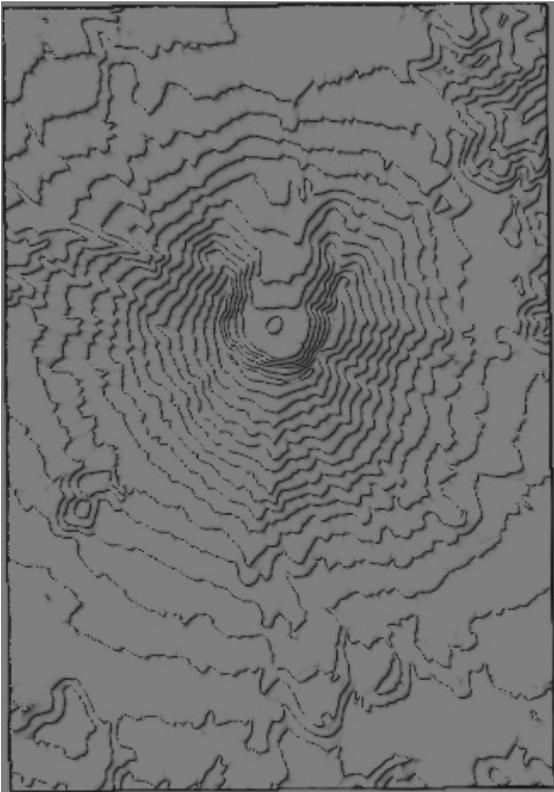
(d)



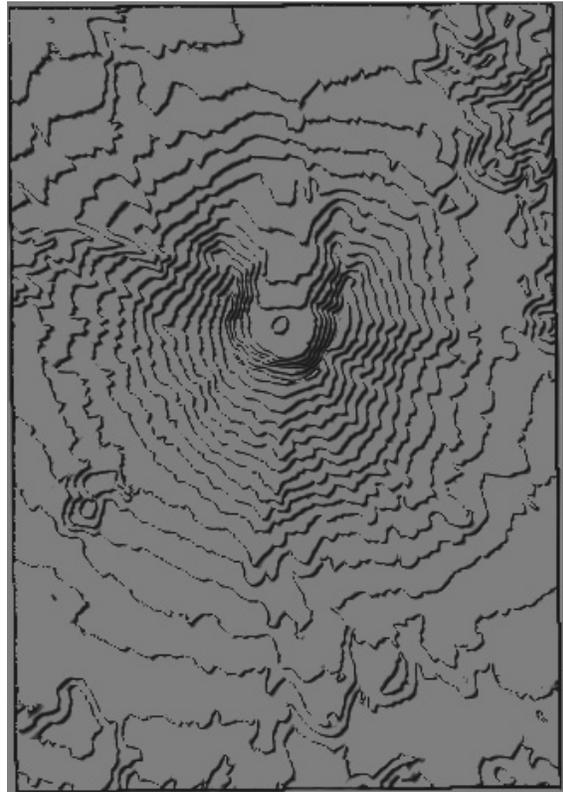
(a)



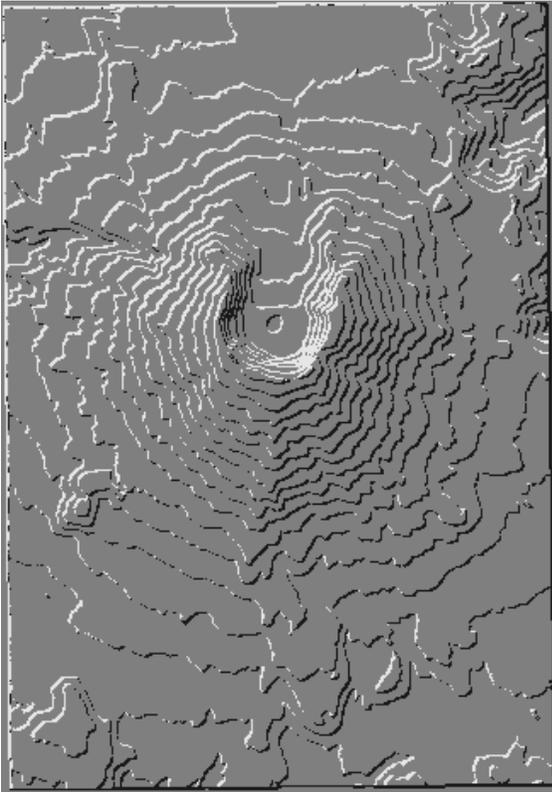
(b)



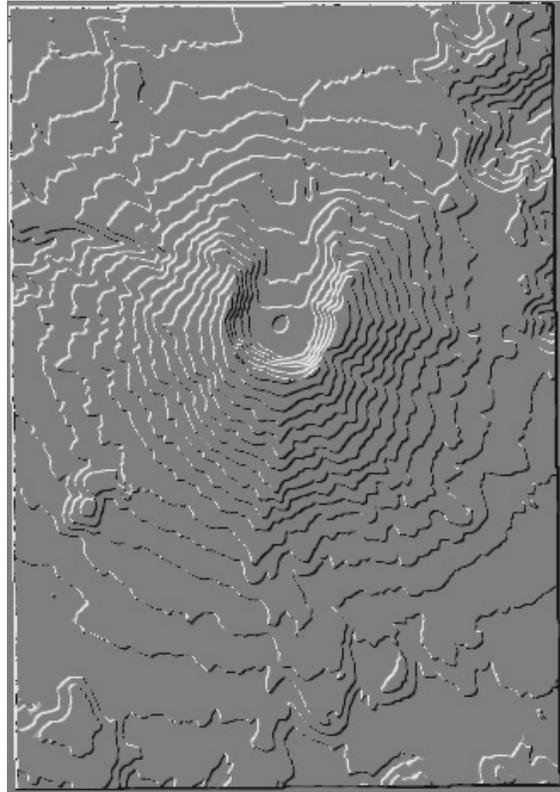
(c)



