

Otto-von-Guericke-Universität Magdeburg
Fakultät für Informatik



Diplomarbeit

**Generierung
charakteristischer Linienzüge
aus 3D-Modellen**

Bert Schönwälder

Institut für Simulation und Graphik

1997

Otto-von-Guericke-Universität Magdeburg
Fakultät für Informatik
Institut für Simulation und Graphik



Diplomarbeit

Generierung charakteristischer Linienzüge aus 3D-Modellen

Verfasser: Bert Schönwälder
geboren 14. Februar 1972 in Zittau
Matrikel-Nummer: 133178

Prüfer: Prof. Dr. Thomas Strothotte

Betreuer: Prof. Dr. Thomas Strothotte
Dipl.-Inf. Stefan Schlechtweg

Beginn: 1. November 1997
Abgabe: 1. Juni 1997

Zusammenfassung

In wissenschaftlichen oder technischen Illustrationen werden häufig Liniengraphiken eingesetzt. Heutigen computergenerierten Liniengraphiken fehlen jedoch viele der Eigenschaften, die mit traditionellen Zeichenmethoden erstellte Graphiken interessant und verständlich machen. Die vorliegende Arbeit untersucht Techniken, welche einige dieser Methoden für computergenerierte Graphiken adaptieren.

Aus einer Analyse traditioneller Methoden heraus wird das Konzept der „charakteristischen Linien“ entwickelt, das die automatische Erzeugung von Konturzeichnungen aus dreidimensionalen Geometriemodellen erlaubt. Die zur Darstellung verwendeten Linien entstehen durch die Auswahl von Kanten eines polygonalen Modells und deren Zusammenfassung zu Kurven anhand von Glättungsinformationen.

Die Charakteristik der Linien kann modellabhängig oder vom Benutzer variiert werden. Das beinhaltet eine beleuchtungsabhängige Variation der Linienstärke, die Hervorhebung von Verdeckungen durch Verkürzung von Linien und die Beeinflussung weiterer vom Zeichenwerkzeug bereitgestellter Attribute.

Vorwort

An dieser Stelle möchte ich mich bei all denen bedanken, die direkten oder indirekten Einfluß auf diese Arbeit hatten.

Zuerst danke ich meiner Mutter, die mir durch ihre liebevolle Unterstützung das Studium überhaupt ermöglicht hat.

Ich bedanke mich bei meinen Betreuern Prof. Strothotte und Stefan Schlechtweg für ihr außerordentliches Verständnis in einer persönlich schwierigen Situation und für ihre kritischen Hinweise sowie wertvollen Anregungen zur Bearbeitung des Themas.

Weiterhin danke ich allen Mitarbeitern des ISG, die auf vielerlei Art Anteil an dieser Arbeit haben – besonders aber Andreas, Axel und Ralf für ihre konstruktiven Ratschläge, Maic für die Unterstützung bei der Modellierung der Beispielbilder und Michael, der seine Begeisterung für Smalltalk auf mich übertrug und immer ein offenes Ohr für meine Fragen hatte.

Besonderer Dank gilt Lars Schumann, der mir in vielen Diskussionen neue Impulse für meine Arbeit gab und mich bis zuletzt tatkräftig unterstützt hat.

Rita hat während des ganzen Diploms versucht, alle familiären Verpflichtungen von mir fern zu halten, mein Dank dafür sei ihr gewiß.

Ich versichere, daß ich die vorliegende Arbeit allein und nur unter Zuhilfenahme der angegebenen Quellen angefertigt habe.

Magdeburg, 1. Juni 1997

Bert Schönwälder

Inhaltsverzeichnis

1	Einführung	3
2	Traditionelle Liniengraphik	6
2.1	Lehrbeispiel I: Wissenschaftliche Illustration	8
2.2	Lehrbeispiel II: Technische Illustration	9
2.3	Weitere Beispiele	12
2.4	Zusammenfassung	13
3	Computergenerierte Liniengraphik	15
3.1	Pixelorientierte Verfahren	15
3.2	Analytische Verfahren	18
3.3	Auswertung	22
4	Charakteristische Linien	24
4.1	Generierung von Kurven	25
4.1.1	Vorbetrachtungen	25
4.1.2	Kantenselektion	27
4.1.3	Kurvenextraktion	28
4.2	Variation der Linienattribute	31
4.2.1	Lichteinfluß	32
4.2.2	Silhouettenbetonung	33
4.2.3	Wichtigkeit	34
4.2.4	Depthcueing	34
4.2.5	Verallgemeinerung	35
4.3	Variation der Endbedingungen einer Linie	35

4.3.1	Lücken	36
4.3.2	Fülldreiecke	37
5	Implementierung	39
5.1	Überblick	39
5.2	Das Modell	40
5.3	Transformation des Modells und Eckenklassifikation	41
5.4	Bestimmung sichtbarer Kanten und Flächen	41
5.5	Kurvengenerierung	43
5.6	Attributierung	45
5.7	Darstellung	45
5.8	PostScript-Ausgabe	46
5.9	Performance	47
6	Ergebnisse in Bildern	48
7	Diskussion	54
7.1	Kritik	54
7.2	Weiterführende Überlegungen	57
7.3	Fazit	58
	Literaturverzeichnis	59

1 Einführung

Seit ihren Anfängen in den 60er Jahren dieses Jahrhunderts setzt die Computergraphik *Lini*-*en* ein. Schon sehr früh wurden im Rechner dreidimensional repräsentierte Objekte linienhaft dargestellt. Diese wurden auf Vektor-Bildschirmen oder auch mit Plottern ausgegeben. Mit der Entwicklung des Raster-Bildschirms wurde es möglich, nicht nur Linien, sondern auch gefüllte Flächen zur Darstellung dreidimensionaler Objekte zu verwenden. Die daraufhin einsetzende rasante Entwicklung dieser „flächenhaften“ Darstellungen, die heute schon kaum noch von Photographien zu unterscheidende Bilder liefert, drängte andere Darstellungsformen an den Rand.

Erst seit Anfang der 90er Jahre beschäftigt sich die Forschung intensiver mit solchen „nicht-photorealistisch“ genannten Graphiken. Für eine photorealistisch gerenderte Graphik ist die Exaktheit der physikalischen Simulation des Bildentstehungsprozesses in einer photographischen Kamera das oberste Ziel. Dem gegenüber wird im Nicht-Photorealismus Wert darauf gelegt, eine dem beabsichtigten Einsatz des Bildes angemessene Darstellung zu wählen. Dabei wird auf unterschiedliche Stile in der Darstellung zurückgegriffen. Häufig orientieren sich diese Stile an traditionellen Mal- und Zeichenmethoden.

Ein wichtiges Teilgebiet des Nicht-Photorealismus ist die Erzeugung von Liniengraphiken. Das grundlegende Gestaltungsmittel dieser Graphiken ist das Wechselspiel von Linie¹ und freier Fläche. Liniengraphiken erlauben weitreichende Charakterisierungen von dargestellten Objekten. Die Bandbreite reicht von flüchtigen, ungenauen, entwurfsartigen Skizzen bis hin zu exakten technischen Zeichnungen.

In dieser Arbeit wird ein Verfahren entwickelt, mit dem sogenannte „Konturzeichnungen“ computergestützt erstellt werden können, mit dem Ziel, in illustrativen Darstellungen verwendet zu werden. Konturzeichnungen werden häufig zum Illustrieren verwendet, da ihre schematische Erscheinung der gezielten Vermittlung von Informationen entgegenkommt. Diese Graphiken setzen Linien sehr sparsam ein, es werden nur die Objektkonturen und einige innere Linien dargestellt. Insbesondere enthalten sie keine Schraffuren. Der Gestaltung jeder einzelnen Linie kommt somit eine besondere Bedeutung zu. Dadurch wird der Charakter des dargestellten Subjektes hervorgehoben.

Den Ausgangspunkt für die Erzeugung einer Liniengraphik im Rahmen dieser Arbeit bildet ein dreidimensionales Geometriemodell. Es werden polygonale Modelle verwendet, da diese weithin verfügbar und effizient handhabbar sind. Die eigentliche Modellierung kann mit einem externen Werkzeug vorgenommen werden, sie ist nicht Bestandteil dieser Arbeit. Modelle sind meist strukturiert. Sie bestehen aus einzelnen Teilmodellen, den Objekten.

¹ Liniendefinition nach [Sch95, Sch97]

Für die liniengraphische Darstellung wird das Konzept der *charakteristischen Linien* entwickelt. Charakteristisch sollen dabei Linien genannt werden, die

- ein Objekt selbst in seiner räumlichen Struktur erkennbar machen und
- das Objekt in Relation zu anderen Objekten setzen.

Für die Erkennbarkeit der Form ist die *Auswahl* und *Plazierung* von Linien maßgebend. Unterstützt wird sie zum Beispiel durch eine Anpassung der Linienbreite in Abhängigkeit von der Schattierung des Objektes. Die Wahl der *Linienausprägung* – also beispielsweise breit/schmal, ungenau/exakt oder durchgezogen/gestrichelt – trägt zur Charakterisierung des Objektes bei. Werden für unterschiedliche Objekte verschiedene Ausprägungen verwendet, setzt sie das in eine Relation. Dieses Verhältnis kann räumlicher (z. B. Vorder-/Hintergrundobjekte), aber auch gestalterischer Natur sein (z. B. Fokussierung auf spezielle Objekte).

Die Ausgabe der Graphik erfordert ein Linienwerkzeug, das die Darstellung von Linienzügen gestattet, die in ihrer Gestaltung und Genauigkeit variabel sind. Solch ein Werkzeug wurde in der Diplomarbeit von LARS SCHUMANN entwickelt [Sch97]. Es erwartet die Übergabe eines mit Attributen wie Druck oder Sättigung versehenen Zeichenpfades in Form einer Liste von Kurvenstützpunkten. Diese werden mit einem wählbaren Linienstil gezeichnet. Über den Stil können verschiedene Darstellungsarten realisiert werden, wie das Stricheln einer Linie oder stochastische Abweichungen vom exakten Verlauf.

Die Ausgabe erfolgt vektororientiert und ist daher unabhängig von der physischen Auflösung des Ausgabegerätes. Ein hochqualitativer Druck mit einer Photosatzmaschine ist nicht aufwendiger zu erzeugen als eine relativ niedrig aufgelöste Bildschirmdarstellung. Die gebotene Qualität des jeweiligen Ausgabegerätes kann daher ohne erhöhten Berechnungsaufwand genutzt werden.

Für die Gestaltungsmöglichkeiten in Liniengraphiken liefern traditionelle Zeichentechniken viele Anregungen. Diese werden in Kapitel 2 näher untersucht. Es wird je ein Lehrbeispiel aus den Bereichen wissenschaftliche und technische Illustration gegeben und um andere, aus unterschiedlichen Quellen stammende Beispiele ergänzt.

Verwandte Arbeiten zum Thema computergenerierter Liniengraphiken werden in Kapitel 3 vorgestellt. Dabei finden speziell diejenigen Verfahren Beachtung, die von dreidimensionalen geometrischen Modellen ausgehen. Die Arbeiten werden unter dem Aspekt der Umsetzung (oder auch *möglichen* Umsetzung) der besprochenen Zeichentechniken betrachtet.

Ausgehend von den in Kapitel 2 gefundenen Darstellungsvariationen und den untersuchten bisherigen Arbeiten wird in Kapitel 4 das Konzept der charakteristischen Linien entwickelt. Es wird ein neues Verfahren zur Kurvengenerierung entworfen und Möglichkeiten der visuellen Attributierung der Linien analysiert.

Ausgewählte Implementierungsdetails werden in Kapitel 5 erläutert. Auch auf Einzelheiten der eingesetzten Werkzeuge und im Vorfeld entstandenen Komponenten wird hier eingegangen.

In Kapitel 6 wird anhand von Bildbeispielen die Leistungsfähigkeit des implementierten Verfahrens demonstriert.

Abschließende Gedanken werden in Kapitel 7 präsentiert. Das Verfahren wird einer Kritik unterzogen, anschließend Möglichkeiten zur Verbesserung und Weiterentwicklung aufgezeigt.

2 Traditionelle Liniengraphik

Liniengraphiken haben eine lange Tradition. Schon in den ersten Höhlenzeichnungen finden sich Darstellungen, die u. a. die Konturen von Beutetieren wiedergeben. Die primäre Verwendung von Linien (z. B. in frühen Kinderzeichnungen) ist bis heute nicht vollständig erklärt. Sie ist um so erstaunlicher, da beim Sehen eigentlich Flächen wahrgenommen werden: „Die Wirklichkeit, soweit sie sichtbar ist, kennt keine Linien, sie zeigt dem Auge nur aneinander grenzende Flächen und Flecken.“ [Wae12].

Dennoch wird durch das Umreißen mit einer *Konturlinie* der dargestellte Gegenstand jedermann begreifbar. Dem liegt eine gestaltpsychologische Tatsache zugrunde, die von ARNHEIM so formuliert wird: „Eine [. . .] Regel besagt, daß die geschlossene Fläche Figur werden will, während die umschließende als Fläche gesehen wird. Dies schließt eine weitere Regel ein, nach der unter bestimmten Bedingungen immer die kleinere Fläche zur Figur wird.“ [Arn74]. Auf dieser Figur-Grund-Beziehung beruht die Fähigkeit von Liniengraphiken, räumliche Körper abzubilden.

Liniengraphiken werden heutzutage in vielfältiger Weise eingesetzt. Ein spezielles Anwendungsgebiet sind Illustrationen (nach [LI94] die „bildliche Darstellung und Ausdeutung vorgegebener Textinhalte“). In dieser Arbeit sollen computergestützt Liniengraphiken erzeugt werden, die sich an wissenschaftliche und technische Illustrationen anlehnen.

„Wissenschaftliches Illustrieren bedeutet das Erstellen von präzisen Zeichnungen und anderen graphischen Darstellungen, die dem wissenschaftlichen Autor ein Hilfsmittel zur Kommunikation sind. [. . .] In diesem Kommunikationsprozeß sollte der Illustrator eine exakte Arbeit liefern, die trotzdem ausgewogen erscheint und in der gestalterischen Bearbeitung des Subjektes ansprechend ist.“¹

Bei wissenschaftlichen Illustrationen geht es also darum, ein Objekt so klar und ansprechend wie möglich zu präsentieren. Dafür ist es notwendig, von überflüssigen Details zu abstrahieren und sich auf die Darstellung der wesentlichen Eigenschaften zu beschränken. Das Zeichnen mit der Feder ist dabei „die wichtigste zu beherrschende Technik eines wissenschaftlichen Illustrators“².

¹ “Scientific illustration is the production of drawings of measured accuracy and other graphic images that help the scientist-author to communicate. [. . .] In this communication process the illustrator should produce accurate work that is also pleasing to the eye in terms of balance and artistic handling of the subject.” [Hod89, S. xi]

² “The most common type of line illustrations is done with pen-and-ink, the use of which is the most important technique for the scientific illustrator to master.” [Hod89, S. 89]

Ähnliche Kriterien für eine gelungene Präsentation gelten für technische Illustrationen. Technische Objekte sind meist einfacher und klarer strukturiert, oftmals liegen CAD-Modelle vor. In den Zeichnungen kommen – häufiger als in wissenschaftlichen Illustrationen – zusätzliche abstrakt-graphische Elemente³ vor.

Sowohl bei den wissenschaftlichen als auch bei den technischen Illustrationen stellen Liniengraphiken ein wichtiges Stilmittel dar. Mit ihnen kann besonders klar und deutlich der darzustellende Sachverhalt oder Gegenstand illustriert werden.

Die Ausprägung der Linie, ob als Kontur oder zum Schattieren verwendet, liefert dem Betrachter dabei Informationen über Licht, Schattierung und Struktur, die zusammengenommen den räumlichen Eindruck ergeben. Oft ist zur adäquaten Darstellung eines Objektes eine „Konturzeichnung“ ausreichend [Hod89]. Sie enthalten keine Schraffuren, sondern stellen nur gewisse Objektkanten dar.

Die zur Darstellung der Kanten eingesetzten Linien werden in Linientypen unterschieden (entsprechendes gilt für die Kanten selbst). Dabei ist die Beziehung zum Objekt signifikant (vgl. [Kos93]):

- *Konturlinien*. Sie umreißen ein Objekt und trennen es von anderen bzw. dem Hintergrund. Da an dieser Stelle ein Objekt oder der Hintergrund verdeckt wird, werden sie auch als *Verdeckungslinien* bezeichnet. Gehört der verdeckte Teil zum Objekt selbst, wird dies eine *Selbstverdeckung* genannt. Andernfalls gehört die Linie zur *Silhouette*.
- *Innere Linien*. Sie zeichnen Merkmale der Objektoberfläche nach, wie zum Beispiel die Kanten eines Würfels. Auch *Binnenkonturen* oder *Strukturlinien* sind übliche Bezeichnungen.

Einen dritten Typus bilden die in Konturzeichnungen nicht vorkommenden Schraffurlinien. In Abbildung 2.1 werden die Typen in einem einfachen Beispielobjekt durch unterschiedliche Linienbreiten kenntlich gemacht.



Abbildung 2.1: Liniertypen

Liniengraphiken bieten aber auch technologische Vorteile. Eine *reine* Liniengraphik ist nach [Hod89] eine Zeichnung, die keine Grautöne enthält. Im allgemeinen besteht sie allein aus tiefschwarzen Linien auf weißem Untergrund. Die Reproduktion einer Linienzzeichnung ist oft besser – oder bei gleicher Qualität billiger – als die einer tonalen (Grautöne enthaltenden) Zeichnung, da der druckereitechnische Schritt des Screenings (die Rasterung der Vorlage) eingespart wird.

³ Strich-Punkt-Linien, Pfeile etc. [SS97]

Leider sind verbale Beschreibungen, wie Dinge zu zeichnen sind, selten so detailliert oder allgemeingültig, daß man nach ihrer Lektüre gute Zeichnungen produzieren könnte. Scheinbar wird dieses Wissen hauptsächlich durch genaues Betrachten und Nachahmen professioneller Zeichnungen erworben. Im folgenden wird daher an Beispielen untersucht, welche Techniken in wissenschaftlichen und technischen Illustrationen verwendet werden, um Konturzeichnungen verständlicher oder ansprechender zu gestalten.

Dazu werden zunächst zwei Lehrbeispiele präsentiert, die die verwendeten Techniken demonstrieren. Es folgen einige weitere Beispiele unterschiedlicher Herkunft.

2.1 Lehrbeispiel I: Wissenschaftliche Illustration

Aus einem Handbuch für wissenschaftliche Illustratoren [Hod89] stammt das erste Beispiel (Abbildung 2.2). Es wurde speziell zur Illustration diverser Techniken erstellt, mit denen in einer Liniengraphik die Form eines Objektes charakterisiert werden kann. In den folgenden Erläuterungen beziehen sich die Nummern in Klammern auf die jeweilige Nummer in der Abbildung.

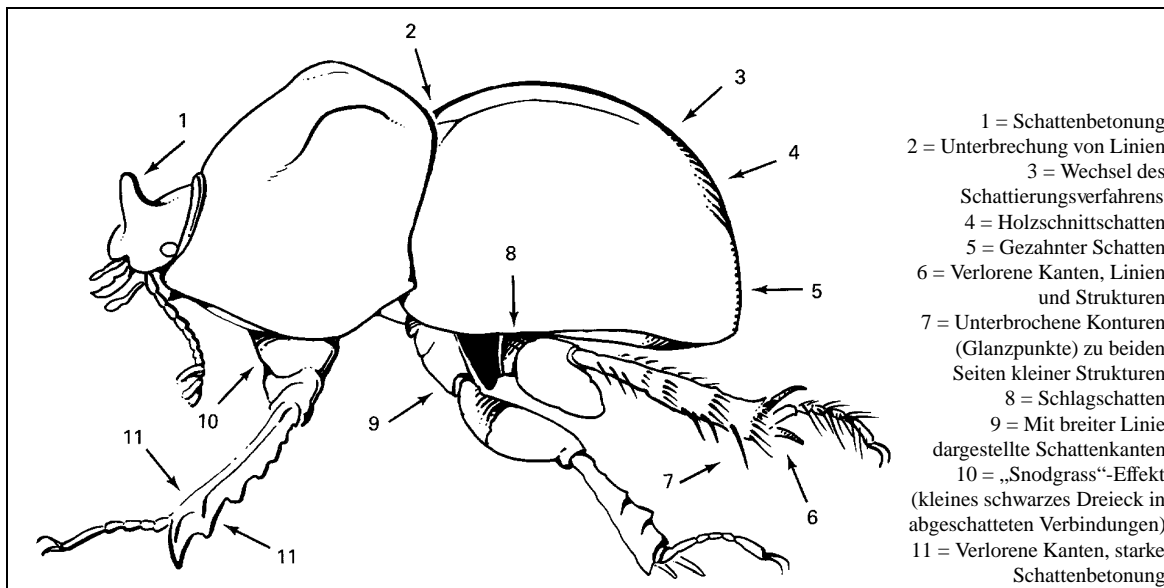


Abbildung 2.2: Linienzzeichnung eines Mistkäfers [Hod89]

Generell ist festzustellen, daß Linien an markanten Stellen plaziert werden. Dazu wird zum einen der Objektrand mit Konturlinien nachgezeichnet. Zum anderen werden innere Linien zur Visualisierung von scharfen Erhöhungen oder Einbuchtungen der Oberfläche eingesetzt, wie z. B. an der Rückenlinie. Sehr feine Objekte werden durch Objekt- statt Konturlinien dargestellt (z. B. die Haare an den Beinen). Dabei wird mit einer Linie keine Fläche eingefasst, sondern die Linie selbst steht für das dargestellte Objekt (vgl. [Arn74, S. 218 ff.]).

Die Akzentuierung der Konturen suggeriert verschiedene Lichtbedingungen. Dabei werden Linien in Schattenregionen stärker gezeichnet, in helleren schwächer (z. B. 1, 11). Außer dieser Linienverbreiterung werden auch gefiederte (4) bzw. gezahnte (5) Linien eingesetzt.

Die Fiederung erzeugt zusätzlich den Eindruck einer Objektrundung. Da die Fiederung bzw. Zahnung im Gegensatz zur Verbreiterung einer Linie asymmetrisch ist, wird schon an einer einzelnen Linie sichtbar, auf welcher Seite das Objektinnere liegt. Dadurch wird dem Betrachter das Erkennen zu einem Objekt gehörender Linien vereinfacht.

Eine Abschwächung von Linien wird durch die Verminderung ihrer Breite erreicht. Mit diesen dünneren Linien werden beispielsweise weniger wichtige Merkmale gezeichnet (wie die eben erwähnte Rückenlinie). Sie schaffen auch einen sanften Übergang von der Fläche zu Auswölbungen (z. B. das Horn am Kopf oder die Höcker auf dem Rücken). Hauptsächlich aber werden sie an den stärker beleuchteten Stellen verwendet. Das kann bis hin zu unterbrochenen Kanten in Bereichen von Glanzlichtern führen (11).

Unterbrochene Linien werden auch zur visuellen Trennung von hintereinander liegenden Objekten verwendet. Wenn eine Linie unter einer Objektkontur verschwindet, kann sie vorher unterbrochen werden (2). Besonders bei feinen Details ist dieser Freiraum zu beiden Seiten des kleinen Objekts notwendig (7).

Eine andere Möglichkeit der Hervorhebung sich verdeckender Linien ist die Platzierung eines kleinen schwarzen Dreiecks an der Verbindungsstelle (u. a. 10). Dies könnte auch als eine helligkeitsabhängige lokale Verbreiterung der verdeckten Linie angesehen werden.

Über diese die Linienausprägung betreffenden Verfahren hinaus ist auch der allgemeine Aufbau der Graphik interessant. So wurde das Objekt von einer Position aus dargestellt, die es besonders gut erkennbar macht. Das Wissen des Betrachters um die symmetrische Natur des Objekts ausnutzend, wurden die dem Betrachter abgewandten Extremitäten weggelassen, da sie keine neue Information enthalten, sondern eher die Klarheit des Bildes beeinträchtigen würden. Viele Details wurden eliminiert. So wurden beispielsweise die Mundwerkzeuge nur schematisch angedeutet, es fehlen die kleineren Härchen und Grübchen. Eine Konvention, den Lichteinfall betreffend, findet sich auch hier: meist wird eine die Szene von links oben beleuchtende Lichtquelle benutzt.

