Studienarbeit

Vertiefende Analyse eines Gestalts-Constraints von Aloimonos und Shulman.

(CV-Bericht 8)

Volker Rodehorst

Technische Universität Berlin
Fachbereich Informatik
Fachgebiet Computer Vision

Prof. R. Klette

15. April 1993

Inhaltsverzeichnis

Zusammenfassung

In dieser Studienarbeit wurde die Verwendbarkeit des Beleuchtungs-Constraints von Aloimonos und Shulman untersucht. Dabei wird die Oberflächengestalt aus der Beleuchtungsrichtung, der Lichtintensität und lokaler Verschiebung ermittelt (Shape from Shading and Motion). Als starke Einschränkungen sind z.B. die Parallelprojektion, Lambert-Reflektion und die Bestimmbarkeit des optischen Flußes gegeben.

Das aus dem integrativen Verfahren isolierte Constraint wurde mit unterschiedlichen Lösungsansätzen für Grauwert-Bildfolgen implementiert und getestet. In den Experimenten wurden ideale Objekte unter Laborbedingungen mit einem 3D-Modellierer synthetisch generiert, um die Verschiebungsvektorbestimmung zu vereinfachen.

Sind bei der gemessenen Bewegung Rotationsanteile um alle Achsen vorhanden, so ist bei den meisten Beleuchtungsrichtungen eine gute Berechnung der Oberflächengestalt möglich. Damit konnte gezeigt werden, daß das Beleuchtungs-Constraint anhand seiner Aussagekraft sogar als eigenständiges Verfahren verwendet werden kann.

Kapitel 1

Einleitung

Die Modellierung von dreidimensionalen Objekten im Rechner ist eine schwierige Aufgabe und erfordert gerade bei natürlich wirkenden Körpern neben einer künstlerischen Begabung auch einen technischen und zeitlichen Aufwand. Um diese Arbeit zu vereinfachen beschäftigt man sich in der Computer Vision mit automatischen Objektrekonstruktionsverfahren. Dabei wird die Gestalt anhand von Abbildern einer Szene durch Ableitung tiefenabhängiger Merkmale und Extraktion von Oberflächenorientierungen gewonnen.

Die Motivation für diese Studienarbeit stammt aus dem Buch *Integration of Visual Modules* [ALO89]. In dem Kapitel 3 mit dem Thema *Shape from Shading and Motion* stellen Aloimonos und Shulman ein integratives Verfahren zur Gestaltserkennung vor. Die drei kombinierten Verfahren nutzen interessante Objekteigenschaften und Aufnahmecharakteristika auf geschickte Art und Weise aus. Der Einstieg wird jedoch durch ein unübersichtliches Formelwerk erschwert und die Wechselwirkungen sind durch die mathematische Verknüpfung nicht leicht zu veranschaulichen.

Bei der Vorlesung Computer Vision an der TU-Berlin [KLE92] wurde dieser interessante Zugang aufgegriffen und leicht überarbeitet. In dieser Studienarbeit soll nun das erste Constraint aus dem kombinierten Verfahren herausgelöst und die isolierte Verwendbarkeit anhand einer Implementation analysiert werden. Das Constraint enthält eine Beziehung zwischen der dreidimensionalen Oberflächengestalt und der Beleuchtungsrichtung, der Lichtintensität sowie der lokalen Verschiebung im Bild.

Die Bezeichnung *Beleuchtungs-Constraint* entstammt dem integrativen Verfahren, wo es ursprünglich zur Bestimmung der Beleuchtungsrichtung verwendet wurde. Obwohl auch die Bewegung eine wichtige Komponente darstellt, wird dieser Name in Anlehnung an den Originaltext weiterverwendet.

Diese Studienarbeit ist wie folgt gegliedert. In Kapitel 2 wird nach einer kurzen Begriffsbestimmung und der Abgrenzung durch zum Teil recht scharfe Einschränkungen das Constraint für die verwendete Szenen- und Abbildungsgeometrie hergeleitet. Anschließend erfolgt in Kapitel 3 die Beschreibung der untersuchten Lösungsverfahren. Nach einigen Problemen mit dem ersten Ansatz wird dort ein verbessertes Verfahren und zum Vergleich ein vollkommen anderer Lösungsansatz vorgestellt.