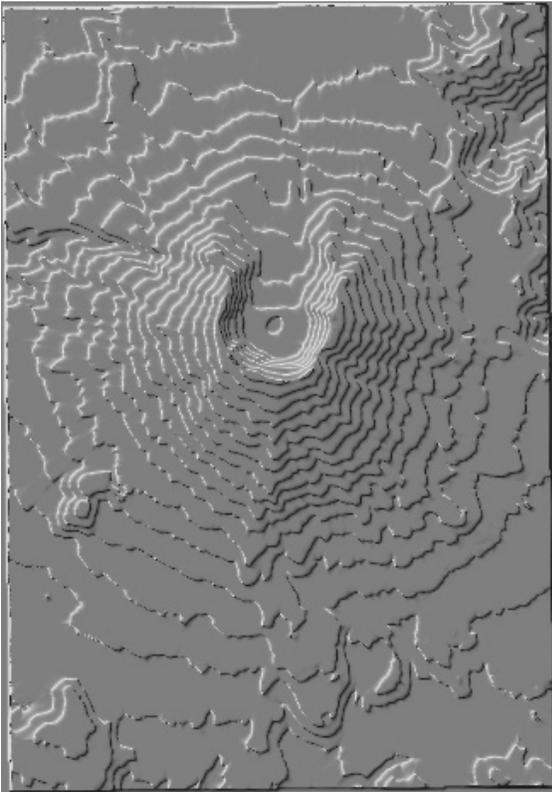
(d)



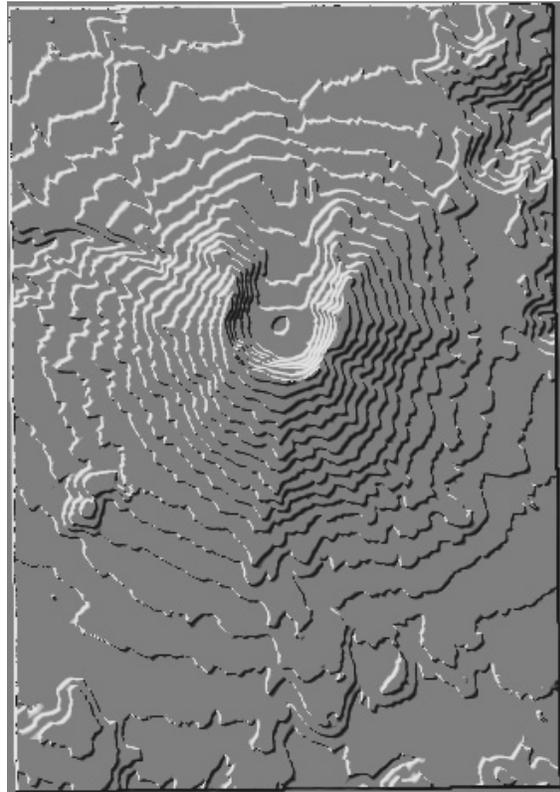
(a)



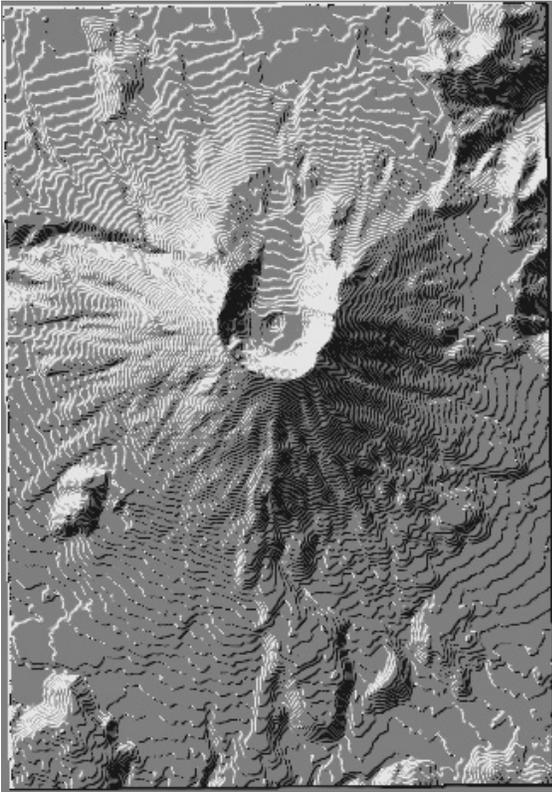
(b)



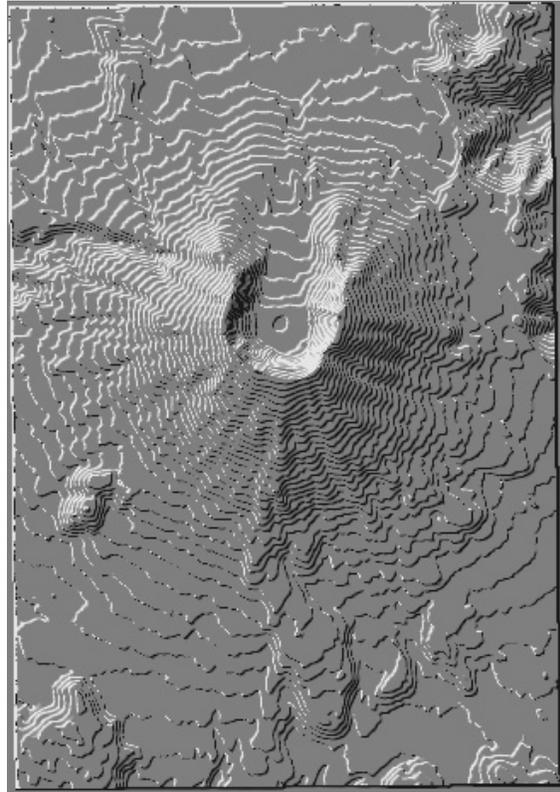
(c)



