

# Studienarbeit

Thema:

Ein Testbed zur Untersuchung von Glättungsalgorithmen für  
Freiformflächen

Vorgelegt von: Roland Göcke

Abgabedatum: 20.07.97

Betreuer: Prof. Dr. Heidrun Schumann

Dr. Holger Theisel

# Inhaltsverzeichnis

<b>1. EINLEITUNG</b> .....	<b>4</b>
1.1. AUFGABENSTELLUNG.....	4
1.2. FREIFORMFLÄCHEN.....	4
1.2.1. <i>Tensorproduktflächen</i> .....	5
1.2.2. <i>Zusammengesetzte Flächen - Problem der Stetigkeit</i> .....	6
<b>2. QUALITÄTSANALYSE - SURFACE INTERROGATION</b> .....	<b>8</b>
2.1. PROBLEMHINTERGRUND.....	8
2.2. VERFAHREN DER QUALITÄTSANALYSE.....	9
2.2.1. <i>Krümmungsplots</i> .....	9
2.2.2. <i>Fokalfächen</i> .....	10
2.2.3. <i>Reflexionslinien</i> .....	11
2.2.4. <i>Orthotomics</i> .....	12
2.2.5. <i>Isophoten</i> .....	12
<b>3. GLÄTTEN VON FREIFORMFLÄCHEN - SURFACE FAIRING</b> .....	<b>13</b>
3.1. GLÄTTUNGALGORITHMEN.....	15
3.1.1. <i>Minimierung der Differenzen der 3. partiellen Ableitungen</i> .....	16
3.1.2. <i>Minimierung der Differenzen der Isophotenkrümmungen</i> .....	17
<b>4. ANFORDERUNGEN AN DAS TESTBED</b> .....	<b>18</b>
4.1. BESCHREIBUNGSFORMAT FÜR DIE FLÄCHEN.....	18
4.2. GRAPISCHE AUSGABE.....	19
4.2.1. <i>Graphische Ausgabe der Freiformflächen</i> .....	19
4.2.2. <i>Graphische Ausgabe von Eigenschaften der Flächen</i> .....	21
4.3. AUSWAHL DES ZU GLÄTTENDEN FLÄCHENBEREICHES.....	21
4.4. PARAMETERSTEUERUNG.....	21
<b>5. PRAKTISCHE UMSETZUNG</b> .....	<b>22</b>
5.1. ALLGEMEINES.....	22
5.2. EINBINDUNG DER OPEN GL UNTER TCL/TK.....	23
5.3. BESCHREIBUNGSFORMAT ENFF UND DATEIVERWALTUNG DES TESTBEDS.....	24
5.4. BENUTZUNGSSCHNITTSTELLE.....	25
5.4.1. <i>Parametersteuerung</i> .....	25
5.4.2. <i>Graphische Ausgabe - Rendering</i> .....	27
5.4.3. <i>Bewegung des Augpunktes und der Lichtquelle</i> .....	30
5.4.4. <i>Umgesetzte Freiformflächen</i> .....	31
5.4.5. <i>Auswahl des zu glättenden Flächenbereichs</i> .....	32
5.5. FLÄCHENGLÄTTUNG.....	32

5.6. DATENSTRUKTUREN.....	34
5.7. RECHENZEITEN.....	34
5.8. PROGRAMMSTRUKTUR .....	36
<b>6. TESTFLÄCHEN.....</b>	<b>36</b>
<b>7. ERGEBNISSE FÜR DIE TESTFLÄCHEN .....</b>	<b>39</b>
<b>8. SCHLUßFOLGERUNGEN UND AUSBLICK.....</b>	<b>44</b>
<b>LITERATURVERZEICHNIS.....</b>	<b>46</b>
<b>SELBSTÄNDIGKEITSERKLÄRUNG .....</b>	<b>48</b>

# **1. Einleitung**

## **1.1. Aufgabenstellung**

Das Ziel der Studienarbeit war der Entwurf und die Implementierung eines Testbeds zur Untersuchung von Glättungsalgorithmen für Freiformflächen. Glättung (Surface Fairing) von Freiformflächen bedeutet, gewisse Kontrollpunkte, die die Geometrie der Fläche festlegen, geringfügig so zu verändern, daß eine glattere Fläche erzeugt wird. Die Glattheit einer Fläche (Smoothness) wird über Verfahren der Qualitätsanalyse (Surface Interrogation) bestimmt.

Mit Hilfe des Testbeds sollen Flächen, die als Dateien vorliegen, eingelesen werden. Danach werden diese Flächen graphisch ausgegeben, z.B. als Drahtmodell oder als mittels Raytracing berechnetes Bild. Weiterhin sollen bestimmte Eigenschaften der Flächen berechnet und dargestellt werden, z.B. die Isophoten der Flächen. Für die Anwendung der Glättungsalgorithmen muß ein zu glättender Flächenbereich ausgewählt werden. Auf den ausgewählten Flächenbereich wird dann der zu untersuchende Glättungsalgorithmus angewandt. Bestandteil der Studienarbeit ist die Anwendung eines einfachen Glättungsalgorithmus auf die akquirierten Testflächen und die Auswertung der erhaltenen Ergebnisse.

## **1.2. Freiformflächen**

Freiformflächen sind heutzutage die gängige Methode in vielen Designabteilungen in der Industrie. Typische Anwendungen liegen z.B. in der Automobilindustrie, im Flugzeugbau und im Schiffbau.

Eine gute und tiefgehende Erläuterung des Konzepts der Freiformflächen findet sich in [Farin 1990]. Freiformflächen bieten eine präzise und kompakte Form der Oberflächenbeschreibung. Sie beschreiben die Oberfläche genau, im Gegensatz zur Approximation einer Oberfläche durch ein Polygonnetz. Dank der begrenzten Anzahl von Designparametern lassen sich auch komplizierte Flächen einfach handhaben. Die Form einer Freiformfläche wird durch die Lage ihrer Kontrollpunkte beschrieben.

Weitverbreitet sind die sogenannten Tensorproduktflächen, die im folgenden näher betrachtet werden sollen.

### 1.2.1. Tensorproduktflächen

Tensorproduktflächen sind eine Abbildung des rechteckigen (u,v)-Parameterraumes in den dreidimensionalen euklidischen Raum. Die bekanntesten und am weitest verbreiteten Tensorproduktflächen sind die Bezierflächen und die B-Splineflächen.

Intuitiv läßt sich eine Fläche als der Ort einer Kurve bezeichnen, die sich durch den Raum bewegt und während der Bewegung ihre Form ändert. Zur Definition einer Bezierfläche setzen wir voraus, daß diese Kurve eine Bezierkurve mit konstantem Grad  $m$  ist. Damit ist die Kurve zu jedem Zeitpunkt durch die Lage ihrer Kontrollpunkte eindeutig bestimmt. Bewegt sich die Kurve, so bewegt sich auch jeder Kontrollpunkt. Die Bewegung jedes Kontrollpunktes erfolgt wiederum auf einer Kurve. Man setzt nun voraus, daß auch diesen Kurven Bezierkurven sind und daß sie alle den gleichen Grad  $n$  besitzen. Damit erhalten wir die Beschreibung eines Punktes auf einer Bezierfläche nach der folgenden Formel

$$\mathbf{x}(u, v) = \sum_{i=0}^m \sum_{j=0}^n \mathbf{b}_{i,j} \cdot B_i^m(u) \cdot B_j^n(v)$$

wobei  $u$  und  $v$  ein rechtwinkliges Parametergebiet aufspannen im (u,v)-Parameterraum, das in den dreidimensionalen euklidischen Raum abgebildet wird. Die  $\mathbf{b}_{i,j}$  sind die Kontrollpunkte der Fläche,  $B_i^m$  bzw.  $B_j^n$  sind die Bernsteinpolynome

$$B_i^m(u) = \binom{m}{i} u^i (1-u)^{m-i} \quad (\text{analog für } B_j^n)$$

Durch den Grad der beschreibenden Kurven ist die Anzahl der Kontrollpunkte festgelegt. Sie liegt immer um 1 höher als der Grad der Kurve, d.h. für eine Kurve vom Grad  $m$  werden  $m+1$  Kontrollpunkte benötigt und folglich für eine Bezierfläche  $(m+1) * (n+1)$  Kontrollpunkte.

Bezierflächen haben folgende Eigenschaften: affine Invarianz, konvexe Hüllen-Eigenschaft, variationsmindernde Eigenschaft, Interpolation in den Eckpunkten  $\mathbf{b}_{0,0}$ ,  $\mathbf{b}_{m,0}$ ,  $\mathbf{b}_{0,n}$ ,  $\mathbf{b}_{m,n}$ , Approximation der restlichen Bezierpunkte (Kontrollpunkte), Randkurven sind Bezierkurven, Invarianz gegenüber affinen Transformationen des Parametergebietes, vgl. [Farin 1990].

Werden statt der Bezierkurven B-Spline-Kurven benutzt, so erhält man die sogenannten B-Spline-Flächen

$$\mathbf{x}(u, v) = \sum_{i=0}^m \sum_{j=0}^n \mathbf{d}_{i,j} \cdot N_i^m(u) \cdot N_j^n(v).$$

Vorteil der B-Spline-Flächen gegenüber den Bezierflächen ist die lokale Wirksamkeit ihrer Kontrollpunkte, wohingegen die Kontrollpunkte der Bezierflächen global (aber pseudo-lokal) wirken.

Rationale Bezierkurven und -flächen bzw. B-Spline-Kurven und -Flächen (wie z.B. die NURBS-(Non-Uniform-Rational-B-Spline)-Flächen sind in den letzten Jahren zum Standard im Bereich der Kurven- und Flächenbeschreibung für CAD-Systeme geworden. Hierbei werden den Kontrollpunkten Gewichte zugeordnet. Diese Gewichte sind zusätzliche Gestaltungsparameter. Stehen alle Gewichte im Verhältnis 1:1, so erhält man die nichtrationalen Standardkurven und -flächen. Von Bedeutung ist das Verhältnis der Gewichte zueinander, nicht ihre absolute Größe. Wird ein Gewicht im Verhältnis zu den anderen Gewichten vergrößert, so wird die Kurve näher an den korrespondierenden Kontrollpunkt herangezogen. Das Verändern der Gewichte bedeutet keine Verschiebung der Kontrollpunkte. Hier die Formeln für die rationalen Flächen

rationale Bezierfläche:

$$\mathbf{x}(u, v) = \frac{\sum_i \sum_j w_{i,j} \cdot \mathbf{b}_{i,j} \cdot B_i^m(u) \cdot B_j^n(v)}{\sum_i \sum_j w_{i,j} \cdot B_i^m(u) \cdot B_j^n(v)}$$

rationale B-Spline-Fläche:

$$\mathbf{x}(u, v) = \frac{\sum_i \sum_j w_{i,j} \cdot \mathbf{d}_{i,j} \cdot N_i^m(u) \cdot N_j^n(v)}{\sum_i \sum_j w_{i,j} \cdot N_i^m(u) \cdot N_j^n(v)}$$

Die  $\mathbf{b}_{i,j}$  bzw.  $\mathbf{d}_{i,j}$  sind die Kontrollpunkte der Fläche,  $B_i^m$  bzw.  $B_j^n$  die schon bekannten Bernsteinpolynome,  $N_i^m$  bzw.  $N_j^n$  die Spline-Polynome und  $w_{i,j}$  die Gewichte.

Viele der Eigenschaften nichtrationaler Flächen gelten ebenso für rationale Flächen.

### 1.2.2. Zusammengesetzte Flächen - Problem der Stetigkeit

Häufig werden beim Entwurf kompliziert geformte oder großflächige Freiformflächen benötigt. Zu ihrer Beschreibung sind eine Vielzahl von Kontrollpunkten nötig. Theoretisch wäre es möglich, mit Hilfe dieser Kontrollpunkte eine einzige Fläche zu definieren. Dieses Vorgehen hätte jedoch gravierende Nachteile. Mit der Anzahl der Kontrollpunkte erhöht sich der Grad der Fläche. Je höher der Grad einer Fläche, desto schwieriger wird die praktische Handhabung und desto größer wird der Rechenaufwand. Außerdem besitzt jeder Kontrollpunkt zwar pseudo-lokalen, damit aber doch globalen Einfluß.

Lösung des Problems ist das Zusammensetzen von komplizierten Flächen aus einfacheren Teilflächen, den sogenannten Patches. Dieses Vorgehen ist eine Erweiterung des Konzepts der Spline-Kurven auf Spline-Flächen. Bei den Spline-Kurven werden die Kontrollpunkte am Ende der Splines vorgegeben. Die weiteren Kontrollpunkte lassen sich dann aus bestimmten Forderungen an die Stetigkeit der Übergänge von einer Spline-Teilkurve zur nächsten bestimmen. Häufig werden kubische Teilkurven benutzt und analog für Flächen bikubische Teilflächen.

Beim Zusammensetzen der Teilflächen spielt die Stetigkeit an den Patchübergängen eine große Rolle. Gesucht sind möglichst glatte Übergänge. Deshalb wird oft die Forderung nach  $C^2$ -Stetigkeit an den Übergängen der Teilflächen gestellt. Bei bikubischen Teilflächen läßt sich die Lage der inneren Kontrollpunkte so berechnen, daß  $C^2$ -Stetigkeit erreicht wird. Dabei gibt der Designer die genaue Lage der Eckpunkte (äußere Kontrollpunkte) des Patches vor. Dort wird die Fläche genau interpoliert, die Lage der inneren Kontrollpunkte wird so berechnet, daß die gewünschte Stetigkeit erreicht wird. Die inneren Kontrollpunkte werden durch die Fläche approximiert.

Die allgemeine Bedingung für  $C^r$ -Stetigkeit von Bezier-Spline-Flächen in Richtung des Parameters  $u$  läßt sich wie folgt ausdrücken [Farin 1990]

$$\left(\frac{1}{\Delta_{I-1}}\right)^r \Delta^{r,0} \mathbf{b}_{m-r,j} = \left(\frac{1}{\Delta_I}\right)^r \Delta^{r,0} \mathbf{b}_{m,j}$$

mit  $j = 0, \dots, n$

$$\Delta_I = u_{I+1} - u_I$$

$$\Delta^{r,0} \mathbf{b}_{i,j} = \Delta^{r-1,0} \mathbf{b}_{i+1,j} - \Delta^{r-1,0} \mathbf{b}_{i,j}$$

Dies bedeutet, daß zwei benachbarte Bezier-Spline-Flächen  $C^r$ -stetig über die gemeinsame Grenze sind genau dann, wenn alle Zeilen ihrer Kontrollpunkte als Polygone von stückweisen,  $C^r$ -stetigen Bezier-Spline-Kurven interpretiert werden

können, d.h. wenn  $C^r$ -Stetigkeit der parametrischen Linien  $v = \text{const}$  vorliegt. Die allgemeine Bedingung für  $C^r$ -Stetigkeit von Bezier-Spline-Flächen in Richtung des Parameters  $v$  läßt sich auf analoge Art und Weise aufstellen.

Ebenso lassen sich die Bedingungen für  $C^r$ -Stetigkeit von B-Spline-Flächen aufstellen, auf die hier jedoch nicht näher eingegangen werden soll.