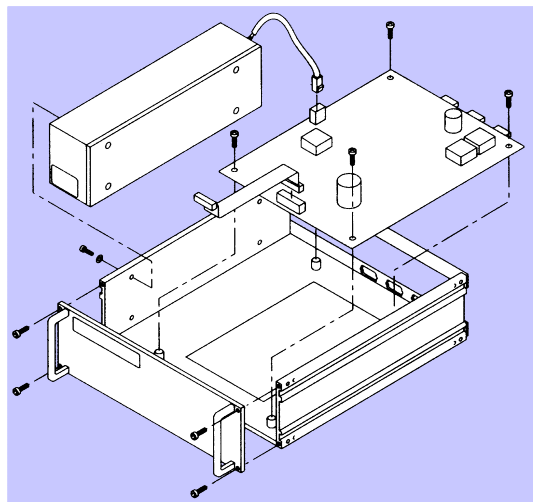
2.2 Lehrbeispiel II: Technische Illustration

Die Verfeinerung einer technischen Illustration verdeutlichen die folgenden Graphiken aus [Zei96]. In diesem Artikel werden zum einen Techniken zur Detailreduktion, zum andern Verfahren der Anreicherung von technischen Liniengraphiken beschrieben.

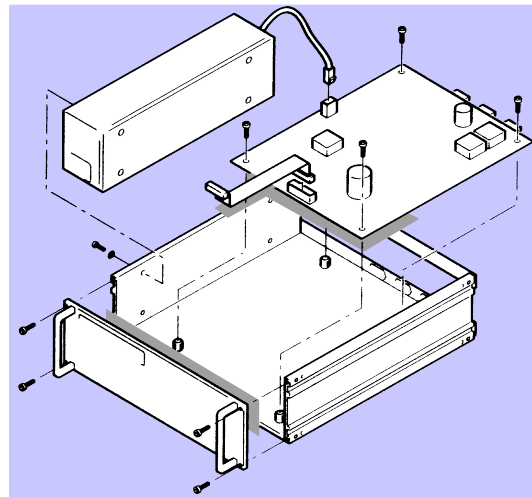
Nicht näher betrachtet wird hier der Prozeß der Detailreduktion, Abbildung 2.3(a) zeigt das Ergebnis dieses Vorganges. Es wurden beispielsweise Lüftungsöffnungen nur angedeutet, die Bauteile sind auf geometrische Grundformen reduziert, es wurden nur markante Bauteile dargestellt usw.⁴

Gezeichnet wurde mit undifferenzierten, gleichstarken Linien. Dadurch wirkt die Graphik „flach“. In Abbildung 2.3(b) wurden die Linien dagegen differenziert eingesetzt. Dies wird im folgenden anhand von Gegenüberstellungen von Details aus beiden Graphiken näher untersucht. Zuerst verdienen aber einige allgemeinere Beobachtungen ihre Erwähnung.

⁴ Mit der automatischen Reduktion von Darstellungsdetails befaßt sich beispielsweise [KR95]



(a) Undifferenzierte Graphik

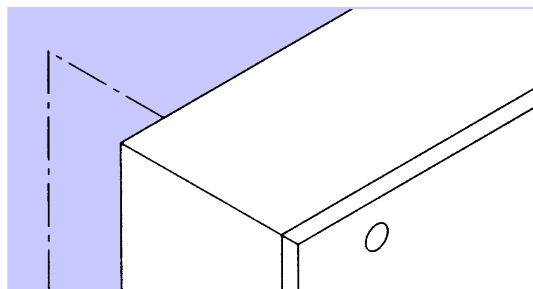


(b) Angereicherte Graphik

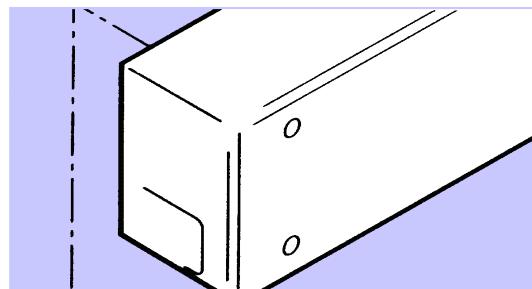
Abbildung 2.3: Illustrationen aus [Zei96]

Das gewählte Beispiel demonstriert generell in technischen Illustrationen eingesetzte Methoden. So wird eine isometrische Projektion verwendet, die – im Gegensatz zu einer Zentral- oder Kavalierverspektive – die Längenverhältnisse erhält. Es wurde eine Explosionsdarstellung genutzt, in der die Einzelteile räumlich voneinander separiert sind. Zusätzlich zu den die physischen Körper darstellenden Linien kommen Strich-Punkt-Linien zum Einsatz, die virtuelle Achsen bilden und hier Schraubverbindungen darstellen.

Bemerkenswert ist die Einfärbung des Hintergrundes in den Abbildungen. Dadurch wird die Erkennbarkeit der Objekte verbessert, da sie sich gut vom Hintergrund abheben. Ohne diese farbliche Hinterlegung ist an einer einzelnen Kante nicht feststellbar, in welcher Beziehung sie zum Objekt steht, d. h. ob sie eine innere Kante ist oder die Silhouette beschreibt. Bei einer Silhouettenkante ist dabei auf einen Blick erkennbar, zu welchem Objekt die Linie gehört.



(a) vorher



(b) nachher

Abbildung 2.4: Unterschiedliche Linienarten [Zei96]

Die erste Differenzierung wird durch eine unterschiedliche Wahl der Strichstärke für Kontur- bzw. innere Linien erreicht (Abbildung 2.4). Durch die Verstärkung der Konturkanten wird sowohl der Zusammenhalt des Objektes in sich, als auch dessen Abgrenzung nach außen hervorgehoben. Die inneren Linien wurden dünner ausgeführt, sie sind durchbrochen. Das verleiht ihnen eine niedrigere visuelle Priorität. Die Durchbrüche der Kanten, die nicht den

Winkel zwischen zwei Flächen darstellen, sind noch größer. Dadurch wird eine visuelle Rangordnung der Linien etabliert.

In Abbildung 2.5 ist eine weitere optische Differenzierung zu erkennen. Wie schon in der wissenschaftlichen Illustration wurden hier Teile durch eine schmale weiße Fläche voneinander getrennt. Auch die zusätzlichen Strich-Punkt-Linien wurden auf diese Weise hervorgehoben.

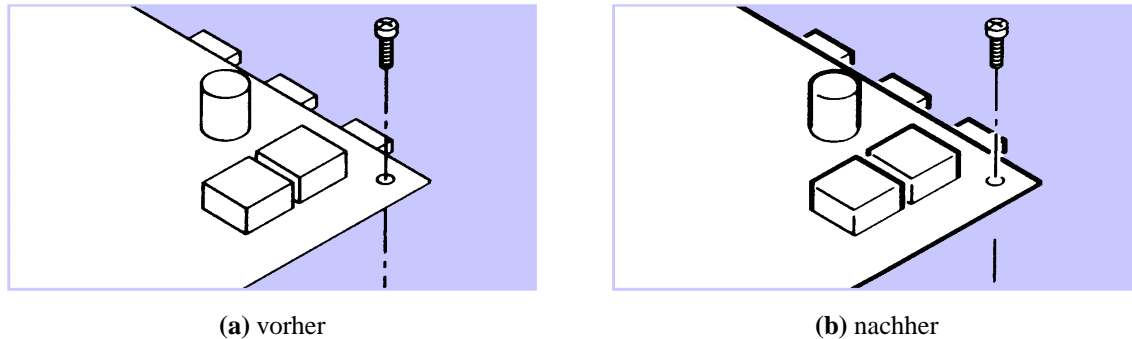


Abbildung 2.5: Trennlinien [Zei96]

Des weiteren wird erkennbar, daß die breitere Umrißlinie nur an den Stellen der Objektkontur verwendet wurde, an denen das betreffende Teil nicht direkt auf ein anderes aufsetzt (an den Bauteilen auf der Platine sichtbar). Damit wird der direkte Kontakt dieser Teile mit der größeren Fläche verdeutlicht. Die breite Umrißlinie wird auch nicht bei sehr kleinen, detailliert dargestellten Teilen verwendet (die Schrauben), da diese dann nicht mehr erkennbar wären.

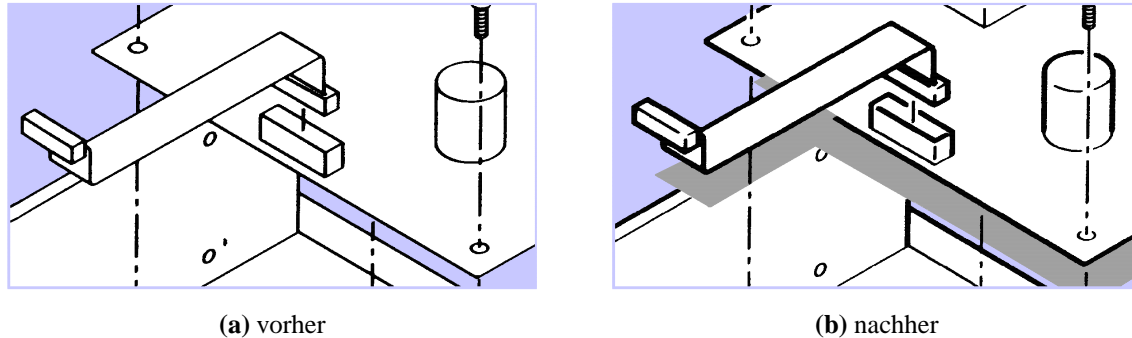


Abbildung 2.6: Leerflächen [Zei96]

Eine noch stärkere visuelle Trennung der einzelnen Baugruppen wird durch Leerflächen erreicht (Abbildung 2.6). Diese künstlichen Flächen wirken zum einen wie räumliche Objekte, sie wurden genau an den Stellen plaziert, wo sich die Projektionen zweier verschiedener Baugruppen überlappen. Zum anderen ist dadurch aber auch jede Baugruppe insgesamt von Flächen umgeben, die Flächen haben also eine ähnlich integrierende Wirkung wie der Hintergrund (siehe oben).

Ein interessantes Detail ist die in dieser Abbildung vorhandene Buchse, in die das Flachbandkabel gesteckt werden soll. Bei ihr wird durch die breite Darstellung der oberen inneren Kanten der Hohlraum deutlich gemacht.

2.3 Weitere Beispiele

Diversen Publikationen wurden die folgenden Illustrationen entnommen. Sie wurden wegen der in ihnen verwendeten Methoden zur unterschiedlichen Charakterisierung von Linien ausgewählt.

Aus einem Medizinlehrbuch stammt die Illustration der Freiheitsgrade der Gehörknöchelchen (Abbildung 2.7). Die Linienstärke variiert in dieser Zeichnung nur unwesentlich (rechts unten ist eine Verbreiterung der Kontur erkennbar). Es wurden aber Lücken an den Stellen gelassen, wo eine Linie verdeckt wird. Auch die Drehpfeile wurden durch einen weißen Saum von den Drehachsen getrennt. Um die Achsen in ihrer gesamten Länge darzustellen, hat man sie gestrichelt, solange sie im Inneren von bzw. hinter Objekten verlaufen. Am Steigbügel ist eine innere Linie durchbrochen dargestellt, dies verleiht ihr eine abgerundete Erscheinung.

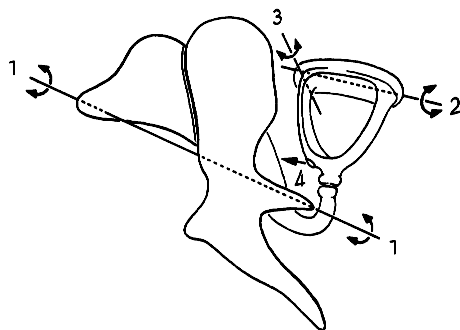


Abbildung 2.7: Gehörknöchelchen: Hammer, Amboß und Steigbügel

Die Montage eines Kurbelwellensterrades (aus einem Reparaturhandbuch, [VW83]) wird durch Abbildung 2.8 illustriert. Beschriftete Pfeile, auf die im Text Bezug genommen wird, zeigen auf die Feder und die dazugehörige Nut. Für das Hauptobjekt (das zu montierende Zahnrad) wurde eine stärkere Linie benutzt. Dadurch hebt es sich vom Hintergrundobjekt ab. Zusätzlich führen keine Linien bis direkt an dieses Teil heran. Im Bild sind außerdem einige durchbrochene Linien zu finden, die für abgerundete Kanten stehen.

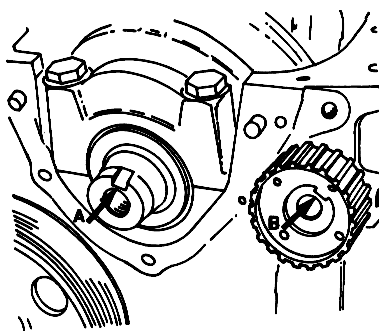


Abbildung 2.8: Aus einer Reparaturanleitung [VW83]

Zuletzt sollen noch einige Illustrationen aus dem Bildwörterbuch [Dud77] betrachtet werden. In diesem Nachschlagewerk werden Begriffe durch Bilder erklärt. Dabei kommen ausschließlich Liniengraphiken zum Einsatz, die extrem vereinfacht und auf das Wesentliche beschränkt sind (Abbildung 2.9, vergrößert).

Die Entwicklerdose (a) weist eine lichtabhängig variierte Linienstärke auf, die rechts unten verlaufenden Kanten wurden breiter dargestellt. An der Stelle, wo die rechte Kontur unter dem Rand verschwindet, wurde ein Fülldreieck plaziert. Es wirkt wie ein kleiner auf die Dose geworfener Schatten. Der Rand des Deckels hat in der Mitte eine größere Lücke, wahrscheinlich soll ein Glanzlicht erzeugt werden.

Der Lederbalg am FuÙe des Schaltknüppels (b) wurde in sehr weichen, runden Formen gezeichnet. Auch hier suggerieren die durchbrochenen Linien ein Glanzlicht (vorne links), die schwarzen Dreiecke einen Schattenwurf (rechts). Auch auf den Schaft wird durch den Knauf ein Schatten geworfen, dargestellt durch die breite Linie am Ansatz.

Für die Konturen der Matratzen (c) wurde eine breite Linie verwendet. Dies hebt die einzelnen Matratzen voneinander ab, stellt aber auch einen Gegensatz zu dem mit dünnen Linien gezeichneten Hund her. Die dünnen inneren Linien, die nicht bis in die Ecken gezogen wurden, lassen eine weiche Kante ahnen.

Auch das Material eines Objektes kann in beschränktem Umfang mit einer Konturzeichnung deutlich gemacht werden. Dies ist an dem Pferd (d) gut zu erkennen. Die aus vielen Stricheln zusammengesetzte Kontur stellt eine Umsetzung für das Fell dar. Durch die unterschiedliche Linienqualität werden die Objekte leicht voneinander unterscheidbar. An den Rädern des Gestells ist wieder die Darstellung weicher Kanten durch nicht bis zum Objektrand durchgezogene Linien zu sehen.

2.4 Zusammenfassung

Die vorgestellten Beispiele zeigen, daß mit einem präzisen Einsatz weniger Linien und ihrer dezenten Attributierung informative und ansprechende Darstellungen zu erzielen sind. Durch die Auswahl der dargestellten Linien werden die Objekte charakterisiert. Die Linienausprägung deutet beispielsweise auf Lichtverhältnisse oder die Wichtigkeit von Objekten hin.

Solche Graphiken bieten damit ein hohes Potential für eine gezielte Beeinflussung der Bild-

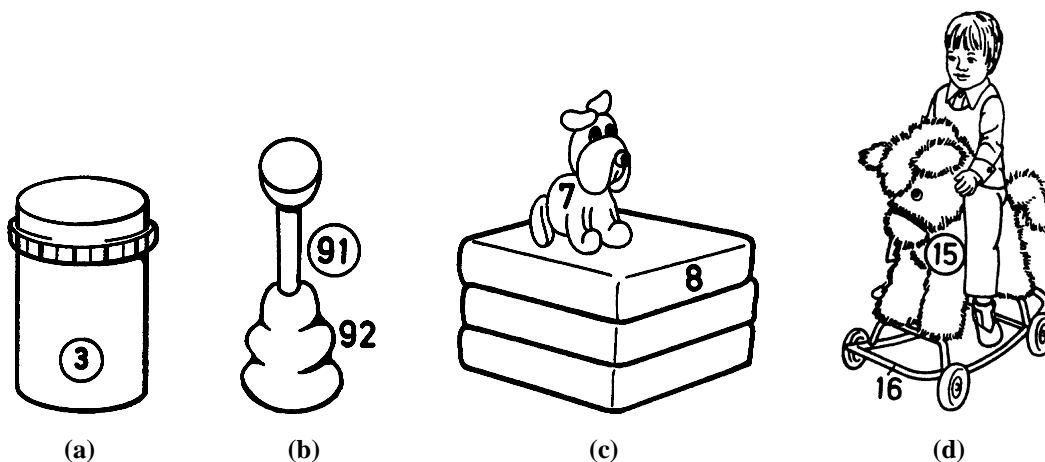


Abbildung 2.9: Illustrationen aus dem Bildwörterbuch [Dud77]

aussage. Eine Umsetzung mit dem Computer erscheint teilweise sehr schwierig. So würde beispielsweise der Einsatz der Trennflächen wie in Abbildung 2.6 für nicht-triviale Modelle kaum automatisch möglich sein, da ihr Einfügen eventuell größere Modelländerungen zur Folge hätten. Andere Techniken erscheinen wiederum leichter realisierbar.

In dieser Arbeit sollen besonders die hier gefundenen Techniken zur Darstellung, Hervorhebung und optischen Trennung von Objekten durch in ihren Attributen variierte Linienzüge Beachtung finden.

Die erste Frage ist dabei die nach der Auswahl und Platzierung der Linien. Die Linienklassifikation kann durch die Auswertung der Informationen des Geometriemodells erfolgen. Da im Geometriemodell alle Kanten der Objekte – und nicht nur die sichtbaren – enthalten sind, müssen diese erst bestimmt werden. Zur Darstellung sollten diejenigen sichtbaren Kanten herangezogen werden, die mit Kontur- oder inneren Linien korrespondieren. Andere eventuell im Modell vorhandene Kanten sollten keine Ausgabe erzeugen.

Hervorhebungen in Abhängigkeit von der Beleuchtung können durch die Auswertung eines Beleuchtungsmodells erreicht werden. Die dabei gefundene Helligkeit kann in eine Variation der Liniendarstellung eingehen, beispielsweise durch Anpassung der Linienbreite. Auch andere Objektbetonungen sind so erzielbar.

Nicht nur die Veränderung der Linienbreite, auch eine Modifikation der Liniendarstellung selbst wurde eingesetzt. So wurde beispielsweise durch eine Zahnung eine Objektrundung angedeutet oder mit Stricheln Materialhinweise gegeben. Der Stil einer gezeichneten Linie sollte daher beeinflusst werden können.

Die Unterbrechung von Linien an Verdeckungen stellt eine wirksame Möglichkeit der visuellen Trennung von Objekten dar. Wenn für eine Linie bekannt ist, daß ihr Ende verdeckt wird, kann eine solche Unterbrechung automatisch erzeugt werden.

Auch die in der technischen Illustration verwendete Einfärbung von Objekt oder Hintergrund erscheint umsetzbar. Dazu müssen allerdings nicht nur Informationen über die sichtbaren Kanten, sondern auch über die sichtbaren Flächen verfügbar sein.

Inwiefern diese Methoden beim heutigen Entwicklungsstand der computergenerierten Liniengraphik schon Berücksichtigung fanden, wird im nächsten Kapitel untersucht.

3 Computergenerierte Liniengraphik

Mit liniengraphischen Darstellungen beschäftigt sich eine große Zahl von Arbeiten der Computergraphik. Schon Anfang der 60er Jahre wurde der erste Algorithmus zur Entfernung verdeckter Linien vorgestellt [Rob63]. Später widmete sich die Forschung hauptsächlich dem Ziel, immer realistischere Bilder zu erzeugen (im Sinne einer Simulation der Photographie), und dieser Trend dauert noch an. Seit Anfang der 90er Jahre ist ein zunehmendes Interesse an alternativen Visualisierungsmethoden zu beobachten.

Im folgenden sollen Arbeiten betrachtet werden, die Linien bewußt als Stilmittel zur Darstellung dreidimensionaler Objekte einsetzen. Ausgeklammert bleiben dabei allerdings die vielfältigen Algorithmen zur Bestimmung sichtbarer Linien, da sie sich nicht in den generierten Linien unterscheiden, sondern nur im Verfahren, diese zu erzeugen. Beispielhaft seien [App67, Wil72, Wri72, WA77, Gri79, HE80, Ran87] erwähnt. Auch die rein auf Bildern arbeitenden, meist interaktiven Methoden (wie die Generierung von Federzeichnungen nach Photographien [SABS94]) werden nicht näher berücksichtigt.

Es lassen sich zwei generelle Herangehensweisen bei der (halb-) automatischen Erzeugung von Liniengraphiken ausmachen. Zum einen sind das die pixelorientierten Verfahren, die unter Anwendung von Bildverarbeitungsmethoden linienartige Pixelgebilde generieren (Abschnitt 3.1). Die andere Gruppe verwendet eine analytische Repräsentation der sichtbaren Kanten bzw. Flächen zur Bilderzeugung (Abschnitt 3.2). Vertreter beider Methoden werden im folgenden vorgestellt.

3.1 Pixelorientierte Verfahren

Als prototypisch für die Klasse der pixelbasierten Methoden kann die Arbeit von SAITO und TAKAHASHI gelten [ST90]. Sie widmen sich dem Ziel, computergenerierte Bilder „verständlicher“ zu machen (*comprehensible rendering*). Dazu verwenden sie G-Buffer¹. Das ist eine spezielle Datenstruktur, die für jeden Bildpunkt während des Renderns über den Farbwert (r, g, b) hinausgehende Informationen speichert (i. A. geometrischer Natur). Beispiele für G-Buffer sind die Objekt-ID, Bildkoordinaten (nur z), Texturkoordinaten (u, v) , Weltkoordinaten (x, y, z) und Normalen (x, y, z) .

Die zusätzlichen Kanäle werden als Bilder aufgefaßt und mittels Bildverarbeitungs-Operatoren

¹ Das „G“ steht für *geometric*, wird aber auch im Englischen [ge:] gesprochen zur besseren Unterscheidung vom „J“

in neue Bilder umgesetzt. Dabei kommen hauptsächlich Kantendetektoren und Schwellwert-Operatoren zum Einsatz.

So liefert beispielsweise ein die C^0 -Unstetigkeiten des z -Kanals darstellender Filter ein Bild, in dem die Pixel an Objektkonturen gesetzt sind (da hier die Tiefendifferenz benachbarter Pixel relativ groß ist). Bei entsprechend gewählter Filtergröße entstehen Linien, deren Breite abhängig von der Tiefendifferenz ist. Wird statt der Differenz deren Änderung verwendet (d. h. die zweite statt der ersten Ableitung), so ergeben sich auch Linien an den inneren Kanten (den „Knicken“ der Oberfläche). Schraffurähnliche Strukturen erzeugt die periodische Einfärbung bestimmter Werte von Kanälen wie z. B. der u - und v -Parameter. Die so generierten Bilder werden für Hervorhebungen im (r, g, b) -Bild genutzt, können aber auch für sich allein stehen (Abbildung 3.1).

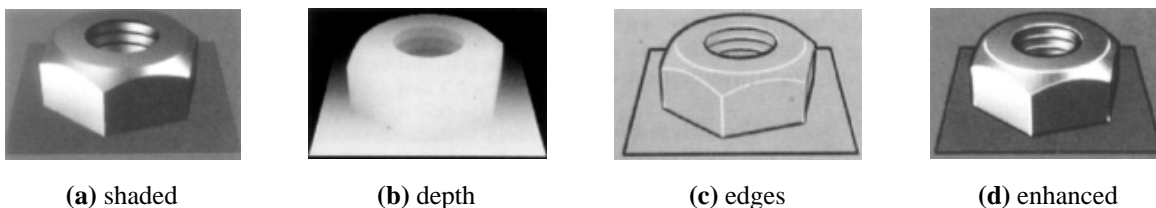


Abbildung 3.1: Anreicherung eines gerenderten Bildes [ST90]

Da die Erzeugung von Liniengraphiken für die Autoren nur einen Zwischenschritt zur Anreicherung einer photorealistischen Darstellung bildet, gehen sie nicht näher auf mögliche Variationen ein. Diese erscheinen auch nur in eingeschränktem Maße möglich. So dürfte es schwierig sein, Linien unabhängig voneinander zu behandeln, um zum Beispiel innere Linien anders als Konturlinien darzustellen. Bemerkenswert ist hingegen, daß überhaupt durch reine Pixeloperationen liniengraphische Bilder entstehen können. Dafür sind die dreidimensionalen Informationen im G-Buffer nötig, da zum einen Konturen im Graubild nicht notwendigerweise von Objektkanten herrühren, zum anderen Objektkanten nicht notwendigerweise Konturen im Graubild erzeugen.