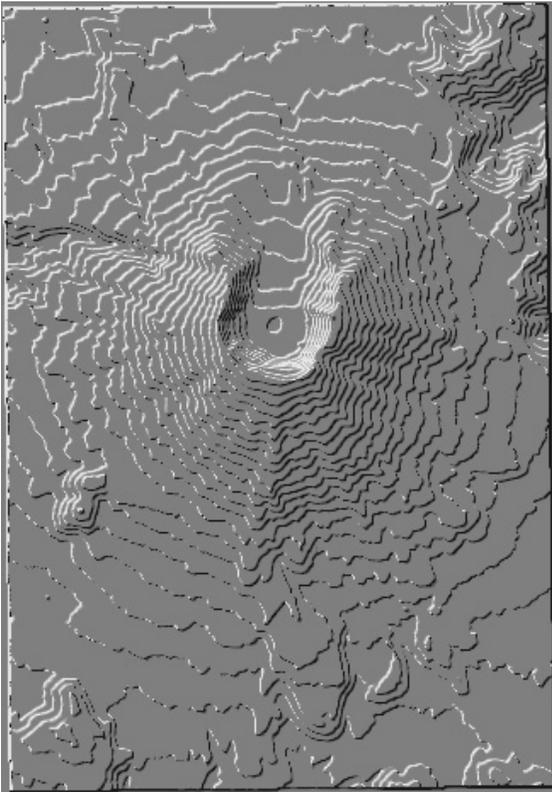
(d)



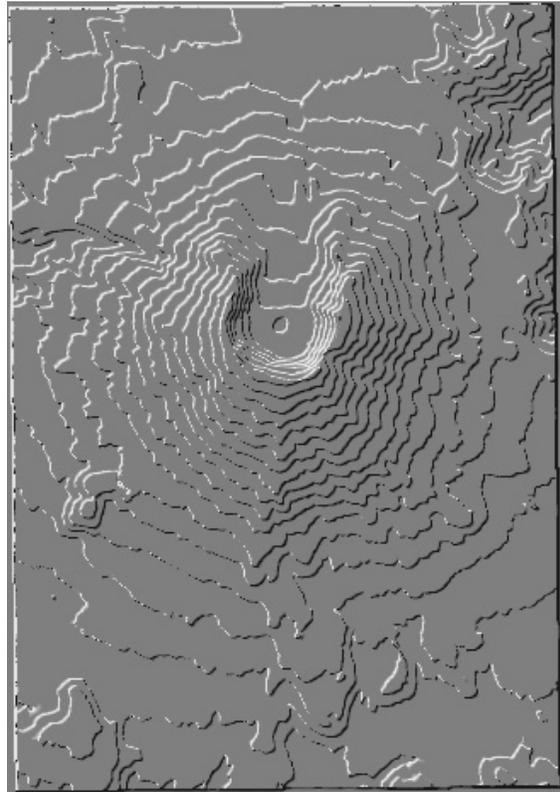
(a)



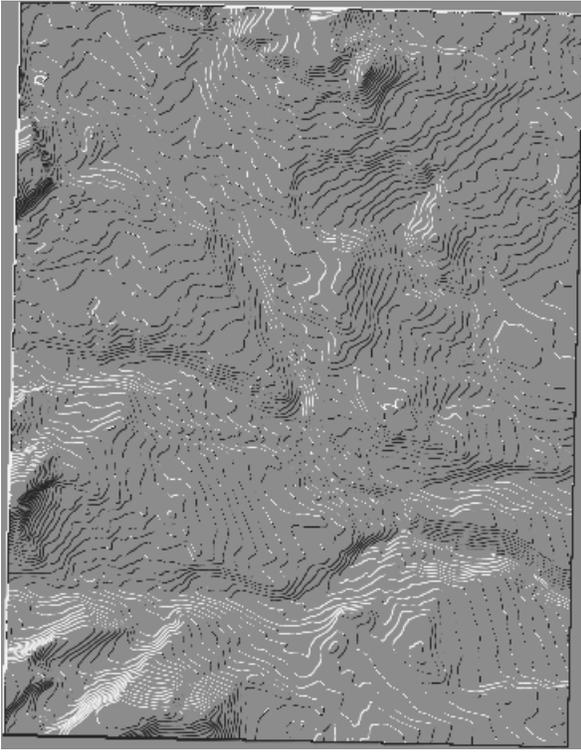
(b)



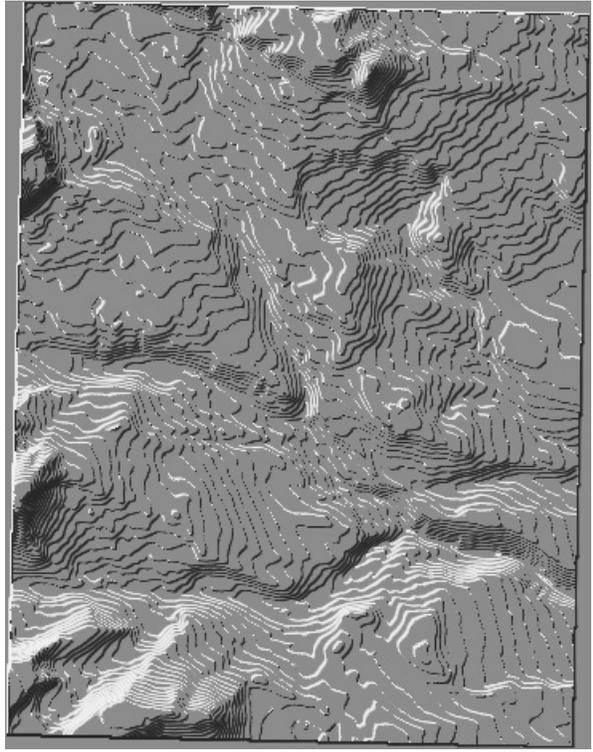
(c)