Für die häufig genutzten bikubischen Bezier-Spline-Flächen ergeben sich dadurch folgende Übergangsbedingungen zum stetigen Zusammensetzen von Teilflächen. Angenommen, zwei bikubische Bezier-Spline-Flächen sind durch Kontrollpunkte  $\mathbf{b}^-_{i,j}$  bzw.  $\mathbf{b}^+_{i,j}$  ( $i, j = 0, \dots, 3$ ) und Parametergebiete  $[u_0, u_1] \times [v_0, v_1]$  bzw.  $[u_1, u_2] \times [v_0, v_1]$  gegeben. Gesucht werden die Bedingungen zum stetigen Zusammensetzen beider Flächen entlang der gemeinsamen Parameterlinie  $u = u_1$ .

- $C^0$ -Stetigkeit der parametrischen Linien  $v = \text{const}$ , d.h.

$$\mathbf{b}^-_{3,j} = \mathbf{b}^+_{0,j} \quad (j = 0, \dots, 3)$$

- $C^1$ -Stetigkeit der parametrischen Linien  $v = \text{const}$ , d.h.

$$\mathbf{b}^-_{2,j}, \mathbf{b}^-_{3,j} = \mathbf{b}^+_{0,j}, \mathbf{b}^+_{1,j} \text{ sind kollinear und es gilt für } j = 0, \dots, 3$$

$$\frac{\|\mathbf{b}^-_{3,j} - \mathbf{b}^-_{2,j}\|}{\|\mathbf{b}^+_{1,j} - \mathbf{b}^+_{0,j}\|} = \frac{u_2 - u_1}{u_1 - u_0}$$

- $C^2$ -Stetigkeit der parametrischen Linien  $v = \text{const}$ , d.h. es existiert ein Punkt  $\mathbf{d}$ , so

$$\text{daß } \frac{\|\mathbf{d} - \mathbf{b}^-_{2,j}\|}{\|\mathbf{b}^-_{2,j} - \mathbf{b}^-_{1,j}\|} = \frac{u_2 - u_1}{u_1 - u_0}$$

$$\text{und } \frac{\|\mathbf{b}^+_{2,j} - \mathbf{b}^+_{1,j}\|}{\|\mathbf{b}^+_{1,j} - \mathbf{d}\|} = \frac{u_2 - u_1}{u_1 - u_0}$$

und  $\mathbf{d}$  ist kollinear zu  $\mathbf{b}^-_{2,j}$  und  $\mathbf{b}^-_{1,j}$  sowie zu  $\mathbf{b}^+_{1,j}$  und  $\mathbf{b}^+_{2,j}$ .

## 2. Qualitätsanalyse - Surface Interrogation

### 2.1. Problemhintergrund

Eine entworfene Freiformfläche kann am Bildschirm perfekt aussehen, jedoch trotzdem Qualitätsmängel aufweisen. Eine Fehlerquelle kann die begrenzte Auflösung der graphischen Darstellung sein, so daß kleine Mängel visuell nicht



wahrgenommen werden. Eine andere Fehlerquelle können Digitalisierungsfehler sein. Sie entstehen z.B. dann, wenn die Datenwerte beim Entwurf von einem digitalen Eingabegerät kommen (z.B. Datentablett). Drittens können Mängel entstehen, wenn Kontrollpunkte interaktiv verschoben werden während des Flächenentwurfs.

Es ist daher notwendig, die Qualität der Fläche durch indirekte Mittel zu kontrollieren. Ziel ist das Erkennen und möglichst automatische Beseitigen der Qualitätsmängel. Das Problem dabei ist, daß es kein exaktes Maß für die Qualität einer Freiformfläche gibt. Häufig werden jedoch folgende Kriterien gestellt:

- Stetigkeit an den Übergängen von Teilflächen bei zusammengesetzten Flächen, z.B. wird oft  $C^2$ -Stetigkeit bei bikubischen Teilflächen gefordert,
- Konvexität, d.h. möglichst wenig Wechsel zwischen konvex und konkav,
- Vermeidung von Flachpunkten, das sind Punkte, in denen sowohl die Gauss-Krümmung als auch die mittlere Krümmung gleich Null sind. Hier die Definitionen der Gauss-Krümmung  $K$

$$K = \kappa_1 \cdot \kappa_2$$

und der mittleren Krümmung  $H$

$$H = \frac{\kappa_1 + \kappa_2}{2}$$

mit  $\kappa_1$  und  $\kappa_2$  sind die Hauptkrümmungen der Fläche.

## **2.2. Verfahren der Qualitätsanalyse**

Es existiert bisher kein Verfahren, das für alle Kriterien gleich gut anwendbar ist. Jedes Verfahren ist für einige Kriterien besser geeignet, für andere weniger gut. Bekannte Verfahren sind Krümmungsplots, Fokalflächen, Reflexionslinien, Orthotomics und Isophoten. Diese Verfahren werden in den nächsten Abschnitten kurz vorgestellt.

### **2.2.1. Krümmungsplots**

Das Verfahren der Krümmungsplots wurde von Farin [1990] vorgeschlagen. Der Krümmungsplot ist der Graph resultierend aus der Krümmung abgetragen gegen die

Bogenlänge. Für Freiformflächen lassen sich u.a. die Gauss-Krümmung und die mittlere Krümmung anwenden. Farin führt außerdem eine vorzeichenbehaftete Krümmung ein, bei der Vorzeichenwechsel Flachpunkte anzeigen.

Er definiert eine qualitativ gute Kurve wie folgt: Eine Kurve ist qualitativ gut, wenn der Krümmungsplot stetig ist und nur aus wenigen monotonen Stücken besteht. Bereiche mit monotoner Krümmung sind voneinander getrennt durch Punkte mit Extremkrümmungen. Daher sollte die Anzahl von Punkten mit Extremkrümmungen für eine qualitativ gute Kurve möglichst gering sein. Extremkrümmungen sollten nur dort auftreten, wo sie vom Designer ausdrücklich erwünscht sind. Unglücklicherweise bietet diese Definition keine genaue mathematische Handhabung. Sie mag für den Entwurf praktikabel sein, nicht jedoch für möglichst automatische Glättungsalgorithmen.

### 2.2.2. Fokalflächen

Fokalflächen sind spezielle Linienkongruenzen. Sie wurden von Pottmann u.a. [1991] auf dem Gebiet der Visualisierung eingeführt, um skalare Werte, die auf einer Oberfläche vorliegen, zu visualisieren. Sie können jedoch ebenso zur Qualitätsanalyse von Freiformflächen genutzt werden. Angenommen, es liegen die skalaren Werte  $f(u,v)$  auf der Fläche  $\mathbf{x}(u,v)$  vor und sollen dargestellt werden. Die Fokalfläche  $\mathbf{s}(u,v)$  ist dann definiert als

$$\mathbf{s}(u,v) = \mathbf{x}(u,v) + f(u,v) \cdot \mathbf{n}(u,v)$$

mit  $\mathbf{n}(u,v)$  ist die normalisierte Oberflächennormale. Die Fokalfläche ist damit ein Spezialfall der Normalen-Kongruenzen [Hagen u.a. 1992].

Eine Linienkongruenz kann als eine Menge von Linien betrachtet werden, die zwei Flächen berühren, die Fokalflächen der Linienkongruenz. Die Kontaktpunkte zwischen einer solchen Linie und den beiden Fokalflächen heißen Fokalfpunkte dieser Linie. Die Fokalfpunkte einer Normalen-Kongruenz sind gerade die Zentren der Hauptkrümmungen der Fläche. Für den Sonderfall der Kugel degenerieren die Fokalflächen zu einem Punkt.

Fokalflächen eignen sich zur Erkennung von Bereichen ungewollter Krümmungen. Generell sind Flachpunkte diejenigen Punkte einer Fläche, in denen sowohl die Gauss-Krümmung als auch die mittlere Krümmung Null sind. Flachpunkte liegen bei den Fokalflächen vor, wenn  $\mathbf{s}(u,v) = \mathbf{x}(u,v)$ , d.h. wenn die Fokalfläche die Fläche berührt.

### 2.2.3. Reflexionslinien

Reflexionslinien sind eine Standardmethode der Qualitätsanalyse, insbesondere in der Automobilindustrie. Mit ihrer computertechnischen Umsetzung lassen sich auch Verfahren zur Korrektur lokaler Oberflächenunregelmäßigkeiten entwickeln [Klass 1980, Kaufmann & Klass 1988].

Bei der Reflexionslinienmethode lassen sich Flächenbereiche mit ungewollter Krümmung anhand von Unregelmäßigkeiten in der Form von reflektierten, parallelen Lichtlinien erkennen. Sei  $\mathbf{x}(u,v)$  eine parametrische Flächenbeschreibung und  $\mathbf{n}(u,v)$  der Normalenvektor der Fläche. Sei weiterhin eine Lichtgerade  $\mathbf{l}$  gegeben mit

$$\mathbf{l}(t) = \mathbf{l}_0 + t \cdot \mathbf{s} \quad (t \in \mathbb{R}).$$

Die Reflexionslinie ist das reflektierte Bild der Lichtgeraden  $\mathbf{l}$  auf der Fläche  $\mathbf{x}$ , welches vom festem Augpunkt  $\mathbf{a}$  aus gesehen wird. Dabei gilt das Reflexionsgesetz für ideal spiegelnde Flächen, d.h. der Winkel zwischen dem Normalenvektor  $\mathbf{n}$  und dem Vektor gebildet aus einem Punkt der Lichtlinie  $\mathbf{l}$  und einem Flächenpunkt  $\mathbf{p}$  ist gleich dem Winkel zwischen dem Normalenvektor  $\mathbf{n}$  und dem Vektor gebildet aus dem Augpunkt  $\mathbf{a}$  und dem Flächenpunkt  $\mathbf{p}$  (Einfallswinkel = Ausfallswinkel). Zur Berechnung der Reflexionslinie kann auf folgende Beziehung zurückgegriffen werden

$$\frac{\mathbf{a}}{\|\mathbf{a}\|} + \frac{\mathbf{b}}{\|\mathbf{b}\|} = 2 \mathbf{n} \langle \mathbf{n}, \frac{\mathbf{b}}{\|\mathbf{b}\|} \rangle$$

mit  $\mathbf{g} = \mathbf{p} - \mathbf{a}$  und  $\mathbf{h} = \mathbf{l} - \mathbf{p}$  und  $\langle, \rangle$  bezeichnet das Skalarprodukt.

Um die Qualität einer Fläche zu begutachten, verwendet man eine Schar paralleler Lichtgeraden, die in einer Ebene liegen. Flächenunregelmäßigkeiten führen zu Unregelmäßigkeiten der Reflexionslinien. Jedoch ist festzustellen, daß sich die Reflexionslinienmethode vor allem für das Erkennen globaler Unregelmäßigkeiten eignet, d.h. zur Begutachtung der ästhetischen Qualität einer Fläche, da feine, lokal begrenzte Flächenunregelmäßigkeiten u.U. nicht detektiert werden [Farin 1990].

#### 2.2.4. Orthotomics

Sei  $\mathbf{x}(u,v)$  die parametrische Beschreibung einer Fläche, sei  $\mathbf{p}$  ein Punkt, der nicht auf  $\mathbf{x}$  und auch auf keiner Tangentenebene von  $\mathbf{x}$  liegt. Dann wird  $\mathbf{p}$  von einer Tangentenebene von  $\mathbf{x}$  reflektiert und der Abstand zwischen  $\mathbf{p}$  und der Tangentenebene von  $\mathbf{x}$  wird mit dem Faktor  $k$  multipliziert. Man erhält die Parameterform der  $k$ -orthotomischen Fläche von  $\mathbf{x}$  bezüglich  $\mathbf{p}$  aus

$$\mathbf{y}_k(u,v) = \mathbf{p} + k \langle \mathbf{x}(u,v) - \mathbf{p}, \mathbf{n}(u,v) \rangle \mathbf{n}(u,v)$$

wobei  $\mathbf{n}$  die Flächennormale und  $\langle, \rangle$  das Skalarprodukt bezeichnet.

Die  $k$ -orthotomische Fläche  $\mathbf{y}_k$  der Fläche  $\mathbf{x}$  bezüglich  $\mathbf{p}$  hat eine Singularität im Punkt  $(u_0, v_0)$  genau dann, wenn die Gauss-Krümmung für  $\mathbf{x}(u_0, v_0) = 0$  ist oder in  $(u_0, v_0)$  ihr Vorzeichen wechselt [Hagen u.a. 1992]. Orthotomics eignen sich damit für den Konvexitätstest.

#### 2.2.5. Isophoten

Isophoten sind Linien, deren Punkte die gleiche Lichtintensität besitzen. Sei  $\mathbf{x}(u,v)$  die Beschreibung einer Fläche in Parameterform und  $\mathbf{r}$  die Richtung der parallelen Lichtstrahlen, dann gilt für eine Isophote folgende Bedingung

$$\langle \mathbf{n}(u,v), \mathbf{r} \rangle = \cos \alpha = \text{const}$$

mit  $\mathbf{n}$  ist die Flächennormale, o.B.d.A.  $\|\mathbf{n}\| = \|\mathbf{r}\| = 1$ ,  $\alpha$  ist der eingeschlossene Winkel von  $\mathbf{n}$  und  $\mathbf{r}$ , und  $\langle, \rangle$  bezeichnet das Skalarprodukt [Poeschl 1984]. Anschaulich ausgedrückt ist eine Isophote die Linie einer Fläche, bei der der Winkel  $\alpha$  konstant ist.

Isophoten haben folgende Eigenschaften. Sie eignen sich zur Darstellung von Unregelmäßigkeiten in der ersten und zweiten Ableitung und in der Gauss-Krümmung einer Fläche. Die Unregelmäßigkeiten werden dabei an den Übergängen der Teilflächen der Freiformfläche deutlich. Isophoten sind einfach zu berechnen. Bei einer festen Lichtrichtung  $\mathbf{r}$  geht durch jeden Punkt der Fläche  $\mathbf{x}(u,v)$  genau eine Isophote. Eine Ausnahme hiervon bilden kritische Punkte, durch die entweder keine Isophote oder mehrere Isophoten verlaufen (Abb. 2.1.).



Abb. 2.1. Isophoten für kritische Punkte. Links ein Punkt, durch den keine Isophote verläuft. Rechts ein Punkt durch den mehrere Isophoten verlaufen.

Eine wichtige Eigenschaft der Isophoten ist die folgende Beziehung: Ist die Fläche  $C^n$ -stetig, dann ist die Isophote  $C^{n-1}$ -stetig. Damit läßt sich die Qualität einer Fläche beurteilen, da an den Übergängen der Teilflächen Stetigkeitssprünge auftreten, wenn nicht gewisse Stetigkeitsbedingungen eingehalten wurden.

Für ebene Bereiche einer Fläche sind alle Normalen parallel und Isophoten existieren nicht. Für diesen Fall funktioniert die Isophotenmethode nicht. Hingegen können für gekrümmte Flächenbereiche gute Ergebnisse erwartet werden. Die Flachpunkte einer Fläche werden erkannt. Es sind diejenigen kritischen Punkte, die bei Änderung der Lichtrichtung  $\mathbf{r}$  ihre Position auf der Fläche  $\mathbf{x}(u,v)$  nicht ändern. Daraus resultiert die Forderung an das Testbed, die Position der Lichtquelle bzw. des Augpunktes verändern zu können. Letzteres ist jedoch rechentechnisch bedeutend aufwendiger, da die Sichtbarkeitsberechnungen komplett neu durchgeführt werden müßten. Die Interaktivität würde bei diesen hohen Zeitanforderungen verloren gehen. Sie bleibt jedoch erhalten, wenn einfach die Position der Lichtquelle verändert wird. In diesem Fall können die zuvor berechneten Normalenvektoren wieder genutzt werden. Eine sehr schnelle Berechnung der Isophoten ist damit möglich, die durchaus Echtzeitanforderungen genügt je nach Anzahl der insgesamt zu berechnenden Bildpunkte.