Mit einem modifizierten Raytracer und ähnlichen Methoden wie den eben beschriebenen erzeugt LEISTER kupferstichartige Darstellungen [Lei94]. Um die für diese Graphiken typischen Schraffurlinien zu erzeugen, wird für jedes Objekt der Szene eine Schar paralleler Ebenen definiert, entlang deren Schnitt mit der Objektoberfläche schraffiert werden soll. Dazu wird außer dem „normalen“ per Raytracing ermittelten RGB-Bild ein zusätzliches $\kappa\lambda\eta$ -Bild erzeugt, wobei κ ein Schraffurmaß ist (Nähe zu einer Schraffurlinie), λ die normalisierte Länge des Strahls und η ein Maß des Winkels zwischen der Oberflächennormalen und dem Strahl. Von der Helligkeit abhängig unterschiedlich breite Schraffurlinien entstehen durch die Anwendung eines Schwellwertoperators auf die Verknüpfung des κ -Kanals mit der Intensität des RGB-Bildes. Objektkonturen werden wie bei SAITO/TAKAHASHI beschrieben bestimmt (λ entspricht dabei z), innere Kanten ebenso, allerdings im η -Kanal (Abbildung 3.2).

Die Konturen sind in den erzeugten Bildern, wie für Kupferstiche typisch, weniger wichtig und werden deshalb auch nicht gesondert behandelt. Die Schraffurlinien variieren dagegen nicht nur im Abstand, sondern auch beleuchtungsabhängig in der Breite. An den Glanzlichtern sind sie sogar durchbrochen. Inwiefern dies auch für die Konturlinien möglich wäre, bleibt bei LEISTER offen.

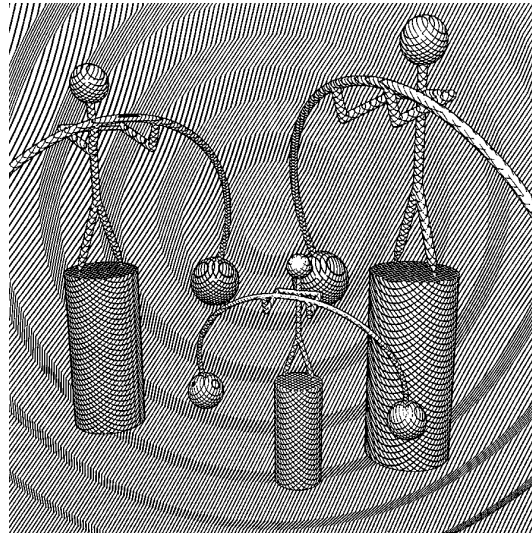


Abbildung 3.2: Mobilés als computergenerierter Kupferstich [Lei94]

Nicht ausschließlich pixelorientiert arbeitet dagegen das von SCHOFIELD vorgestellte interaktive Illustrationssystem „Piranesi“ [Sch94]. Die Bilder entstehen dabei nicht durch Bildverarbeitungs-Methoden, sondern der Benutzer „malt“ mit einer Reihe von Werkzeugen das Bild. Dadurch werden Bildelemente (*marks*) erzeugt, die größer als ein Pixel sind.² Während des Malens werden die Werkzeuge von einer den G-Buffern sehr ähnlichen Datenstruktur beeinflusst. Diese EPixel-Matrix (von *enriched pixels*) kann beispielsweise die beim Rendern ermittelte Helligkeit, die Objekt-ID oder Schattenmasken speichern und enthält, als Wichtigstes, den z -Buffer. Anhand der Objekt-ID kann z. B. das Zeichnen auf gewisse Objekte beschränkt werden. Verwendet man ein für Diskontinuitäten des z -Buffers sensitives Werkzeug, entsteht eine Linienzeichnung. Schraffuren oder Texturen können auch perspektivisch korrekt aufgebracht werden.

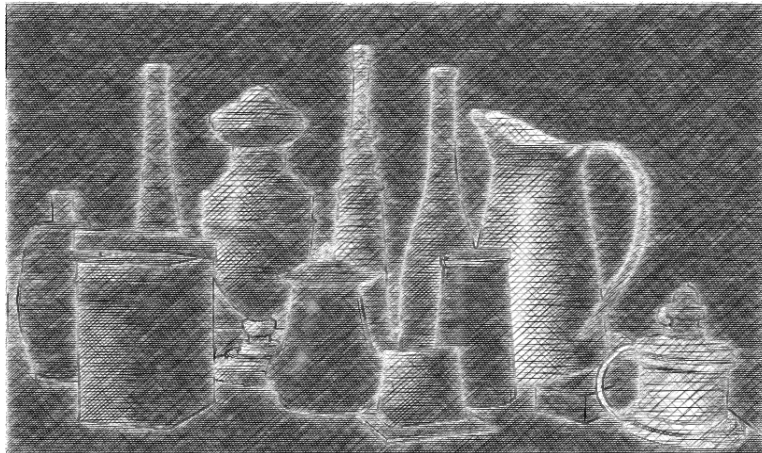


Abbildung 3.3: Interaktiv erstelltes Stilleben [Sch94]

Durch die interaktive Bilderzeugung sind alle Aktionen lokal begrenzt. Damit können im Prinzip viele mit traditionellen Medien erreichbare Darstellungen nachempfunden werden

² Diese Eigenschaft wird von SCHOFIELD als charakteristisch für nicht-photorealistische Darstellungen angesehen.

(notfalls werden sie Pixel für Pixel gezeichnet). Die Unterstützung durch das System beschränkt sich darauf, bestimmte Bildbereiche zu schützen (damit sie nicht versehentlich übermalt werden) oder Parameter des Zeichenwerkzeuges an die jeweilige Bildposition anzupassen. Durch geeignete Wahl dieser Parameter können zum Beispiel die Konturen hervorgehoben werden. Eine automatische Erzeugung einer Konturzeichnung ist aber nicht vorgesehen.

3.2 Analytische Verfahren

Im Gegensatz zu den pixelorientierten Verfahren basieren die analytischen (oder vektororientierten) Verfahren nicht auf einer diskretisierten Modellbeschreibung. Vielmehr benutzen sie direkt die im 3D-Modell gespeicherten Informationen.

Ein Verfahren zur Darstellung von Linien ähnlich den im Kapitel 2 beschriebenen Lücken an Verdeckungen beschreibt APPEL [ARS79]. Seine *haloed lines* (Abbildung 3.4) ersetzen interessanterweise einen Algorithmus zur Bestimmung sichtbarer Kanten. Eine einfache Wireframe-Repräsentation reicht aus, da keine Flächeninformation gespeichert sein muß.

Zur Erzeugung dieser Darstellung wird jede Kante vor dem Zeichnen mit allen vor ihr verlaufenden Kanten geschnitten. Am Schnittpunkt liegende Intervalle werden als unsichtbar markiert. Die verbleibenden, sichtbaren Segmente werden dargestellt. Wenn der *Halo* entsprechend groß gewählt wird, werden hintere Kanten komplett entfernt. Dies entspricht allerdings nicht einer Entfernung verdeckter Kanten, da sich der Halo auch über die (gedachten) Objektgrenzen hinaus erstreckt.

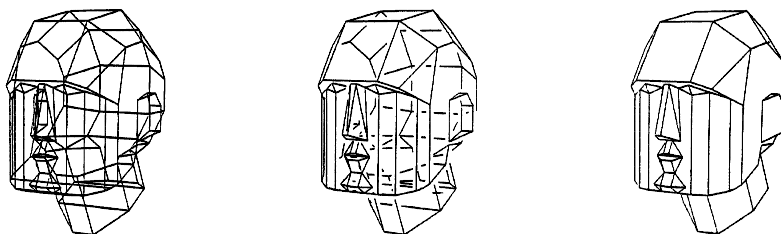


Abbildung 3.4: Kopf mit allen Kanten, Halo bzw. entfernten Kanten [ARS79]

In dieser relativ frühen Arbeit ging es primär darum, die Rechenzeit für die Entfernung verdeckter Kanten einzusparen. Mit dem Verfahren können Drahtgittermodelle gut illustriert werden. Faßt man jedes Liniensegment als dreidimensionales Objekt („Drahtstück“) auf, entspricht die Darstellung exakt einem Verfahren zur Konturunterbrechung an Verdeckungen.

Diese Idee wird von ELBER wieder aufgegriffen und auf parametrische Modelle erweitert [Elb95]. Zur Verstärkung des Tiefeneindrucks wird mittels Depthcueing die Helligkeit bzw. Breite der dargestellten Linien verändert. Statt der Kanten eines polygonalen Modells werden bei Freiformflächen isoparametrische Kurven verwendet (Abbildung 3.5).

Da die Flächeninformation bei dieser Konvertierung verlorengeht, kann im Gegensatz zu polygonalen Modellen (die in ELBERS Arbeit mit ähnlichen Mitteln illustriert werden) keine Eliminierung der von Flächen verdeckten Kanten mehr angewandt werden. Da zudem –

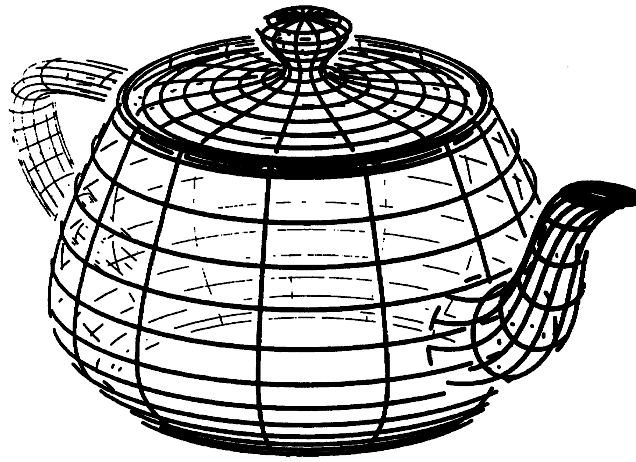


Abbildung 3.5: Illustration des Utah-Teapots [Elb95]

anders als die Kanten von Polygonen – die isoparametrischen Linien im allgemeinen nicht auf dem sichtbaren Rand des Objektes liegen, ist die Objektkontur so nicht darstellbar. Das äußert sich in einer durchbrochenen, unvollständig wirkenden Silhouette (z. B. am Henkel deutlich sichtbar).

Durch eine explizite Bestimmung der Silhouettenkurve von parametrischen Flächen wird dieses Problem gelöst. Diese Technik verwenden DOOLEY und COHEN in ihrem System zur automatischen Illustration von 3D-Geometriemodellen [DC90]. Ihr Ziel ist es, ähnlich wie SAITO und TAKAHASHI eine photorealistische Darstellung mit einer Liniengraphik zu überlagern (Abbildung 3.6). Dadurch sollen vor allem bei herkömmlichem Rendern unsichtbare Strukturen sichtbar gemacht werden.

Als Modelle dienen DOOLEY und COHEN Freiformflächen, denen zusätzlich eine Wichtung gegeben wird (*importance*). Besondere Beachtung findet die Klassifikation der Kurvensegmente und deren Endpunkte.

Die Kurven werden unterschieden in *boundary lines* (Patch-Ränder), *silhouette lines* (die eben erwähnten durch die Objektwölbung entstehenden Konturen), *discontinuity lines* (Unstetigkeiten, von den Autoren auch als Knicke (*fold*) bezeichnet) und *isoparametric lines* (auf dem Patch verlaufende Linien, die keine Unstetigkeiten darstellen). Ein Segment ist der Teil einer Kurve, in dessen Verlauf keine *boundary* oder *silhouette lines* diese Kurve schneiden. Im Verlauf eines Segments ändert sich also nicht die Anzahl der verdeckenden bzw. verdeckten Flächen. Diese Anzahl in Verbindung mit der Wichtigkeit sowie dem Segmenttyp entscheiden über die Darstellung des Segments (Linienstärke und durchgezogene bzw. gestrichelte Linien). Die dafür zu verwendenden Linienattribute werden interaktiv vorgegeben. Nicht spezifizierte Kombinationen werden aus den definierten abgeleitet.

Die Segmentenden werden je nach Zu- oder Abnahme des Verdeckungsgrades unterschiedlich dargestellt. Auf der stärker verdeckten Seite wird die verdeckte Linie nicht bis an die verdeckende Kontur gezeichnet (*open end*), auf der anderen Seite schließt die Linie an die Kontur an (*closed*). Hier kann die Linie zusätzlich verbreitert werden (*thickened*). Eine auslaufende *silhouette line* wird verjüngt dargestellt (*tapered*).

Dieses System verwirklicht somit einige der in Kapitel 2 identifizierten Techniken. Insbeson-

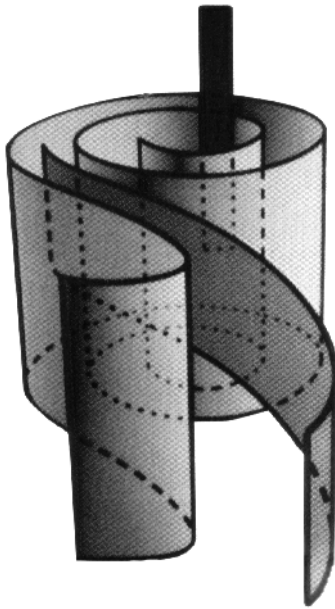


Abbildung 3.6: Mit Linien überlagerte Flächendarstellung [DC90]

dere werden die Linientypen ähnlich klassifiziert und zur differenzierten Darstellung ausgewertet. Das Hauptaugenmerk legen die Autoren aber auf die Darstellung von verdeckten Strukturen, die in einem herkömmlichen Rendering nicht sichtbar sind. So werden Lücken nur auf der normalerweise unsichtbaren Seite einer Verdeckung gelassen. Algorithmische Schwierigkeiten ergeben sich nach Aussage der Autoren aus der direkten Verwendung von Freiformflächen in der Bestimmung sichtbarer Segmente.

Eher künstlerischen Anspruch als diese für technische Illustrationen gedachten Graphiken haben die von WINKENBACH und SALESIN generierten Federzeichnungen [WS94]. Aus polygonalen Modellen werden die sichtbaren Flächen durch einen analytischen BSP-Renderer bestimmt und sowohl als 2D-BSP-Baum als auch in einer *Map* gespeichert. Ein herkömmliches Rendern derselben Szene liefert eine Grauwert-Referenz. Dieser Grauton wird durch das Zeichnen von aus einzelnen Strichen aufgebauten prozeduralen Texturen schrittweise angenähert (Abbildung 3.7).

Diesen sorgfältig abgestimmten *prioritized stroke textures* verdanken die erzeugten Bilder ihre stark an manuelle Federzeichnungen erinnernde Erscheinung. In einer Textur für Mauerwerk werden beispielsweise erst die Umrisse der Ziegel dargestellt (sie haben die höchste Priorität), es folgen Schraffuren einzelner Steine. Ist der gewünschte Grauton noch nicht erreicht, setzen größere Kreuzschraffuren ein.

Die Texturen werden auf allen zumindest teilweise sichtbaren Flächen generiert, an dem BSP-Baum geclippt und die verbleibenden Stücke gezeichnet. Aus den Polygonkanten der Map werden Umrisse erzeugt, die dargestellt werden, falls der Kontrast zwischen den beiden angrenzenden Flächen relativ klein ist.

Das Verfahren wird in [WS96] auf parametrische Oberflächen, wie NURBS oder Rotationskörper, ausgedehnt. Die Bestimmung sichtbarer Flächen erfolgt wie gehabt, nachdem die Freiformflächen polygonalisiert wurden. Deren Triangulierung ist nicht adaptiv, nur die

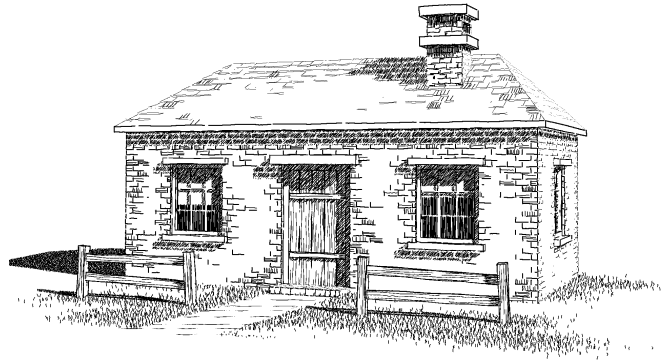


Abbildung 3.7: Haus als computergenerierte Federzeichnung [WS94]

Kontur wird soweit verfeinert, daß keine Ecken sichtbar werden.

Anders ist die Erzeugung von Schraffuren gelöst, da die *prioritized stroke textures* nur für große planare Flächen geeignet sind. Die Linien verlaufen jetzt entlang isoparametrischer Linien, sie werden im Bild ausgedünnt und in der Breite variiert, um den Grauton des Referenzbildes zu erzeugen (*controlled-density hatching*). Auch die Umrisse mußten anders erzeugt werden, da in der Map jetzt vor allem Triangulierungskanten enthalten sind. Dazu werden aus allen Kanten die relevanten ausgewählt (Konturlinien und Unstetigkeiten darstellende innere Kanten), soweit wie möglich verkettet und dargestellt.

Die Darstellung der Objektkonturen ist bei WINKENBACH/SALESIN sekundär, ihr Hauptziel ist die texturierte Darstellung der Flächen. Dennoch wird der Ausführung von Konturlinien Beachtung geschenkt. Sie werden stärker gezeichnet, wenn der Helligkeitsunterschied der angrenzenden Flächen relativ gering ist. Abgeschwächt werden sie in sehr hellen Bildregionen. Allerdings wird nicht nach verschiedenen Linientypen unterschieden.

Liniengraphiken ganz anderer Art erzeugt der von STROTHOTTE et al. vorgestellte „Sketchrenderer“ [SPR⁺94]. Es werden bewußt ungenaue, künstlich verfremdete Bilder generiert (Abbildung 3.8). Sie kommen beispielsweise in Architekturskizzen zum Einsatz.

Mittels eines BSP-Renderers wird aus einem polygonalen Geometriemodell für eine bestimmte Ansicht ein Sichtbarkeits-Bild generiert. Dieser G-Buffer speichert für jeden Bildpunkt, welche Fläche sichtbar ist. Die Polygonkanten und auf die Polygone aufgebrachte Schraffur- und Schattenlinien werden an dem Sichtbarkeits-Bild geclippt. Die entstehenden Liniensegmente werden mit Liniestilen versehen gezeichnet.

Ein Liniensstil definiert dabei eine stückweise polynomiale Kurve, durch die ein darzustellendes Geradensegment ersetzt wird. Die Kurve wird in Orientierung und Größe dem Segment angepaßt. Dem Liniensstil kann ein Breiten- und Helligkeitsverlauf gegeben werden. Durch die mögliche Definition stochastischer Abweichungen des Stils wird der skizzenhafte Eindruck erreicht [Sch92].

Zur Verfeinerung der Bildaussage wurden diverse Techniken eingesetzt. So können Schatten, wie in Handzeichnungen, von der sie erzeugenden Fläche abhängig dargestellt werden

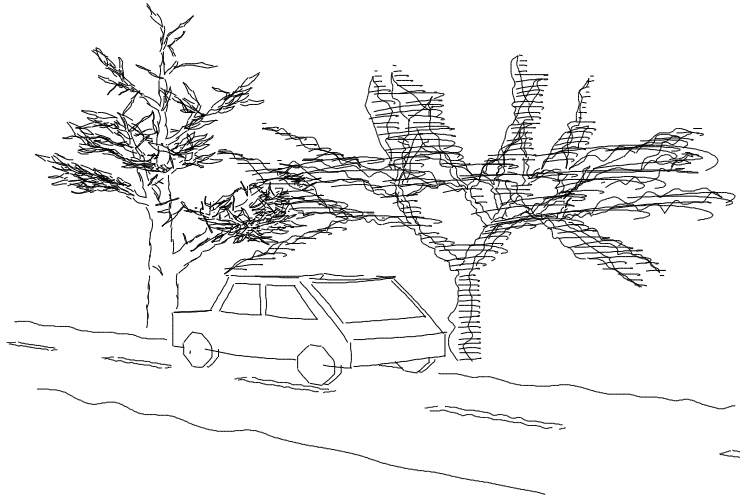


Abbildung 3.8: Computergenerierte Skizze [SPR⁺94]

[Raa94]. Die Aufmerksamkeit des Betrachters kann auf bestimmte Bildbereiche fokussiert werden [PS95]. Zur Betonung von Bildteilen wird eine Beschränkung zeichnerischer Ressourcen eingeführt [SS96].

Durch den Einsatz von Linienstilen können selbst grobe Modelle, wie sie in frühen Entwicklungsstadien eines Produktes üblich sind, ansprechend dargestellt werden. Die skizzenhafte Darstellung unterstützt dabei den unfertigen Charakter der Objekte. Die Darstellung der sichtbaren Kanten eines polygonalen Modells erzeugt eine Konturzeichnung. Beeinflusst werden kann zum Beispiel der Linienstil pro Objekt, was eine visuelle Differenzierung der Objekte ermöglicht. Nicht möglich ist dagegen eine vom Konturtyp abhängige Darstellung, da die darzustellenden Liniensegmente nicht attribuiert sind.

3.3 Auswertung

Alle vorgestellten Verfahren bieten die Möglichkeit der Darstellung von Kontur- und Binnenlinien, wie sie in Kapitel 2 als Elemente von Konturzeichnungen gefunden wurden.

Die pixelorientierten Methoden erzeugen sie durch Operatoren der Bildverarbeitung in erster Linie aus Bildern, die die Tiefe jedes Bildpunktes speichern (z -Buffer). Auch die Kombination mit anderen G-Buffern ist möglich. Die Verwendung von G-Buffern ist auch deswegen vorteilhaft, weil sie mit etablierten Algorithmen zur Bestimmung sichtbarer Flächen erzeugt werden können. Diese Aufgabe kann auch Hardware-Renderern übertragen werden. Die gewonnenen Linienbilder können mit relativ geringem Aufwand in flächenhafte Bilder integriert werden. Allerdings sind sie in der Auswahl und Beeinflussung der Darstellung stark eingeschränkt, da die resultierenden „Linien“ nicht im einzelnen manipuliert werden können.

Vielfältige Darstellungsweisen sind dagegen mit den analytischen Methoden möglich, wie besonders die drei zuletzt vorgestellten Arbeiten belegen. Durch die explizite Verfügbarkeit der Linien bieten sie wohl das größere Potential für die gezielte Beeinflussung ihrer Darstellung.

Keine der Arbeiten beschäftigte sich ausdrücklich mit Konturzeichnungen. Am nächsten kommen ihnen die Linienillustrationen von DOOLEY/COHEN. In ihrer Arbeit finden sich zum einen die Klassifikation von Linien in Typklassen, die unterschiedlich dargestellt werden können. Zum anderen berücksichtigen sie die Darstellung von Linienenden gesondert. Probleme bereitet ihnen insbesondere die Verwendung parametrischer Oberflächen.

In der Arbeit von WINKENBACH/SALESIN finden Konturlinien nur am Rande Erwähnung. Dennoch ist die Variation der Linienstärke in Abhängigkeit vom Flächenkontrast bemerkenswert.

Die Verwendung von Linienstilen im Sketchrenderer ermöglicht eine sehr flexible Gestaltung der dargestellten Linien. Diese kann aber nicht von einer einzelnen Kante abhängig gemacht werden. Die Darstellung ist auf Polygonkanten eingeschränkt, was nur für die verwendeten groben Modelle ansprechende Ergebnisse produziert. Runde oder detaillierte polygone Objekte enthalten aber viele kurze Kanten, die in der Darstellung zusammengefaßt werden müßten.

Zur computergestützten Erzeugung von Konturzeichnungen, die die Verwendung von Techniken ähnlich den traditionellen Zeichentechniken erlauben, erscheint eine Kombination und Erweiterung dieser Ansätze sinnvoll. Dazu wird das Konzept der charakteristischen Linien entwickelt, welches im nächsten Kapitel vorgestellt wird.