Diesen etwas mathematischen Abschnitten der Theorie folgt dann ein praxisbezogener Teil, wo sich aber auch Probleme und Erfahrungen im Umgang mit diesen mathematischen Hilfsmitteln widerspiegeln. So enthält Kapitel 4 die praktische Umsetzung in einen Algorithmus und einige bei der Implementation getroffenen Realisierungsentscheidungen. Eine kurze Beschreibung der entwickelten Programme und ihre Handhabung wird in Kapitel 5 gegeben. Das Kapitel 6 zeigt schließlich die in den Experimenten erzielten Versuchsergebnisse auf. Der ausführlich kommentierte Quelltext zu den Programmen befindet sich am Ende dieser Studienarbeit.

Kapitel 2

Theoretische Grundlagen

In diesem Kapitel soll das Beleuchtungs-Constraint von Aloimonos und Shulman eingeführt werden. Die Gleichungen und der nachfolgende Beweis unterscheiden sich in einigen Punkten von der Original-Literatur [ALO89, KLE92]. Diese Abweichungen sind jedoch nicht in der Beweisidee begründet, sondern wurden durch eine konsistente Modellbildung mit entsprechender Szenengeometrie erforderlich.

2.1 Modell von Szenenraum und Abbildung

Dieser Abschnitt soll für einige Begriffsbestimmungen genutzt werden, wobei dem kritischen Leser bereits auffallen wird, daß durch einige Annahmen recht scharfe Einschränkungen impliziert werden.

2.1.1 Kameramodell

Wir wollen ein dreidimensionales Objekt auf einem zweidimensionalen Bildschirm betrachten. Allgemein wird unter einem *Bild* eine Funktion f in zwei Ortskoordinaten x, y verstanden, deren Funktionswert $f(x, y)$ als Skalar einen Grauwert repräsentiert. Weiter benötigt man eine Transformation, die die Abbildung eines Punktes in n Dimensionen auf einen Punkt geringerer Dimension durchführt.

Die einfachste Methode besteht im Weglassen einer der Koordinaten, was man als *parallele* oder *orthogonale Projektion* bezeichnet. Die Festlegung erfolgt durch die Angabe einer Projektionsebene und des Projektionszentrums, was bei diesem Abbildungsverfahren ein Fernpunkt ist (d.h. er liegt im Unendlichen).

- Verwendet wird ein *linkshändiges Kamera-zentriertes Koordinatensystem*. Die xy-Bildebene liegt parallel zur XY-Ebene und steht senkrecht zur Z-Achse, die der optischen Achse entspricht.
- Durch die Verwendung der *Parallelprojektion* ergeben sich die Projektionsgleichungen:

$$x = X \qquad \text{und} \qquad y = Y. \tag{2.1}$$

2.1.2 Beleuchtung

Die Aufnahme natürlicher Objekte setzt den Einsatz einer Lichtquelle voraus. Die *Intensität* des Lichts, das von einer Oberfläche empfangen wird, hängt von der Beschaffenheit der Lichtquellen und von der Struktur und Materialart der beleuchteten Fläche ab.

Ein Teil des einfallenden Lichts wird vom Objekt absorbiert und der Rest wird entweder reflektiert oder, bei transparenten Körpern, teilweise durchgelassen. Zur Vereinfachung wird von einem parallel einfallenden Licht ausgegangen und von einem Schattenwurf abstrahiert.

- Der Richtungsvektor zur *punktförmigen Lichtquelle* $\mathbf{s} = (s_1, s_2, s_3)$ sei für den Bereich der Szenenobjekte als *konstant* angenommen und als Einheitsvektor auf die Länge $|\mathbf{s}| = 1$ *normiert*.

2.1.3 Szenenobjekte

Für die Darstellung von *Oberflächenorientierungen* ist es üblich, die Normale der Tangentialebene in einem betrachteten Punkt anzugeben. Um die Dimension der Repräsentation zu vermindern, stellt man die Normale durch den Vektor $(p, q, -1)$ dar, wobei p und q die ersten partiellen Ableitungen der Fläche nach x und y sind. Die durch p und q gegebene Koordinatenebene wird als *Gradientenraum* bezeichnet und jeder Punkt im Gradientenraum beschreibt eine Oberflächenorientierung.