Weiterhin läßt sich die Isophotenkrümmung für jeden Punkt der Fläche  $\mathbf{x}(u,v)$  berechnen. Dies ist von Bedeutung für die Methode der Flächenglättung durch Minimierung der Differenzen der Isophotenkrümmungen (s. Abschnitt 4.1.2.).

### 3. Glätten von Freiformflächen - Surface Fairing

Nachdem Flächenunregelmäßigkeiten erkannt wurden, möchte man diese möglichst einfach beseitigen. Dazu muß die Fläche verändert werden, d.h. die die Fläche definierenden Kontrollpunkte müssen verschoben werden. Durch das Zulassen kleiner Verschiebungen der Kontrollpunkte versucht man, die Qualitätsmängel zu beseitigen. Wichtig ist hierbei, daß das Design der Fläche so wenig wie möglich, aber so viel wie nötig geändert wird und daß sich die Stetigkeit der Fläche an den Übergängen der Teilflächen nicht verschlechtert. Die erste Forderung ist der Grund dafür, daß nur kleine Verschiebungen erlaubt werden. Häufig sind dies Werte im Bereich kleiner als 1mm. Die Einhaltung der zweiten Forderung läßt sich bei der Glättung kontrollieren. Für B-Spline-Flächen läßt sich diese Forderung einfach verwirklichen, da die Kontrollpunkte (de Boor-Punkte) nur lokale Wirkung haben. Bei Bezier-Spline-Flächen hingegen besitzen die Kontrollpunkte (Bezierpunkte) nur eine pseudo-lokale Wirkung, das bedeutet insgesamt globale Wirkung. Würde nur ein Kontrollpunkt verschoben, so kommt es zu ungewollten Schwingungen der Fläche. Deswegen müssen benachbarte Kontrollpunkte mitverschoben werden. Sei z.B. eine  $C^2$ -stetige bikubische Bezier-Spline-Fläche mit uniformer Parametrisierung gegeben. Soll der Kontrollpunkt  $\mathbf{k}_{i,j} = \mathbf{b}_{3i,3j}$  um den Vektor  $\mathbf{r}$  verschoben werden, so müssen die benachbarten Bezierpunkte im Abstand von bis zu 3 Bezierpunkten vom Kontrollpunkt wie folgt geändert werden:

$$\begin{aligned} \mathbf{k}'_{i,j} &= \mathbf{b}'_{3i,3j} = \mathbf{b}_{3i,3j} + \mathbf{r} \\ \mathbf{b}'_{3i+p,3j+q} &= \mathbf{b}_{3i+p,3j+q} + \mathbf{r} \quad \text{für } \max\{\|p\|, \|q\|\} = 1 \\ \mathbf{b}'_{3i+p,3j+q} &= \mathbf{b}_{3i+p,3j+q} + \mathbf{r} / 2 \quad \text{für } \max\{\|p\|, \|q\|\} = 2 \\ \mathbf{b}'_{3i+p,3j+q} &= \mathbf{b}_{3i+p,3j+q} + \mathbf{r} / 4 \quad \text{für } \max\{\|p\|, \|q\|\} = 3 \end{aligned}$$

Die restlichen Bezierpunkte bleiben unverändert.

Das Verschieben der Kontrollpunkte kann u.a. interaktiv erfolgen, bis z.B. die Isophoten oder Reflexionslinien „gut“ aussehen. Jedoch ist dies ein sehr zeitaufwendiges Verfahren und erfordert eine gewisse Übung des Anwenders. Gesucht werden daher automatische Verfahren, die einfach zu bedienen sind, effektiv arbeiten und möglichst parametrisierungsunabhängig sind. Einige vorgestellte automatische Verfahren sind die Minimierung der Biegeenergie, die Minimierung der Differenzen der 3. Partiellen Ableitungen oder die Minimierung der Differenzen der Isophotenkrümmungen um den Kontrollpunkt. Bei diesen Verfahren wird interaktiv ein Flächenbereich zur Glättung ausgewählt, auf den dann der Glättungsalgorithmus angewandt wird. In Zukunft interessant sind Verfahren, die

ebenso die Auswahl des zu glättenden Flächenbereiches automatisch bewerkstelligen.

Im allgemeinen werden beim Entwurf der Fläche gewisse Kontrollpunkte festgelegt, in denen die Fläche die Punkte genau interpolieren soll. Dies sind die Eckpunkte der Teilflächen, aus denen sich die Gesamtfläche zusammensetzt. Die restlichen Kontrollpunkte werden dann so berechnet, daß sie den gewünschten Stetigkeitsbedingungen an den Übergängen der Teilflächen entsprechen. Für die Flächenglättung sind die gemeinsamen Kontrollpunkte in den Eckpunkten der Teilflächen von Interesse. Sie sollen verschoben werden.

### **3.1. Glättungsalgorithmen**

Generell besteht ein Glättungsalgorithmus aus folgenden Teilschritten. Erstens, Berechnung des Qualitätsmaßes an den ausgewählten Kontrollpunkten. Im zweiten Schritt wird dieser skalare Wert minimiert. Dabei erhält man eine Verschiebungsvektor (Richtung und Weite). Anschließend wird die Verschiebung durchgeführt. Abschließend wird die veränderte Fläche erneut der Qualitätsanalyse unterworfen. Da die Glättung ein iterativer Prozeß ist, wird das Ergebnis mit einem bestimmten Endkriterium verglichen. Ist dieses nicht erfüllt, so wird ein weiterer Iterationsschritt durchgeführt. Ein mögliches und sinnvolles Endkriterium ist z.B. das Unterschreiten eines Schwellwertes durch die Differenz aus alter und neuer Kontrollpunktposition oder das Erreichen einer vorgegebenen Anzahl von Iterationsschritten. Weiterhin sollte die Iteration beendet werden, wenn die maximal erlaubte Verschiebungsweite erreicht wurde.

Ein anderer Ansatz ist, die Flächenglättung auf die Kurvenglättung zurückzuführen [Farin 1990]. Hierzu betrachtet man die Fläche entlang der  $u$ - bzw.  $v$ -Parameterlinien, an denen die Fläche durch Kontrollpunkte definiert sind. Nun glättet man zunächst die isoparametrischen Kurven in Richtung des Parameters  $u$  ( $v = \text{const}$ ), dann die isoparametrischen Kurven in Richtung des Parameters  $v$  ( $u = \text{const}$ ) mit der oben beschriebenen Methode der Kurvenglättung, d.h. erst zeilenweise glätten, dann spaltenweise glätten. Im allgemeinen sollen die dabei entstehenden Flächen „glatter“ sein. Farin schlägt vor, die Krümmungsplots für die Flächenglättung zu nutzen. Man sucht hierbei nach dem größten Anstieg in der Krümmungsfunktion und versucht an dieser Stelle die Kurve zu glätten. Dazu wird der an dieser Stelle

liegende Kontrollpunkt so verschoben, daß aus den beiden Kurvenstücken links und rechts vom Kontrollpunkt ein einziges Kurvenstück wird, d.h. der „störende“ Kontrollpunkt entfernt werden könnte (not a knot-condition). Kurvenmängel können nur an Kontrollpunkten auftreten, da die Kurven zwischen den Kontrollpunkten  $C^\infty$ -stetig sind.

Im folgenden sollen die beiden Glättungsverfahren vorgestellt werden, die für die vorliegende Studienarbeit ausgewählt wurden. Hierzu betrachten wir eine Fläche gegeben in ihrer parametrischen Form  $\mathbf{x}(u,v)$ .

### **3.1.1. Minimierung der Differenzen der 3. partiellen Ableitungen**

Bei diesem Verfahren betrachten wir einen gemeinsamen Kontrollpunkt von vier aneinanderstoßenden Teilflächen, d.h. den Punkt, an dem die „Ecken“ aller vier Flächen zusammenstoßen. Abbildung 3.1. zeigt die Anordnung der Teilflächen. Für jede Teilfläche können die 3. partiellen Ableitungen  $uuu$  bzw.  $vvv$  der Parameter  $u$  bzw.  $v$  in eben diesem Kontrollpunkt berechnet werden. Durchläuft man die vier Teilflächen in einer kreisförmigen Reihenfolge, so lassen sich für beide Parameter  $u$  und  $v$  die Differenzen der 3. partiellen Ableitungen im Kontrollpunkt für jeweils zwei aufeinanderfolgende Teilflächen berechnen. Die Quadrate dieser Differenzen werden aufsummiert. Die entstehende Summe ist der gesuchte skalare Wert, der zu minimieren ist. Hintergrund ist, daß beim völlig glatten Übergang zweier Teilflächen die 3. partiellen Ableitungen gleich sind, d.h. die Differenzen gleich Null sind, folglich die Summe auch Null ist. Ist dies für alle Übergänge der Fall, so hat man an diesem Kontrollpunkt eine geglättete Fläche erreicht. Daher wird versucht, die Summe zu minimieren, wobei die maximal erlaubte Verschiebungsweite eines Kontrollpunktes zu beachten ist.

Die Betrachtung der Differenzen der 3. partiellen Ableitungen ist das einfachste Verfahren, um Flächeneigenschaften 3. Ordnung zu behandeln. Nachteil dieses Verfahrens ist die Abhängigkeit von der gewählten Parametrisierung.



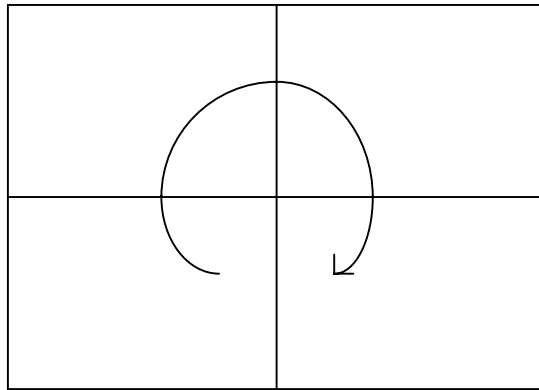


Abb. 3.1. zeigt die Anordnung der Teilflächen und die Reihenfolge des Durchlaufens für die Berechnung des Optimierungsmaßes.

### 3.1.2. Minimierung der Differenzen der Isophotenkrümmungen

Wie zuvor betrachten wir einen gemeinsamen Kontrollpunkt von vier aneinanderstoßenden Teilflächen (Abb. 3.1.). Für jede Teilfläche läßt sich die Krümmung der Isophote am Kontrollpunkt berechnen. Durchläuft man nun wieder die vier aneinanderstoßenden Teilflächen kreisförmig, so läßt sich jeweils die Differenz der Isophotenkrümmungen im gemeinsamen Kontrollpunkt zweier aufeinanderfolgender Teilflächen berechnen. Diese Differenzen werden aufsummiert und die entstehende Summe ist das gesuchte Optimierungsmaß. Für glatte Übergänge an den Teilflächen wird erwartet, daß die Isophotenkrümmungen den gleichen Wert haben, d.h. die Differenzen gleich Null sind und somit auch die Summe gleich Null ist. Sind die Isophotenkrümmungen zweier aufeinanderfolgender Teilflächen unterschiedlich, so bedeutet dies, daß die Isophoten an den Übergängen unstetig sind und damit eine Unstetigkeit der Fläche vorliegt. Zur Flächenglättung wird folglich versucht, die Summe zu minimieren, wobei wiederum die maximal erlaubte Verschiebungsweite zu beachten ist.

Die Minimierung der Differenzen der Isophotenkrümmungen ist ein weiteres einfaches Verfahren, um Flächeneigenschaften 3. Ordnung zu behandeln. Es besitzt den Vorteil, unabhängig von der gewählten Parametrisierung zu sein.

## **4. Anforderungen an das Testbed**

In diesem Abschnitt werden die allgemeinen Anforderungen an das Testbed definiert. Die praktische Umsetzung dieser Forderungen wird im Abschnitt 5 beschrieben.

Das Testbed soll die Untersuchung von Glättungsalgorithmen für Freiformflächen ermöglichen. Dazu muß das Testbed vorhandene Flächen, die in Dateien gespeichert sind, einlesen können. Es ist nicht Aufgabe des Testbeds, Freiformflächen zu entwerfen, da diese Testumgebung für den Zweck der Untersuchung von Glättungsalgorithmen geschaffen wurde.

Weiterhin soll es mit dem Testbed möglich sein, die eingelesenen Flächen bzw. Teile von ihnen graphisch auszugeben. Dabei waren zwei Varianten der graphischen Ausgabe gefordert, einmal als Drahtmodell (Wireframe-Struktur) und zum anderen mittels Raytracing zur Berechnung der Sichtbarkeit und als Schattierungstechnik.

Um Glättungsalgorithmen auf Freiformflächen bzw. Teile von ihnen sinnvoll anwenden zu können, ist es unerlässlich, Eigenschaften der Flächen zur Begutachtung der Qualität zu berechnen und graphisch auszugeben. In der vorliegenden Arbeit wurden Isophoten als Mittel der Qualitätsanalyse (Surface Interrogation) angewandt. Anhand der dargestellten Eigenschaften kann der Anwender visuell feststellen, welcher Flächenbereich für die Glättung ausgewählt werden soll. Das Testbed muß folglich die interaktive Auswahl eines solchen Flächenbereiches unterstützen. Auf dem ausgewählten Flächenbereich können dann die zu untersuchenden Glättungsalgorithmen angewandt werden und auf ihre Nützlichkeit untersucht werden.

### **4.1. Beschreibungsformat für die Flächen**

Das auszuwählende Beschreibungsformat für die in Dateien gespeicherten Flächen sollte die leichte Editierbarkeit der Flächen durch den Anwender gewährleisten, da das Testbed selbst keine Möglichkeit zur Generierung neuer Flächen bietet. Gesucht wird ein einfach zu handhabendes Beschreibungsformat, das ASCII-Text-Dateien liefert, die auch für den Anwender lesbar sind. Desweiteren kann ein allgemeines Beschreibungsformat dazu dienen, den Austausch von Flächen zu anderen Programmen sicherzustellen.

## **4.2. Graphische Ausgabe**

### **4.2.1. Graphische Ausgabe der Freiformflächen**

Nach der Klassifikation von Boldt [1994] lassen sich drei grundlegende Verfahren zur Visualisierung von Freiformflächen unterscheiden, die Isokurvenextraktion, die Unterteilungsverfahren und die Iterationsverfahren. Für die vorliegende Arbeit von Interesse sind die Isokurvenextraktion und die Iterationsverfahren. Unterteilungsverfahren teilen die Fläche in kleinere Teilflächen (z.B. Drei- oder Vierecke), die entstandene Polyederapproximation wird dann visualisiert.

Bei der Isokurvenextraktion (isoparametric curves) wird die Fläche entlang von Linien im  $(u,v)$ -Parameterraum mit  $u=\text{const}$  bzw.  $v=\text{const}$  in einzelne Bildpunkte zerlegt. Dabei wird der  $(u,v)$ -Parameterraum mit einem regulären Gitter überzogen und die  $(u,v)$ -Gitterpunkte werden in den dreidimensionalen euklidischen Raum abgebildet. Dies kann bei Bezierflächen z.B. über den de Casteljau-Algorithmus erfolgen. Die Flächenpunkte lassen sich über den jeweils freien Parameter berechnen und man erhält durch Verbinden der Flächenpunkte die Approximation einer isoparametrischen Kurve. Die graphische Darstellung der isoparametrischen Kurven erzeugt ein Drahtmodell der Fläche. Je mehr Flächenpunkte berechnet werden, d.h. je feiner das Gitter ist, desto besser ist die Approximation der Fläche. Nachteil des Drahtmodells (Wireframe-Modell) ist, daß verdeckte Kanten und Flächen nicht entfernt werden, d.h. kein Hidden Line Hidden Surface Removal. Umso wichtiger ist die Notwendigkeit, sich die Fläche von verschiedenen Blickpunkten aus ansehen zu können, um dadurch die Verdeckungsbeziehungen zu erkennen.