4 Charakteristische Linien

Wie Kapitel 2 zeigte, werden in liniengraphischen Illustrationen eine Reihe von Techniken verwendet, die durch geeignete Variationen in der Darstellung von Linien Objekte verständlich und interessant präsentieren. Im letzten Kapitel wurde festgestellt, daß diese Techniken in der Computergraphik erst ansatzweise umgesetzt wurden. Dabei stellen die vorgestellten Arbeiten nur den aktuellen Forschungsstand computergenerierter Liniengraphiken dar – kommerziell üblich ist dagegen eine undifferenzierte Kantendarstellung, wie beispielsweise die Drahtgitter-Darstellung in vielen Modellierungssystemen.

Linien, die durch eine differenzierte Darstellung Objekte einer Szene charakterisieren, sollen *charakteristische Linien* genannt werden. Sie haben dabei die zwei eingangs erwähnten Funktionen – die Darstellung der Form eines Objektes und die Visualisierung seiner Relation zu anderen Objekten. Dafür sind die *Auswahl* und *Plazierung* der Linien sowie ihre *Gestaltung* maßgeblich. Dieses Kapitel beschreibt die entwickelten Konzepte zur automatischen Erzeugung derartiger Linien aus polygonalen Modellen.

In dieser Arbeit werden zwei Werkzeuge eingesetzt. Zum einen ist das der vom Autor implementierte analytische Renderer, auf dessen Ausgabe (der Beschreibung sichtbarer Kanten und Flächen) die Erzeugung charakteristischer Linien aufsetzt. Zum anderen wird zum Darstellen der erzeugten Linien das von L. SCHUMANN entwickelte parametrisierbare Linienmodell genutzt [Sch97]. Diesem wird eine attributierte Liste von Stützpunkten (der Pfad) übergeben. Beim Zeichnen kann zusätzlich eine Störkurve (der Stil) überlagert werden.

Zunächst wird eine Methode entwickelt, um aus sichtbaren Kantensegmenten Kurven zu generieren (Abschnitt 4.1). Dies ist notwendig, da die Auflösung der polygonalen Repräsentation des zu illustrierenden Modells nicht direkt auf die Darstellung Einfluß haben sollte.

Dann wird untersucht, wie die Attributierung dieser Linien geeignet variiert werden kann, um diverse der in Kapitel 2 vorgestellten traditionellen Techniken für die Computergraphik nutzbar zu machen. Dabei werden nach bestimmten Kriterien Attribute des Zeichenpfades verändert (Abschnitt 4.2). Für die betrachtete Art von Liniengraphiken ist dabei die Beeinflussung des Breitenverlaufes der Linienzüge von besonderer Bedeutung.

Da die Behandlung der Linienabschlüsse über eine einfache Modifizierung der Linienattribute hinausgeht, ist diesem Punkt ein eigener Abschnitt gewidmet (Abschnitt 4.3).

Die folgenden Ausführungen werden durchgängig anhand eines Beispielmodells¹ illustriert, das viele in einer Szene auftretende Konstellationen vereint (Abbildung 4.1). Es enthält mehrere konvexe und konkave Objekte (i-„Basis“, i-Punkt, s, g), es treten Verdeckungen auf. In

¹ „isg“ ist das Kürzel des Instituts für Simulation und Graphik, an dem diese Arbeit entstand.

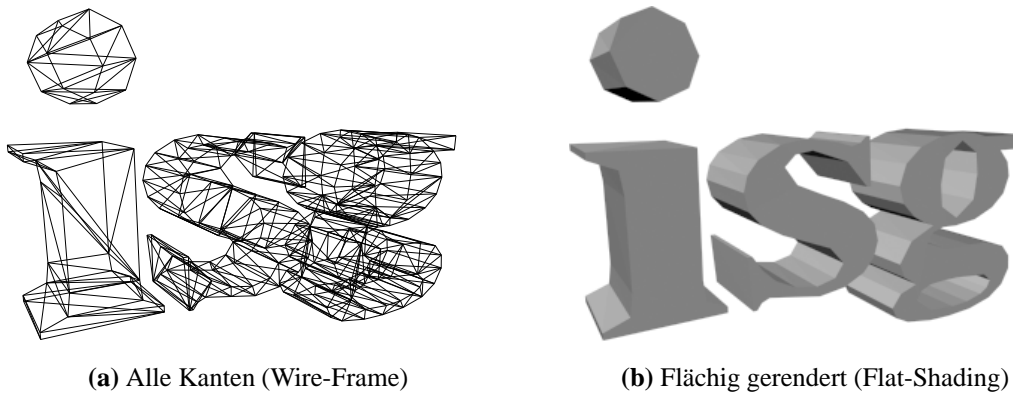


Abbildung 4.1: Das Beispielmodell

der endgültigen Darstellung müssen gerade, gekrümmte und geschlossene Linien verwendet werden, es existieren lange und kürzere Linienzüge. Da viele Darstellungsartefakte nur bei groben Modellen sichtbar werden, wurde es relativ grob polygonalisiert (504 Dreiecke).

4.1 Generierung von Kurven

Aus den vorhandenen sichtbaren Segmenten soll eine „glatte“, Kurven enthaltende Darstellung erzeugt werden. Nach einigen Vorbetrachtungen zur Erzeugung von Linien aus polygonalen Repräsentationen wird in Abschnitt 4.1.2 beschrieben, welche der vorhandenen Kanten für die Kurvengenerierung ausgewählt werden müssen, darauf folgt die Präsentation eines neuen Verfahrens zur Kurvenextraktion (Abschnitt 4.1.3).

4.1.1 Vorbetrachtungen

Den Ausgangspunkt für die Erzeugung liniengraphischer Darstellungen im Rahmen dieser Arbeit bilden polygonale Modelle. Diese bieten gegenüber anderen Repräsentationen (wie beispielsweise Freiformflächen oder implizite Oberflächen) u. a. folgende Vorteile:

- Polygonale Repräsentationen sind weit verbreitet. Es gibt sowohl eine Vielzahl von Modellen, als auch von Systemen, die diese erzeugen (Modellierer, 3D-Scanner) oder verarbeiten (Renderer) können.
- Die Verarbeitung von polygonalen Modellen kann mit Algorithmen erfolgen, die mittlerweile Standard in der Computergraphik sind.
- Andere Repräsentationsformen können relativ einfach in polygonale Modelle konvertiert werden, während der umgekehrte Fall nicht immer zutrifft. Die implementierten Verfahren können daher auch für diese Modelle verwendet werden.

Ein Objekt in einem polygonalen Modell besteht aus einem Polygonnetz, d. h. die Polygone sind zusammenhängend. Damit teilen aneinandergrenzende Polygone eine Kante, was (in Anlehnung an die Klassifikation im Kapitel 2) folgende vom Betrachterstandpunkt (also der gewählten Kamera und damit Projektion) abhängige Kantenklassifikation ermöglicht:

- *Konturkanten* gehören zu genau einer Vorderfläche (eine dem Betrachter zugewandte Fläche). Auf der anderen Seite ist eine Fläche sichtbar, deren Objektzugehörigkeit eine weitere Unterscheidung ermöglicht:
 - Bei *Silhouettenkanten* gehört die dort sichtbare Fläche nicht zum gleichen Objekt wie die Kante selbst.
 - Bei *Selbstverdeckungskanten* dagegen ist eine zum gleichen Objekt gehörende Fläche verdeckt.
- *Innere Kanten* haben zu beiden Seiten Vorderflächen.

Wenn das Modell trianguliert wurde, also Flächen mit mehr als drei Eckpunkten in Dreiecke zerlegt wurden, entstehen zusätzliche Kanten. Diese werden als Triangulierungskanten bezeichnet.

Um eine Liniengraphik zu erzeugen, müssen zunächst die sichtbaren Kanten bestimmt werden. Ein Hidden-Line-Algorithmus liefert im Normalfall eine ungeordnete Folge sichtbarer Liniensegmente (die Projektion der im Modell vorhandenen Kanten), die durch einen Anfangs- und Endpunkt definiert sind. Diese Information ist für eine nicht-triviale Darstellung unzureichend, da das simple Zeichnen aller Segmente eher verwirrend denn informativ wirkt, wie in Abbildung 4.2(a) gezeigt.

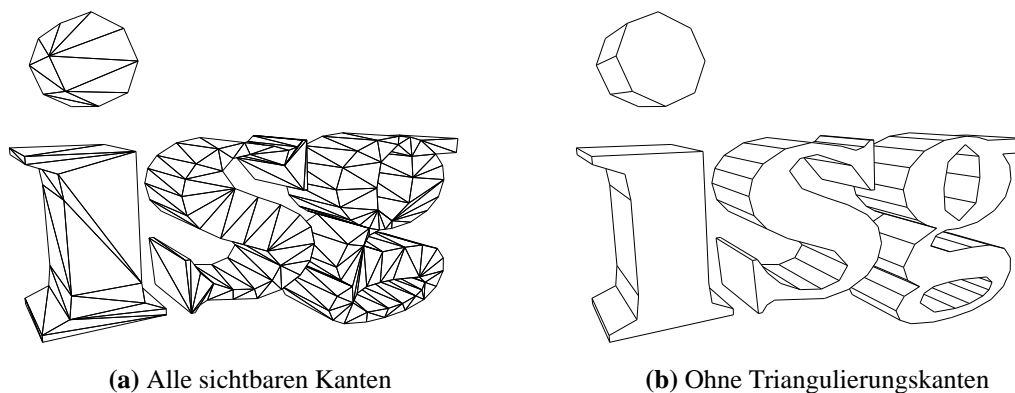


Abbildung 4.2: Sichtbare Kantensegmente

Wünschenswert sind also weitergehende Informationen:

- die Segmente sollten Objekten zugeordnet sein (um die Objekte unterschiedlich darstellen zu können),
- zusammengehörige Segmente müssen sich finden lassen (um Kurvenzüge zu generieren),
- es sollte ersichtlich sein, was ein Segment repräsentiert (entsprechend der Kantenklassifikation)

In Abbildung 4.2(b) ist demonstriert, daß schon das Weglassen der Triangulierungskanten eine deutliche Verbesserung des visuellen Eindrucks beim Betrachter bewirkt, da die in einer Ebene liegenden aneinander angrenzenden Dreiecke optisch zu Flächen zusammengefaßt werden. Bei einer flächenhaften Darstellung sind diese Artefakte nicht sichtbar, da alle aus

einer Fläche resultierenden Dreiecke die gleiche Normale besitzen und somit gleich erscheinen (vgl. Abbildung 4.1(b)).

4.1.2 Kantenselektion

Der polygonale Aufbau des Modells bleibt auch nach dem Weglassen der Triangulierungskanten deutlich sichtbar. Das kann durchaus erwünscht sein (beispielsweise als Modellierhilfe). Eigentlich stellen die sichtbaren Kanten aber nur Artefakte der polygonalen Approximation des „idealen“ Modells dar. Dieser Abschnitt beschreibt eine weitergehende Selektion der Kanten für den Zweck einer liniengraphischen Darstellung.

In der flächenhaften Computergraphik besteht ein übliches Verfahren der Darstellung von durch Polygone approximierten Flächen in der Anwendung interpolierender Schattierungsverfahren, wie GOURAUD- oder PHONG-Shading [FvDFH90]. Dabei wird an den Grenzen benachbarter Polygone ein kontinuierlicher Übergang erzeugt (Abbildung 4.3(a)). Die Silhouette bleibt davon unbeeinflusst. Sie spiegelt nicht die Rundung der Flächen wider, sondern ist deutlich sichtbar polygonal.

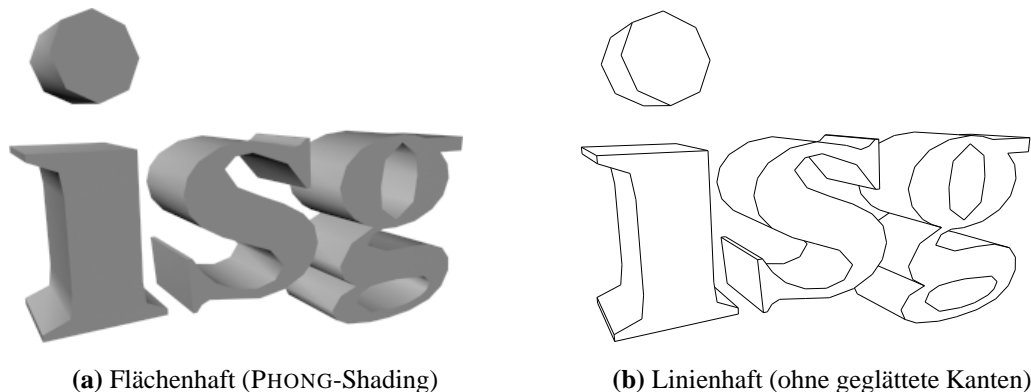


Abbildung 4.3: Abrundung der Flächen

Eine Umsetzung dieses Verfahrens mit liniengraphischen Mitteln veranschaulicht Abbildung 4.3(b). Für diese Darstellung wurden genau die Kanten selektiert, die entweder nur zu einem sichtbaren Polygon gehören (die also auf dem Rand liegen), oder zwischen deren beiden angrenzenden Polygonen nicht interpoliert werden soll. Die erste Bedingung stellt sicher, daß alle Objektkonturen gezeichnet werden, die zweite erzeugt Binnenstrukturen. Es wurden also insbesondere auch geglättete Kanten für die Darstellung selektiert, und zwar genau die, die am Objektrand liegen (sonst wären beispielsweise am i-Punkt die auf dem Zylindermantel liegenden Kanten weggelassen worden).

In den folgenden Ausführungen wird eine Kante, die einen glatten Übergang zwischen zwei Flächen repräsentiert (die also durch die Approximation einer gewölbten Fläche entstanden ist), eine *geglättete* Kante genannt, während eine Kante, die auch einer Kante im ursprünglichen Objekt entstammt, als *scharfe* Kante bezeichnet wird.

Die Information über eine eventuelle Glättung wird üblicherweise für Flächenpaare gespeichert und beim Modellieren erzeugt. Für Primitive (Kugeln, Kegel, Zylinder etc.) ist eine

algorithmische Zuweisung von Glättungsinformationen üblich. Alternativ dazu kann man sie von Hand editieren.

Fehlt die Glättungsinformation, kann trotzdem bei vielen Modellen eine automatische Bestimmung erfolgen. Das geschieht über den Winkel, um den sich die Normalen zweier benachbarter Polygone unterscheiden. Ist die Differenz relativ klein (ein üblicher Wert ist 30°), werden die Polygone als Teile einer gewölbten Oberfläche angesehen.

Diese Annahme muß aber nicht korrekt sein. Ein achtseitiges Prisma (wie der i-Punkt in der Abbildung) kann z. B. sowohl die Annäherung eines Zylinders als auch genau ein achtseitiges Prisma darstellen. Ebenso kann ein Zylinder durch einen Quader approximiert sein. Obwohl alle benachbarten Flächen bei ihm den gleichen Winkel einschließen, müßte über die Mantelflächen interpoliert werden, dagegen zwischen Mantel- und Deckflächen nicht.²

4.1.3 Kurvenextraktion

Weitere Verbesserungen der Darstellung erfordern mehr Aufwand. Störend wirken vor allem die durch die grobe Approximation deutlich sichtbaren Knicke in den Konturen. Normalerweise (d. h. in photorealistischen Darstellungen) würde man jetzt die Anzahl der Polygone erhöhen oder zu Freiformflächen übergehen, wodurch auch die Silhouette des Objektes den glatten Verlauf der Oberfläche widerspiegelt. Dies steigert aber den Berechnungsaufwand enorm.

Eine Liniengraphik bietet eine interessante Alternative, die *ohne* eine Abänderung des Modells auskommt. Wenn die einzelnen Kontursegmente nicht einzeln, sondern „in einem Zug“, also durch Interpolation der Endpunkte der Segmente, gezeichnet werden, entsteht ein Bild wie in Abbildung 4.4.

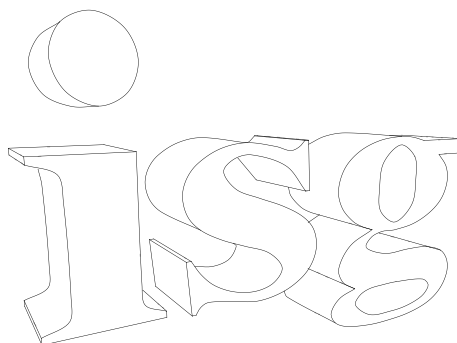


Abbildung 4.4: Extrahierte Kurven

Die Kurvenzüge charakterisieren ein gewölbtes Objekt besser als einzelne Geradensegmente, weil nicht irreführenderweise ein kantiges Objekt suggeriert wird. Sollte dagegen das Modell, das die Approximation des eigentlichen Objektes darstellt, visualisiert werden, wäre eine kantige Darstellung angemessen.

² An diesem Beispiel würden G-Buffer-Methoden komplett fehlschlagen: die im z -Buffer enthaltenen Informationen spiegeln nur die Winkelverhältnisse der Flächen wider und lassen keine Rückschlüsse auf eine eventuell erwünschte Glättung zu.

Die gewünschte Zusammenfassung der einzelnen Segmente zu Kurven ist allerdings algorithmisch bei weitem nicht so unproblematisch, wie es scheinen mag. Am Beispiel des i-Punktes, dessen zylindrische Form als achtseitiges Prisma mit geglätteten Mantelkanten repräsentiert ist, soll das verdeutlicht werden (Abbildung 4.5):

Von seinen insgesamt 42 Kanten bleiben nach Entfernung verdeckter und Triangulierungskanten fünfzehn sichtbare Kanten: acht bilden die Deckfläche, vier liegen auf dem Mantel, drei gehören zur verdeckten Rückseite (a). Ohne die beiden geglätteten inneren Kanten des Mantels findet man schnell drei zusammenhängende Ketten, die in zwei Punkten zusammentreffen (b).

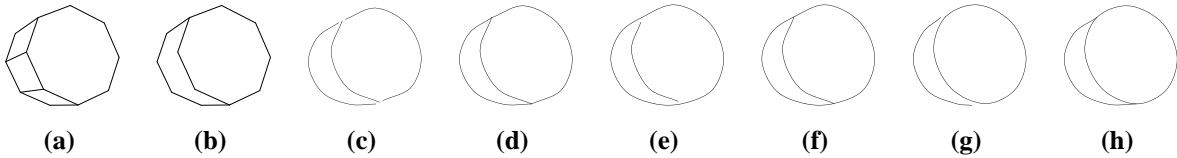


Abbildung 4.5: Kurvenextraktion. In (c, e, g) wurden die Kurven der besseren Unterscheidbarkeit wegen verkürzt dargestellt, (d, f, h) zeigen jeweils die realen Kurven.

Diese drei Ketten könnten jetzt einzeln mit Kurven gezeichnet werden (c, d). Das ist aber nicht das gewünschte Resultat, vielmehr soll eine geschlossene Kurve die Deckfläche darstellen, eine zweite den Rest. An Punkten, wo mehrere Segmente aufeinandertreffen, muß also eine sinnvolle Verkettung stattfinden (oder auch nicht, falls es eine Ecke ist).

Versuche, diese Aufgabe rein geometrisch zu lösen, führen nicht immer zum Erfolg, da sich Annahmen über die Stetigkeit der Fortsetzung von Linienzügen nicht allgemeingültig halten lassen. Eine stetige Fortsetzung würde im Beispiel die äußere Kontur zusammenfassen (e, f).

Zur Generierung zusammenhängender Kurvenzüge aus den sichtbaren Segmenten wurde eine neues Verfahren entwickelt, das keinerlei geometrische Informationen verwendet, sondern allein auf den im Modell gespeicherten Glättungsinformationen basiert. Die Ausgabe für das Beispielmmodell besteht in der gewünschten Zusammenfassung der Segmente zu einer geschlossenen und einer offenen Kurve (g, h).

Entwicklung des neuen Verfahrens

Das Grundgerüst des Algorithmus zur Kurvengenerierung ist recht einfach. Je zwei aneinandergrenzende sichtbare Segmente teilen sich den gemeinsamen Eckpunkt, so daß die Segmentverfolgung kein Problem darstellt. Es wird über „offene“ Enden der gefundenen Ketten Buch geführt, die Segmente werden in beliebiger Reihenfolge abgearbeitet und an die jeweils passende Kette angefügt. Die Punkte in jeder gefundenen Kette bilden die Stützpunkte der zu zeichnenden Kurven. Das genaue Verfahren ist im Implementierungskapitel, Abschnitt 5.5, beschrieben.

Das eigentliche Problem ist, an einem Kettenende zu entscheiden, ob es fortgesetzt werden soll, d. h. ob zwischen zwei aufeinanderfolgenden Segmenten eine glatte oder eckige Verbindung besteht. Der erste Ansatz, dies von der Winkeldifferenz zwischen den beiden Segmenten abhängig zu machen, wurde verworfen. Er führt nicht zu den gewünschten Ergebnissen (siehe Abbildung 4.5 (e, f)).

In einem zweiten Ansatz wurde versucht, aus der Glättungsinformation der sichtbaren Kanten ein Kriterium abzuleiten, das diese Entscheidung ermöglicht. Ein Kriterium wäre, an allen Stellen wo nur zwei Kanten aufeinandertreffen, eine glatte Fortsetzung zu erzeugen, und wenn mehr Kanten zusammenkommen, die geglätteten Kanten zu separieren. Dies führt am Beispiel des i-Punktes zum gewünschten Ergebnis, da hier an genau zwei Stellen mehr als zwei Kanten zusammenstoßen, von denen je eine die nicht zu verfolgende Mantelkante ist. Daher werden die Kanten der Deckfläche zusammengefaßt, es entsteht eine geschlossene Kurve.

Dieser Ansatz „rundet“ aber unzulässigerweise alle Ecken, von denen nur zwei Kanten sichtbar sind (wenn also eine Kante zu einer nach hinten zeigenden Fläche gehört). Anhand der Glättungsinformation der *sichtbaren* Segmente ist hier keine korrekte Entscheidung möglich. Die *unsichtbaren* Segmente müßten mit einbezogen werden.

Aufgrund dieser Betrachtungen wurde ein dritter Ansatz verfolgt, der letztendlich das gewünschte Resultat erzielte. Er geht von der Überlegung aus, daß eine Objektkontur so weit mit einer Kurve gezeichnet werden kann, bis eine *Ecke* erreicht ist.

Eine Ecke ist dabei definiert als ein Punkt, an dem mehr als zwei scharfe Kanten zusammenreffen. Ein Eckpunkt eines Würfels wird also als Ecke betrachtet, da drei scharfe Kanten von ihm ausgehen (Triangulierungskanten, so sie nicht schon entfernt sind, werden als geglättet angesehen). Dagegen hat das Modell eines Zylinders keine Ecken, da sich immer nur maximal zwei scharfe Kanten treffen. In einem Kugelmodell existieren nur geglättete Kanten, daher gibt es keine Ecken (Abbildung 4.6).

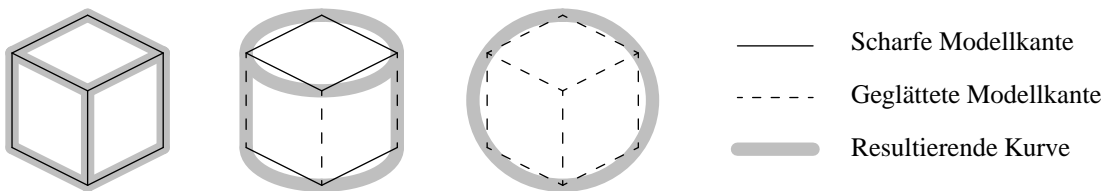


Abbildung 4.6: Unterschiedliche Glättungsinformationen (schematisch)

Das Verfahren läßt sich in zwei Phasen unterteilen:

1. Im Modell (vor der Entfernung verdeckter Kanten) werden alle Punkte als Ecke bzw. Nicht-Ecke klassifiziert. Dies geschieht in Abhängigkeit von der Anzahl der scharfen Kanten, die diesen Punkt teilen. Treffen weniger als drei solcher Kanten aufeinander, wird der Punkt nicht als Ecke betrachtet. Dieser Schritt ist nur einmal pro Modell, also unabhängig von der gewählten Ansicht, nötig.
2. Zur Kurvengenerierung aus den selektierten sichtbaren Segmenten werden diese, wie oben beschrieben, soweit verkettet, bis ein als Ecke klassifizierter Punkt auftritt oder kein anschließendes Segment mehr vorhanden ist. Die Fortsetzung der Kurve an einer Nicht-Ecke hängt von der Zahl und Art der involvierten Kanten ab. Sind hier nur zwei Kanten im Spiel, werden sie verkettet. Bei mehr als zwei Kanten werden zwei scharfe Kanten verbunden, geglättete Kanten enden. Mehr als je zwei scharfe bzw. geglättete Kanten können nicht auftreten³.