- Für die Objekte in der Szene seien stückweise planare Oberflächensegmente π mit dem *Gradienten* (p,q) im Punkt (X,Y,Z) angenommen. Die entsprechende *Oberflächennormale* $\mathbf{n}$ sei als Einheitsvektor auf die Länge $|\mathbf{n}| = 1$ *normiert*.
- Die Szenenobjekte seien *physikalisch feste Körper* (rigid bodies).
- Für die Oberfläche aller Objekte in der Szene wird *Lambert-Reflexion* (perfekt diffus) mit einer *konstanten Reflektanzkonstanten* ρ (Albedo) angenommen, so daß sich folgende Abbildung ergibt:

$$f(x, y) = \rho \cdot \mathbf{n}(x, y) \cdot \mathbf{s}. \tag{2.2}$$

2.1.4 Bewegung

Als *Bildfolge* bezeichnet man eine geordnete Menge von Abbildern einer Szene. Veränderungen zwischen den Bildern zeigen sich in Grauwertänderungen und können somit detektiert werden. Die Ursache für diese *Bilddifferenzen* kann in der Bewegung von Objekten oder der Veränderung der Betrachterposition (Eigenbewegung) begründet sein.

- Die 3D-Bewegung muß durch die Bewegungskomponenten der *3D-Rotation* als Drehung um den Ursprung und der *3D-Translation* beschreibbar sein.
- Mit der *lokalen Verschiebung* $(u(x, y), v(x, y))$ zwischen den Bildern f_1 und f_2 wird die Bewegung des im Bildpunkt (x, y) in f_1 abgebildeten Szenenpunktes (X, Y, Z) zum Bildpunkt $(x + u(x, y), y + v(x, y))$ in f_2 beschrieben.

Abbildung 2.1: Szenen- und Abbildungsgeometrie.

2.2 Das Beleuchtungs-Constraint

Es seien f_1 und f_2 zwei Abbilder mit der lokalen Verschiebung $(u(x,y), v(x,y))$ von f_1 zu f_2. Das Bildwertverhältnis für den in den Bildern f_1 und f_2 abgebildeten Szenenpunkt (X, Y, Z) wird mit

$$r = \frac{f_2(x + u(x,y), y + v(x,y))}{f_1(x,y)}, \qquad f_1(x,y) \neq 0. \tag{2.3}$$

und einige Bewegungsdifferenzen der lokalen Verschiebung werden mit

$$\begin{aligned} u_x &= u(x+1,y) - u(x,y), & v_x &= v(x+1,y) - v(x,y), \\ u_y &= u(x,y-1) - u(x,y), & v_y &= v(x,y-1) - v(x,y). \end{aligned} \tag{2.4}$$

bezeichnet. Für die weiteren Darlegungen werden folgende Abkürzungen eingeführt

$$\begin{aligned} A &= (1+u_x)(v_y - 1) - v_x u_y, & D &= u_y^2 + v_y(v_y - 2), \\ B &= u_y s_2 - s_1(v_y - 1), & E &= v_x^2 + u_x(u_x + 2), \\ C &= v_x s_1 - s_2(1 + u_x), & F &= 2u_y(1 + u_x) + 2v_x(v_y - 1). \end{aligned} \tag{2.5}$$

Durch Lichtstärke, lokale Verschiebung und Gestalt ist für die Beleuchtung folgendes Polynom zweiten Grades in zwei Variablen gegeben:

$$\boxed{ap^2 + bq^2 + cpq + dp + eq + f = 0.} \tag{2.6}$$

mit den Koeffizienten

$$\begin{aligned} a &= r^2 s_1^2 - B^2, & d &= -2rs_3 s_1 (r + A), \\ b &= r^2 s_2^2 - C^2, & e &= -2rs_3 s_2 (r + A), \\ c &= 2(r^2 s_1 s_2 + BC), & f &= s_3^2 (A(2r + A) + r^2) + DC^2 + BCF + EB^2. \end{aligned}$$

2.3 Beweis

Der Punkt $P = (x, y)$ im Bild f_1 stellt den Szenenpunkt (X, Y, Z) auf einem Oberflächensegment π mit dem Gradienten (p, q) dar. Es sei E die Tangentialebene im Punkt (X, Y, Z). Die Vektoren $\mathbf{a} = (1, 0, p)$ und $\mathbf{b} = (0, -1, -q)$ liegen in E und lassen sich in Komponentendarstellung mit den Standardbasisvektoren $\mathbf{i}, \mathbf{j}$ und $\mathbf{k}$ ausdrücken.