Probleme sind die Abstandswahl zwischen den isoparametrischen Kurven und damit zusammenhängend die Wahl der zu berechnenden Punkte pro Parameterlinie, um die Fläche möglichst gut zu approximieren. Dieses Problem läßt sich auch als das Problem des Over- bzw. Undersamplings bezeichnen. Bei einem zu großen Abstand zwischen den Isokurven können kleine, aber u.U. wichtige Flächendetails verlorengehen. Bei zu kleinem Abstand sind die einzelnen isoparametrischen Kurven nicht mehr voneinander zu trennen und somit ist die Form der Fläche nur schwer auszumachen. Bei zu geringer Anzahl von zu berechnenden Flächenpunkten wirkt

die Darstellung der Fläche eckig und approximiert damit nur schlecht die wahre Gestalt der Fläche. Eine zu hohe Anzahl führt zu einem hohen Rechenaufwand, ohne zu einer Verbesserung der Genauigkeit der Darstellung beizutragen.

Die Iterationsverfahren bieten eine iterative Berechnung der Sichtbarkeit. Dazu wird ein Strahl ausgehend vom Augpunkt durch jeden Pixel der Bildfläche in die dreidimensionale Szene hineingeschossen und der mögliche Schnittpunkt zwischen Strahl und Freiformfläche mittels Iteration berechnet. Es handelt sich folglich um ein numerisches Verfahren der Annäherung an die wahre Position des Schnittpunktes und damit des Flächenpunktes. Nach Boldt unterscheidet man die Eliminationsmethode und das Verfahren der Abstandsminimierung. Die Iterationsverfahren entsprechen dem Verfahren des Raycastings für Sichtbarkeitsberechnungen und lassen sich damit gut mit dem Verfahren des Raytracings als Schattierungstechnik kombinieren. Damit lassen sich qualitativ hochwertige Bilder berechnen und darstellen. Gebräuchliche Verfahren zur Iteration sind Newton-Verfahren, Quasi-Newton-Verfahren und Gradientenverfahren. Die Iterationsverfahren bestimmen den zum jeweiligen Augpunkt am nächsten liegenden Schnittpunkt mit der Fläche. Dadurch werden verdeckte Kanten und Flächen eliminiert (Hidden Line Hidden Surface Removal) und eine realistischere Darstellung der Szene als durch das Drahtmodell erreicht.

Ein genereller Nachteil der Iterationsverfahren ist die u.U. auftretende Nichtkonvergenz, d.h. in Abhängigkeit vom Anfangswert kann es vorkommen, daß kein Schnittpunkt mit der Fläche gefunden wird, obwohl ein Schnittpunkt existiert.

Weiterhin stellt sich an das Testbed die Forderung, die eingelesene Fläche aus verschiedenen Blickwinkeln betrachten zu können. Damit der Anwender feststellen kann, welcher Flächenbereich zu glätten ist, muß er sich die Fläche von allen Seiten anschauen können. Dazu läßt sich entweder die Orientierung der Fläche oder die Lage des Augpunktes verändern. Im ersten Fall müssen die Positionen aller Kontrollpunkte Neuberechnet werden. Im zweiten Fall ändert sich die Lage und Orientierung des Sichtvolumens. Beiden Fällen gemein ist die anschließende Neuberechnung der Darstellung, da sich die Abbildung der Szene auf die Bildmatrix verändert hat.

#### **4.2.2. Graphische Ausgabe von Eigenschaften der Flächen**

Um die Glattheit einer generierten Fläche zu begutachten, ist es unerlässlich, Eigenschaften der Flächen zu berechnen und zu visualisieren. Das Testbed muß folglich neben der Darstellung der Flächen auch die Darstellung von Eigenschaften zur Qualitätsanalyse von Freiformflächen unterstützen (siehe Abschnitt 2).

Für die vorliegende Arbeit stand die Forderung der Nutzung von Isophoten als Mittel der Qualitätsanalyse. Fehler in der Glattheit von Flächen lassen sich bei Nutzung von Isophoten besonders gut erkennen, wenn die Isophoten für unterschiedliche Lichtrichtungen berechnet und dargestellt werden. Daraus resultiert die Forderung an die Testumgebung, die Position der Punktlichtquelle verändern zu können.

#### **4.3. Auswahl des zu glättenden Flächenbereiches**

Durch die Visualisierung bestimmter Eigenschaften der Freiformfläche läßt sich erkennen, welcher Flächenbereich zu glätten ist. Die Testumgebung muß folglich die Auswahl eines solchen Flächenbereiches ermöglichen. Wurde ein Flächenbereich ausgewählt, so ist festzustellen, welche Kontrollpunkte in diesen Bereich abbilden. Betrachtet man die Teilflächen, aus denen die Gesamtfläche zusammengesetzt ist, so sind für die Glättung diejenigen Kontrollpunkte von Interesse, an denen vier Teilflächen zusammenstoßen, d.h. die Eckpunkte der Teilflächen (siehe Abschnitt 3).

#### **4.4. Parametersteuerung**

Eine wichtige Forderung an das Testbed ist die Möglichkeit der Steuerung bestimmter Parameter. Dazu gehören die Parameter zur Steuerung der Position des Augpunktes, der Position der Lichtquelle, der Anzahl und damit des Abstandes der isoparametrischen Kurven für die Berechnung des Drahtmodells, die „Breite“ der Isophotenlinien und die maximal erlaubte Verschiebung eines Kontrollpunktes im Verlauf der Glättung. Wo immer möglich sollte die Veränderung eines Parameters nicht zur kompletten Neuberechnung der Darstellung führen, um eine effektive, interaktive Nutzung sicherzustellen.

## 5. Praktische Umsetzung

Dieser Abschnitt beschreibt die praktische Umsetzung der in den vorangegangenen Abschnitten vorgestellten Theorie und der Anforderungen an das Testbed.

### 5.1. Allgemeines

Das Testbed sollte unter Nutzung der Open GL (Graphics Library) auf einer Silicon Graphics erstellt werden. Die Open GL ist eine weitverbreitete Bibliothek von Funktionen zur Programmierung graphischer Darstellungen.

Für die Testumgebung wird eine Benutzungsschnittstelle in Form einer graphischen Oberfläche (Graphical User Interface) benötigt. Eine Möglichkeit der Implementierung einer derartigen Oberfläche besteht mit den zahlreichen Funktionen der X-Bibliothek. Die Programmierung mit der X-Bibliothek ist jedoch relativ aufwendig. Daher wurde für die vorliegende Testumgebung auf die Programmiersprache Tcl und die Toolkit Tk, kurz Tcl/Tk, von Ousterhout [1993] zurückgegriffen. Die Toolkit Tk ist in Tcl geschrieben und stellt die Funktionalität zur Programmierung der graphischen Oberfläche bereit (Widgets, Eventhandling, etc.). Tcl/Tk bietet eine sehr einfache Handhabung der Programmierung von graphischen Oberflächen. Insbesondere die Handhabung der graphischen Ein- und Ausgabe-prozesse wird erleichtert. Ein Nachteil von Tcl/Tk ist die interpretative Arbeitsweise, d.h. ein Programm wird erst zur Laufzeit übersetzt und abgearbeitet. Damit liegt die benötigte Rechenzeit bei Ausführung eines Programms wesentlich höher als bei einem zuvor compilierten Programm, wie es z.B. bei der Programmierung der Oberfläche mit Hilfe der X-Bibliothek möglich wäre. Dieser Nachteil kann jedoch umgegangen werden, da es die Programmiersprache Tcl erlaubt, eigene C-Funktionen als neue Tcl-Kommandos einzubinden und zu nutzen. Hierdurch ist es möglich, rechenintensive Funktionen in C zu programmieren und in compilierter Form im Tcl/Tk-Programm zu nutzen, während gleichzeitig die Vorteile von Tcl/Tk bei der Verwaltung der Benutzungsschnittstelle genutzt werden können.

Weiterhin umgeht man auf diese Weise einen weiteren Nachteil der Tcl-Sprache, daß alle Variablen grundsätzlich Strings sind und außer Listen und Feldern von

Strings keine Möglichkeit zum Aufbau komplizierterer Datenstrukturen besteht. Zur Verarbeitung numerischer Daten können Strings numerisch evaluiert werden. Für den Programmierer interessante (und damit oft kompliziertere) Datenstrukturen können jedoch in den einzubindenden C-Funktionen definiert und genutzt werden.

Tcl-Interpreter können auch von C-Programmen aus gestartet und genutzt werden. Über die Initialisierung bestimmter Tcl-Routinen läßt sich die Kommunikation zwischen dem C-Programm und der unter Tcl/Tk verwalteten Benutzungsschnittstelle herstellen. Damit ist es möglich, sämtliche rechenintensive Funktionen in C zu programmieren und nur ihre Ergebnisse bei Ausführung des Programms an die graphische Oberfläche weiterzuleiten, die dann für die entsprechende graphische Repräsentation sorgt. Gleichzeitig verwaltet die Benutzungsschnittstelle die Eingaben des Nutzers und startet die entsprechenden C-Funktionen zur Bearbeitung der Nutzereingaben.

## **5.2. Einbindung der Open GL unter Tcl/Tk**

Für die Nutzung der Open GL unter Tcl/Tk steht die Togl-(Tcl and Open GL)-Bibliothek zur Verfügung [Quelle]. Sie stellt die Verbindung zwischen der Verwaltung der graphischen Benutzungsschnittstelle und den Befehlen der Open GL zur Graphikprogrammierung her. Dazu wird ein neues Window angelegt, das ein sogenanntes Togl-Widget enthält, welches die Renderingbefehle der Open GL in eine entsprechende graphische Darstellung umwandelt und Eingaben im Togl-Widget als Events an Tcl/Tk weiterleitet. Tcl/Tk ist dann für die Eventsteuerung verantwortlich. Mit Togl kann die gesamte Funktionalität der Open GL unter Tcl/Tk genutzt werden.

Ein an der TU Illmenau entwickeltes Programm (TIGER) bietet eine komplette Umsetzung der Open GL unter Tcl/Tk an [Barth 1996]. Damit können sämtliche Befehle der Open GL durch den Tcl-Interpreter genutzt werden. Diese Variante wurde jedoch nicht für die vorliegende Studienarbeit genutzt, da der gesamte Ablauf und die Berechnungen für die graphische Ausgabe interpretativ erfolgen würde und damit einen erheblich höheren Rechenaufwand nach sich ziehen würde. Da die Zeitanforderungen für das Testbed eine wichtige Größe darstellen, wurde eine andere Umsetzung gewählt.

Hierbei wird die Togl-Bibliothek im C-Programm mit eingeschlossen. Über Initialisierungsroutinen wird die Kommunikation zwischen einem zuvor gestarteten Tcl-Interpreter und den Togl-Funktionen beim Start des C-Programms hergestellt. Anschließend können neue Togl-Widgets definiert werden und die Rendering-ergebnisse in sie umgeleitet werden.

### **5.3. Beschreibungsformat ENFF und Dateiverwaltung des Testbeds**

Für das Testbed wurde nach einem einfach zu handhabenden Beschreibungsformat gesucht, das für den Anwender lesbare ASCII-Text-Dateien beschreibt. Ein solches Format ist das Extended Neutral File Format (ENFF) von Eric Haynes [1993]. Es baut auf dem von ihm entwickelten Neutral File Format (NFF) auf [Haynes 1988] und erweitert es u.a. um die Beschreibung von Bezierflächen und NURBS. Das NFF wurde als minimale Szenenbeschreibungssprache entwickelt, um verschiedene Renderingalgorithmen zu testen.

Das ENFF beschreibt die Geometrie und die grundlegenden Oberflächeneigenschaften (Farbe, Schattierungsparameter) von Objekten, die Position von Punktlichtquellen mit ihren Eigenschaften (Lichtfarbe, Intensität), die Hintergrundfarbe und das Sichtvolumen. Das ENFF erfüllt damit die gestellten Bedingungen für das Dateibeschreibungsformat für das Testbed. Es wurde am Fachbereich Informatik der Universität Rostock bereits mehrfach für die Beschreibung von Freiformflächen genutzt. Prinzipiell bietet das ENFF keine besonderen Vor- oder Nachteile gegenüber anderen Formaten, die es für den Einsatz in der vorliegenden Testumgebung prädestinieren würden. Auch andere Beschreibungsformate wären möglich gewesen, wie z.B. das OBJ-File-Format aus Wavefronts Advanced Visualizer. Andere Beschreibungsformate wie das Materials-and-Geometry-(MGF)-Format oder das DXF-AutoCAD-Format hätten erst um die Beschreibungsmöglichkeiten für Freiformflächen erweitert werden müssen.

Für das Testbed wird eine Teilmenge der Beschreibungsmöglichkeiten des ENFF benötigt, nämlich die Beschreibung der Viewing-Parameter, der Lichtparameter, der Farbparameter, der Materialeigenschaften der Flächen sowie der Geometrie von Bezier- und NURBS-Flächen.

Wird eine ENFF-Datei zum Lesen geöffnet, so werden die Daten in eine interne Datenstruktur umgewandelt und gespeichert. Die Datei kann entsprechend der



ENFF-Spezifikation Kommentarzeilen enthalten. Eine Datei muß die Viewingparameter enthalten, um akzeptiert zu werden. Weiterhin müssen für die darzustellenden Flächen wenigstens einmal die Farbwerte und Materialeigenschaften definiert werden. Die einzelnen Datenwerte werden in der Reihenfolge der Spezifikation des ENFF erwartet. Grobe Fehler in dieser Reihenfolge bzw. fehlende Datenwerte werden erkannt und das Einlesen der Datei abgebrochen. Fehlerhafte Reihenfolgen können zu Fehlern in der Darstellung führen. In diesem Fall obliegt es dem Anwender, für eine Korrektur in der Datei zu sorgen. Am Ende einer Datei können die Datenwerte der Sichtbarkeitsberechnungen, die RGB-Werte des Raytracings als Schattierungstechnik und die Skalarprodukte der Isophotenberechnung stehen, sofern sie zum Zeitpunkt der Speicherung der Datei berechnet waren. Dies ermöglicht eine erheblich schnellere Berechnung der Darstellung und eine bereits einmal berechnete Szene braucht nicht ein zweites Mal berechnet werden.

Vom Testbed aus kann die aktuelle Szene gespeichert werden. Hierzu werden neben den Daten, die die Szene und die Objekte beschreiben, auch die Ergebnisse der Sichtbarkeitsberechnungen, die RGB-Werte des Raytracings als Schattierungstechnik und die Skalarprodukte der Isophotenberechnung in die Datei geschrieben, sofern letztere zum Zeitpunkt der Speicherung berechnet waren.

#### **5.4. Benutzungsschnittstelle**

Der Aufbau der Benutzungsschnittstelle der Testumgebung ist zweigeteilt. Das erste Window umfaßt die graphische Oberfläche zur Steuerung der Testumgebung sowie der Parameter für die graphische Ausgabe (Abb. 5.1.). Das zweite Window enthält das oben beschriebene Togl-Widget, das verantwortlich für das Rendering ist.