³ Mehr als zwei scharfe Kanten können nicht auftreten, da der Punkt sonst eine Ecke darstellte. Mehr als zwei geglättete Kanten ebensowenig, da diese nur selektiert werden, wenn sie am Objektrand liegen

Zusätzlich zu der automatischen Eckenklassifizierung im ersten Schritt wäre auch eine manuelle Eingabe denkbar. Die Punktklassifizierung müßte dazu als Zusatzinformation im Modell gespeichert werden. Mit einem speziellen Werkzeug könnte man diese Klassifikation nach Belieben ändern, um z. B. eine Ecke zu erzeugen, obwohl nur ungeglättete Kanten aufeinandertreffen. Ein Beispiel dafür fände sich etwa an der Spitze eines Kegels (siehe Abbildung 4.7).



Abbildung 4.7: Manuelle Eckenklassifikation

Zumindest ein potentielles Problem wurde bisher nicht erwähnt (für eine ausführlichere Betrachtung siehe Kapitel 7). Es wurden nur Punkte betrachtet, die im Modell enthalten waren. Diese konnten anhand der Glättungsinformation der Kanten klassifiziert werden.

Beim Prozeß der Bestimmung sichtbarer Kanten entstehen aber auch neue Punkte, nämlich an Verdeckungen. Auf der verdeckten Kante wird ein Punkt erzeugt. Die extrahierte Kurve endet hier. Ob auch auf der verdeckenden Kante ein Punkt generiert wird, hängt vom verwendeten Hidden-Line-Removal-Algorithmus und den gewünschten Segmentattributen ab (siehe Abschnitt 4.2.2). Dieser Punkt sollte nicht als Kurvenstützpunkt verwendet werden, da er im Gegensatz zu den beiden Endpunkten der Kante eventuell nicht auf der interpolierenden Kurve liegt, und somit eine sichtbare Verformung der generierten Kurve zur Folge hätte (siehe Abbildung 4.8).

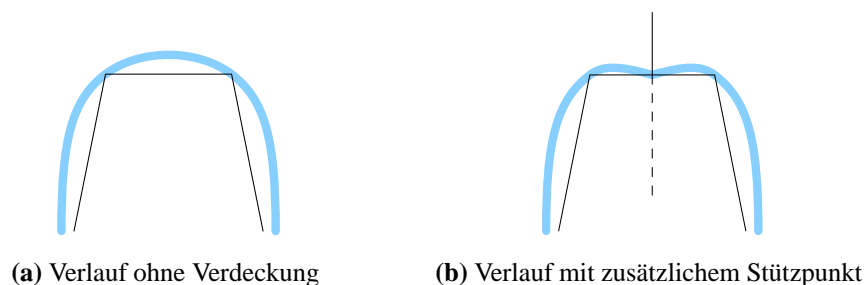


Abbildung 4.8: Verformung der Kurve

4.2 Variation der Linienattribute

Wie die Beispiele in Kapitel 2 gezeigt haben, kommt der Variation der Darstellung der Linien eine besondere Rolle zu. Die Breite einer Linie kann beispielsweise dazu verwendet werden, Licht- und Schattenverhältnisse zu klären (und so der Graphik „Tiefe“ zu geben), Objekte besser voneinander unterscheidbar zu machen oder besonders zu betonen. Die Arbeiten zum

Sketchrenderer zeigten, daß auch Genauigkeit und Anzahl der zur Darstellung verwendeten Striche einen großen Einfluß auf eine graphische Präsentation haben.

Eine entsprechende Umsetzung wird im Rahmen dieser Arbeit durch die Verwendung des parametrisierbaren Linienmodells aus [Sch97] ermöglicht. Im Gegensatz zum im Sketchrenderer verwendeten Linienmodell [Sch92] ist bei diesem nicht nur eine Attributierung des Stils, sondern auch des Pfades möglich. Beim Zeichnen des Pfades mit einem Stil werden die in beiden gespeicherten Attribute miteinander gemischt und das Resultat dargestellt. Die Linienbreite beispielsweise hängt sowohl vom Pfad als auch vom verwendeten Stil ab. Es können längere oder kürzere, genauere oder unschärfere Striche Verwendung finden. Auch die Helligkeit einer Linie kann beeinflußt werden.

Im folgenden werden einige Beispiele für derartige Attributierungen vorgestellt. Dabei wird meist die Linienbreite beeinflußt, da dies für den angestrebten Stil der Schwarz-Weiß-Liniengraphiken am besten geeignet erscheint. Es wäre aber auch ohne weiteres möglich, statt der Breite beispielsweise die Helligkeit der Linie zu verändern.

4.2.1 Lichteinfluß

In mehreren Beispielen war die Variation der Linienbreite in Abhängigkeit von der Beleuchtung zu sehen (vgl. Abbildungen 2.2, 2.4, 2.9(a) auf Seite 8 ff.). Dabei stehen breitere Linien für dunklere Gebiete, dünnere für helle. In besonders hellen Teilen wurden Linien sogar durchbrochen dargestellt.

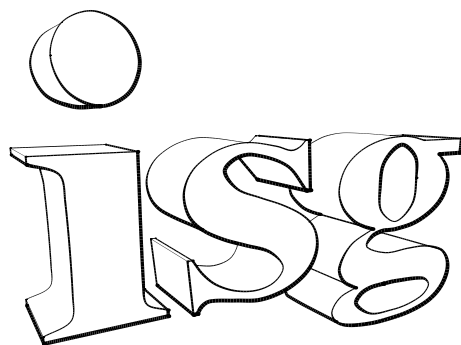


Abbildung 4.9: Lichtabhängige Linienbreite

Eine Umsetzung zeigt Abbildung 4.9. Hier wurde ein Beleuchtungsmodell mit unendlich weit entfernter Lichtquelle verwendet und für jede Kante in Abhängigkeit von einem Normalenvektor ausgewertet, der aus den Normalen der angrenzenden Polygone gemittelt wurde. Dies erbrachte bessere Resultate als die Bestimmung der Helligkeit aus Normalen in den Eckpunkten, wie es in der flächigen Computergraphik üblich ist (dort werden allerdings auch die Flächen statt der Kanten dargestellt).

Obwohl die Darstellung in der Abbildung unter Annahme einer sich links oben hinten befindenden Lichtquelle mathematisch korrekt ist, erscheint sie leicht unnatürlich. Das liegt daran, daß man aus handgezeichneten Bildern eher Darstellungen gewohnt ist, die das Objekt im Ganzen behandeln. Die Linienstärke nimmt dann kontinuierlich von der dem Licht

zugewandten zur abgewandten Seite zu. Mit einem Beleuchtungsmodell, das die Entfernung zur Lichtquelle berücksichtigt, kann dieser Effekt nachgebildet werden.

Möglicherweise könnte eine globale Beleuchtungssimulation gute Ergebnisse liefern. Die Helligkeiten würden in einem Vorverarbeitungsschritt ermittelt. Beim Zeichnen könnte ohne weitere Berechnung direkt auf diese gespeicherte Helligkeitsinformation zugegriffen werden.

4.2.2 Silhouettenbetonung

Wie beispielsweise in Abbildung 2.9(c) auf Seite 13 zu sehen ist, wird manchmal eine Technik verwendet, die die äußeren Konturen von Objekten gegenüber innen liegenden Kanten betont. Dadurch wird die Abgrenzung der Objekte voneinander und der Zusammenhalt jedes einzelnen Objekts hervorgehoben. Die derartig variierte Version des Beispielmodells zeigt Abbildung 4.10.

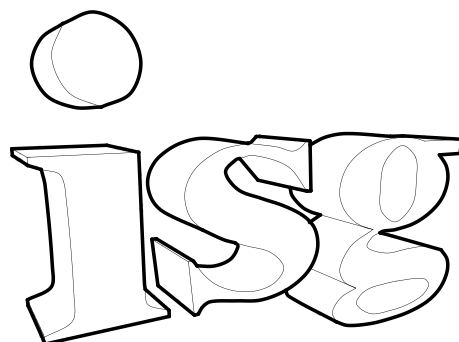


Abbildung 4.10: Hervorgehobene Silhouetten

Zur Erzeugung dieser Betonungen sind an den Segmenten zweierlei Informationen vonnöten. Zum einen ist eine Unterscheidung in innere und Randkanten erforderlich. Innere Kanten werden dabei von zwei sichtbaren Polygonen geteilt (*shared edges*), Randkanten gehören nur zu einer sichtbaren Fläche. Diese Unterscheidung sagt nur aus, ob an einer Kante eine Verdeckung auftritt oder nicht. Sie reicht deshalb nicht zur Hervorhebung der Silhouette aus. Wird allerdings von einer Randkante eine Fläche desselben Objektes verdeckt, zu dem die Kante gehört, gehört diese Kante nicht zur Silhouette. Diese wird nur von jenen Kanten gebildet, die andere Objekte bzw. den Hintergrund verdecken. Die zweite benötigte Unterscheidung ist also eine Klassifizierung der Randkanten in Kanten der Silhouette und solche, die eine Selbstverdeckung darstellen. Diese zweite Klassifizierung erzwingt die Unterteilung einer Randkante in zwei Segmente (ein Rand- und ein Silhouettensegment), falls diese eine Silhouettenkante verdeckt, wie in Abschnitt 4.1.3 erwähnt wurde.

Ein Problem tritt in Abbildung 4.10 zu Tage. Da die verwendeten Linienstile bisher keine harten Übergänge der Breite erlauben, mußte für diese Darstellung das Bild in einzelne Kurvenzüge aufgelöst werden (z. B. am i-Punkt sichtbar). Eine bessere Lösung wäre die Einführung „harter“ Attributwechsel in das Linienmodell. Für die Pfadgeometrie könnte dann die übliche Interpolation verwendet werden.

4.2.3 Wichtigkeit

Die Fokussierung von Bildbereichen durch Beeinflussung der Liniendarstellung ist relativ einfach möglich. Dies kann auf zwei Arten realisiert werden. Zum einen können die Objekte je nach Wichtigkeit mit unterschiedliche Stilen gezeichnet werden. Dabei werden ganze Objekte hervorgehoben (Abbildung 4.11).

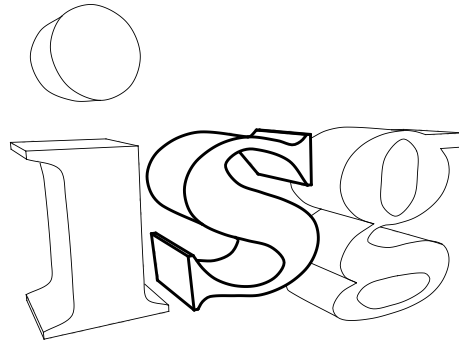


Abbildung 4.11: Hervorgehobenes Objekt

Zum zweiten können natürlich auch die Attribute des Pfades beeinflusst werden. Damit kann ein kontinuierlicher Übergang von wichtigen zu unwichtigen Bereichen visualisiert werden. Dies läßt sich über eine Kodierung der Entfernung zum Point-of-Interest in die Linienbreite erreichen. Dabei ist sowohl eine Bestimmung der Entfernung im Bild- als auch im Objektraum denkbar. Die dreidimensionale Entfernungsbestimmung scheint sinnvoller, da im (zweidimensionalen) Bild eng benachbart dargestellte Objekte nicht notwendigerweise räumlich nah beieinander liegen müssen.

Noch wirkungsvoller wäre eine derartige Fokussierung, wenn sie mit einer Modifizierung des Detaillierungsgrades der nicht im Mittelpunkt des Interesses stehenden Objekte einhergehen würde. Derartiges ist in vielen Handzeichnungen anzutreffen. Dies stellt sicher einen Forschungsschwerpunkt für weitergehende Arbeiten dar.

4.2.4 Depthcueing

Eine in der Computergraphik seit langem verwendete Möglichkeit der Visualisierung der Tiefenverhältnisse in einer Liniengraphik ist das Depthcueing. Dabei wird typischerweise die Entfernung einer Linie zum Betrachter durch die Änderung der Helligkeit der Linie dargestellt. In der Malerei spricht man von der „athmosphärischen Perspektive“ (die Farben weiter entfernter Objekte gehen ins Bläuliche über, der Kontrast nimmt ab). In Zeichnungen trifft man eher auf die im letzten Abschnitt erwähnte Möglichkeit der Detailreduktion.

Die Möglichkeit zur Anwendung des Depthcueings besteht auch mit dem hier vorgestellten System. Abbildung 4.12 demonstriert die Umsetzung der Tiefeninformation jedes Modellpunktes in die Linienbreite.

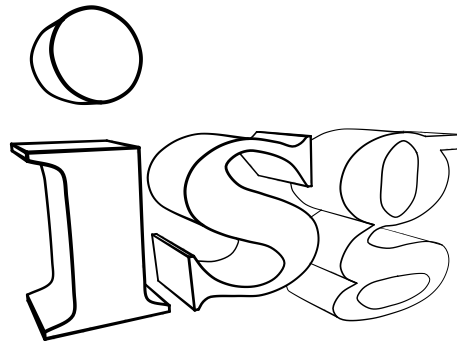


Abbildung 4.12: Entfernungsabhängige Linienbreite

4.2.5 Verallgemeinerung

Die erläuterten Methoden zur Charakterisierung von Linien durch Änderung der Zeichenattribute sind nicht voneinander separiert, sondern es lassen sich Gemeinsamkeiten ausmachen. Diese Gemeinsamkeiten ergeben einen guten Rahmen für die Änder- und Erweiterbarkeit des Systems.

Prinzipiell werden im Modell vorhandene oder sich aus der Sicht auf das Modell ergebende Parameter in bestimmter Art und Weise in Linienattribute überführt. Damit gibt es drei Variablen: der zu visualisierende Parameter, das beeinflusste Attribut und die Übertragungsfunktion, die beide ineinander überführt.

In der Herkunft der Parameter gibt es Unterschiede: sie sind entweder segment- oder punktgebunden, d. h. ein Parameter ist über die Länge eines Segments konstant (wie z. B. die Eigenschaft, Silhouettensegment zu sein) oder er ist an den Modellpunkten definiert und wird über den Verlauf des Segments interpoliert (wie die Tiefeninformation).

Das abhängige Linienattribut kann frei aus der Menge der dem verwendeten Linienmodell eigenen Attribute gewählt werden. Im Rahmen dieser Arbeit wurde hauptsächlich die Linienbreite modifiziert.

Die Übertragungsfunktion bildet das Intervall der möglichen Parameterwerte in das Intervall der möglichen Attributwerte ab. Als günstig hat es sich erwiesen, die Funktion so auszulegen, daß sie definierte Grenzwerte hat, also z. B. eine minimale und maximale Linienbreite einzustellen erlaubt. Das Parameterintervall muß nicht notwendigerweise im gesamten Bild konstant sein, es ist möglich, dieses dynamisch für jedes Objekt aus der Menge der tatsächlich vorkommenden Parameterwerte zu bestimmen.

4.3 Variation der Endbedingungen einer Linie

Sehr wirkungsvoll ist in den betrachteten traditionellen Illustrationen der Einsatz zweier Techniken, um räumlich hintereinanderliegende Linien an Verdeckungen hervorzuheben. Zum einen wird dabei eine Lücke in der verdeckten Kante gelassen (bzw. nachträglich ausgeradiert), zum anderen ist der Einsatz eines kleinen schwarzen Fülldreiecks zu beobachten.

Letzteres kommt an Verbindungen zum Einsatz, an denen das weiter vorn liegende Objekt verdeckte Linien abschattet.

4.3.1 Lücken

Die Lücke am Ende einer verdeckten Linie wird dadurch erzeugt, daß diese Linie nicht bis ganz an die verdeckende Kontur gezeichnet wird, oder daß nachträglich die Lücke durch Erzeugen einer Trennlinie (z. B. mit einem Messer oder weißer Farbe) entsteht.

Für die Breite der Lücke konnte keine durchgehende Erklärung gefunden werden. Diese ist nicht einmal innerhalb einer Zeichnung konstant oder vom Entfernungsunterschied abhängig (wie anfänglich vermutet wurde).

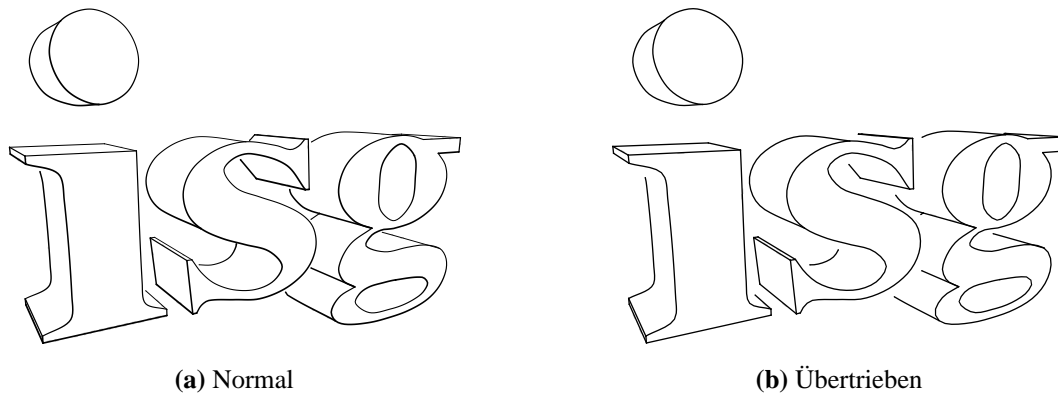


Abbildung 4.13: Lücken durch Linienkürzung

Ein durch Kürzen der verdeckten Linien erzeugtes Bild zeigt Abbildung 4.13. Die Größe der Lücke ist in diesem Bild konstant und wurde direkt als Betrag der Länge verwendet, um den die Linien gekürzt wurden. Statt einer unveränderlichen Lückenbreite könnten auch Werte verwendet werden, die beispielsweise abhängig vom Segmenttyp oder der Tiefendifferenz der Segmente sind.

In [Elb95] wird darauf hingewiesen, daß der Winkel zwischen den beiden Segmenten beachtet werden sollte, um die Lücken gleich groß erscheinen zu lassen. Das läßt sich zwar leicht in die Berechnung des Linienkürzungsbetrages einbeziehen, hat aber einen entscheidenden Nachteil. Wenn der Winkel sehr klein wird, also eine Linie sehr flach von einer Kontur verdeckt wird, wird mit dieser Methode die Lücke so groß, daß der Linienendpunkt sehr weit vom eigentlichen Verdeckungspunkt entfernt liegt. Darunter leidet der Gesamteindruck der Verdeckungsstelle.

Auch das zweite Verfahren, die nachträgliche Erzeugung der Trennlinie, wurde untersucht. Dafür wurde, nach dem Zeichnen aller Linien am äußeren Rand von Konturlinien die eine andere Linie verdecken, eine weiße Linie gezogen (siehe Abbildung 4.14, die grau hinterlegt wurde, um die Trennlinien sichtbar zu machen). Auch hier muß der wirkliche Verlauf der dargestellten Linie berücksichtigt werden, daher muß das verwendete Linienmodell diese Art der zusätzlichen Konturierung unterstützen. Da dies noch nicht der Fall ist, wurde für diese Abbildung eine provisorische Konturierung implementiert (was die Artefakte im Bild erklärt).

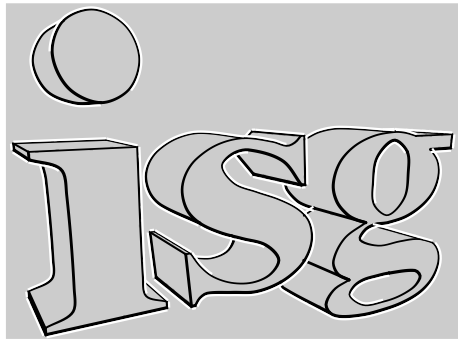


Abbildung 4.14: Lücken durch Trennlinien

Für die hier betrachteten Konturzeichnungen liefert die erste Variante ausreichende Ergebnisse. Die zweite Variante ist rechentechnisch aufwendiger. Außerdem erzeugt das gerade Abschneiden von Linien bei ihr steril wirkende Linienenden.

4.3.2 Fülldreiecke

Das Füllen der Verbindung zweier Linien mit einem kleinen Dreieck, nach einem seiner Verfechter auch „Snodgrassing“ genannt [Hod89], stellt eine interessante und effektvolle Methode der Darstellung von Linienenden an Verdeckungen dar. In der Computergraphik wurde es schon in [DC90] eingesetzt, allerdings nur in Form einer Linienverbreiterung im Verdeckungsfall.

In den in Kapitel 2 betrachteten Beispielen kam das Dreieck nur dann zum Einsatz, wenn die Tiefendifferenz der beiden Linien relativ gering war. Möglicherweise suggeriert das Dreieck einen kleinen, auf die verdeckte Linie fallenden Schatten, der diese am Ende verbreitert.

Die Größe des Dreiecks ist dabei im gesamten Bild ungefähr gleich, seine Basis entspricht ungefähr der drei- bis fünffachen Linienstärke. Die Kanten schließen glatt an die beiden Liniensegmente an. Die Form des Fülldreieckes hängt auch von der Art der verdeckten Linie ab. Es führt niemals aus dem Objekt heraus (siehe Abbildung 2.9(a) auf Seite 13), d. h. bei einer Konturlinie kommt, im Gegensatz zu einer inneren Linie, nur ein „halbes“ Dreieck auf einer Seite der Linie zum Einsatz.

Die für die Erzeugung des Dreiecks nötigen Informationen sind vorhanden. Es sind zwei Kanten beteiligt, die verdeckende und die verdeckte Kante. Zur Darstellung der Fülldreiecke wurde ein Verfahren entworfen, das im folgenden kurz beschrieben wird. Da das Linienwerkzeug dieses Verfahren noch nicht unterstützt, konnten bisher keine Beispielbilder erstellt werden.

Das Verfahren soll eine möglichst freie Parametrisierbarkeit des Dreiecks zulassen. Dabei sollen Form und Größe beeinflussbar sein. Das Fülldreieck wird durch zwei Teildreiecke erzeugt. Wenn die verdeckte Linie eine Konturlinie ist, wird das äußere Halbdreieck weggelassen. Um einen glatten Übergang zu erreichen, wird eine Seite eines Teildreiecks durch eine kubische Beziér-Kurve erzeugt, die beiden anderen liegen an den Konturen der Linie an (siehe Abbildung 4.15(a)). Die Länge L der beiden geraden Schenkel des Dreiecks (für einen Schenkel in der Abbildung eingezeichnet) bestimmt seine Größe. Mit ihr läßt sich auch eine

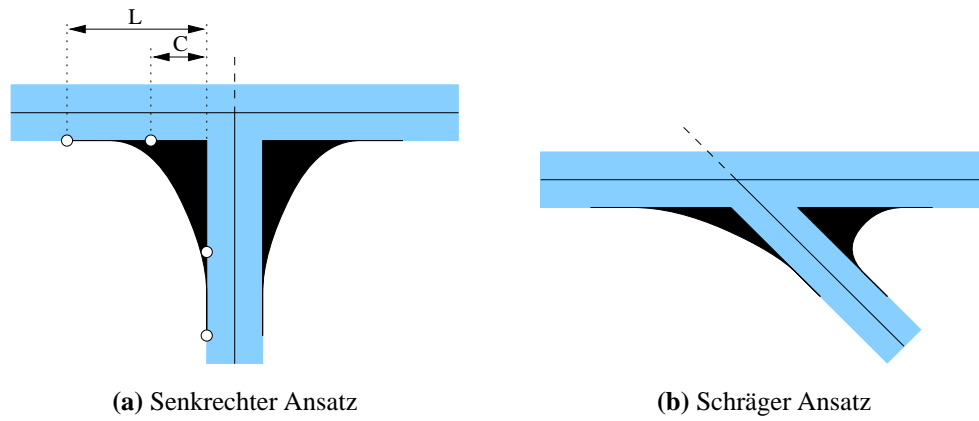


Abbildung 4.15: Fülldreiecke

gestreckte oder gestauchte Form wählen. Der Parameter C , eine Zahl zwischen 0 und 1, gibt das Teilungsverhältnis des jeweiligen Schenkels an, an dem der innere Punkt der Bezierkurve liegt. Mit ihm kann die Form des Dreiecks variiert werden. In Abbildung 4.15(b) ist zu erkennen, daß auch bei asymmetrischer Teilung eine ansprechende Form erzielbar ist.