$$\mathbf{a} = a_1\mathbf{i} + a_2\mathbf{j} + a_3\mathbf{k}, \qquad \mathbf{b} = b_1\mathbf{i} + b_2\mathbf{j} + b_3\mathbf{k}.$$

Nach Definition des Vektorprodukts von $\mathbf{a}$ und $\mathbf{b}$ in rechtwinkligen kartesischen Koordinaten folgt

$$\mathbf{a} \times \mathbf{b} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{vmatrix} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ 1 & 0 & p \\ 0 & -1 & -q \end{vmatrix} = -\mathbf{k} + p \cdot \mathbf{i} + q \cdot \mathbf{j} = (p, q, -1) = \mathbf{n}. \tag{2.7}$$

Mit $\mathbf{a}$ und $\mathbf{b}$ ist also auch eine mögliche Repräsentation der Gestalt gegeben. Die *Parallelprojektion* von $\mathbf{a}$ in die Bildebene ist der Vektor $\mathbf{i}$ von dem Punkt $P = (x, y)$ nach dem Punkt $Q = (x + 1, y)$, die *Parallelprojektion* von $\mathbf{b}$ in die Bildebene ist der Vektor $-\mathbf{j}$ von dem Punkt $P = (x, y)$ nach dem Punkt $R = (x, y - 1)$. Nach der lokalen Verschiebung u, v von Bild f_1 zu f_2 sind die Punkte P, Q, R nun in den Positionen:

$$\begin{aligned} P' &= (x + u(x, y), y + v(x, y)), \\ Q' &= (x + 1 + u(x + 1, y), y + v(x + 1, y)), \\ R' &= (x + u(x, y - 1), y - 1 + v(x, y - 1)). \end{aligned} \tag{2.8}$$

Die im Bild f_2 zu $\mathbf{a}, \mathbf{b}$ entspechenden Vektoren $\mathbf{a}'$, $\mathbf{b}'$ führen in der Bildebene also von P' nach Q' bzw. von P' nach R'. Nach Differenzbildung der neuen Positionen (2.8)

$$\begin{aligned} \mathbf{a}' &= ((x + 1 + u(x + 1, y)) - (x + u(x, y)), (y + v(x + 1, y)) - (y + v(x, y)), \lambda), \\ \mathbf{b}' &= ((x + u(x, y - 1)) - (x + u(x, y)), (y - 1 + v(x, y - 1)) - (y + v(x, y)), \mu), \end{aligned}$$

besitzen sie mit Einführung der Abkürzungen (2.4) die Darstellung

$$\mathbf{a}' = (1 + u_x, v_x, \lambda) \qquad \text{und} \qquad \mathbf{b}' = (u_y, v_y - 1, \mu). \tag{2.9}$$

wobei λ und μ noch zu bestimmen sind. Wegen der Annahme der Bewegung *fester Szenenobjekte* gilt

$$|\mathbf{a}|^2 = |\mathbf{a}'|^2 \qquad \text{und} \qquad |\mathbf{b}|^2 = |\mathbf{b}'|^2.$$

Aus $|\mathbf{a}|^2 = 1^2 + 0^2 + p^2 = 1 + p^2$ und $|\mathbf{a}'|^2 = (1 + u_x)^2 + v_x^2 + \lambda^2$ folgt

$$\begin{aligned} (1 + p^2) &= (1 + u_x)^2 + v_x^2 + \lambda^2, \\ \lambda^2 &= (1 + p^2) - (1 + 2u_x + u_x^2) - v_x^2, \end{aligned}$$

und aus $|\mathbf{b}|^2 = 0^2 + (-1)^2 + (-q)^2 = 1 + q^2$ und $|\mathbf{b}'|^2 = u_y^2 + (v_y - 1)^2 + \mu^2$ folgt

$$\begin{aligned} (1+q^2) &= u_y^2 + (v_y - 1)^2 + \mu^2, \\ \mu^2 &= (1+q^2) - u_y^2 - (v_y^2 - 2v_y + 1), \end{aligned}$$

womit sich zusammenfassend für λ und μ folgende Gleichung ergibt:

$$\lambda^2 = p^2 - 2u_x - u_x^2 - v_x^2 \qquad \text{und} \qquad \mu^2 = q^2 - u_y^2 - v_y^2 + 2v_y. \tag{2.10}$$