##### **5.4.1. Parametersteuerung**

Die graphische Oberfläche besitzt zwei Hauptaufgaben. Zum einen erfolgt hier die Ablaufsteuerung des Testbeds durch den Anwender, zum anderen die Parametersteuerung. Die Ablaufsteuerung durch den Anwender erfolgt menügesteuert. Der

Anwender kann Menüoptionen zum Lesen oder Schreiben einer ENFF-Datei, zur Wahl des Renderingmodus (Wireframemodell, Raytracing) bzw. zur Darstellung von Flächeneigenschaften (Isophoten), zur Steuerung der Position des Blickpunktes und der Lichtquelle sowie zur Ausführung eines Glättungsalgorithmus (Minimierung der Differenzen der 3. partiellen Ableitungen bzw. der Isophotenkrümmungen am Kontrollpunkt) auswählen. Weiterhin steht eine Menüoption zur Hilfe bereit, die eine kurze Erläuterung der Funktionsweise des Testbeds gibt. Die Menüsteuerung ist so angelegt, daß nur die zum jeweiligen Zeitpunkt sinnvollen Optionen durch den Anwender ausgewählt werden können, wodurch die Testumgebung robust gegenüber Fehleingaben des Anwenders ist. So muß z.B. erst eine Datei geladen werden, bevor die Menüoptionen zur Wahl des Renderingsmodus, der Positionierung von Blickpunkt und Lichtquelle, und der Flächenglättung zur Verfügung stehen.

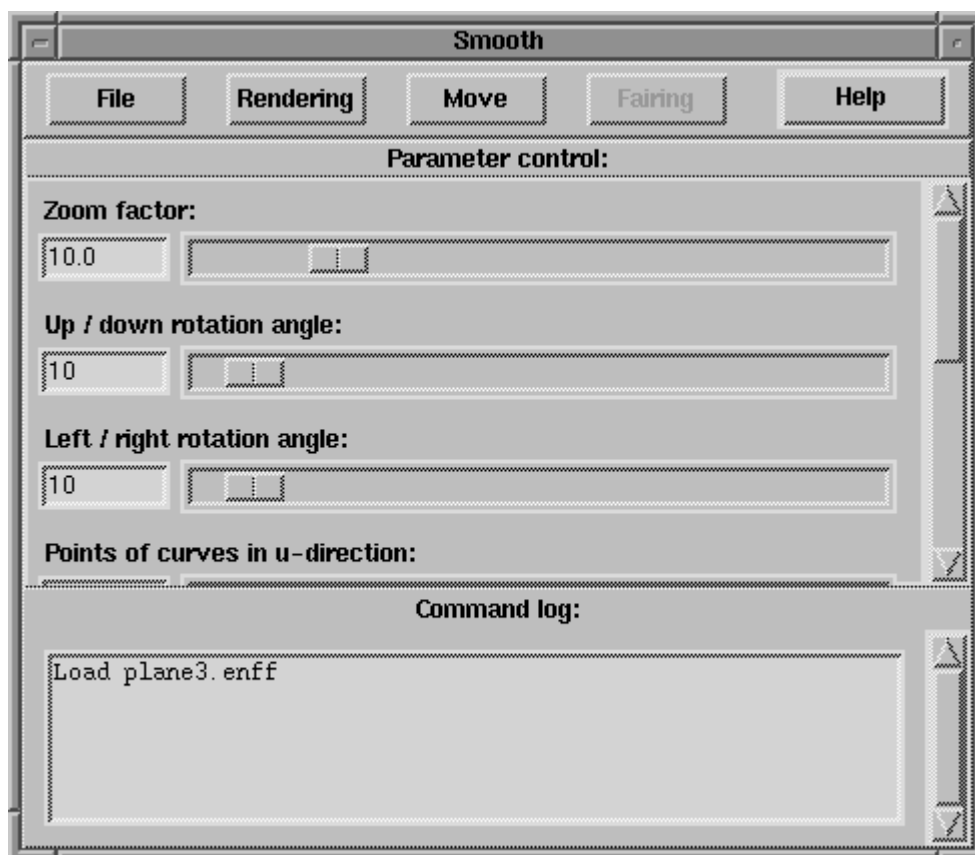


Abb. 5.1. Die graphische Benutzungsoberfläche zur Programmablaufsteuerung und Parametersteuerung.

Im zweiten Abschnitt der graphischen Oberfläche erfolgt die eigentliche Parametersteuerung. Durch den Anwender können folgende Parameter der Testumgebung gesteuert werden:

- ein Zoomfaktor für das Vergrößern bzw. Verkleinern der Darstellung,
- die Rotationswinkel für die Auf-/Ab- bzw. Links-/Rechts-Bewegung des Augpunktes bzw. der Punktlichtquelle,
- die Anzahl der zu berechnenden Punkte pro isoparametrischer Kurve in u- bzw. v-Richtung einer Freiformfläche bei der Wireframe-Darstellung,
- die Breite eines Isophotenintervalls, und
- die maximal erlaubte Verschiebungsweite eines Kontrollpunktes bei der Flächenglättung.

Der jeweilige Parameter kann sowohl über ein Eingabefeld direkt eingegeben werden, als auch über einen Slider gesteuert werden. Die Änderung einiger Parameter bewirkt eine sofortige Änderung in der graphischen Darstellung. Dies betrifft die Anzahl der zu berechnenden Punkte pro isoparametrischer Kurve und den Parameter für die Breite eines Isophotenintervalls. Da hierbei keine rechentechnisch aufwendigen Schnittpunktberechnungen durchgeführt werden müssen, können die Veränderungen unmittelbar sichtbar gemacht werden.

Weiterhin umfaßt die graphische Oberfläche einen Bereich, der die ausgeführten Befehle protokolliert (Command Log), so daß der Anwender Veränderungen der graphischen Darstellung auch später noch nachvollziehen kann.

#### **5.4.2. Graphische Ausgabe - Rendering**

Das Rendering der dreidimensionalen Szene erfolgt über das Togl-Widget. Beim Einlesen einer neuen Datei wird ein Fenster, das das Togl-Widget enthält, mit der in der Datei angegebenen Auflösung geöffnet. Wurden keine Werte angegeben, so wird ein Fenster in der Standardgröße 200x200 geöffnet.

Im Testbed stehen zwei Möglichkeiten des Rendering zur Verfügung, die Darstellung einer Fläche als Wireframe-Struktur und ein mittels Raytracing zur Sichtbarkeitsberechnung und als Schattierungstechnik berechnetes Bild. Die Wireframe-Struktur dient der schnellen Darstellung der grundlegenden Form der

Fläche(n). Nur in diesem Modus kann der Augpunkt verschoben werden, wohingegen die punktförmige Lichtquelle in jedem Darstellungsmodus bewegt werden kann. Die Wireframe-Darstellung ist dazu gedacht, daß der Anwender die ihn interessierende Position und Blickrichtung auf die Fläche findet. Die für die Qualitätsanalyse und Flächenglättung benötigten Funktionen setzen dann jedoch die Sichtbarkeitsberechnung mittels Raytracing voraus.

Für die Wireframe-Darstellung wird kein gesondertes Beleuchtungsmodell benötigt, da das entstehende Gitter mit einer vom Anwender in der Datei definierten Farbe gezeichnet wird. Es wird die in Abschnitt 4.2.1. beschriebene Methode der Isokurvenextraktion benutzt.

Wurden mittels Raytracing als Verfahren zur Bestimmung der Sichtbarkeit die sichtbaren Punkte bestimmt, so können diese Ergebnisse genutzt werden, um die Lichtintensität des darzustellenden Flächenpunktes zu berechnen (Raytracing als Schattierungstechnik). Dazu wird vom getroffenen Flächenpunkt ein weiterer Strahl (Sekundärstrahl) in Richtung der idealen Spiegelung geschossen. Trifft der geschossene Strahl eine Fläche, so wird der Farbwert in diesem neuen Schnittpunkt berechnet und in der spiegelnden Komponente des zuvor berechneten Punktes (von dem der Strahl aus geschossen wurde) berücksichtigt. Trifft der Strahl auf kein Objekt, so läuft er aus der Szene und kein Anteil wird in der spiegelnden Komponente berücksichtigt. Diese Prozedur ließe sich rekursiv für jeden erhaltenen Schnittpunkt fortsetzen, jedoch werden selten mehr als ein oder zwei Rekursionen durchgeführt, da der prozentuale Anteil an der insgesamt zu berechnenden Lichtintensität mit zunehmender Rekursionstiefe immer geringer wird. Gleichzeitig steigt der Rechenaufwand erheblich an, da die Strahlverfolgung und die Berechnung möglicher Schnittpunkte sehr aufwendig und rechentechnisch teuer sind. Für die vorliegende Testumgebung wurde eine einfache Rekursionstiefe gewählt, d.h. nur vom jeweils ersten Schnittpunkt mit der Fläche aus werden Sekundärstrahlen geschossen. Im allgemeinen dürfte unserer Meinung nach mehrfache Spiegelung bei den zu betrachtenden Flächen selten auftreten.

Eine Darstellung von Glanzlichtern (Highlights) wie z.B. bei der Phong-Schattierung erfolgt bei der spiegelnden Reflexion nur für den seltenen Fall, daß ein reflektierter Strahl direkt durch die Lichtquelle läuft. Es wäre jedoch möglich, eine spiegelnde Komponente wie bei der Phong-Schattierung zu berechnen und hinzuzufügen.

Weiterhin werden für die Berechnung der Lichtintensität die diffuse Reflexion nach dem Lambertschen Gesetz sowie eine ambiente Komponente berücksichtigt. Die Berücksichtigung von Sekundärstrahlen in Richtung der idealen Brechung unter Anwendung des Transparenzgrades ist für das Testbed nicht erforderlich, da uns bei der Flächenglättung insbesondere die Eigenschaften auf der Fläche interessieren und photorealistische Bilder von zweitrangigem Interesse sind. Es wurde auf eine Implementierung verzichtet, um die Rechenzeit beim Raytracing so gering wie möglich zu halten.

Über die definierende Datei können jeder Teilfläche Farbparameter und Materialeigenschaften zugeordnet werden. Erstere werden als RGBA-Werte angegeben, letztere über die diffuse Reflexionskomponente  $K_d$ , die spiegelnde Reflexionskomponente  $K_s$  und die ambiente Komponente  $K_a$  definiert. Daneben kann die farbliche Zusammensetzung und die Intensität des Lichtes der Lichtquelle definiert werden, wodurch sich unterschiedliche Beleuchtungseffekte erreichen lassen.

Zur Darstellung der Szene ist eine Lichtquelle in der Datei zu definieren. In der Testumgebung ist genau eine Lichtquelle erforderlich. Mehr Lichtquellen stellen zwar prinzipiell kein Problem für das Raytracing dar, machen aber für die Berechnung der Isophoten in der Qualitätsanalyse wenig Sinn, da wir dort eine, feste Lichtrichtung benötigen. Es wäre möglich, eine Lichtquelle für die isophotenberechnung zu markieren und ansonsten alle Lichtquellen bei der Raytracing-Schattierung zu berücksichtigen. Auf diese Variante wurde jedoch in der vorliegenden Arbeit verzichtet, da der Hauptzweck des Testbeds die Untersuchung verschiedener Glättungsalgorithmen ist.

Der Raytracer wurde selbst entwickelt. Für die Sichtbarkeitsberechnungen mittels Raytracing wurde auf die Implementierung eines Newton-Iterationsverfahrens durch Boldt in seiner Dissertation am Fachbereich Informatik [1994] zurückgegriffen. Dieses Verfahren findet fast immer korrekt den Schnittpunkt. Es kann jedoch u.U. passieren, daß das Iterationsverfahren nicht konvergiert, obwohl ein Schnittpunkt gefunden werden müßte. In diesem Fall treten „Löcher“ in der Flächendarstellung auf. Es handelt sich hierbei um ein Startwertproblem.

Die Berechnung und Darstellung der Isophoten als Mittel der Qualitätsanalyse gestaltet sich nach erfolgter Sichtbarkeitsberechnung sehr einfach, da für jeden

sichtbaren Punkt die Normale der Fläche in diesem Punkt mitberechnet wurde. Als Richtung der parallelen Lichtstrahlen wird der Vektor von der Lichtquelle zum Blickpunkt (nicht zu verwechseln mit dem Augpunkt) definiert. Statt den eingeschlossenen Winkel zwischen Flächennormale und Lichtvektor explizit zu berechnen, wird das Skalarprodukt aus den beiden Vektoren gebildet. In der Praxis werden keine einzelnen Isophoten berechnet, sondern man testet, ob das berechnete Skalarprodukt in bestimmte Intervalle fällt. Die Breite eines solchen Isophotenintervalls läßt sich durch den Anwender steuern. Je nach Zugehörigkeit zu einem Intervall erhält der sichtbare Punkt einen Farbwert zugewiesen. Dadurch wird die gesamte Fläche von Isophotenbändern überzogen und die Qualität der Fläche kann begutachtet werden.

Zur Beschleunigung der Darstellung werden einmal berechnete Werte der sichtbaren Punkte in dynamisch verwalteten Datenstrukturen gespeichert (siehe Abschnitt 5.5.). Die dynamische Verwaltung ist notwendig, da unterschiedliche Fenstergrößen möglich sind und somit eine unterschiedliche Anzahl von Pixeln berechnet werden muß. Nach Veränderungen in der Szene oder Darstellung werden nur die veränderten Werte neu berechnet. Ansonsten wird bei Veränderungen auf die bereits berechneten Werte zurückgegriffen, was zu einer erheblichen Geschwindigkeitserhöhung bei der Darstellung führt. Insbesondere ist es dadurch möglich, interaktiv in kurzer Zeit die Position der Lichtquelle bei der Isophotendarstellung zu verändern und die Auswirkungen unmittelbar zu verfolgen (siehe Abschnitt 5.4.3.). Bei kleinen Fenstergrößen wird Echtzeit erreicht, bei einer größeren Anzahl von Pixeln liegt die Rechenzeit hierfür im Bereich weniger Sekunden.

Generell wird der Anwender vor jeder Berechnung, die längere Rechenzeiten nach sich zieht, gefragt, ob er fortfahren oder abbrechen möchte. Den größten Anteil an der Rechenzeit haben die Sichtbarkeitsberechnungen. Allgemein ist die Dauer der Berechnungen abhängig von der Auflösung (Anzahl der Pixel) und der Komplexität der Szene (Anzahl der Teilflächen). Im Abschnitt 5.7. erfolgen nähere Erörterungen zu den Rechenzeiten.

### **5.4.3. Bewegung des Augpunktes und der Lichtquelle**

Die Möglichkeit der Bewegung des Augpunktes wird benötigt, um sich die eingeleseene Fläche von verschiedenen Positionen aus anzuschauen. Die positionelle Veränderung der Lichtquelle ist wichtig für unterschiedliche Beleuchtungen der Szene beim Raytracing und insbesondere für die Qualitätsanalyse mittels Isophoten. Häufig erkennt man erst durch die Bewegung der Lichtquelle Unstetigkeiten in den Isophoten, die auf Unstetigkeiten an den Übergängen der Teilflächen zurückzuführen sind.