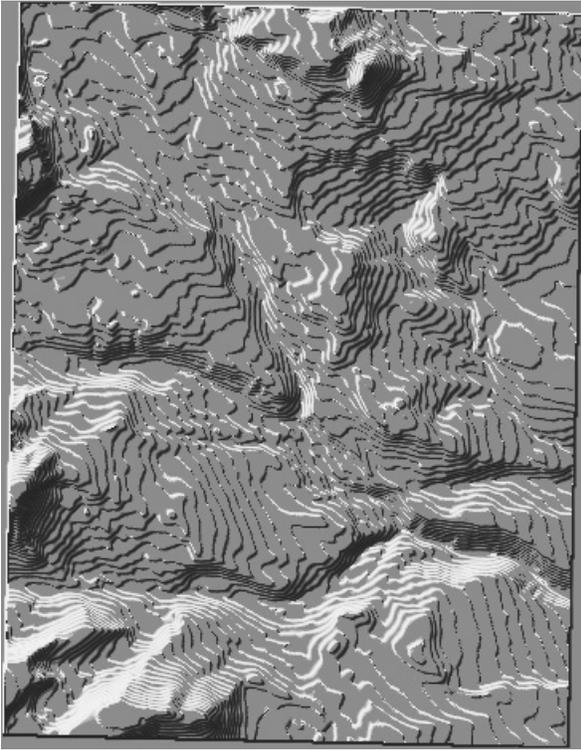
(d)



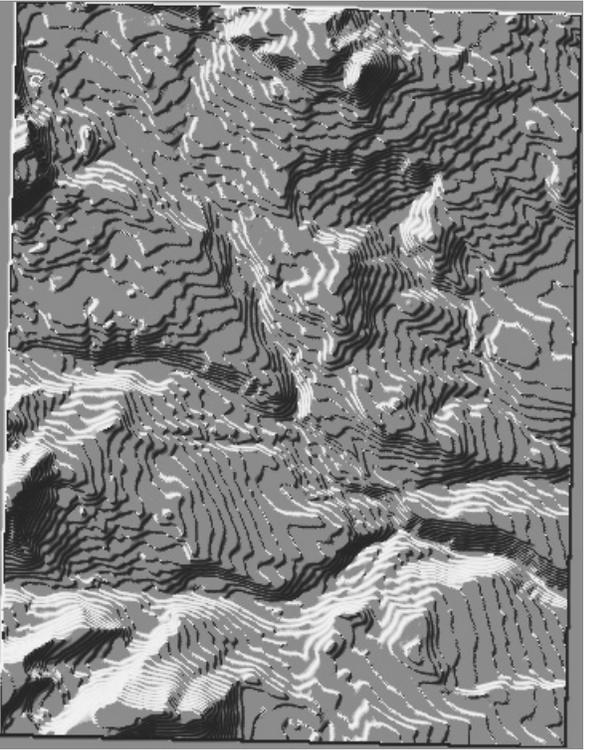
(a)



(b)



(c)