5 Implementierung

In diesem Kapitel werden einige Details der Implementierung der im letzten Kapitel vorgestellten Konzepte besprochen. Dazu gehört auch eine kurze Vorstellung der verwendeten Komponenten und Werkzeuge, die in der Arbeitsgruppe „Graphik“ des ISG entstanden.

Die Implementierung erfolgte vollständig in VisualWorks, einer auf vielen Plattformen (u. a. PCs unter Windows NT/95, Sun Workstations unter Solaris, SGI Workstations unter Irix) verfügbaren Smalltalk-Variante. Außer dem Vorteil der Plattformunabhängigkeit bietet es eine komfortable Entwicklungsumgebung und ermöglicht die vergleichsweise einfache Integration der unter verschiedener Autorenschaft entstandenen Komponenten.

Der Aufbau des Kapitels orientiert sich an der im folgenden Abschnitt kurz vorgestellten Rendering-Pipeline. Danach werden einzelne Aspekte unterschiedlich detailliert beleuchtet.

5.1 Überblick

Abbildung 5.1 illustriert den groben Ablauf der Erzeugung einer liniengraphischen Darstellung aus einem 3D-Modell.

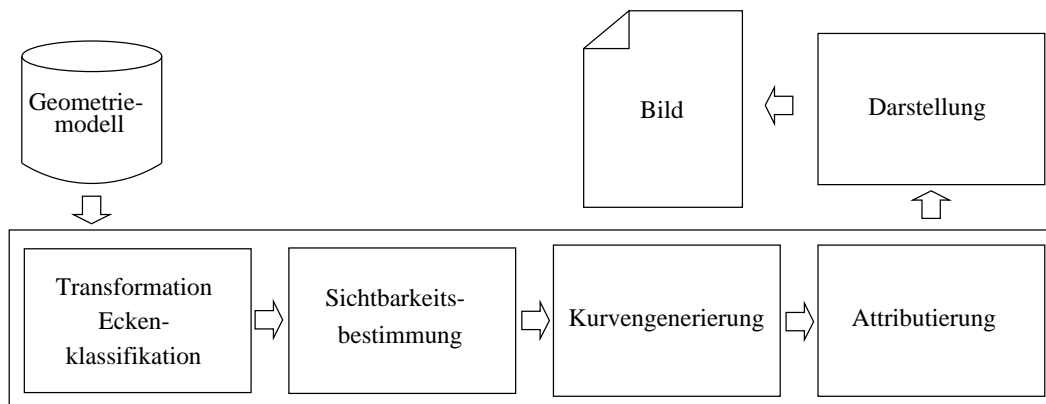


Abbildung 5.1: Die Rendering-Pipeline

Der erste Schritt ist das Laden des Geometriemodells in die interne Repräsentation. Es folgt die Transformation des Modells in die für den analytischen Renderer notwendigen Datenstrukturen. Dabei erfolgt die Projektion der Eckpunkte entsprechend der gewählten Kamera, die Erzeugung von Kanten und die Klassifikation der Punkte in Ecken und Nicht-Ecken. Als

nächstes findet die Ermittlung sichtbarer Kanten und Flächen statt. Dabei werden zusätzliche Informationen in den erzeugten Segmenten gespeichert. Mit Hilfe dieser Informationen erfolgt anschließend die Auswahl der darzustellenden Segmente, die dann zu Kurvenzügen verkettet werden. Der letzte Schritt vor der Linienübergabe an das Zeichenwerkzeug besteht in der Attributierung der Kurven.

5.2 Das Modell

Den Ausgangspunkt des Renderings bildet das Geometriemodell. Es repräsentiert die geometrische (räumliche) Struktur des darzustellenden Gegenstandes. In dieser Arbeit werden polygonale Modelle betrachtet. Sie sind insofern universell, als daß sich andere Repräsentationsformen (wie Freiformflächen, Rotationskörper, implizite Oberflächen usw.) relativ einfach in polygonale Strukturen überführen lassen.

Das Modell wird aus einer Datei in eine interne Datenstruktur eingelesen. Die Datei wird mit einem externen Modellierungswerkzeug erstellt.

Für die interne Modellrepräsentation wurde eine in der Arbeitsgruppe entwickelte Klassenbibliothek genutzt. Eine Erweiterung dieser Strukturen betrifft das zusätzliche Einlesen der Glättungsinformation aus einer 3DS-Datei¹ und deren interne Speicherung und Verwertung.

Für das Einlesen von Geometriedateien stehen mehrere Parserklassen (pro Dateiformat eine) bereit. Diese lesen eine Datei ein und liefern eine Menge von attribuierten Geometrieobjekten zurück. Im Rahmen dieser Arbeit wurde der 3DS-Dateien lesende `ThreeDSParser` so erweitert, daß er auch die (bisher unberücksichtigten) Glättungsinformationen einliest.

Das 3DS-Format ist ein hierarchisches, binäres Format. Es enthält einzelne *Chunks* (Datenblöcke), die wiederum Subchunks enthalten können. Ein Chunk wird durch einen Header eingeleitet. Dieser enthält eine den Chunktyp identifizierende ID und die Länge des Chunks. Dadurch ist es einem Parser möglich, unbekannte Chunks zu überspringen.

Die Glättungsinformationen sind in einem speziellen Chunk in Form eines 32 Bit Integers pro Dreieck eines Objektes (im 3DS-Format sind alle Flächen trianguliert) enthalten. Die einzelnen Bits korrespondieren mit sogenannten *smoothing groups*, d. h. ein gesetztes Bit zeigt an, daß das Dreieck zu einer bestimmten Glättungsgruppe gehört. Beim interpolierenden Shading wird zwischen genau den aneinandergrenzenden Flächen interpoliert, die mindestens einer gemeinsamen Gruppe angehören.

Der `ThreeDSParser` erzeugte ursprünglich Instanzen der Klasse `GeoVertexFace` (die eine durch eine Sequenz von Punkten definierte Fläche repräsentiert). Um zusätzlich den eingelesenen Glättungsparameter zu speichern, wurde von dieser die neue Klasse `GeoSmoothedVertexFace` abgeleitet. Diese Klasse definiert Methoden zum Zugriff auf diesen Parameter, ebenso wie zum Testen der Glättung zwischen zwei Flächen. Dieser Test erfolgt in der Methode „`isSmoothTo: anotherSmoothedVertexFace`“ und ist erfolgreich, wenn die

¹ Dies ist das von 3D-Studio, einem weit verbreiteten Modellierungs- und Animationssystem auf PC-Basis, verwendete Dateiformat. Es enthält, im Gegensatz zu vielen anderen Formaten, die benötigten Glättungsinformationen.

bitweise UND-Verknüpfung der beiden Glättungsparameter ungleich Null ist ($smoothing_1 \wedge^{bit} smoothing_2 \neq 0$).

5.3 Transformation des Modells und Eckenklassifikation

Zur Anwendung des Renderalgorithmus (siehe Abschnitt 5.4) werden andere, speziell auf diesen Renderer zugeschnittene Datenstrukturen benötigt. Das betrifft hauptsächlich die Klasse `SGVertex`, die eine Liste von an diesem Punkt zusammentreffenden Kanten hält, sowie die Klasse `SGInputEdge`, die diese Kanten repräsentiert.

Da zur Klassifikation der Vertices in Ecken/Nicht-Ecken auch die verdeckten Flächen nötig sind, muß dieser Schritt vor der Entfernung verdeckter Kanten und Flächen erfolgen. Auch das Entfernen nicht sichtbarer Rückseiten kann erst nach der Eckenklassifikation erfolgen. Daher werden zunächst für alle im Modell vorhandenen Flächen Kanten und Punkte erzeugt.

Die Speicherung der Kanten in einem `SGVertex` vereinfacht die Eckenklassifikation. Eine Kante wird dabei als scharf angesehen, wenn ihre beiden angrenzenden Flächen nicht *smooth* zueinander sind. Das Verfahren erläutert der folgende Pseudocode.

```
foreach Objekt in Modell do
  foreach Vertex in Objekt do
    Summe := 0
    foreach Kante an Vertex do
      if Kante ist scharf then Summe := Summe + 1 fi
    od
    if Summe > 2 then Vertex sei Ecke fi
  od
od
```

Aus Gründen der Effizienz erfolgt die Speicherung der Eckenklassifikation nicht in dem Vertex selbst, sondern in den anschließenden Kanten. Diese halten ein Flagfeld, das noch ungenutzte Bits enthält. Eine Kante kann per `isSmoothAtStart` bzw. `isSmoothAtEnd` nach der Klassifikation ihres Start- bzw. Endpunktes befragt werden.

5.4 Bestimmung sichtbarer Kanten und Flächen

Der im Vorfeld der Diplomarbeit implementierte SECHREST/GREENBERG-Algorithmus [SG81] zur Bestimmung sichtbarer Kanten wurde dahingehend erweitert, daß er zusätzliche Informationen in den sichtbaren Segmenten speichert.

Seine hohe Effizienz bezieht dieser Algorithmus aus der Nutzung von Kohärenzeigenschaften des Modells. So wird die Sichtbarkeit von Kanten an die benachbarten weitergegeben. Selbst an Schnittpunkten kann meist ohne Tiefenvergleich die Sichtbarkeit bestimmt werden. Ein großer Nachteil ist jedoch, daß er einander durchdringende Flächen ausschließt. Dieses

Problem kann durch eine Vorverarbeitung des Modells beseitigt werden, die ein durchdringungsfreies Modell erzeugt. Der Algorithmus wurde dahingehend erweitert, daß viele während des Renderns anfallende Informationen gespeichert werden. Das ermöglicht später eine differenzierte Liniendarstellung.

Wie von anderen Scan-Line-Algorithmen (Verfahren, die ein Bild zeilenweise generieren) bekannt, verwendet der SECHREST/GREENBERG-Algorithmus eine *Active Edge Table* (AET, Liste aktiver Kanten). Diese wird aber nicht in Pixelschritten aktualisiert, sondern wenn bestimmte Ereignisse (*Events*) auftreten. Das können Vertex-Events (Punkte, an denen Kanten beginnen oder enden) oder Crossing-Events (Punkte, an denen sich Kanten schneiden) sein. Die projizierte Szene wird von unten nach oben durchlaufen, an einem Vertex-Event werden die hier endenden sichtbaren Kanten ausgegeben (d. h. in die Ausgabeliste eingetragen). Die an diesem Vertex beginnenden Kanten ersetzen die hier endenden Kanten in der AET. Sie übernehmen deren Sichtbarkeit. An einem Crossing-Event wird entschieden, ob dadurch eine Kante verdeckt oder sichtbar wird. Die beiden Kanten werden in der AET vertauscht. Der sichtbare Teil einer unsichtbar gewordenen Kante wird ausgegeben.

Anders als im originalen Algorithmus wird auch die verdeckende Kante am Kreuzungspunkt in zwei sichtbare Segmente geteilt, da ab dieser Stelle eine andere Fläche verdeckt wird (das ermöglicht beispielsweise eine Unterscheidung in Silhouettenkanten und solche, an denen das Objekt selbst verdeckt wird).

Für die Rekonstruktion der Flächen werden im Verlauf des Algorithmus verkettete Ausgabe-Vertices erzeugt. Dies geschieht an allen Stellen, wo zwei sichtbare Segmente einen Winkel einschließen, d. h. wo eine Ecke eines Ausgabepolygons erzeugt wird. Um das bei anderen Algorithmen notwendige aufwendige Zuordnen von Löchern zu Konturen zu vermeiden, werden diese durch einen speziellen Kantentyp mit der Kontur verbunden. Mit diesen Informationen ist im Anschluß eine einfache Zusammenfassung der Ausgabe-Vertices zu Polygonen möglich.

Das Ergebnis des Verfahrens besteht in einer nach Objekten sortierten Liste von mit Zusatzinformationen versehenen sichtbaren Kanten und Flächen. Die Flächen können konkav sein und Löcher haben.

Die erzeugten sichtbaren Kanten sind von der Klasse `VisibleLine`, sie referenzieren ihren Start- und Endpunkt sowie ein Flagfeld, das ihren Typ festlegt. Kantentypen sind rechte, linke oder innere Kante (die Kanten sind gerichtet). Diese werden in 2 Bits mit den Masken `LeftMask` und `RightMask` kodiert. Diese Typangabe bezieht sich auf die Anzahl der Flächen nach der Rückseiten-Entfernung. Das Flag mit der Maske `SmoothMask` gibt an, ob das Segment von einer glatten oder scharfen Kante stammt. Ein Silhouettensegment, also eine rechte oder linke Kante, die keine Selbstverdeckung darstellt, hat das der `SilhouetteMask` entsprechende Flag gesetzt. Die Flags `ContStartMask` und `ContEndMask` bezeichnen solche Segmente, die durch das Aufteilen einer Kante an einem Schnittpunkt entstanden, deren Ende also fortgesetzt (*continued*) ist. Die die Eckenklassifikation speichernden Flags der `SmoothStartMask` und `SmoothEndMask` wurden schon erwähnt.

5.5 Kurvengenerierung

Aus den wie eben beschrieben gewonnenen sichtbaren Kantensegmenten sollen Kurven generiert werden. Dazu erfolgt erst die Auswahl der darzustellenden Segmente, dann ihre Verkettung und danach die Erzeugung von Kurvenstützpunkten.

Die Segmentselektion erfolgt anhand der Flags der `VisibleLines`. In Abschnitt 4.1.2, Seite 27 wurde die Selektionsbedingung erläutert, sie läßt sich so auf eine einfache Formel bringen:

Selektiere Segment \iff *Segment ist nicht inneres* \vee *Segment ist scharf*

Aus den ausgewählten Segmenten werden Ketten gebildet, die später „in einem Strich“ dargestellt werden sollen. Das Ende (bzw. der Anfang) eines Segment wird als „offen“ angesehen, wenn an dieses noch ein anderes Segment „paßt“, also die Kette an dieser Stelle noch nicht zu Ende ist. Sinngemäß gilt das auch für ganze Ketten: ist der Anfang des ersten Segmentes bzw. das Ende des letzten offen, ist die Kette an dieser Stelle offen. Diese Einteilung in offene und geschlossene Enden aus der Eckenklassifikation ist das eigentlich Neue an diesem Verfahren. Es fließen aber auch noch andere Parameter mit in diese Entscheidung ein. So wird zum Beispiel ein geteiltes Segment (das die `ContEndMask` gesetzt hat), garantiert am Ende offen sein.

Die Verkettung erfolgt getrennt für jedes Objekt. Es wird eine Liste mit offenen Kettenenden geführt. Die Enden selbst dienen dabei als Schlüssel eines `Dictionary`. Da ein `Dictionary` als Hash-Tabelle implementiert ist, ist die Suche nach freien Enden in dieser Liste ausreichend effizient. Der Algorithmus nimmt immer ein Segment aus der Liste sichtbarer Segmente und fügt es an eine Kette aus der Liste offener Ketten an. Wird dabei eine Kette abgeschlossen, wird diese ausgegeben. Das genaue Verfahren ist dem folgenden Pseudocode zu entnehmen.

```
proc Verkette_Segmente
  begin
    Liste der offenen Ketten := leere Liste
    foreach Sichtbares Segment do
      bestimme ob Segmentenden offen sind
      if beide Segmentenden abgeschlossen
        then gib Segment als Kette aus
      elsif beide Segmentenden offen
        then Behandle_beidseitig_offenes_Segment
        else Behandle_einseitig_offenes_Segment
      fi
    od
  end.
```

Die Behandlung der beiden offenen Segmenttypen geben die folgenden Fragmente wieder.

```

proc Behandle_einseitig_offenes_Segment
  begin
    if offen am Ende then drehe Segment fi
    suche offenen Segmentanfang in Kettenliste
    if gefunden
      then verkette Segment mit Kette
        if Kette ist jetzt abgeschlossen
          then gib Kette aus
        fi
      else erstelle neue Kette aus Segment
        setze Kette in Liste
      fi
    end.

proc Behandle_beidseitig_offenes_Segment
  begin
    suche offene Enden in Kettenliste
    if kein offenes Kettenende gefunden
      then erstelle neue Kette aus dem Segment
        setze Kette in Liste
      elsif ein offenes Kettenende gefunden
        then nimm Kette aus Liste
          verkette Segment mit Kette
          setze Kette in Liste
        else zwei offene Kettenenden gefunden
          if beide Enden sind in der gleichen Kette
            then verbinde Kette mit Segment zu Ring
              gib Kette aus
            else nimm Ketten aus Liste
              verbinde Ketten mit Segment
              if Kette ist jetzt abgeschlossen
                then gib Kette aus
              fi
            fi
          fi
        fi
      end.

```

Nach der Erzeugung der Ketten erfolgt die Umwandlung in Kurvenstützpunkte. Dies geht mit deren Attributierung einher (siehe nächster Abschnitt). Bei der Umwandlung müssen zyklische und nicht-zyklische Ketten unterschieden werden. Auch ist zu berücksichtigen, daß die durch Segmentteilung entstandenen Vertices nicht als Stützpunkte verwendet werden. Dies veranschaulicht der folgende Pseudocode.


```

proc Wandle_Kette_in_Kontrollpunkte
  begin
    if Kette ist nichtzyklisch
      then Erzeuge Kontrollpunkt aus Startpunkt des ersten Segments
    fi
    foreach Segment in Kette do
      if Segmentende ist nicht durch Teilung entstanden
        then Erzeuge Kontrollpunkt aus Endpunkt des Segments
      fi
    od
  end.

```

5.6 Attributierung

Das im letzten Abschnitt angesprochene Erzeugen der Kontrollpunkte aus den Start- bzw. Endpunkten von Segmenten ist die Stelle, an der die Attributierung der Kontrollpunkte erfolgt. Diese Attribute verwendet das im nächsten Abschnitt vorgestellte Zeichenwerkzeug, um die Erscheinung der dargestellten Kurve zu modifizieren.

Die Umwandlung von Segmentparametern in Linienattribute ist vergleichsweise einfach. Es muß nur eine angemessene Funktion zur Übertragung gefunden werden. Eine angemessene Skalierung der Linienbreite ist dagegen schwieriger.

Gelöst wurde das Problem durch die Einführung einer minimalen und maximalen Linienbreite für das Bild. Dafür werden in einem ersten Schritt die minimalen und maximalen zu visualisierenden Parameterwerte des Bildes (oder des Objektes) gesammelt. Die letztendlich verwendete Linienbreite ergibt sich im zweiten Durchlauf aus

$$\text{Linienbreite} = \text{Breite}_{\min} + \frac{\text{Parameter} - \text{Parameter}_{\min}}{\text{Parameter}_{\max} - \text{Parameter}_{\min}} (\text{Breite}_{\max} - \text{Breite}_{\min})$$

5.7 Darstellung

Zur Darstellung der erzeugten attributierten Kurven wird das von L. SCHUMANN entwickelte Linienmodell benutzt [Sch97]. Die Eingabelinie beinhaltet neben der Beschreibung des geometrischen Verlaufs weitere Attribute wie Druck und Sättigung. Dieses Merkmal stellt eine wesentliche Voraussetzung für die vorliegende Arbeit dar. Sowohl der geometrische Verlauf als auch der Verlauf der Attribute wird durch Kurven beschrieben.

Die gezeichnete Linie ist eine Überlagerung aus Pfad und Stil. Der Pfad ist die Eingabelinie und beschreibt die ursprüngliche Bewegung des Zeichenwerkzeuges. Er wird vom Stil gestört. Diese Störungen untergliedern sich in zwei Klassen, die der Geometrie und die der Attribute.

Die Störung der Geometrie wird mit Hilfe zweier Kurven beschrieben, einer Störkurve und einer Referenzkurve. Die Überlagerung resultiert aus allen Operationen, die notwendig sind,

die Referenzkurve dem Verlauf des Pfades anzupassen. Die Geometrie des Pfades und der Störkurve wird durch stückweise polynomiale Kurven beschrieben. Diese benötigen drei Parameter:

- Eine Liste von Kontrollpunkten, die den generellen Verlauf der Kurve beschreibt,
- eine Polynombasis, sie gibt an, wie die Kontrollpunkte interpretiert werden sollen (beispielsweise durch Geraden verbunden werden, mit Spline interpoliert oder approximiert werden),
- und ein Flag, das angibt, ob die Kurve zyklisch ist.

Die eigentliche Überlagerung geschieht punktweise. Dazu werden Pfad, Störkurve und Referenzkurve als parametrische Kurven aufgefaßt und für den gleichen Parameter überlagert. Es wird für den entsprechenden Parameter die Differenz von Stör- und Referenzkurve bestimmt. Dieser Differenzvektor ist definiert durch seine Länge und den Winkel zwischen ihm und der Referenzkurve. Der Differenzvektor wird an dem entsprechenden Pfadpunkt abgetragen. Dabei wird der Winkel bezüglich der Tangente des Pfades in dem Pfadpunkt gemessen. Es ergibt sich eine entsprechende Anpassung an den Verlauf des Pfades. Die Überlagerung wird durch entsprechend dichte Berechnung der Pfadpunkte als Polygonzug ausgegeben.

Pfad und Störkurve können neben den geometrischen Informationen auch zusätzliche Attribute wie Druck und Sättigung speichern. Dabei gibt es mehrere mögliche Beschreibungsarten. In dieser Arbeit werden die Attributwerte an den Kontrollpunkten des Pfades angegeben. Für die Attribute können aber auch unabhängige Kontrollpunkte gewählt werden.

Die so ermittelten Attribute werden in physische Attribute konvertiert (Helligkeit und Breite). Die Breite wird an dem aus der Geometrieüberlagerung gewonnenen Polygonzug abgetragen und kann so gezeichnet werden. Dazu werden einzelne Vierecke mit der jeweils bestimmten Helligkeit ausgegeben.

5.8 PostScript-Ausgabe

VisualWorks selbst stellt Routinen zur Graphik-Ausgabe in PostScript-Dateien bereit. Diese wurden von dem Zeichenwerkzeug benutzt.

Da in dieser Arbeit hauptsächlich die Linienbreite variiert wurde, ist eine Aufteilung der darzustellenden Linie in einzelne Vierecke unnötig. Dies ist nur notwendig, um verschiedene Grauwerte darstellen zu können. Daher wurde eine spezialisierte Routine entwickelt, die einsetzt, wenn der Linienzug keinen Helligkeitsverlauf beinhaltet, d. h. wenn die Sättigung an allen Kontrollpunkten gleich ist. Daraus resultiert eine Verkleinerung der PS-Dateien auf ungefähr die Hälfte, da Koordinaten nicht doppelt ausgegeben werden.

Diese Änderung erzeugt allerdings Polygone mit sehr vielen Eckpunkten. Die in VisualWorks vorhandene Routine zur PS-Ausgabe legt erst alle Koordinaten auf den Stack, bevor der Befehl zum Zeichnen gegeben wird. Aufgrund einer beschränkten Stacktiefe erzeugen einige Drucker dabei einen Stacküberlauf. Daher wurde eine Methode entwickelt, die direkt

spezialisierten PostScript-Code schreibt und nicht die in VisualWorks verfügbaren Routinen benutzt.

5.9 Performance

Die Performance des Gesamtsystems ist für ein Batch-Rendering-System ausreichend. Typische Zeiten auf einem PC (Pentium 90 Mhz unter Windows NT) liegen unter einer Minute. Als Beispiel sei das Modell der Knochen eines menschlichen Fußes angeführt. Es besteht aus 24 Objekten mit insgesamt 4204 Dreiecken. Die Darstellung dieses Modells mit beleuchtungsabhängigen Linienstärken und Lücken an den Enden erzeugt 265 Linienzüge und dauerte 35.1 Sekunden. Diese teilen sich wie folgt auf:

Prozeß	Zeit in Sek.	Anteil in Prozent
Laden des Modells	4.2 s	12.0%
Bestimmung sichtbarer Kanten	22.9 s	65.2%
Kurvengenerierung und -attributierung	0.5 s	1.4%
Darstellung der Linien	7.5 s	21.4%

Die meiste Zeit verbrauchen dabei die Sichtbarkeitsbestimmung und das eigentliche Zeichnen der Segmente. Die Kurvengenerierung und -attributierung fallen kaum ins Gewicht. Wenn die Performance des Zeichenwerkzeugs noch verbessert wird, ist somit eine interaktive Manipulation des Bildes möglich (solange nicht neu gerendert werden muß).

Im nächsten Kapitel werden einige Beispiele vorgestellt, die die Leistungsfähigkeit der implementierten Verfahren demonstrieren.

6 Ergebnisse in Bildern

In diesem Kapitel werden einige mit den in dieser Arbeit entwickelten Verfahren generierte Bilder präsentiert. Generell wurden dabei 3DS-Modelle gerendert, und als Encapsulated PostScript ausgegeben.