Die Bewegung des Augpunktes und der Lichtquelle erfolgt als Rotationsbewegung um ein festes Rotationszentrum, den Blickpunkt (View-At-Punkt, Blickrichtung definierender Punkt). Der Punkt wird in horizontaler oder vertikaler Richtung um das Rotationszentrum bewegt, wobei der Rotationswinkel jeweils zwischen  $0^\circ$  und  $180^\circ$  beliebig gewählt werden kann. Ferner wurde ein Zooming implementiert. Dabei wird der Punkt auf der Geraden durch Augpunkt und Blickpunkt in Richtung Rotationszentrum oder in entgegengesetzter Richtung bewegt, wodurch der Bildausschnitt vergrößert oder verkleinert wird.

Als Rotationszentrum wurde der Blickpunkt gewählt. Es ist daher darauf zu achten, daß der Blickpunkt und der Mittelpunkt eines Objektes oder einer Fläche ungefähr zusammenfallen, da dann die Rotation des Augpunktes um das Rotationszentrum einer Rotation des Objektes um den selben Punkt entspricht. Diese Einschränkung bei der Wahl des Blickpunktes ist unserer Meinung nach nicht gravierend, da die Lage des Blickpunktes auf dem durch den Blickrichtungsvektor definierten Strahl vom Augpunkt ausgehend an beliebiger Stelle erfolgen kann und doch die gleiche Blickrichtung ergibt. Jedoch sollten weder der Augpunkt noch die Position der Lichtquelle mit dem Rotationszentrum zusammenfallen, da dann keine Bewegung nach dem gerade beschriebenen verfahren möglich ist.

Eine Bewegung des Augpunktes ist nur im Wireframe-Modus möglich, da anschließend umfangreiche Neuberechnungen der Sichtbarkeit durchgeführt werden müssen. Die Position der Punktlichtquelle kann in allen Renderingmodi der Testumgebung geändert werden, da die Sichtbarkeit erhalten bleibt und somit der Rechenaufwand weitaus geringer ist. Kurze Antwortzeiten sind hierbei üblich.

#### **5.4.4. Umgesetzte Freiformflächen**

Bisher implementiert wurden die Funktionen für nichtrationale Bezierflächen. Sie bieten eine gute Grundlage für die Arbeit mit dem Testbed. In der ENFF-Spezifikation sind weiterhin die NURBS-Flächen enthalten, d.h. rationale Flächen. Da alle Programmteile modular aufgebaut sind, sollte ihre Einbindung in die bestehenden Funktionen ohne weiteres möglich sein. Bei der Berechnung der Flächenpunkte sind dann die Gewichte der einzelnen Kontrollpunkte zu beachten.

#### **5.4.5. Auswahl des zu glättenden Flächenbereichs**

Befindet sich der Anwender in der Isophoten-Darstellung, so kann er einen Glättungsalgorithmus zur Flächenglättung auswählen. Der Anwender wird zur Markierung des Flächenbereiches in der graphischen Darstellung aufgefordert. Wurde ein Flächenbereich mit Hilfe eines „Gummi“-Rechtecks markiert, so berechnet das Testbed diejenigen Kontrollpunkte, die in diesen Bereich abbilden. Zu jedem sichtbaren Punkt wurde in der internen Datenstruktur die Teilfläche vermerkt, zu dem der Kontrollpunkt gehört. Für alle Bildpunkte wird getestet, ob sie einen sichtbaren Punkt der Fläche enthalten. Ist dies der Fall, dann werden die Eckpunkte der Teilfläche untersucht, ob sie in den markierten Bereich abbilden. Nur die Eckpunkte der Teilflächen, die in den Bereich abbilden und gemeinsamer Eckpunkt von vier Teilflächen sind, sind für die Flächenglättung von Interesse (vergleiche Abschnitt 3.).

Die ermittelten Kontrollpunkte werden nach mehreren Kriterien sortiert, um die folgende Glättung zu vereinfachen. Im ersten Schritt werden diejenigen Kontrollpunkte zusammengruppiert, die auf ein und denselben Flächenpunkt abbilden. Dies müssen immer vier Punkte (jeweils einer von den vier aneinanderstoßenden Teilflächen) oder ganzzahlige Vielfache davon sein. Letzteres gilt für den (entarteten) Fall, daß mehr als ein Kontrollpunkt einer Teilfläche auf den Flächenpunkt abbildet, das bedeutet, daß mindestens zwei Kontrollpunkte einer Teilfläche gleich sind. Im zweiten Schritt werden die Kontrollpunkte innerhalb einer Gruppe so angeordnet, daß sie aufeinanderfolgend einem kreisförmigen Durchlaufen der vier Teilflächen entsprechen (siehe Glättungsalgorithmen, Abschnitt 3.).

#### **5.5. Flächenglättung**



Die Flächenglättung ist ein iterativer Prozeß, bei dem versucht wird, das Qualitätsmaß zu optimieren. Dazu werden in jeder Iteration Kontrollpunkte verschoben und anschließend das Qualitätsmaß neu berechnet, bis das Optimum erreicht wurde. Das Qualitätsmaß (und damit das Optimierungsmaß) sind so zu wählen, daß bei Erreichen des Optimums eine glattere Fläche erhalten wird.

Bei der Flächenglättung wird auf die zuvor sortierte Liste der ausgewählten Kontrollpunkte zurückgegriffen. Für jede Gruppe von Kontrollpunkten wird das Qualitätsmaß am gemeinsamen Flächenpunkt berechnet. Dieses Maß durchläuft dann ein Optimierungsverfahren. In der vorliegenden Arbeit wurde ein Gradientenabstiegsverfahren als Optimierungsmethode implementiert., wobei die Gradienten über Zentralkdifferenzen approximiert werden. Ergebnis der Optimierung ist ein Verschiebungsvektor (Richtung und Weite). Die Kontrollpunkte aller beteiligten Flächen werden entsprechend verschoben (und je nach Art der Freiformfläche die benachbarten Kontrollpunkte ebenso, siehe Abschnitt 3.). Anschließend wird mit der nächsten Gruppe von Kontrollpunkten fortgefahren. Erreicht eine Gruppe die maximal erlaubte Verschiebungsweite, so wird sie markiert und in einem eventuellen nächsten Glättungsschritt nicht mehr berücksichtigt.

Wurde jede Gruppe von Kontrollpunkten bearbeitet und existiert mindestens eine Gruppe, die noch nicht markiert ist, so wird ein neuer Glättungsschritt gestartet. Ein weiteres Abbruchkriterium dieses iterativen Glättungsalgorithmus ist das Unterschreiten eines Schwellwertes der Differenz aus der vorherigen und der derzeitigen Kontrollpunktposition, da in diesem Fall das Optimum erreicht wurde.

Nach beendeter Glättung wird die Darstellung der veränderten Fläche neu berechnet, wozu eine erneute Berechnung der Sichtbarkeit (hoher Rechenaufwand!) erforderlich ist. Der Anwender kann dann visuell kontrollieren, ob die Flächenglättung nach seinen Vorstellungen verlaufen ist. Ist er mit dem Ergebnis nicht zufrieden, so kann der Parameter zur Steuerung der Verschiebungsweite geändert werden oder ein anderer Flächenbereich zur Glättung ausgewählt werden und ein neuer Versuch unternommen werden.

Bisher wurde als Glättungsalgorithmus die Minimierung der Differenzen der 3. partiellen Ableitungen am Kontrollpunkt implementiert. Ein Großteil der dabei benutzten Funktionen sollte sich auf für die Umsetzung der Minimierung der Differenzen der Isophotenkrümmungen am Kontrollpunkt nutzen lassen, die derzeit

noch nicht implementiert wurde. Bei beiden Verfahren handelt es sich um lokale Verfahren, d.h. das Qualitäts- und Optimierungsmaß wird immer nur am jeweiligen Kontrollpunkt berechnet. Denkbar und zu untersuchen wären globale Verfahren, die den Wert des Qualitäts- und Optimierungsmaßes an den anderen Kontrollpunkten der Fläche ebenso berücksichtigt.

## **5.6. Datenstrukturen**

Um den erheblichen Zeitaufwand bei der Durchführung der Sichtbarkeitsberechnungen so wenig wie möglich zum Tragen kommen zu lassen, werden die berechneten sichtbaren Punkte und ihre Normalen in einer internen Datenstruktur verwaltet. Weiterhin werden die RGB-Werte des Raytracing als Schattierungstechnik und die Skalarprodukte der Isophotenberechnung gespeichert, sofern sie jeweils berechnet wurden. Solange die Position des Augpunktes, die Geometrie der Flächen und die Größe des Darstellungsbereiches nicht verändert werden, wird auf diese interne Datenstruktur zurückgegriffen, wodurch die Renderingzeiten minimalisiert werden konnten und je nach Anzahl der darzustellenden Pixel im Bereich weniger Sekunden oder darunter liegen.

Die interne Datenstruktur besteht aus zwei einfach verketteten Listenstrukturen. Die erste Liste (Zeilenliste) enthält die Pointer zu den Listen der zweiten Struktur (Punktliste). Die Zeilenliste enthält damit immer den Pointer auf die Datenwerte des ersten Pixels einer jeden Zeile des Darstellungsbereiches. Die Punktliste enthält jeweils die Punkte einer Zeile. Diese Datenstruktur hat den Vorteil eines schnelleren, wahlweisen Zugriffs auf einzelne Bildzeilen als eine sequentielle Abspeicherung. Sie stellt eine Mischung aus völlig wahlfreiem Zugriff und und komplett sequentiell Zugriff dar.

Desweiteren existiert eine Datenstruktur, die die Teilflächen mit ihren Kontrollpunkten enthält. Diese Datenstruktur wird beim Einlesen einer Fläche erzeugt. Sie wird bei allen Zugriffen auf die Koordinaten der Kontrollpunkte genutzt.

## **5.7. Rechenzeiten**

Bei der Implementierung der in dieser Arbeit erläuterten Verfahren hat sich herausgestellt, daß die Berechnung der Sichtbarkeit mittels Raytracing der rechen- und damit zeitintensivste Prozess ist. Ihr Anteil an der Gesamtberechnungszeit beträgt ungefähr 80-90%. Dies resultiert daraus, daß für jeden Bildpunkt ein Strahl in die Szene geschossen wird. Das sind bereits bei einer kleinen Fläche von 100x100 Pixeln 10000 Primärstrahlen. Jede Teilfläche muß mit jedem Strahl getestet werden, ob ein Schnittpunkt zwischen ihnen besteht. Desweiteren müssen bei den Iterationsverfahren teilweise viele Iterationen durchgeführt werden, bevor die Iteration konvergiert.

Die Sichtbarkeitsberchnungen müssen sowohl für ein Raytracing-Bild als auch für die Isophotenberechnung durchgeführt werden. Demgegenüber werden die Berechnungen der RGB-Werte für das Raytracing-Bild und die Isophotenberchnung nur durchgeführt, wenn sie benötigt werden, d.h. der Anwender diese Option gewählt hat.

Nach den Sichtbarkeitsberechnungen ist die Berechnung der RGB-Werte für das Raytracing-Bild der nächstgrößere Teil der Gesamtrechnzeit. Dies liegt an der Berechnung der einzelnen Komponenten, wobei insbesondere die spiegelnde Komponente einen großen Anteil, da bei ihrer Berechnung Sekundärstrahlen vom zuvor ermittelten Schnittpunkt in Richtung der idealen Reflexion geschossen werden. Für diese Sekundärstrahlen muß wieder eine Schnittpunktberechnung mit allen Teilflächen durchgeführt werden.

Die Berechnung der Isophoten ist sehr schnell, da hierzu nur das Skalarprodukt aus Lichtvektor und Normalenvektor berechnet werden muß. Danach erfolgt der Test, in welches Intervall das Skalarprodukt fällt und daraus resultierend die Zuweisung eines Farbwertes. Ebenso schnell läßt sich die Wireframe-Darstellung berechnen, da hierfür nur verhältnismäßig wenig Punkte ermittelt werden müssen. Bei beiden Darstellungen lassen sich in Abhängigkeit von der Anzahl der darzustellenden Bildpunkte Echtzeit bzw. sehr geringe Rechenzeiten im Bereich weniger Sekunden erreichen.

Typische Rechenzeiten für die Berechnung der Sichtbarkeit und anschließend der Isophoten liegen für einen 100x100 Pixel großen Darstellungsbereich bei geringer Szenenkomplexität bei ca. 5 Minuten auf einer SGI Indigo2. In der vorliegenden Arbeit wurde auf die Implementierung der Schnittpunktberechnungsroutinen des Raytracers von Boldt [1994] zurückgegriffen. Aufgabe der Studienarbeit

war es nicht, Verbesserungen in diesen Routinen zu erreichen, zumal die Rechenzeit sehr stark von der Anzahl der Pixel und der Komplexität der Szene abhängig ist.

## 5.8. Programmstruktur

Das Testbed liegt als Programm **Smooth** vor. Es wird über die Shell **smooth.sh** gestartet. Das Hauptprogramm **smooth.c** enthält die Initialisierungsroutinen für den Tcl-Interpreter und das Togl-Widget. Weiterhin werden hier alle neuen (eigenen) Tcl-Kommandos und die Callback-Routinen des Togl-Widgets initialisiert. Anschließend wird die Kontrolle an das Tcl/Tk-Programm **sm.tcl** übergeben, das den weiteren Ablauf steuert. Die eigenen Tcl-Kommandos und ihre Subroutinen sind im Unterprogramm **commands.c** enthalten. Die Togl-Callback-Routinen liegen im Unterprogramm **callbacks.c**. Die Konvertierung der Daten im ENFF in die interne Datenstruktur beim Einlesen einer Szene aus einer Datei und zurück beim Speichern einer Szene in einer Datei erfolgt mit den Funktionen im Unterprogramm **convert.h**. Die Datenstruktur ist in der Bibliothek **data.h** gespeichert. Die Flächenglättung wird von den Routinen des Unterprogramms **fair\_routs.c** ausgeführt. Gewisse mathematische Hilfsroutinen zur Arbeit mit Vektoren sind in im Unterprogramm **help\_routs.c** definiert.

## 6. Testflächen

Die Testflächen für das Testbed wurden selbst erstellt. Sie dienen neben dem Testen der Funktionalität vor allem der Untersuchung von Glättungsalgorithmen. Folgende Testflächen wurden erstellt: **plane3.enff**, **sphere2.enff**, **sphquadr.enff** und **sphcubic.enff**. Es handelt sich dabei um nichtrationale Bezierflächen. **Plane3.enff** beschreibt eine ursprünglich vollkommen ebene Fläche bestehend aus vier Teilflächen, die jeweils durch 4x4 Kontrollpunkte definiert werden. Die Abstände zwischen den Kontrollpunkten sind gleichmäßig, was einer uniformen Parametrisierung der Bezierfläche entspricht. Eine völlig ebene Fläche mit uniformer Parametrisierung erfüllt an den Übergängen die Bedingungen der  $C^2$ -Stetigkeit. Für eine derartige Fläche würde die Isophotenmethode versagen, da alle Normalen-

vektoren in ein und dieselbe Richtung zeigen. Verschieben wir jedoch den gemeinsamen Kontrollpunkt aller vier Teilflächen („Mittelpunkt“ der Gesamtfläche), so erhält die Fläche eine Unregelmäßigkeit („Störung“). Man beachte, daß die benachbarten Kontrollpunkte mitverschoben wurden, um die  $C^2$ -Stetigkeit zu erhalten. Abbildung 6.1. zeigt eine Ansicht des Drahtmodells dieser Fläche. Hierbei wurde der gemeinsame Kontrollpunkt senkrecht zur Ebene um ca.  $1/6$  der Kantenlänge der Gesamtfläche verschoben. Dies stellt eine sehr extreme Flächenunregelmäßigkeit dar, jedoch läßt sich an ihr sehr gut die Wirksamkeit des Glättungsalgorithmus zeigen. Die Ergebnisse werden im nächsten Abschnitt diskutiert.