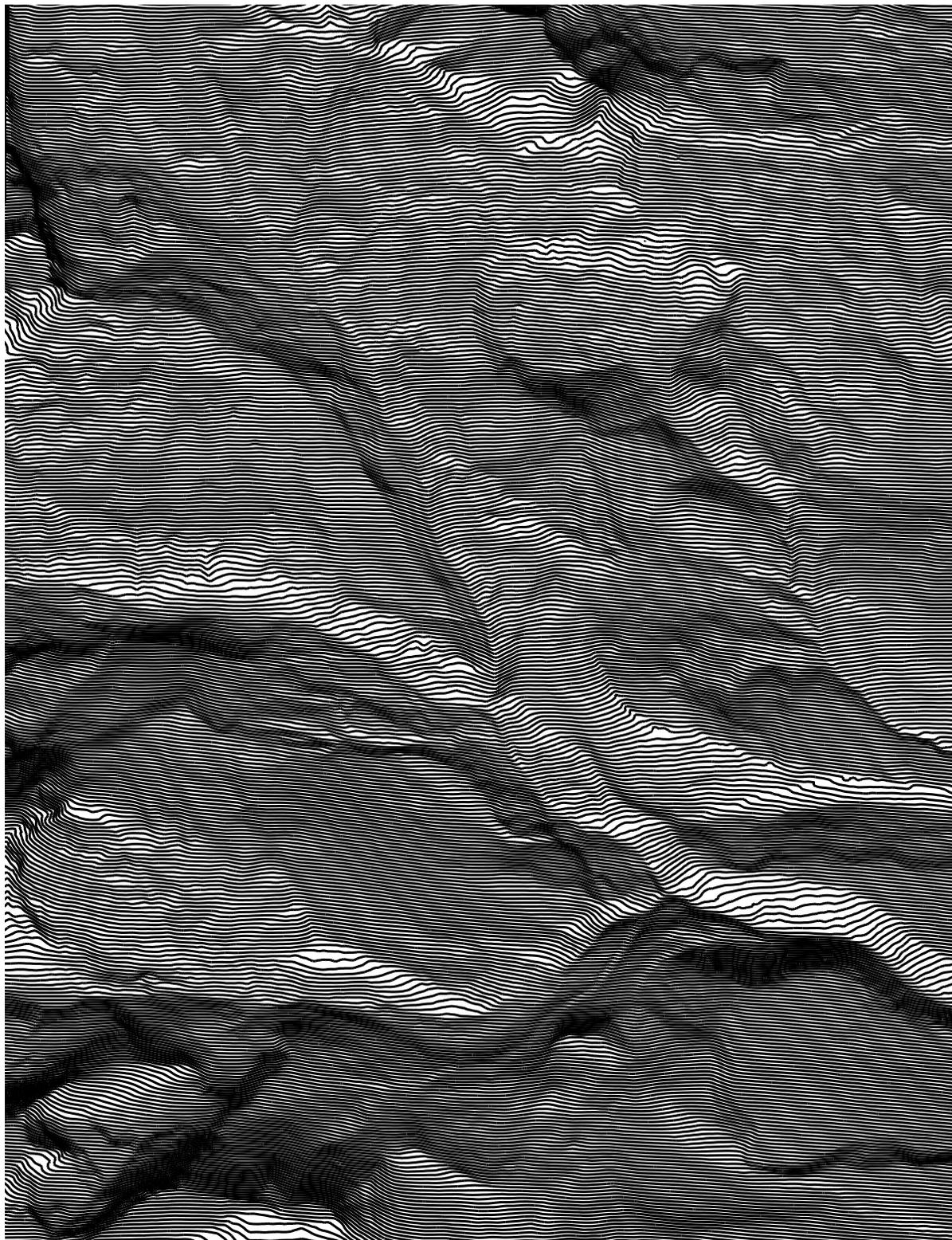


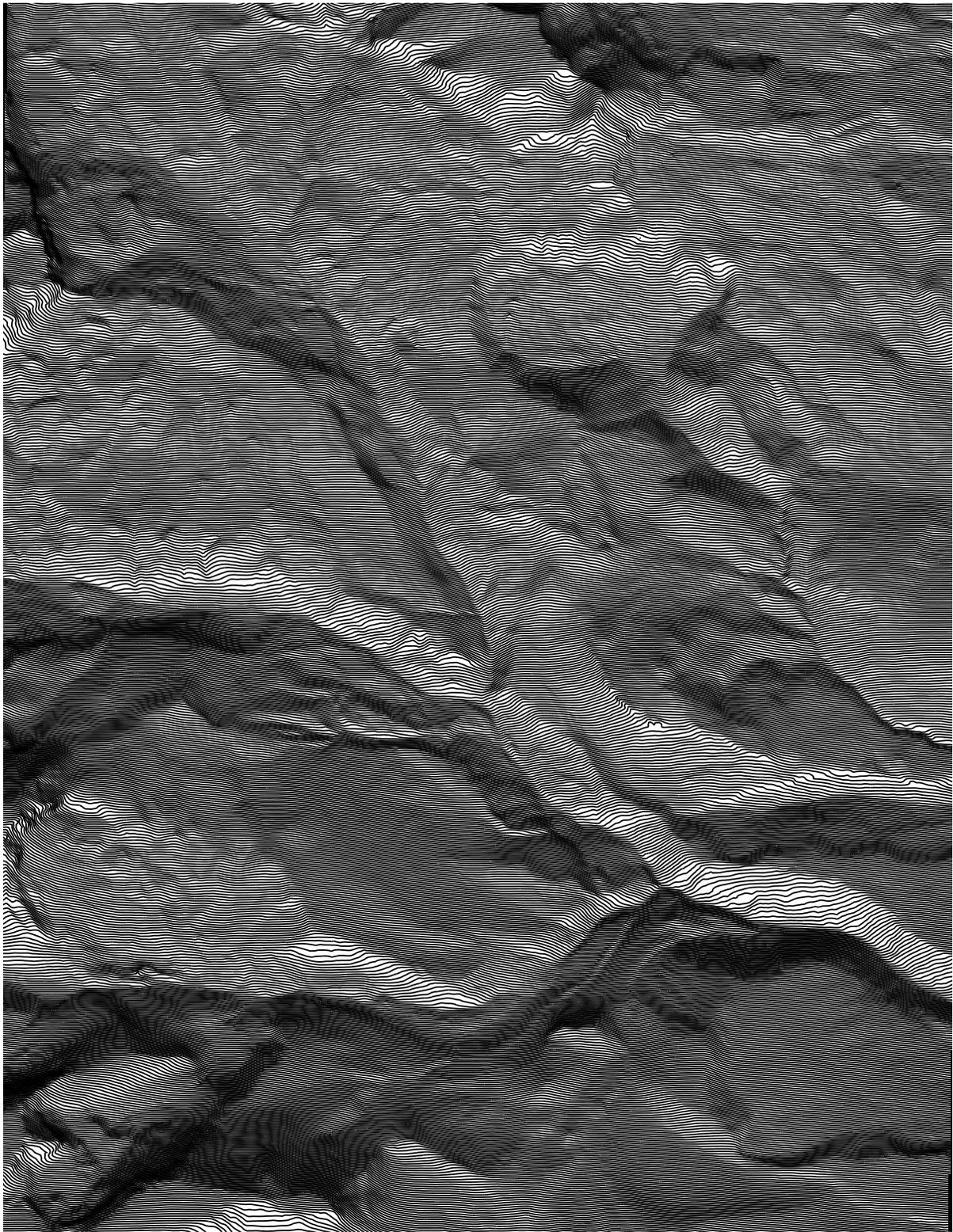
(d)