Den Anfang bildet eine Gruppe springender Delphine (Abbildung 6.1). Im kleineren Bild wurden nur die Kurven bestimmt und mit einem Stil mit Grauverlauf gezeichnet. Außer der Pfadgeometrie wurden keine weiteren Attribute verwendet. Das große Bild dagegen verwendet einen einfachen Stil ohne Grauverlauf, dafür eine beleuchtungsabhängige Linienbreite (Lichtquelle von rechts oben), die durch Attributierung des Pfades erreicht wurde. Die Linienenden wurden an den Verdeckungen gekürzt.

Die auf den ersten Blick vielleicht merkwürdig erscheinende doppelte Unterbrechung des Ringes oben und unten wird durch eine an diesen Stellen befindliche Verdeckung verursacht. Diese Verdeckung ist in der Liniengraphik nicht mehr offensichtlich, da die beteiligten inneren Kanten des Modells nicht mit dargestellt werden (sie sind geglättet).

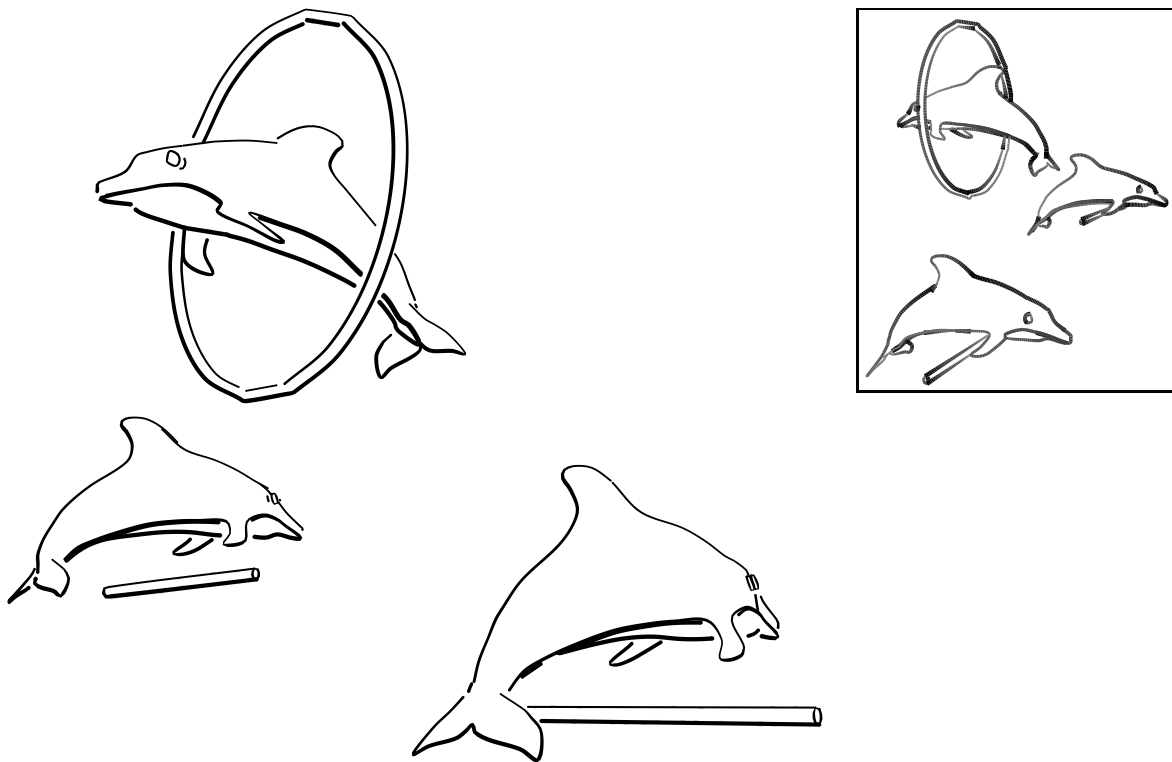


Abbildung 6.1: Delphine

In Abbildung 6.2 wird der Einfluß unterschiedlich gesetzter Glättungsinformationen gezeigt. Im ursprünglichen Modell (a) waren alle Kanten geglättet, so daß nur Verdeckungskanten dargestellt werden. Dies ist besonders am Maul sichtbar. Das Modell wurde modifiziert (b), indem die Innenseite des Mauls eine andere Glättungsgruppe zugewiesen bekam als das Äußere des Kopfes. Dadurch wird eine zusätzliche Linie dargestellt, die den Kopf besser erkennbar macht. Des weiteren wurde auch die Glättung der Wirbel verändert. Die „Zacken“ auf der Oberseite der Wirbel gehen damit nicht mehr glatt in die Wirbel selbst über. Erst so werden sie deutlich erkennbar.

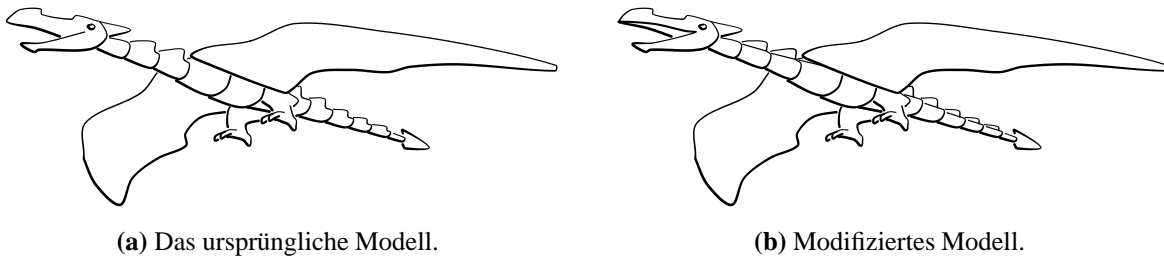


Abbildung 6.2: Drachen

Ein detaillierteres Beispiel aus dem Bereich der Architektur zeigt Abbildung 6.3. Durch den Einsatz von Lücken sind beispielsweise die Geländer der oberen Etage visuell von den teilweise verdeckten, hinteren Objektpartien getrennt. Trotz der relativ breiten Linienführung sind so im rechten Bildteil noch einzelne Streben erkennbar. Nur am linken Giebel, der perspektivisch stark verkürzt ist, überlagern sich die Linien zu einer fast homogenen schwarzen Fläche.

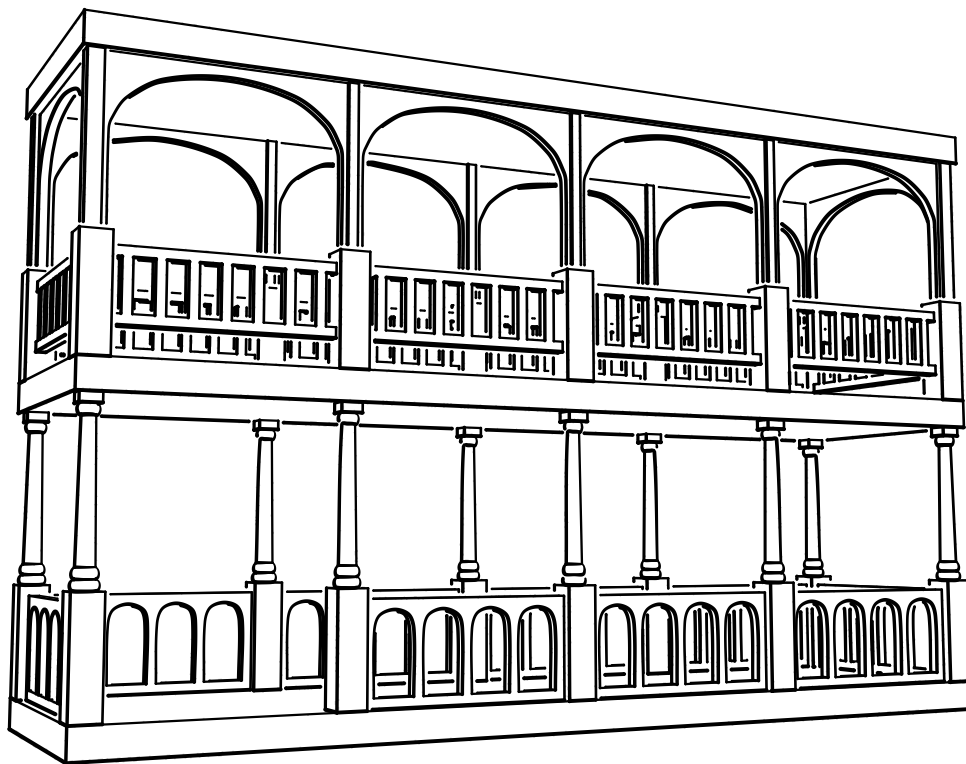
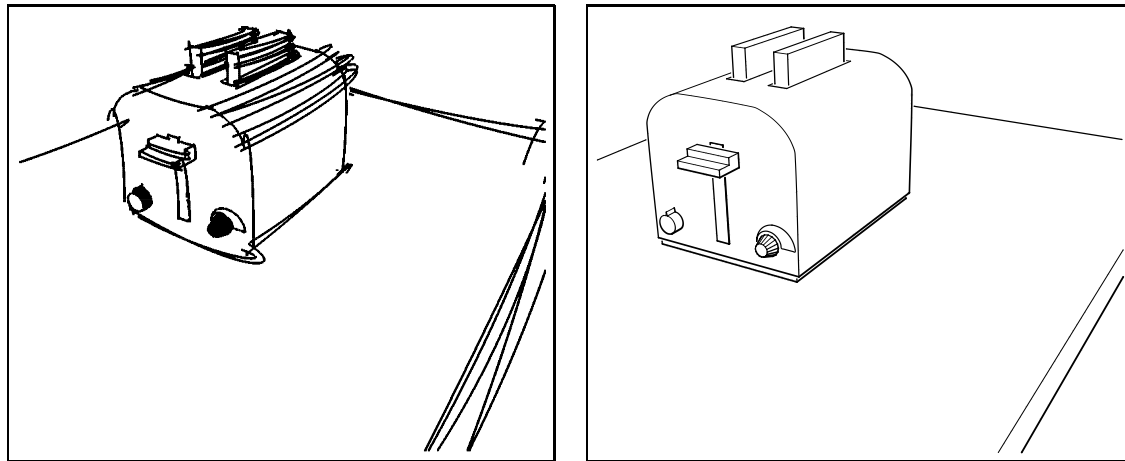


Abbildung 6.3: Belvedere

Die Graphik in Abbildung 6.4(a) wurde mit dem Sketchrenderer erzeugt. Sie enthält auch Schraffuren (Toastscheiben). Der schwungvolle Liniestil verleiht dem Bild eine entwurfsartige, eben skizzenhafte Erscheinung. Im Stil einer technischen Illustration ist (b) gehalten. Die schnörkellose Liniendarstellung macht auch kleinere Details, wie den Einstellregler, erkennbar.

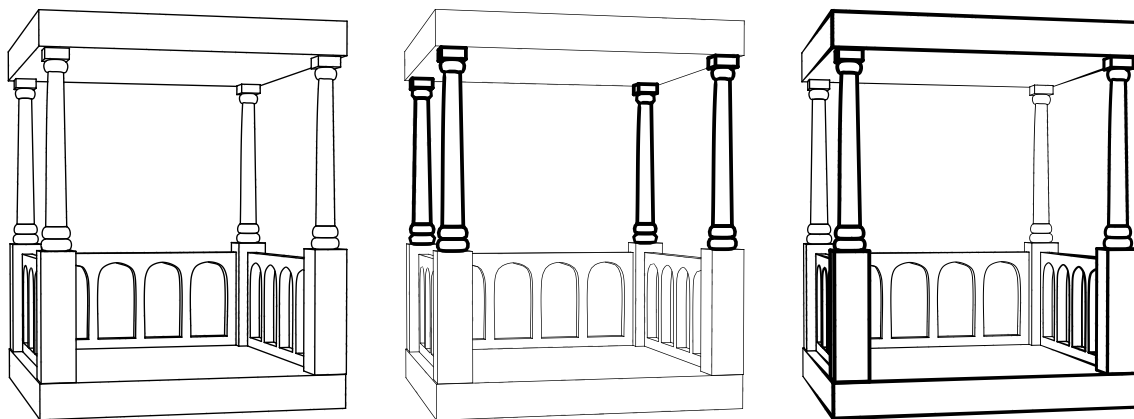


(a) Mit dem Sketchrenderer gerendert

(b) Mit charakteristischen Linien gerendert

Abbildung 6.4: Toaster

In Abbildung 6.5 werden Methoden der Betonung gezeigt. In (a) ist das ohne Hervorhebungen gezeichnete Modell sichtbar. Die Betonung der Säulen in (b) wird durch die Variation der Liniestärke erreicht. Das im (c) gezeigte Depthcueing steht prototypisch für eine Vielzahl weiterer Visualisierungsmöglichkeiten. Hier wird die Entfernung vom Betrachter in der Linienbreite codiert. Genau so gut könnten andere den Eckpunkten zugeordnete numerische Daten verwendet werden.



(a) Ohne Hervorhebungen

(b) Säulen betont

(c) Depthcueing

Abbildung 6.5: Betonungen

In Anlehnung an das Logo des Deutschen Studienpreises der Körber-Stiftung entstand das Modell für Abbildung 6.6. Hier wurde es mit skizzenhaften Strichen gezeichnet. Für Skizzen typisch ist das mehrfache Ansetzen des Zeichenwerkzeuges, wobei sich die einzelnen Striche überlappen. Die Geometrie jedes einzelnen Striches wurde stochastisch modifiziert. Gut zu beobachten ist hier, wie die Striche trotz der zufälligen Störung der Objektrundung folgen.

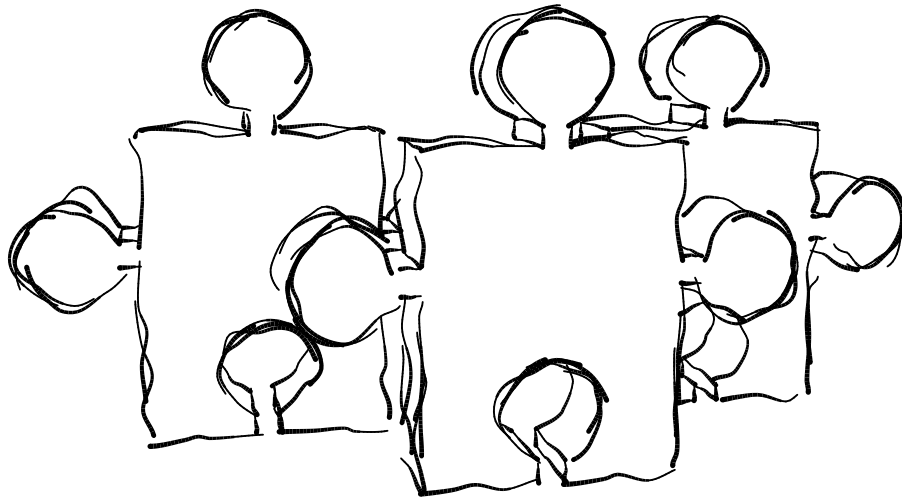


Abbildung 6.6: Skizzenhafter Liniestil

Den Versuchsaufbau des Magdeburger Galgenversuches Otto von Guericke's illustriert Abbildung 6.7. In dem kleineren Bild sind alle Kanten dargestellt um den Modellaufbau zu verdeutlichen. Durch das Setzen der Glättungsinformationen werden die Gewichte, die im Modell einen viereckigen Querschnitt haben, in der Zeichnung rundlich dargestellt. Das trifft auch auf die Kettenglieder zu.

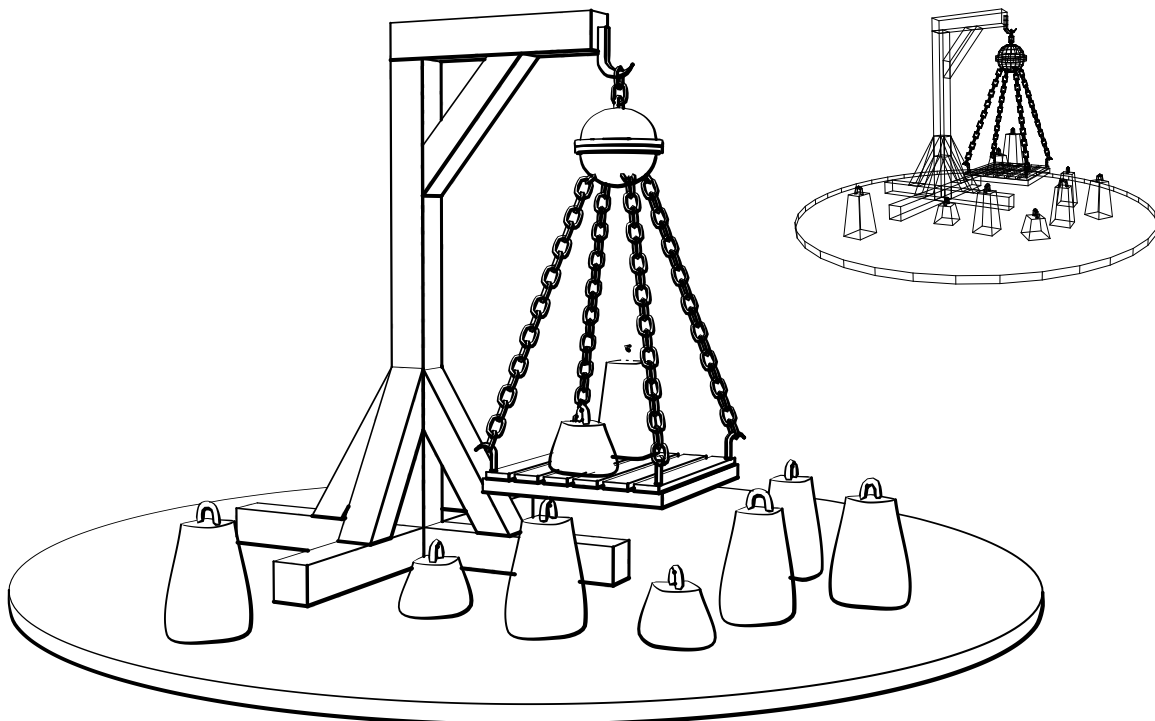


Abbildung 6.7: Galgenversuch

Eine weniger ernsthafte Darstellung zeigt Abbildung 6.8. Zur Erzeugung dieses Bildes wurde der analytische Renderer so manipuliert, daß er an bestimmten Verdeckungen die „falsche“ Alternative wählt. Das Kriterium hierfür ist, daß die Entfernungsdifferenz am Schnittpunkt einen bestimmten Wert überschreitet. Zur Darstellung wurde ein stochastisch veränderter

Linienstil benutzt.

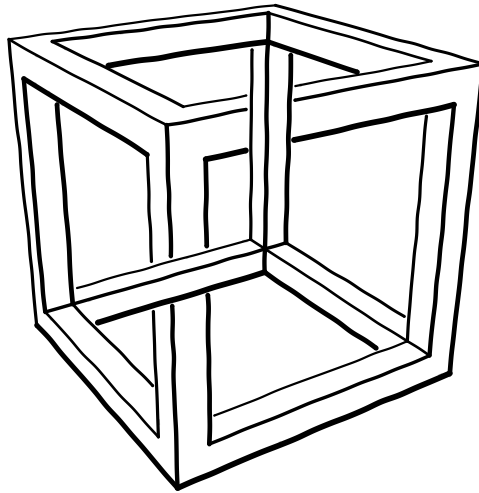
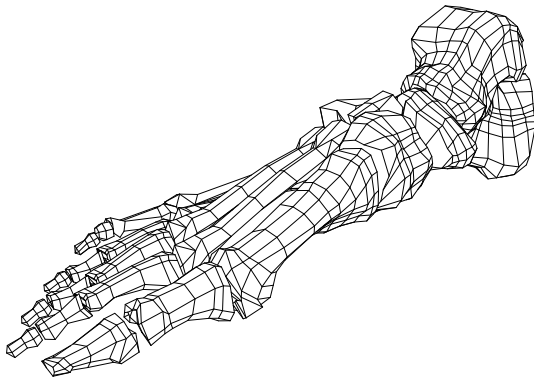


Abbildung 6.8: Unmöglicher Würfel

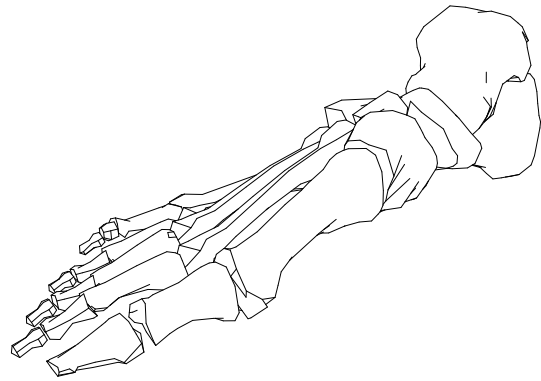
Die Bildserie in Abbildung 6.9 zeigt das Modell der Knochen eines menschlichen Fußes. Die polygonale Struktur ist aus (a) ersichtlich. Es wurden alle sichtbaren Kanten außer den Triangulierungskanten dargestellt. Aus diesen Kanten wurden die Kanten für (b) selektiert und gezeichnet. Es sind dies die Konturen und Unstetigkeiten der Oberfläche. Die aus den Segmenten extrahierten Kurven zeigt (c). Hier wurde die Linienbreite entsprechend der Beleuchtung gewählt. Zusätzliche (hier übertrieben große) Lücken an Verdeckungen weist (d) auf.

Eine Erweiterung der in dieser Arbeit erzeugten Konturzeichnungen zeigen die letzten beiden Bilder. Hier wurden die (normalerweise freien) Flächen zwischen den Linien besonders behandelt. In (e) wurde die Liniengraphik mit einer flächigen Darstellung kombiniert. Da beim Rendern auch eine analytische Beschreibung der sichtbaren Flächen vorliegt, können diese auf einfache Weise als Polygone in PostScript ausgegeben werden. Im Gegensatz zu anderen Verfahren, die solche Bilder erzeugen (wie z. B. bei DOOLEY/COHEN), müssen also keine Pixelbilder generiert werden. Allerdings ist so ohne größeren Aufwand auch nur ein Flat-Shading möglich. Der Grauwert wird dabei durch die Auswertung des Beleuchtungsmodells ermittelt.

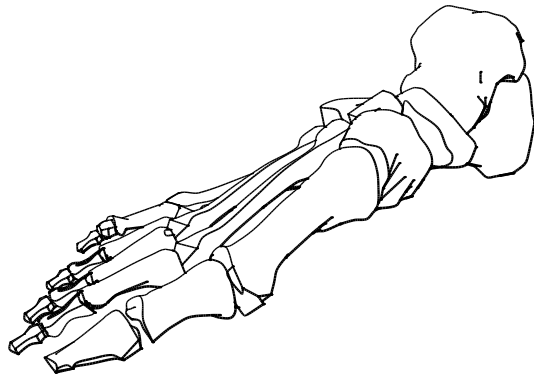
Eine andere Technik der Flächendarstellung, das *Stippling*, zeigt (f). Dabei wird eine Schattierung durch das Füllen von Flächen mit mehr oder weniger dichten, zufällig verteilte Punkten realisiert. Diese Darstellungsart ist häufig in wissenschaftlichen Illustrationen zu finden. Laut [Hod89] hat Stippling gegenüber Schraffuren den Vorteil, daß es nicht ungewollt eine gerichtete Oberflächenstruktur impliziert. Allerdings ist die manuelle Ausführung dieser Technik sehr zeitaufwendig. Für die hier gezeigte Darstellung wurden Punkte in einem Raster erzeugt, dessen Weite von der Helligkeit der jeweiligen Fläche abhängt, und nach einer zufälligen Störung gezeichnet.