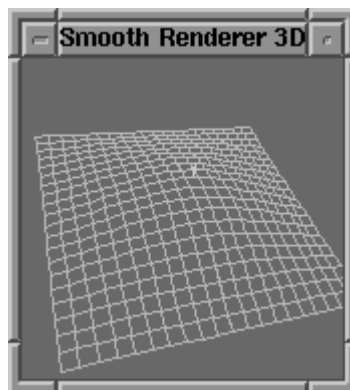


Abb. 6.1. Drahtmodell der ursprünglich ebenen Fläche definiert in plane3.enff. Das Kreuz markiert die Position der Lichtquelle.

Die Flächen **sphere2.enff**, **sphquadr.enff** und **sphcubic.enff** stellen eine Approximation der Oberfläche einer Halbkugel, wobei der Unterschied zwischen den einzelnen Dateien im Grad ihrer Bezierteilflächen besteht. Alle Flächen sind durch vier Teilflächen definiert. Allen Flächen gemein ist die „Entartung“, daß jeweils zwei Eckkontrollpunkte jeder Teilfläche in einem Punkt zusammenfallen und dies ist gerade der gemeinsame Punkt aller vier Flächen („Mittelpunkt“ am Scheitel der Halbkugel). Das Testbed kann auch derartige Fälle erfolgreich bearbeiten.

Die Teilflächen von **sphere2.enff** werden durch jeweils 5x5 Punkte definiert. Dabei wurden alle definierenden Kontrollpunkte auf die Oberfläche einer Halbkugel gelegt, d.h. auch die inneren Kontrollpunkte und nicht nur die Kontrollpunkte in den Ecken. Die erhaltene Fläche entspricht im groben einer Halbkugeloberfläche, jedoch kann sie diese nicht genau darstellen, da bei Splineflächen nur die Eckpunkte der

Teilflächen interpoliert werden, während die inneren Kontrollpunkte nur approximiert werden (Abb. 6.2.). An den Übergängen liegt  $C^0$ -Stetigkeit vor.

Die Datei **sphquadr.enff** enthält die Definition einer halbkugelähnlichen Fläche bestehend aus vier  $3 \times 3$  Teilflächen, d.h. aus biquadratischen Patches. Hier wurden nur die Eckkontrollpunkte auf die Oberfläche der Halbkugel gelegt, während die inneren Punkte so berechnet wurden, daß sich  $C^1$ -Stetigkeit an den Übergängen einstellt (vergleiche Abschnitt 4.2.2.). Die Übergänge sind damit glatter als bei der zuvor betrachteten Fläche. Die erhaltene Fläche stellt eine bessere Approximation der Halbkugeloberfläche dar (Abb. 6.3.).

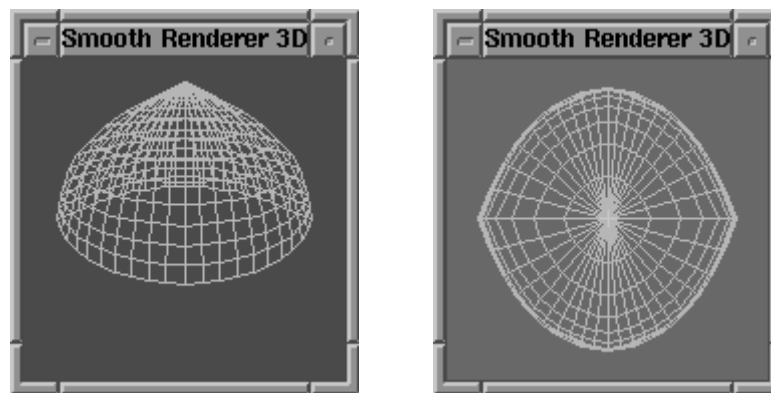


Abb. 6.2. Das linke Bild zeigt eine Ansicht von „schräg oben“ auf die halbkugelähnliche Oberfläche definiert in sphere2.enff. Die relativ grobe Approximation der wahren Gestalt einer Halbkugel ist gut zu erkennen. Das rechte Bild zeigt eine Ansicht senkrecht zur Grundfläche, wobei der Augpunkt genau über dem gemeinsamen Punkt der Teilflächen liegt. Die kreisförmige Grundfläche einer Halbkugel wird nur schlecht angenähert.

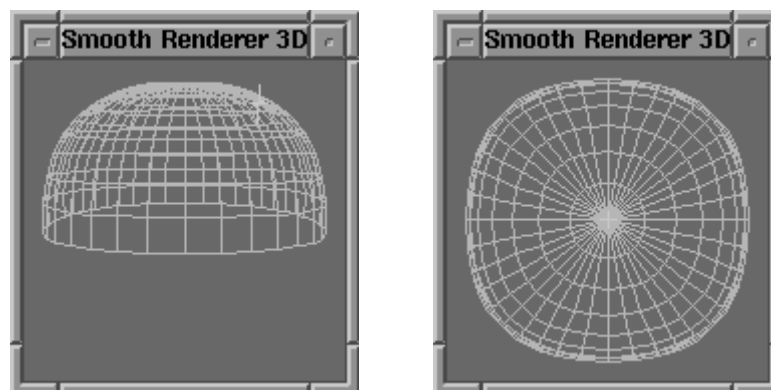


Abb. 6.3. Das linke Bild zeigt wiederum eine Ansicht von „schräg oben“ auf die Fläche definiert in sphquadr.enff. Die Approximation der Halbkugeloberfläche mit biquadratischen Teilflächen fällt wesentlich besser aus als zuvor. Die Übergänge sind wesentlich glatter ( $C^1$ -Stetigkeit). Das rechte Bild zeigt wiederum eine

„Draufsicht“ auf die Halbkugel. Die kreisförmige Grundfläche einer Halbkugel wird hier weitaus besser erreicht.

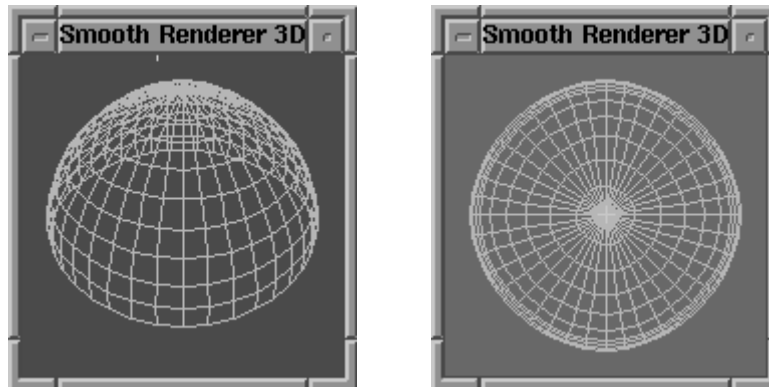


Abb. 6.4. Das linke Bild zeigt wie zuvor eine Ansicht von „schräg oben“ auf das Drahtmodell definiert in `sphcubic.enff`. Im rechten Bild wird eine Draufsicht dargestellt. Die Approximation der Halbkugeloberfläche ist in diesem Fall für die drei betrachteten Flächen am besten. An den Übergängen liegt  $C^2$ -Stetigkeit vor.

Abschließend betrachten wir eine Approximation der Halbkugeloberfläche bestehend aus bikubischen Teilflächen (4x4 Kontrollpunktmatrix), wie sie in der Datei `sphcubic.enff` definiert wurde. Die inneren Kontrollpunkte wurden in diesem Fall so berechnet, daß wir  $C^2$ -Stetigkeit an den Übergängen erhalten (vergleiche Abschnitt 4.2.2.). Man erhält eine noch bessere Approximation der Halbkugelgestalt, ohne diese jedoch genau zu erreichen (Abb. 6.4.).

Es ist festzuhalten, daß es sich bei den letzten drei Flächen wohlgermerkt nur um Approximationen der Oberfläche einer Halbkugel handelt. Mit Hilfe von nicht-rationalen Bezier- und B-Spline-Flächen können keine allgemeinen Kegelschnitte definiert werden. Hierzu wurden die rationalen Flächen eingeführt, deren Gewichte für eine entsprechende Formgebung sorgen.

## 7. Ergebnisse für die Testflächen

Die Abbildungen 7.1. (a) und (b) zeigen die  $C^0$ -stetige Fläche definiert in `sphere2.enff` als Drahtmodell und als Isophotendarstellung auf der Fläche. Man erkennt sehr gut die Unstetigkeit der Isophotenlinien, die auf die  $C^0$ -Stetigkeit der Übergänge der Teilflächen zurückzuführen ist.

Die Abbildungen 7.2. (a) und (b) zeigen die  $C^1$ -stetige Fläche definiert in **sphquadr.enff** als Drahtmodell und als Isophotendarstellung auf der Fläche. In der Isophotendarstellung läßt sich gut die höhere Stetigkeit der Isophoten (nun  $C^0$ -stetig) an den Übergängen der Teilflächen erkennen. Gleichzeitig wird ein Mangel in der Darstellung deutlich, die Schnittpunktberechnungsroutinen des Raytracers (Iterationsverfahren) finden nicht alle Schnittpunkte der Sichtstrahlen mit der Fläche (im rechten, unteren Flächenbereich besonders deutlich zu sehen).

Die Abbildungen 7.3. (a) und (b) zeigen die  $C^2$ -stetige Fläche **sphcubic.enff** als Drahtmodell und ebenso in Isophotendarstellung. Auf Grund der  $C^2$ -Stetigkeit der Flächenübergänge sind die Isophoten nun  $C^1$ -stetig an eben diesen Übergängen. Wie zuvor treten Fehler bei der Schnittpunktberechnung auf (rechter, unterer Flächenbereich).

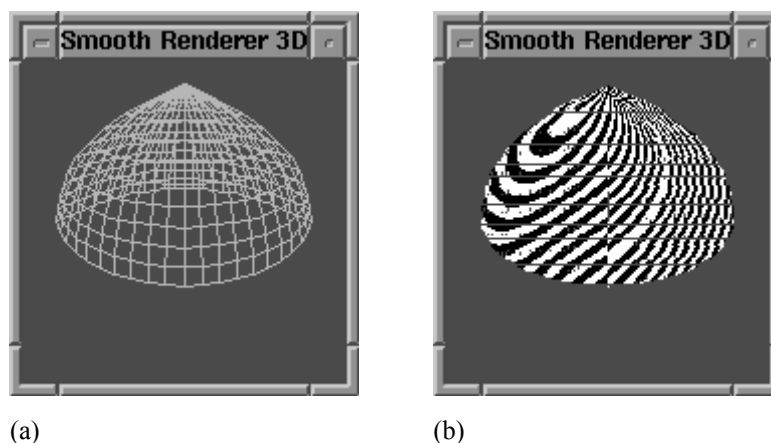


Abb. 7.1. Verschiedene Darstellungen der Fläche definiert in **sphere2.enff**. (a) zeigt eine Seitenansicht auf das Drahtmodell. (b) zeigt die Isophotenverläufe für diese Fläche. Gut zu erkennen ist die Unstetigkeit in der Mitte des Bildes.

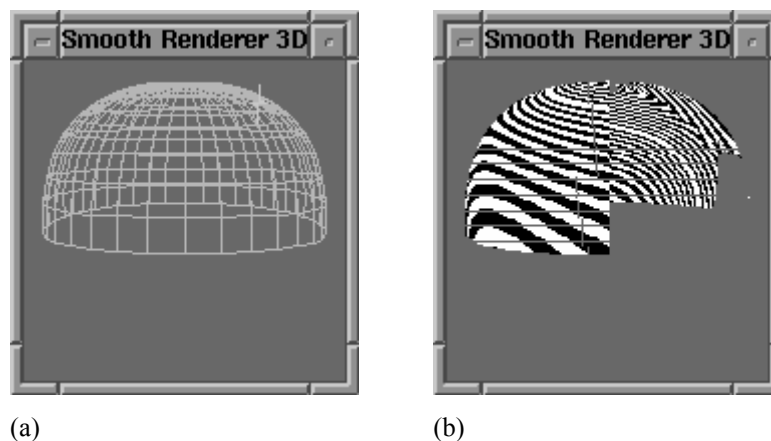


Abb. 7.2. Verschiedene Darstellungen der Fläche definiert in **sphquadr.enff**. (a) zeigt eine Seitenansicht auf das Drahtmodell. (b) zeigt die Isophotendarstellung. An den Flächenübergängen herrscht  $C^1$ -Stetigkeit, für die



Isophoten folglich  $C^0$ -Stetigkeit. Jedoch ist ein Großteil der Fläche im rechten Bildbereich nicht zu erkennen, da die Schnittpunkt-berechnungen nicht konvergierten.

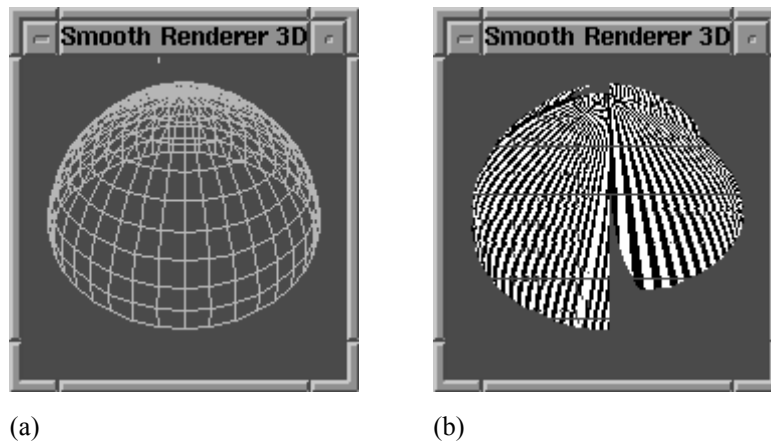


Abb. 7.3. Verschiedene Darstellungen der Fläche definiert in `sphcubic.enff`. (a) zeigt eine Seitenansicht auf das Drahtmodell. (b) zeigt die Isophotendarstellung. An den Flächenübergängen herrscht nun  $C^2$ -Stetigkeit, für die Isophoten folglich  $C^1$ -Stetigkeit. Auch hier ist wieder ein Teil der Fläche nicht sichtbar, da die Schnittpunkt-berechnungen nicht konvergierten.

Die Abbildungen 7.1. bis 7.3. verdeutlichen die Wirksamkeit der Isophoten als Verfahren zur Qualitätsanalyse. Die Isophotendarstellung bietet eine gute Möglichkeit, Unstetigkeiten an Flächenübergängen zu erkennen.

Im folgenden werden die Ergebnisse der Flächenglättung vorgestellt. Dazu wurden die Flächen `plane3.enff` und `sphcubi2.enff` genutzt. Letztere ist eine Abwandlung der in `sphcubic.enff` definierten Fläche in der Hinsicht, daß der gemeinsame Kontrollpunkt aller vier Teilflächen verschoben wurde, um eine Unregelmäßigkeit in der  $C^2$ -stetigen Oberfläche zu erzeugen. Die Verschiebungsrichtung liegt senkrecht zur Halbkugelgrundfläche in Richtung eines vergrößerten Kugelradius. Die Verschiebungsweite beträgt 10% des Halbkugelradius. Bei der Verschiebung des Kontrollpunktes geht die  $C^2$ -Stetigkeit nicht verloren, da die benachbarten Kontrollpunkte mitverschoben wurden (vgl. Abschnitt 3.).