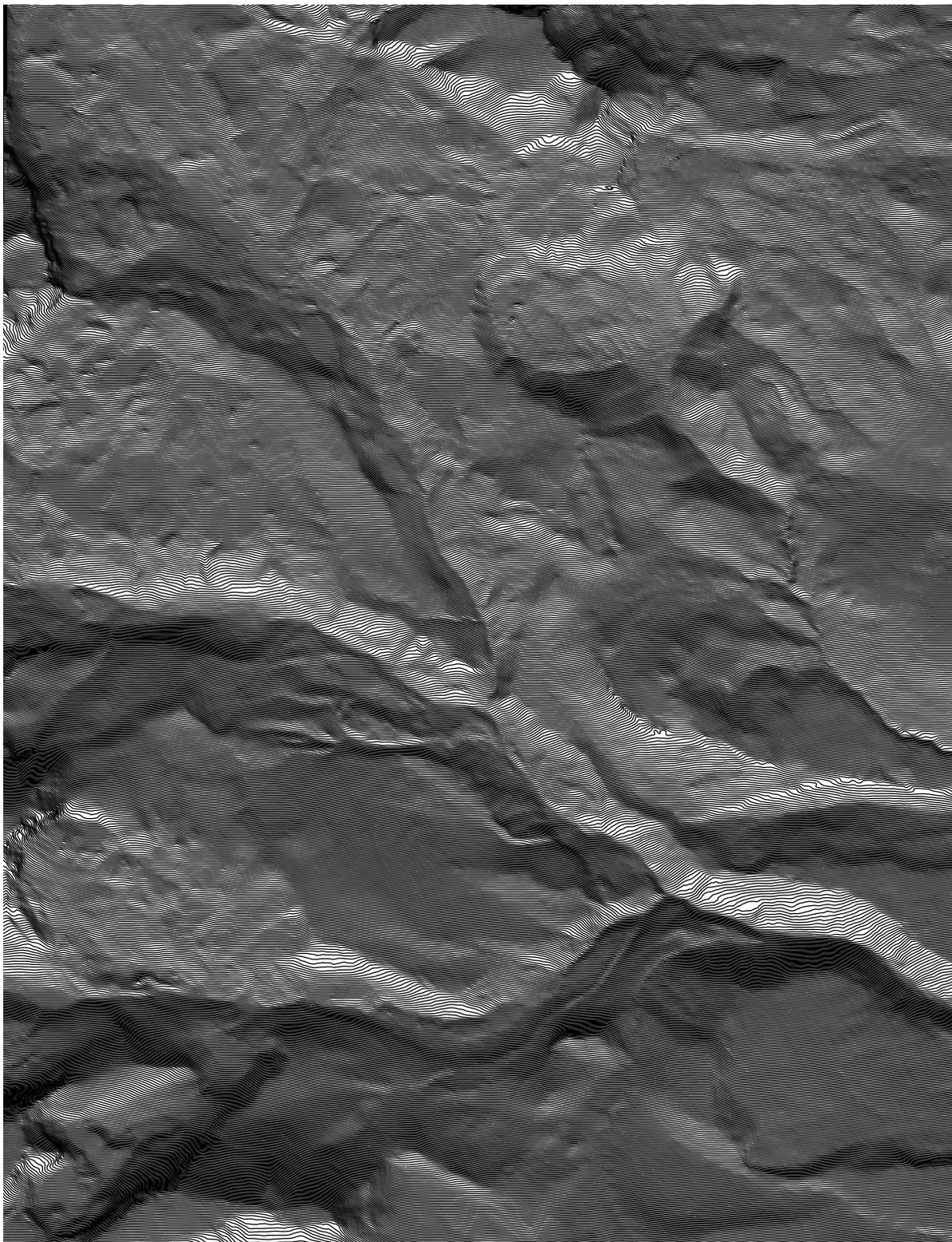


## D Beispiele für orthogonale Reliefs

Auf den folgenden Seiten sind drei orthogonale Reliefs abgebildet, bei denen mit steigender Linienanzahl pro Zentimeter die Linienbreite verringert wird.







# E Symbolverzeichnis

<u>Mathematisches Objekt</u>	<u>Symbol</u>
Skalar	$d$
Punkt	$\mathbf{C}, \mathbf{P}_i$
Vektor	$\mathbf{n}, \mathbf{x}_i$
Menge	$S$
Matrix	$[g_{ij}]$
Matrizelement	$g_{ij}$
Skalarprodukt	$\mathbf{x} \cdot \mathbf{y}$
Kreuzprodukt	$\mathbf{x} \times \mathbf{y}$
partielle Ableitung	$f_x, S_u$
diskrete Faltung	$f \star g$
Boolesche Operatoren	XOR $\oplus$ , UND $\wedge$ , NICHT $\neg$

# Index

- Abstandsraaster, 26
- Blockbild, 3
- buffer
  - edge-, 7
  - enhanced frame, 15
  - F-, *siehe* F-Puffer
  - G-, 14
  - normal, 6, 15, 23
- controlled-density-hatching, *siehe* Schraffierung
- Dreieckmodell, 19
- F-Puffer, 24
- Falllinie, *siehe* Linie
- Formkante, 4
  - als Kante im normal buffer, 6
  - mittels virtual lights, 10
- G-Buffer, *siehe* buffer
- geneigter Umriß (inclined contour), 38
- Halbtönung
  - mit Linien, 38
  - zur Schraffierung, 13
- Halbton
  - als zusammengesetztes Kartenzeichen, 1
- inclined contour, *siehe* geneigter Umriß
- Kantenlinie, *siehe* Linie
- Kartenzeichen, 1
- Kontur, 4
  - linie, 31
  - Relief-, 32
- Lambertscher Reflektierer, 32
- Linie
  - als elementares Kartenzeichen, 1
  - als Gestaltungsmittel, 4
  - Fall-, 3
  - Kanten-, 3
- scheinbare vertikale Überhöhung, 42
- Schraffierung
  - controlled-density-hatching, 17
  - durch Ebenenschnitte, 18
  - durch G-Buffer, 14
  - durch Halbtönung, 13
  - Echtzeit-, 16
  - im Parameterraum, 15, 17
- Schraffur
  - als Gestaltungsmittel, 13
  - als zusammengesetztes Kartenzeichen, 1
- Schraffurlinie
  - Integration, 25
  - Platzierung, 25
- Schraffurmaske, 28
- Sichtbarkeitsproblem
  - im Bildraum, 7
  - im Objektraum, 11
- Silhouette
  - als Kante im Tiefenpuffer, 6, 19
  - einer Polygonfläche, 4
  - einer regulären Fläche, 4
  - innere, 4
  - mittels Dualitätsabbildung, 9
  - mittels Gauß-Abbildung, 8
  - mittels virtual lights, 10
  - subpolygonale, 10
- Strichtextur, 14
- stroke-texture, *siehe* Strichtextur
- Textur
  - als Dithermatrix, 14
  - als tonal-art-map, 17
  - für Schraffurmaske, 27
  - Strich-, 14