Für die Normalenvektoren $\mathbf{n_1}$ und $\mathbf{n_2}$ seien nach Gleichung (2.2) die Abbildungen

$$\begin{aligned} f_1(x,y) &= \rho_1 \cdot \mathbf{n_1}(x,y) \cdot \mathbf{s}, \\ f_2(x + u(x,y), y + v(x,y)) &= \rho_2 \cdot \mathbf{n_2}(x + u(x,y), y + v(x,y)) \cdot \mathbf{s}. \end{aligned} \tag{2.11}$$

Wir verwenden o.B.d.A $\rho_1 = \rho_2 = \rho$, da dieselben Oberflächen abgebildet werden. Es war in Gleichung (2.3)

$$r = r(x,y) = \frac{f_2(x + u(x,y), y + v(x,y))}{f_1(x,y)},$$

vereinbart. Folglich erhält man durch Einsetzen von Gleichung (2.11)

$$\begin{aligned} r(x,y) &= \frac{\rho \cdot \mathbf{n_2}(x + u(x,y), y + v(x,y)) \cdot \mathbf{s}}{\rho \cdot \mathbf{n_1}(x,y) \cdot \mathbf{s}}, \\ r(x,y) \cdot \mathbf{n_1}(x,y) \cdot \mathbf{s} &= \mathbf{n_2}(x + u(x,y), y + v(x,y)) \cdot \mathbf{s}, \\ r(x,y) \cdot \mathbf{n_1}(P) \cdot \mathbf{s} &= \mathbf{n_2}(P') \cdot \mathbf{s}. \end{aligned} \tag{2.12}$$

Ferner gilt für die normierten Normalenvektoren

$$\mathbf{n_1}(P) = \frac{\mathbf{a} \times \mathbf{b}}{|\mathbf{a} \times \mathbf{b}|} \qquad \text{und} \qquad \mathbf{n_2}(P') = \pm \frac{\mathbf{a}' \times \mathbf{b}'}{|\mathbf{a}' \times \mathbf{b}'|}. \tag{2.13}$$

Zur Bestimmung des Vorzeichens wird das Spatprodukt

$$[\mathbf{a}, \mathbf{b}, \mathbf{k}] = \begin{vmatrix} 1 & 0 & p \\ 0 & -1 & -q \\ 0 & 0 & 1 \end{vmatrix} = -1 < 0, \tag{2.14}$$

betrachtet. Es ist auch $[\mathbf{a}', \mathbf{b}', \mathbf{k}'] < 0$, falls bei f_2 gegenüber f_1 keine Umkehrung der Orientierung der Oberflächennormalen erfolgt. Dies wäre nur bei größeren Drehbewegungen oder bei Flächen fast senkrecht zur Bildebene möglich. Bei $\mathbf{n_2}$ sei also das Vorzeichen so definiert, daß die Vereinbarung für Normalen

$$\mathbf{n_2}(P') \cdot \mathbf{k} < 0,$$

gilt. Wegen der Annahme der Bewegung *fester Szenenobjekte* ist $|\mathbf{a}'| = |\mathbf{a}|$ und $|\mathbf{b}'| = |\mathbf{b}|$ sowie $\mathbf{a} \cdot \mathbf{b} = \mathbf{a}' \cdot \mathbf{b}'$, somit folgt auch

$$|\mathbf{a} \times \mathbf{b}| = |\mathbf{a}' \times \mathbf{b}'|,$$

d.h. aus den Gleichungen (2.12) und (2.13) folgt nach dem Kürzen der Beträge der Vektorprodukte

$$r(x,y) \cdot (\mathbf{a} \times \mathbf{b}) \cdot \mathbf{s} = (\mathbf{a}' \times \mathbf{b}') \cdot \mathbf{s},$$

bzw. etwas kürzer durch Verwenden des Spatproduktes

$$r \cdot [\mathbf{a}, \mathbf{b}, \mathbf{s}] = [\mathbf{a}', \mathbf{b}', \mathbf{s}]. \tag{2.15}$$