Die Abbildungen 7.4. (a) - (d) zeigen eine Ansicht des Drahtmodells und die zugehörige Isophotendarstellung der Fläche `sphcubi2.enff` vor und nach der Glättung mittels Minimierung der Differenzen der 3. partiellen Ableitungen am Kontrollpunkt. Der verschobene Kontrollpunkt wurde wieder in Richtung Halbkugeloberfläche geschoben. Aus den Seitenansichten des Drahtmodells läßt sich nur schwer erkennen, daß sich der Radius im Scheitel der Halbkugel wieder ungefähr

dem Radius der Grundfläche angenähert hat. Für diese Flächenform geben die Isophoten kaum Aufschluß über eine Verbesserung oder Verschlechterung der Flächenqualität. Die Rückverschiebung beträgt rund 80% der Störung und wurde in 2 Glättungsschritten erreicht. Es läßt sich (auch visuell) feststellen, daß die Verschiebungsrichtung nicht genau der idealen Richtung entsprach, sondern leicht seitlich versetzt lag. Warum diese Verschiebung geschah, ist bisher unklar. Eine Möglichkeit wäre, daß das Gradientenabstiegsverfahren diese Abstiegsrichtung anstelle der idealen Richtung in einem der ersten Schritte liefert und anschließend das Verfahren nicht mehr genügend Schritte durchführt, um die ideale Richtung und Verschiebungsweite als Ergebnis zu erhalten. In diesem Fall müßten Experimente mit unterschiedlichen Schwellwerten zum Abbruch der Glättungsiteration durchgeführt werden.

Die Abbildungen 7.5. (a) - (d) zeigen eine Seitenansicht des Drahtmodells und eine Draufsicht der Isophotendarstellung auf die in der Datei **plane3.enff** definierten Fläche jeweils vor und nach der Glättung. Wiederum wurde die Minimierung der Differenzen der 3. partiellen Ableitungen am Kontrollpunkt als Glättungsalgorithmus angewandt. Bei dieser Fläche war der gemeinsame Kontrollpunkt um  $1/6$  der Kantenlänge der Gesamtfläche senkrecht zur Fläche verschoben worden, was einer sehr extremen „Störung“ gleichkommt. Wie die Seitenansicht nach der Glättung zeigt, wurde die Störung praktisch beseitigt. Das Ergebnis ist eine fast vollkommen ebene Fläche. Die Rückverschiebung erfolgte senkrecht zur Grundfläche (ideale Richtung) und die Verschiebungsweite betrug ungefähr 99,6% der Störung. Visuell läßt sich kein Unterschied zu einer ebenen Fläche erkennen. Auch die Isophotendarstellung liefert keine Hinweise mehr auf Unregelmäßigkeiten. Es liegt die Vermutung nahe, daß der Glättungsalgorithmus in diesem Fall zu einer vollkommen ebenen Fläche als Optimum führen würde, wenn die Glättungsiteration nicht vorher durch das Endkriterium abgebrochen wird. Sehr gut läßt sich die Mitverschiebung der benachbarten Kontrollpunkte beobachten. Die Methode der Minimierung der Differenzen der 3. partiellen Ableitungen am Kontrollpunkt konnte auch für diese extreme Flächenunregelmäßigkeit ihre Tauglichkeit nachweisen.

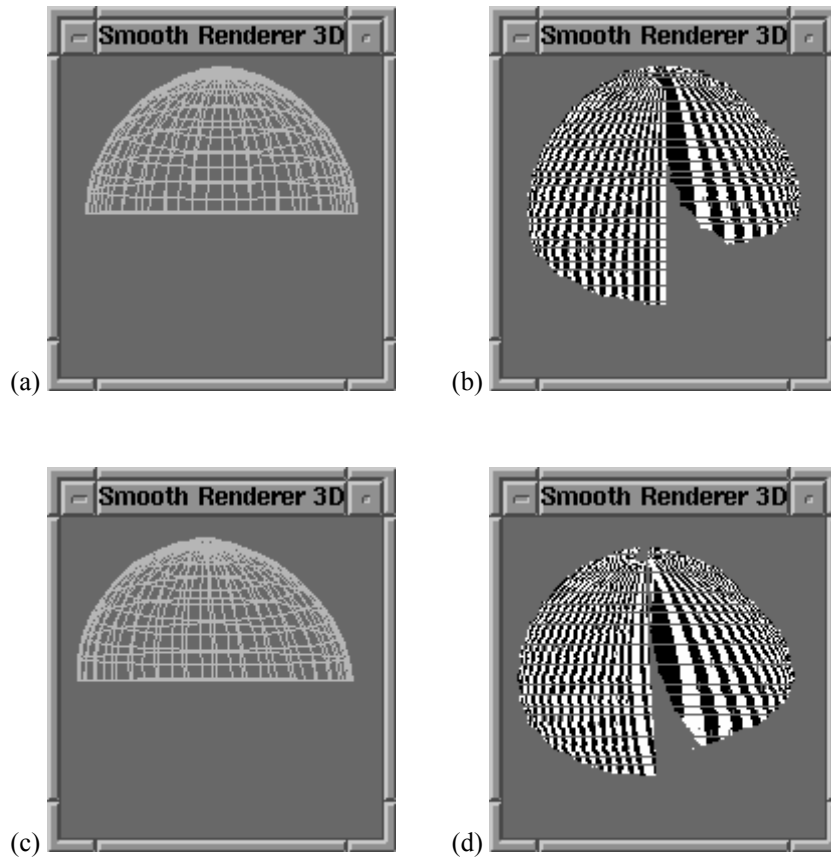
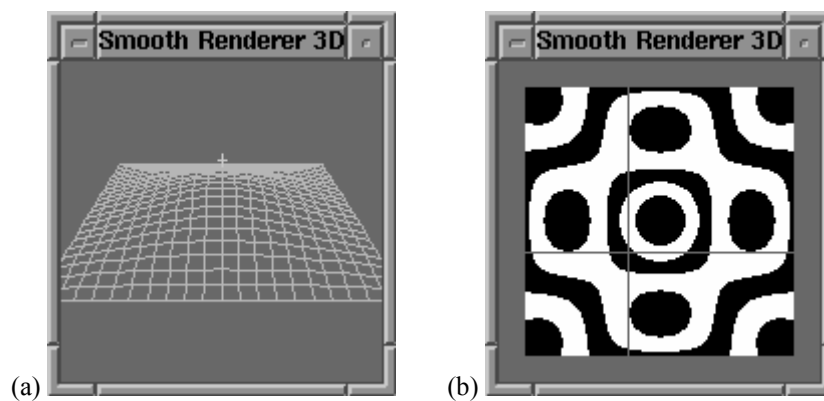


Abb. 7.4. Verschiedene Darstellungen der Fläche definiert in sphcubi2.enff. (a) und (b) zeigen eine Seitenansicht des Drahtmodells und eine Isophotendarstellung vor der Flächenglättung, (c) und (d) danach. Es läßt sich erkennen, daß die Verschiebung des zentralen Kontrollpunktes während der Glättung nicht in idealer Richtung (senkrecht zur Grundfläche), sondern leicht nach links versetzt erfolgte.



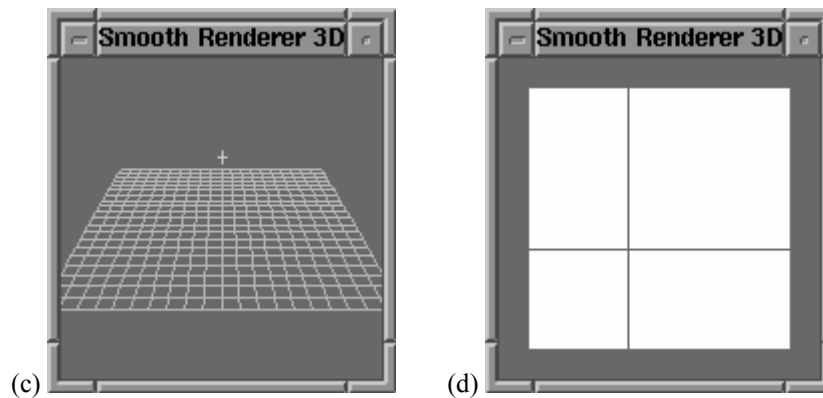


Abb. 7.5. Verschiedene Darstellungen der Fläche definiert in plane3.enff. (a) und (b) zeigen eine Seitenansicht des Drahtmodells bzw. eine Isophotendarstellung (Draufsicht) vor der Flächenglättung, (c) und (d) die Ergebnisse danach. Visuell läßt sich die erhaltene Fläche weder im Drahtmodell noch in der Isophotendarstellung von einer vollkommen ebenen Fläche unterscheiden. Die erreichte Rückverschiebung betrug 99,6% der Störungs-verschiebung in genau entgegengesetzter Richtung.

## 8. Schlußfolgerungen und Ausblick

Das Testbed bietet eine Umgebung zur Untersuchung von Glättungsalgorithmen für Freiformflächen. Es ist modular aufgebaut, so daß einzelne Komponenten ausgetauscht werden können. Das Testbed ist erweiterungsfähig. Fehleingaben des Nutzers werden dadurch weitestgehend eingeschränkt, daß zu jedem Zeitpunkt nur die dann sinnvoll einsetzbare Funktionalität zur Verfügung steht. In der derzeitigen Implementierung wird mit Bezierflächen gearbeitet. Eine Umsetzung von NURBS-Flächen sollte leicht integrierbar sein, wobei bei den Flächenberechnungsroutinen dann die Gewichte der Kontrollpunkte zu berücksichtigen sind. Für nichtrationale Flächen würden dann die Gewichte z.B. alle auf 1 gesetzt (wichtig ist das gleiche Verhältnis zueinander).

Die Ergebnisse für die Testflächen zeigen, daß sowohl die Isophoten als Verfahren der Qualitätsanalyse als auch die Minimierung der Differenzen der 3. partiellen Ableitungen als Glättungsverfahren generell für die Glättung von Freiformflächen anwendbar sind. Die Isophoten ermöglichen ein einfaches Erkennen von Unstetigkeiten an den Flächenübergängen und sind nach den bisher gemachten Erfahrungen für den Stetigkeitstest bei der Qualitätsanalyse gut geeignet.

Als nächstes wäre das Glättungsverfahren der Minimierung der Differenzen der Isophotenkrümmungen am Kontrollpunkt in das Testbed einzubinden und zu untersuchen, um vergleichende Angaben machen zu können. Diesem Verfahren kommt dabei besondere Bedeutung zu, da es parametrisierungsunabhängig ist.

Neben diesen beiden lokalen Glättungsalgorithmen sind im weiteren Verlauf globale Algorithmen zu betrachten, die das Qualitäts- und Optimierungsmaß auch an den anderen Kontrollpunkten berücksichtigen und dadurch globale Maße erhalten. Es wäre zu untersuchen, ob sie bessere Ergebnisse bei der Flächenglättung liefern.

Weiterhin ist die Ursache zu ergründen, warum bei manchen Flächen nicht die ideale Richtung der Kontrollpunktverschiebung bei der Glättung gefunden wird (vergleiche Ergebnisse für **sphcubi2.enff**). Handelt es sich dabei um ein Problem der Optimierungsmethode bzw. des Optimierungsmaßes? Bei kleinen und kleinsten Flächenunregelmäßigkeiten, wie sie eigentlich in der Praxis bei der Flächenglättung vorkommen, fällt diese Abweichung unserer Erfahrung nach nicht ins Gewicht. Jedoch machen sie sich bei den teilweise extremen Unregelmäßigkeiten der Testflächen bemerkbar. Es wäre folglich zu ergründen, ob diese extremen „Störungen“ Ursache des Problems sind und die Optimierung und Verschiebung bei kleinen Unregelmäßigkeiten ideal verläuft.

Eine interessante Aufgabe für die Zukunft bleibt, wie die Auswahl des zu glättenden Flächenbereichs automatisch erfolgen könnte. Eine Idee hierfür ist, den gemeinsamen Kontrollpunkt von je vier Teilflächen für alle Teilflächen zu betrachten, das Qualitätsmaß zu bestimmen und auf diejenigen Kontrollpunkte einen Glättungsalgorithmus anzuwenden, bei denen das Qualitätsmaß eine Toleranzgrenze überschreitet.

Schließlich wäre zu untersuchen, wie die Schnittpunktberechnungsroutinen des Raytracers verbessert werden können, so daß tatsächlich alle Schnittpunkte gefunden werden. Bei der Konvergenz des Iterationsverfahrens scheint der Winkel zwischen Sichtstrahl und Flächennormale eine Rolle zu spielen, denn für unterschiedliche Augpunktpositionen werden dieselben Flächenbereiche einmal gefunden und ein anderes Mal nicht. Wir schlußfolgern daraus, daß es sich um ein Problem der korrekten Wahl des Startwertes handelt. Die Startwerte sind blickpunkt-abhängig.

## Literaturverzeichnis

- Barth, R. [1996]: Grafikprogrammierung mit OpenGL, Bonn 1996.
- Encarnacao, J., Straßer, W. [1988]: Computer Graphics, München 1988.
- Farin, G. [1990]: Curves and Surfaces for Computer Aided Geometric Design, 2<sup>nd</sup> Ed., Boston 1990.
- Hagen, H., Hahmann, S., Schreiber, T., Nakajima, Y., Wördenweber, B., Hollemann-Grundstedt, P. [1992]:  
Surface Interrogation Algorithms, IEEE Computer Graphics & Applications, 9 / 1992, 53-60.
- Haynes, E. [1988]: Specification of the Neutral File Format, Ithaca 1988.
- Haynes, E. [1988]: Extensions of the Neutral File Format, Ithaca 1993.
- Hoschek, J. [1991]: Freeform Tools in CAD Systems, Stuttgart 1991.
- Hoschek, J. , Lasser, D. [1992]:  
Grundlagen der geometrischen Datenverarbeitung, Stuttgart 1992.
- Karltun, P. [1996]: OpenGL Graphics with the X Windows System (Version 1.2), Silicon Graphics Computer Systems, Mountain View 1996.
- Kaufmann, E. , Klass, R. [1988]:  
Smoothing surfaces using reflection lines for families of splines, Computer Aided Design, 20 (6), 1988, 312 - 316.
- Klass, R. [1980]: Correction of local surface irregularities using reflection lines, Computer Aided Design, 12 (2), 1980, 73 - 77.
- Lowes, M., Paulik, A. [1995]:  
Programmieren mit C, Stuttgart 1995.
- Ousterhout, K. [1993]: Tcl and the Tk Toolkit, Berkeley 1993.
- Paul, B., Bederson, B. [1996]:  
Togl - a Tk OpenGL Widget, Version 1.2, 1996.
- Poeschl, T. [1984]:  
Detecting surface irregularities using isophotes, Computer Aided Geometric Design 1, 1984, 163 - 168.
- Segal, M., Akeley, K. [1994]:  
The Design of the OpenGL Graphics Interface, Silicon Graphics Computer Systems, Mountain View 1994.
- Segal, M., Akeley, K. [1995]:

The OpenGL Graphics System - A Specification (Version 1.1),  
Silicon Graphics Computer Systems, Mountain View 1995.

Smith, K.P., Frazier, C. [1995]:  
The OpenGL Graphics System Utility Library,  
Silicon Graphics Computer Systems, Mountain View 1995.

## **Selbständigkeitserklärung**

Hiermit erkläre ich die vorliegende Arbeit selbständig und nur unter Zuhilfenahme der angegebene Literatur erstellt zu haben.

Roland Göcke

Rostock, den 20.07.1997