Tiefenhinweis, 1  
tonal-art-map, 17

virtual lights, 10

# Literatur

- [1] AKENINE-MÖLLER, T. und E. HAINES: *Real-Time Rendering*. A. K. Peters, 2. Auflage, 2003.
- [2] APPEL, A.: *The Notion of Quantitative Invisibility and the Machine Rendering of Solids*. In: *Proceedings ACM National Conference*, Seiten 387–393. Thompson, 1967.
- [3] APPEL, A., F. ROHLF und A. STEIN: *The Haloed Line Effect for Hidden Line Elimination*. In: *Proceedings of SIGGRAPH '79*, Computer Graphics Proceedings, Seiten 151–157. ACM SIGGRAPH, 1979.
- [4] BOLLMANN, J. und W. KOCH (Herausgeber): *Lexikon der Kartographie und Geomatik*. Spektrum, 2001.
- [5] BRONSTEIN, I., K. SEMENDJAJEW, G. MUSIOL und H. MÜHLIG: *Taschenbuch der Mathematik*. Harri Deutsch, 1997.
- [6] BUCHANAN, J. und M. SOUSA: *The edge buffer: A data structure for easy silhouette rendering*. In: *Non-Photorealistic Animation and Rendering 2000 (NPAR '00)*, Seiten 39–42, Annecy, Juni 2000. ACM Siggraph.
- [7] BURROUGH, A.: *Principles of Geographical Information Systems for Land Resources Assessment*. Oxford Science, 1986.
- [8] BUSS, S.: *3-D Computer Graphics*. Cambridge UP, 2003.
- [9] CARD, D und J. MITCHELL: *Non-Photorealistic Rendering with Pixel and Vertex Shaders*. In: ENGEL, W. (Herausgeber): *Vertex and Pixel Shader Tips and Tricks*. Wordware, April 2002.
- [10] CHAIKIN, G.: *An algorithm for high speed curve generation*. *Computer Graphics and Image Processing*, 3:346–349, 1974.
- [11] COCKSHOTT, T.: *Wet and Sticky: A Novel Model for Computer-Based Painting*. Doktorarbeit, University of Glasgow, 1991.
- [12] DECAUDIN, P.: *Cartoon-Looking Rendering of 3D-Scenes*. Technischer Bericht INRIA 2919, Université de Technologie de Compiègne, Juni 1996.
- [13] DEUSSEN, O.: *Nichtrealistische Computergraphik*. *Informatik Spektrum*, 24:71–80, April 2001.

- [14] DEUSSEN, O., J. HAMEL, A. RAAB, S. SCHLECHTWEG und T. STROTHOTTE: *An Illustration Technique Using Intersections and Skeletons*. In: *Proceedings of Graphics Interface '99*, Seiten 175–182, Juni 1999.
- [15] DOCARMO, M.: *Differentialgeometrie von Kurven und Flächen*. Vieweg, 1983.
- [16] EBERLY, D.: *3D Game Engine Design*. Morgan Kaufmann, 2001.
- [17] ELBER, G: *Line Art Illustrations of Parametric and Implicit Forms*. IEEE Transactions on Visualization and Computer Graphics, 4(1):71–81, Januar 1998.
- [18] FELLNER, W.: *Computergrafik*. BI-Wissenschaftsverlag, 1992.
- [19] FLOYD, R. und L. STEINBERG: *An Adaptive Algorithm for Spatial Gray Scale*. Society for Information Display Digest, 17:75–77, 1976.
- [20] FOLEY, J., A. VAN DAM, S. FEINER und J. HUGHES: *Computer Graphics*. Addison-Wesley, 2. Auflage, 1996.
- [21] FREBOLD, G.: *Profil und Blockbild*. Westermann, 1951.
- [22] GEUZAIN, C.: *GL2PS: an OpenGL to PostScript printing library*. <http://www.geuz.org/gl2ps>, 2003.
- [23] GOOCH, B., P. SLOAN, A. GOOCH, S. SHIRLEY und R. RIESENFELD: *Interactive Technical Illustration*. Proceedings of the Conference on the 1999 Symposium on Interactive 3D Graphics, Seiten 31–38, April 1999.
- [24] HAEBERLI, P.: *Paint By Numbers: Abstract Image Representation*. In: *SIGGRAPH 90 Conference Proceedings*, Seiten 207–214, 1990.
- [25] HAKE, G., D. GRÜNREICH und L. MENG: *Kartographie*. Walter de Gruyter, 8. Auflage, 2002. Bak 0 HAKE.
- [26] HERTZMANN, A.: *Introduction to 3D Non-Photorealistic Rendering: Silhouettes and Outlines*. In: GREEN, S. (Herausgeber): *Non-Photorealistic Rendering (Siggraph '99 Course Notes)*. ACM Siggraph, 1999.
- [27] HERTZMANN, A. und D. ZORIN: *Illustrating Smooth Surfaces*. Proceedings of SIGGRAPH 2000, Seiten 517–526, 2000.
- [28] HÖLZEL, F.: *Perspektivische Karten*. In: *Internationales Jahrbuch für Kartographie*. 1963.
- [29] IMHOF, E.: *Kartenverwandte Darstellungen der Erdoberfläche*. In: *Internationales Jahrbuch für Kartographie*. 1963.