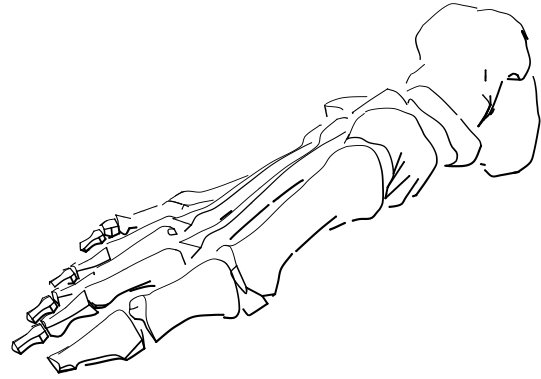
(a) Alle Kanten des verwendeten Modells



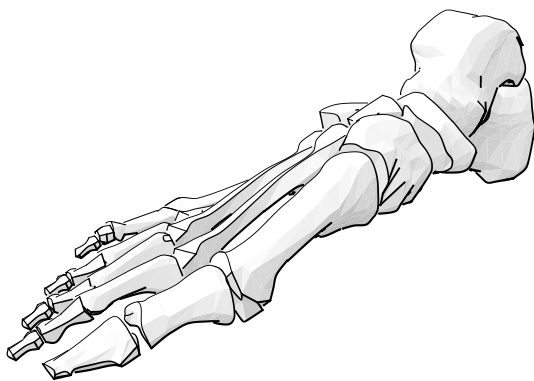
(b) Die Unstetigkeiten, polygonal dargestellt



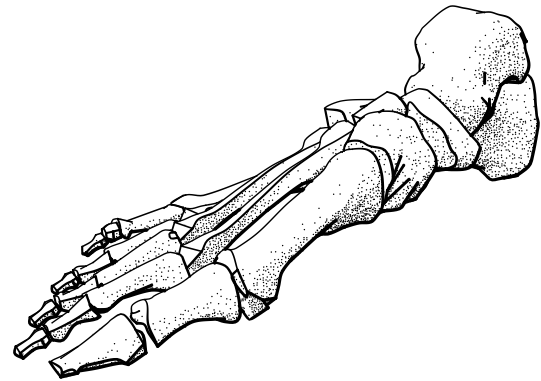
(c) Kurven, Schattierung durch Linienbreite angedeutet



(d) Zusätzlich betonte Verdeckungen (leicht übertrieben)



(e) Hinterlegte Flächen



(f) Stippling

Abbildung 6.9: Fußknochen

7 Diskussion

Es wurden Verfahren entwickelt und implementiert, mit denen Liniengraphiken aus 3D-Modellen erzeugt werden können. Diese orientieren sich in der Erscheinung an traditionellen Liniengraphiken. Dazu wurde das Konzept der charakteristischen Linien eingeführt. Es zeichnet sich dadurch aus, daß die im polygonalen Modell enthaltenen Kanten nicht wie üblich direkt gezeichnet werden. Statt dessen werden aus diesen Kanten Kurvenzüge generiert, die mit Attributen wie Druck und Sättigung versehen werden.

Dabei können die gleichen Modelle wie für photorealistische Render-Verfahren verwendet werden. Die in ihnen enthaltenen Informationen werden nur anders interpretiert und zur Beeinflussung der Darstellung herangezogen. So können auch aus relativ groben Modellen ansprechende Darstellungen erzeugt werden.

Zur Darstellung selbst wird ein Zeichenwerkzeug benutzt, das die übergebenen attributierten Kurvenzüge auf einem physischen Medium zeichnet. Durch die daraus resultierende Auflösungsunabhängigkeit sind für den professionellen Einsatz unabdingbare hochqualitative Ausgabegeräte (wie Photosatzmaschinen) nutzbar, ohne daß erhöhter Berechnungsaufwand entsteht.

Im Laufe der Arbeit wurden Schwächen des gewählten Ansatzes offenbar, die im folgenden dargelegt werden. Es schließen sich Ausführungen zu Ideen an, die bei der Bearbeitung der Aufgabenstellung auftraten, aber keine Berücksichtigung finden konnten.

7.1 Kritik

Das in dieser Arbeit entwickelte Verfahren zur Rekonstruktion von Kurven aus polygonalen, mit Glättungsinformationen versehenen Modellen erhebt nicht den Anspruch auf mathematisch-physikalische Korrektheit. In diesem Punkt steht es in einer Reihe mit vielen computergraphischen Methoden, die trotzdem häufig Verwendung finden – eine ansprechende visuelle Erscheinung hat dabei eine höhere Priorität als die exakte Simulation der Bildentstehung¹.

Der gewählte Ansatz, direkt die Projektion der gegebenen Modellpunkte im zweidimensionalen Bildraum zu interpolieren, leistet dabei gute Dienste. Daß keine schwerwiegenden

¹ In many applications [...] reality is altered for aesthetic effect or to fulfill a naive viewer's expectations. [...] Taking liberties with physics can result in attractive, memorable, and useful pictures! [FvDFH90, S. 605 f.]

Bildfehler auftauchen, ist den typischerweise verwendeten Modellierungsverfahren zu verdanken. Diese sind auf eine polygonale, flächenhafte Darstellung am Bildschirm ausgelegt und produzieren daher Triangulierungen, die dicht genug sind, um im erzeugten Pixelbild „rund“ zu erscheinen. In Bereichen größerer Krümmungen ist dabei ein dichteres Netz vorhanden als in flacheren Regionen, wo größere Polygone Verwendung finden.

Die Modelle sind allerdings im allgemeinen auch nicht so eng trianguliert, daß keine Umwandlung in Kurven nötig wäre. Das liegt an der weitaus höheren Auflösung eines Druckers bzw. Belichtungssystems gegenüber der eines normalen Bildschirms. Zum anderen sollte gezeigt werden, daß *gerade* aus relativ groben Modellen noch annehmbare Liniengraphiken erzeugt werden können (die dabei nicht unbedingt die Grobheit des Modells offenbaren, wie das bei einer photorealistischen Darstellung unumgänglich ist).

Je gröber das Modell ist, um so größer ist die Abweichung der generierten Kurve von dem durch sie angenäherten Polygonzug. Das kann einige unerwünschte Konsequenzen haben. So können sich Kurven benachbarter Objekte überschneiden, während die Polygonzüge sich nicht einmal berühren (siehe Abbildung 7.1).

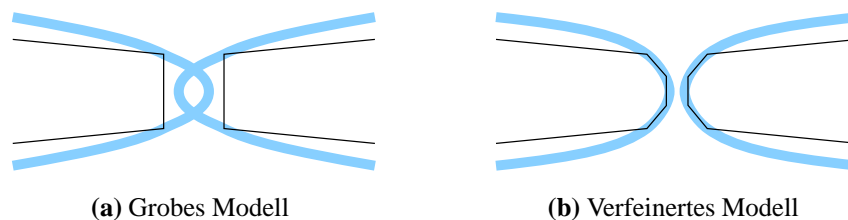


Abbildung 7.1: Einfluß der Modellauflösung

Linien enden im allgemeinen in Ecken, oder es treffen sich Kontur- und Binnenlinien. Bei diesen Konstellationen treten kaum Probleme auf (abgesehen vom Zusammentreffen unterschiedlicher Strichstärken), da alle beteiligten Linien wenigstens den betreffenden Stützpunkt teilen.

Anders sieht das an Verdeckungen aus. Wie schon in Abschnitt 4.1.3 erwähnt, spiegeln hier die generierten Kurven die Geometrie der Polygone nicht ganz adäquat wider. Zwar akzeptiert das Auge gerade bei Liniengraphiken viele Unexaktheiten, doch sollte dieser Fakt, falls es auf Präzision ankommt, nicht leichtfertig ignoriert werden.

Das Problem ist zweiteilig. Zum einen entstehen durch die Krümmung der verdeckenden Kontur unbeabsichtigte Lücken und Überschneidungen. Die verdeckte Linie wird bis zu dem (polygonal exakt bestimmten) Schnittpunkt gezeichnet, den die verdeckende Kurve aber nicht notwendigerweise durchläuft. Je nachdem, ob die verdeckende Kurve nach außen oder innen gewölbt ist, entsteht eine Lücke oder Überschneidung (siehe Abbildung 7.2).

Zum anderen hängt der Kurvenverlauf der verdeckten Linie davon ab, an welcher Stelle das Segment verdeckt wurde. Der „ideale“ Verlauf der Kurve geht von den unverdeckten, im Modell vorhandenen Stützpunkten aus. Es wird aber der Segmentschnittpunkt als letzter Stützpunkt der verdeckten Linie verwendet. Damit weicht der reale Verlauf im Verdeckungsfall mehr oder weniger von dem idealen ab (siehe Abbildung 7.3).

In Einzelbildern fallen diese Artefakte weniger auf. Es wurden aber auch versuchsweise Animationen erstellt. Dabei sind durch die langsame Änderung der Verdeckungen (beispielswei-

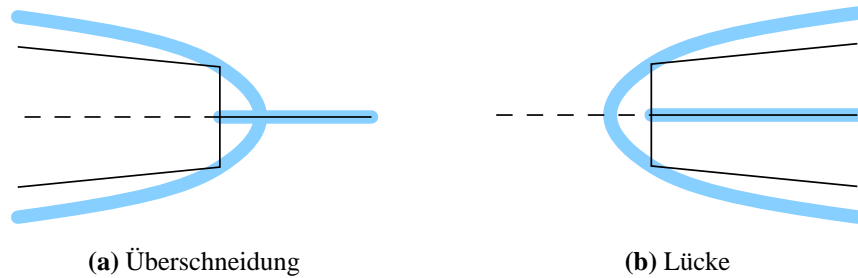


Abbildung 7.2: Ungenauigkeiten durch Krümmung der verdeckten Kontur

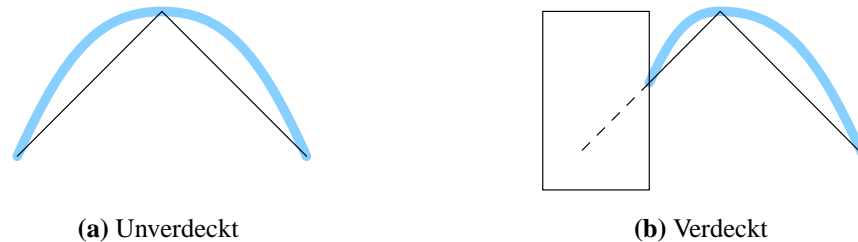


Abbildung 7.3: Verformung durch Verdeckung einer Linie

se bei einer Kamerafahrt) inkonsistente Linien viel leichter auszumachen. Bei der Betrachtung der Bildsequenz fallen auch Sprünge zwischen aufeinanderfolgenden Bildern auf. Das geschieht zum Beispiel, wenn eine frei verlaufende Kante von einem Objekt verdeckt und dadurch in zwei Teile geteilt wird. Da der Stil mit der Pfadlänge skaliert wird, halbiert sich im selben Moment die Amplitude der gezeichneten Linie (sofern keine Strokes eingesetzt werden).

Die Lösung für das erste Problem besteht in einer Bestimmung des Schnittpunktes der beiden Kurven statt der Verwendung des Segmentschnittpunktes. Diese Berechnung ist zusätzlich abhängig vom zum Zeichnen verwendeten Linienmodell; insbesondere kann ein die Pfadgeometrie verändernder Stil zur Anwendung gekommen sein. Es reicht daher nicht aus, den Schnittpunkt der beiden Pfade zu ermitteln, sondern der wirkliche („auf dem Papier zu sehende“) Linienzug muß berücksichtigt werden.

Das zweite Problem kann gelöst werden, wenn jedes teilweise sichtbare Objekt so gezeichnet wird, als wäre es unverdeckt. Die Linien werden anschließend „an den verdeckenden Objekten“ geclippt. „An den verdeckenden Objekten“ steht deshalb in Anführungszeichen, weil auch hier eventuell wieder die Verfremdung dieser Objekte durch Linienstile berücksichtigt werden muß.

Ein weiteres Problem wurde im Zusammenhang mit der Animation offenbar. Bei der hierbei erfolgenden Darstellung aus unterschiedlichen Perspektiven häuften sich Fälle, in denen der Sichtbarkeitsalgorithmus (siehe Abschnitt 5.4, Seite 41) fehlschlug. Seine Robustheit gegenüber auftretenden Spezialfällen bedarf einer Verbesserung. Das könnte durch seine Umstellung auf exakte Arithmetik (bisher werden Fließkomma-Operationen verwendet) erreicht werden (vgl. hierzu [For96]).

7.2 Weiterführende Überlegungen

Bisher ist noch keine Einbindung der in dieser Arbeit vorgestellten Verfahren in ein Illustrationssystem erfolgt. Zur breiteren Nutzung muß eine leicht zu bedienende Oberfläche dafür bereitgestellt werden. Dann könnten auch Fachleute (z. B. Graphiker) das System testen. Dies würde mit Sicherheit neue Anregungen für Weiterentwicklungen geben. Statt einer eigenen Oberfläche könnte die Renderingkomponente eventuell als Plugin für ein gängiges Modellersystem (z. B. Alias oder 3D-Studio) konzipiert werden.

Aber auch die Möglichkeiten der Darstellung selbst sind bei weitem noch nicht ausgeschöpft. Beispielsweise kann untersucht werden, ob weitere Kanten zur Darstellung herangezogen werden sollten. Für spezielle Objekte gibt es hierzu schon Überlegungen. So benutzt SASADA in seinen Landschaftsdarstellungen „Wasserfluß-Linien“, also Linien, denen entlang Wasser einen Berg herunterfließen würde [Sas87].

Auch ist es für manche Darstellung wünschenswert, verdeckte Objekte darzustellen. Dazu könnten die Arbeiten von KAMADA/KAWAI oder DOOLEY/COHEN als Ausgangspunkt dienen [DC90, KK87]. Der analytische Renderer kann relativ leicht so modifiziert werden, daß verdeckte Segmente mit ausgegeben werden.

Werden aus den verdeckten Segmenten auch Polygone rekonstruiert, können diese zur Schattendarstellung verwendet werden. Dazu wird eine Bestimmung der verdeckten Polygone aus Sicht einer gerichteten Lichtquelle (*spot light*) vorgenommen. Das liefert alle im Schatten liegenden Polygonfragmente. Nach einer Sichtbarkeitsbestimmung aus Sicht der Kamera erhält man die sichtbaren Schattenfragmente, die „über“ die normale Zeichnung gezeichnet werden können [AWG78].

Interessant wäre auch die Untersuchung von Verzerrungen in Liniengraphiken. Diese werden häufig bei medizinischen Illustrationen verwendet. Dabei könnten sowohl im dreidimensionalen Modellraum als auch im zweidimensionalen Bildraum Veränderungen vorgenommen werden. Probleme könnte die Konsistenz der verformten Bilder bereiten. Hier müßten gewisse Randbedingungen eingehalten werden. Reine Konturzeichnungen können sicherlich nachträglich behandelt werden, beim Einsatz von Schraffuren würde das aber Auswirkungen auf die Liniendichte und damit auf die Bildhelligkeit haben.

Vielversprechend gerade für Liniengraphiken erscheint eine Kombination mit Techniken zur automatischen Detailreduktion. So können für die Bildaussage unwichtige Objekte weniger detailliert dargestellt oder sogar weggelassen werden. Eine einfachere und effizientere Darstellung für flache bzw. linienartige Objekte (dünne Platten, Haare) wird durch Objektlinien (siehe Kapitel 2) erreicht. Dabei werden beim Darstellen eine oder zwei Dimensionen nicht mit einbezogen (also die Dicke der Platte bzw. der Querschnitt des Haares). Sie werden durch die Breite der Linie selbst wiedergegeben.

Eine Anreicherung der Zeichnung mit Details der Objektoberfläche bietet der Übergang von reinen Kontur- zu Flächenzeichnungen. Dabei werden auch in den in einer Konturzeichnung weißen Flächen Bildelemente plaziert. Diese „modellieren“ die Oberfläche des Objektes. Schon in der Computergraphik eingesetzt wurden Kreuzschraffuren (LEISTER), Texturen und Punkte (WINKENBACH/SALESIN) und gefüllte Flächen (SAITO/TAKAHASHI, DOO-

LEY/COHEN). Einfache, während der Arbeit implementierte Beispiele waren im Beispielkapitel aufgeführt. Eine sich an traditionellen Schraffurverfahren orientierende Technik müßte mehr bildorientiert arbeiten, also zum Beispiel die Schraffurstriche entlang der Objektkonturen im Bild ausrichten. Die bisherigen Verfahren arbeiten mit isoparametrischen Linien oder nach der „Eierschneidermethode“, bei der die Schnittlinien mit einer Ebenenschar dargestellt werden. Letztere könnte bei der Orientierung der Ebenen an einem Skelett des Objektes bessere Ergebnisse liefern.

7.3 Fazit

Liniengraphiken sind ein wichtiges Mittel zur Darstellung dreidimensional repräsentierter Objekte. Sie können mit computergraphischen Mitteln erzeugt werden. Allerdings ist es wohl noch ein weiter Weg, bis solche und andere nicht-photorealistic Graphiken von gängigen Renderingsystemen als gleichberechtigtes Stilmittel behandelt werden. Dazu müssen die eingesetzten Verfahren erst reifen und soviel Aufmerksamkeit erfahren, wie das heute bei photorealistic Techniken der Fall ist. An diesem Weg ein Stück mitgebaut zu haben ist der Lohn dieser Arbeit.

Literaturverzeichnis

- [App67] A. Appel. The Notion of Quantitative Invisibility and the Machine Rendering of Solids. In *Proceedings of the ACM National Conference*, Seiten 387–393, Washington, DC, 1967.
- [Arn74] R. Arnheim. *Art and Visual Perception*. University of California Press, 1974.
- [ARS79] A. Appel, F. J. Rohlf und A. J. Stein. The Haloed Line Effect for Hidden Line Elimination. In *SIGGRAPH '79 Conference Proceedings*, Seiten 151–157, 1979.
- [AWG78] P. Atherton, K. Weiler und D. Greenberg. Polygon Shadow Generation. In *SIGGRAPH '78 Conference Proceedings*, Seiten 275–281, August 1978.
- [DC90] D. Dooley und M. Cohen. Automatic Illustration of 3D Geometric Models: Lines. In *Computer Graphics (Symposium on Interactive 3D Graphics)*, Seiten 77–82, März 1990.
- [Dud77] Duden. *Das Bildwörterbuch*. Dudenverlag, 1977.
- [Elb95] G. Elber. Line Illustrations ∈ Computer Graphics. *The Visual Computer*, 11:290–296, 1995.
- [For96] The Computational Geometry Impact Task Force. Application Challenges to Computational Geometry. Technical Report TR-521-96, Princeton University, April 1996.
- [FvDFH90] J. Foley, A. van Dam, S. Feiner und J. Hughes. *Computer Graphics. Principles and Practice*. Addison Wesley, Reading, MA, 2. Auflage, 1990.
- [Gri79] J. G. Griffiths. Eliminating Hidden Edges in Line Drawings. *Computer Aided Design*, 11(2):71–78, März 1979.
- [HE80] D. Herbison-Evans. How To Merge Hidden Arcs And Then Not Draw Them. *Computers and Graphics*, 5(2-4):79–81, 1980.
- [Hod89] E. R. S. Hodges. *The Guild Handbook of Scientific Illustration*. Van Nostrand Reinhold, 1989.
- [KK87] T. Kamada und S. Kawai. An Enhanced Treatment of Hidden Lines. *ACM Transactions on Graphics*, 6(4):308–323, Oktober 1987.
- [Kos93] W. Koschätzky. *Die Kunst der Zeichnung*. Neuer Pawlak Verlag, Köln, 1993.

- [KR95] A. Krüger und Thomas Rist. Since Less is often More: Methods for Stylistic Abstractions in 3D-Graphics. In *Electronic Proceedings of the ACM Workshop on Effective Abstractions in Multimedia*, San Francisco, California, November 1995.
- [Lei94] W. Leister. Computer Generated Copper Plates. *Computer Graphics Forum*, 13(1):69–77, 1994.
- [LI94] Lexikon-Institut, Hrsg. *Bertelsmann Universal Lexikon*. Bertelsmann Lexikon Verlag GmbH, Gütersloh, 1994.
- [PS95] B. Preim und Th. Strothotte. Tuning Rendered Line-Drawings. In *Winter School of Computer Graphics and Scientific Visualization*, Plzeň, Februar 1995.
- [Raa94] A. Raab. Aufbau und Funktionsweise eines Skizzen-Renderers. Technischer Report, Otto-von-Guericke-Universität Magdeburg, 1994.
- [Ran87] J. R. Rankin. A Geometric Hidden-Line Processing Algorithm. *Computers and Graphics*, 11(1):11–19, 1987.
- [Rob63] L. G. Roberts. Machine Perception of Three Dimensional Solids. TR 315, Lincoln Laboratory, MIT, Mai 1963.
- [SABS94] M. P. Salisbury, S. E. Anderson, R. Barzel und D. H. Salesin. Interactive Pen-and-Ink Illustration. In *SIGGRAPH '94 Conference Proceedings*, Seiten 101–108, 1994.
- [Sas87] T. T. Sasada. Drawing Natural Scenery by Computer Graphics. *Computer Aided Design*, 19:212–218, 1987.
- [Sch92] J. Schumann. Linienqualität in ungenauen Grafiken. Diplomarbeit, Technische Universität Magdeburg, 1992.
- [Sch94] S. Schofield. *Non-photorealistic Rendering: A Critical Examination and Proposed System*. Dissertation, Middlesex University, 1994.
- [Sch95] St. Schlechtweg. Beschränkung zeichnerischer Ressourcen in Liniengrafiken. Diplomarbeit, Otto-von-Guericke-Universität Magdeburg, 1995.
- [Sch97] L. Schumann. Ein parametrisierbares Modell zur Darstellung von Linien. Diplomarbeit, Otto-von-Guericke-Universität Magdeburg, 1997.
- [SG81] S. Sechrest und D. P. Greenberg. A Visible Polygon Reconstruction Algorithm. In *SIGGRAPH '81 Conference Proceedings*, Seiten 17–27, August 1981.
- [SPR⁺94] Th. Strothotte, B. Preim, A. Raab, J. Schumann und D. R. Forsey. How to Render Frames and Influence People. *Computer Graphics Forum*, 13(3):455–466, September 1994.
- [SS96] St. Schlechtweg und Th. Strothotte. Rendering Line-Drawings with Limited Resources. In *Proceedings of GRAPHICON'96*, Seiten 131–137, St. Petersburg, Juli 1996.

- [SS97] C. Strothotte und Th. Strothotte. *Seeing Between the Pixels: Pictures in Human-Computer Interaction*. Springer-Verlag, Berlin-Heidelberg-New York, 1997. Im Druck.
- [ST90] T. Saito und T. Takahashi. Comprehensible Rendering of 3-D Shapes. In *SIGGRAPH '90 Conference Proceedings*, Seiten 197–206, 1990.
- [VW83] *VW Golf Reparaturanleitung*. Verlag Bucheli, Zug/Schweiz, 1983.
- [WA77] K. Weiler und K. Atherton. Hidden Surface Removal using Polygon Area Sorting. *SIGGRAPH '77 Conference Proceedings*, Seiten 214–222, 1977.
- [Wae12] W. Waetzold. *Einführung in die Bildenden Künste*. Leipzig, 1912.
- [Wil72] H. Williamson. Hidden-Line Plotting. *Communications of the ACM*, 15:100–102, 1972.
- [Wri72] T. Wright. Visible Surface Plotting. *Communications of the ACM*, 17:152–157, 1972.
- [WS94] G. Winkenbach und D. H. Salesin. Computer-Generated Pen-and-Ink Illustration. In *SIGGRAPH '94 Conference Proceedings*, Seiten 91–100, 1994.
- [WS96] G. Winkenbach und D. H. Salesin. Rendering Parametric Surfaces in Pen and Ink. In *SIGGRAPH '96 Conference Proceedings*, Seiten 469–476, 1996.
- [Zei96] Weniger ist mehr. In *ZEICHNEN. Fachzeitschrift für Konstruieren und Gestalten*, Januar 1996.

Thesen

- Liniengraphiken sind ein Darstellungsmittel mit dem gezielt Informationen vermittelt werden können.
- Für Illustrationen sind Liniengraphiken in besonderem Maße geeignet.
- Die Analyse von Liniengraphiken, welche mit traditionellen Mitteln erstellt wurden, liefert wertvolle Anregungen für die Umsetzung auf den Computer.
- Zur Erzeugung von Konturzeichnungen sind dreidimensionale Modellinformationen nötig.
- Aus polygonalen Modellen können charakteristische Linienzüge, die Kurven enthalten, erzeugt werden.
- Dazu sind zusätzliche Modellinformationen (wie beispielsweise Glättungsparameter) nötig.
- So erzeugte Liniengraphiken sind unempfindlicher gegen eine Senkung des Detailliertheitsgrades des Modells als flächige Darstellungen.
- Eine Variation der Linienstärke kann zur Verdeutlichung der Lichtverhältnisse angewandt und aus dem Modell bestimmt werden.
- Auch andere Parameter können sinnvoll zur Anreicherung einer Konturzeichnung benutzt werden.