- [30] IMHOF, E.: *Kartographische Geländedarstellung*. Walter de Gruyter, 1968.
- [31] INTERRANTE, V., H. FUCHS und S. PIZER: *Enhancing Transparent Skin Surfaces with Ridge and Valley Lines*. IEEE Visualization, Seiten 52–59, 1995.
- [32] ISENBERG, T., B. FREUDENBERG, N. HALPER, S. SCHLECHTWEG und T. STROTHOTTE: *A Developer's Guide to Silhouette Algorithms for Polygonal Models*. IEEE Computer Graphics and Applications, Seiten 28–37, 2003.
- [33] JOBARD, B. und W. LEFER: *Creating Evenly-Spaced Streamlines of Arbitrary Density*. In: *Proceedings of 8th Eurographics Workshop on Visualization in Scientific Computing*, Seiten 45–55, 1997.
- [34] LAKE, A., C. MARSHALL, M. HARRIS und M. BLACKSTEIN: *Stylized Rendering Techniques for Scalable Real-Time 3D Animation*. In: *Proceedings of NPAR 2000*, Seiten 13–20. ACM Press, Juni 2000.
- [35] LANSDOWN, J. und S. SCHOFIELD: *Expressive Rendering: A Review of Nonphotorealistic Techniques*. IEEE Computer Graphics and Applications, 15(3):29–37, Mai 1995.
- [36] MAC EACHREN, A.: *How maps work*. Guilford, 1995.
- [37] MANOILOV, P., G. MANOILOV und B. DELIJSKA: *Elsevier's Dictionary of Computer Graphics*. Elsevier, 2000.
- [38] MARKOSIAN, L., M. KOWALSKI, S. TRYCHIN, L. BOURDEV, D. GOLDSTEIN und J. HUGHES: *Real-Time Nonphotorealistic Rendering*. Proceedings of SIGGRAPH 97, Seiten 415–420, August 1997.
- [39] MARTÍN, D. und J. C. TORRES: *Rendering Silhouettes with Virtual Lights*. Computer Graphics Forum, 20(4):271–282, 2001.
- [40] MITCHELL, J., C. BRENNAN und D. CARD: *Real-Time Image-Space Outlining for Non-Photorealistic Rendering*. Siggraph 2002 Sketch, San Antonio, Juli 2002.
- [41] MONKHOUSE, F. und H. WILKINSON: *Maps and Diagrams*. Camelot Press, 1963.
- [42] POPINET, S.: *GNU Triangulated Surface Library*. <http://gts.sourceforge.net>, 2003.
- [43] POP, M., G. BAREQUET, C. DUNCAN, M. GOODRICH und S. KUMAR: *Efficient Perspective-Accurate Silhouette Computation*. In: *Proceedings 7th Pacific Graphics Conference*, Seiten 60–68. IEEE CS, 1999.
- [44] PRAUN, E., H. HOPPE, M. WEBB und A. FINKELSTEIN: *Real-Time Hatching*. Proceedings of SIGGRAPH 2001, Seiten 579–584, August 2001.

- [45] PREPARATA, S. und M. SHAMOS: *Computational Geometry*. Springer, 1985.
- [46] RAISZ, E.: *Principles of Cartography*. McGraw-Hill, 1962.
- [47] RASKAR, R. und M. COHEN: *Image Precision Silhouette Edges*. In: *Proc. 1999 ACM Symposium on Interactive 3D Graphics*, Seiten 135–140, 1999.
- [48] RÖSSL, C. und L. KOBBELT: *Line-Art Rendering of 3D-Models*. Proceedings of Pacific Graphics 2000, 2000.
- [49] RÖSSL, C., L. KOBBELT und H.-P. SEIDEL: *Line Art Rendering of Triangulated Surfaces Using Discrete Lines of Curvature*. In: *Proceedings of WSCG 2000*, Seiten 87–96, Oktober 2000.
- [50] SAITO, T. und T. TAKAHASHI: *Comprehensible Rendering of 3-D Shapes*. Computer Graphics (Siggraph '90 Proceedings), 24(4):197–206, August 1990.
- [51] SALISBURY, M., S. ANDERSON, R. BARZEL und D. SALESIN: *Interactive Pen-And-Ink Illustration*. Proceedings of SIGGRAPH 94, Seiten 101–108, July 1994. ISBN 0-89791-667-0. Held in Orlando, Florida.
- [52] SALISBURY, M., M. WONG, J. HUGHES und D. SALESIN: *Orientable Textures for Image-Based Pen-and-Ink Illustration*. Proceedings of SIGGRAPH 97, Seiten 401–406, August 1997.
- [53] SANDER, P., X. GU, S. GORTLER, H. HOPPE und J. SNYDER: *Silhouette Clipping*. Proceedings of SIGGRAPH 2000, Juli 2000.
- [54] SONKA, M., V. HLAVAC und R. BOYLE: *Image Processing, Analysis, and Machine Vision*. Brooks/Cole, 2. Auflage, 1999.
- [55] STRASSMANN, S.: *Hairy Brushes*. Siggraph, 20(4):225–232, 1986.
- [56] STROTHOTTE, T. und S. SCHLECHTWEG: *Non-Photorealistic Computer Graphics*. Morgan Kaufmann, 2002.
- [57] SUTHERLAND, I., R. SPROULL und R. SCHUMACKER: *A Characterization of Ten Hidden-Surface Algorithms*. Computing Surveys, 6(1):1–55, 1974.
- [58] TANAKA, K.: *The Orthographical Relief Method of Representing Hill Features on a Topographical Map*. Geographical Journal, 79(1):213–219, 1932.
- [59] TANAKA, K.: *The Relief Contour Method of Representing Topography on Maps*. Geographical Review, 40:444–456, 1950.

- [60] THROWER, N.: *Extended Uses of the Method of Orthogonal Mapping of Traces of Parallel, Inclined Planes with a Surface, especially Terrain*. In: *Internationales Jahrbuch für Kartographie*. 1963.
- [61] U.S. GEOLOGICAL SURVEY: *Standards for Digital Elevation Models*, 1998.
- [62] VERYOVKA, O. und J. BUCHANAN: *Comprehensive Halftoning of 3D Scenes*. In: *Proceedings of Eurographics '99*, Seiten 13–22, 1999.
- [63] VERYOVKA, O. und J. BUCHANAN: *Halftoning With Image-Based Dither Screens*. *Graphics Interface '99*, Seiten 167–174, Juni 1999.
- [64] WINKENBACH, G. und D. SALESIN: *Rendering Parametric Surfaces in Pen and Ink*. In: RUSHMEIER, HOLLY (Herausgeber): *SIGGRAPH 96 Conference Proceedings*, Annual Conference Series, Seiten 469–476. ACM SIGGRAPH, Addison Wesley, August 1996.