Die Ausführung der entsprechenden Berechnung erfolgt gemäß

$$r \cdot \begin{vmatrix} 1 & 0 & p \\ 0 & -1 & -q \\ s_1 & s_2 & s_3 \end{vmatrix} = \begin{vmatrix} 1+u_x & v_x & \lambda \\ u_y & v_y - 1 & \mu \\ s_1 & s_2 & s_3 \end{vmatrix}, \tag{2.16}$$

wobei λ^2 und μ^2 in (2.10) angegeben sind, und da

$$\begin{aligned} \mathbf{a} \cdot \mathbf{b} &= \mathbf{a}' \cdot \mathbf{b}' = -pq, \\ &= (1+u_x)u_y + (v_y - 1)v_x + \lambda\mu. \end{aligned}$$

gilt für das gemischte Produkt

$$\lambda\mu = -pq - (1+u_x)u_y - (v_y - 1)v_x. \tag{2.17}$$

Aus der Determinatengleichung (2.16) erhält man

$$r(-s_3 + ps_1 + qs_2) = (1+u_x)(v_y - 1)s_3 + v_x\mu s_1 + \lambda u_y s_2 - \lambda(v_y - 1)s_1 - (1+u_x)\mu s_2 - v_x u_y s_3.$$

Somit können in der Gleichung

$$r(-s_3 + ps_1 + qs_2) - s_3((1+u_x)(v_y - 1) - v_x u_y) = (u_y s_2 - s_1(v_y - 1))\lambda + (v_x s_1 - s_2(1+u_x))\mu. \tag{2.18}$$

beide Seiten quadriert werden und die Werte für λ^2, μ^2 und $\lambda\mu$ eingesetzt werden. Zuerst wird nur die linke Seite der Gleichung (2.18) betrachtet und durch eine Abkürzung

$$(r(-s_3 + ps_1 + qs_2) - s_3(\underbrace{(1+u_x)(v_y - 1) - v_x u_y}_{A}))^2,$$

vereinfacht. Durch Zerlegung des Binoms

$$r^2(-s_3 + ps_1 + qs_2)^2 - 2rs_3A(-s_3 + ps_1 + qs_2) + s_3^2A^2,$$

und Auflösen der restlichen geklammerten Potenzen

$$r^2(s_3^2 + p^2s_1^2 + q^2s_2^2 - 2s_3ps_1 - 2s_3qs_2 + 2ps_1qs_2) - 2rs_3A(-s_3 + ps_1 + qs_2) + s_3^2A^2,$$

erhält man nach dem Ausmultiplizieren für die linke Seite den Term

$$r^2s_3^2 + r^2s_1^2p^2 + r^2s_2^2q^2 - 2r^2s_3s_1p - 2r^2s_3s_2q + 2r^2s_1s_2pq + 2rs_3^2A - 2rs_3As_1p - 2rs_3As_2q + s_3^2A^2 \tag{2.19}$$

Anschließend wird die rechte Seite der Gleichung (2.18) durch folgende Abkürzungen vereinfacht

$$(\underbrace{(u_y s_2 - s_1(v_y - 1))}_{B}\lambda + \underbrace{(v_x s_1 - s_2(1+u_x))}_{C}\mu)^2,$$

und durch Zerlegen des Binoms

$$B^2\lambda^2 + 2BC\lambda\mu + C^2\mu^2,$$

und dem Expandieren von λ^2, μ^2 und $\lambda\ \mu$, durch Einsetzen von (2.10) und (2.16), umgeformt

$$B^2(p^2 - 2u_x - u_x^2 - v_x^2) + 2BC(-pq - (1+u_x)u_y - (v_y - 1)v_x) + C^2(q^2 - u_y^2 - v_y^2 + 2v_y).$$

Mit der Einführung von weiteren Abkürzungen

$$B^2(p^2 - \underbrace{(2u_x + u_x^2 + v_x^2)}_{E}) + BC(-2pq - \underbrace{(2(1+u_x)u_y + 2(v_y-1)v_x)}_{F}) + C^2(q^2 - \underbrace{(u_y^2 + v_y^2 - 2v_y)}_{D}),$$

ergibt sich nach dem Ausmultiplizieren für die rechte Seite der Term

$$B^2p^2 - B^2E - 2BCpq - BCF + C^2q^2 - C^2D. \tag{2.20}$$

Durch Zusammenführen der beiden Terme (2.19), (2.20) und Sortieren nach p und q erhält man die Bedingung für die Beleuchtung (2.6)

$$\underbrace{(r^2s_1^2 - B^2)}_{a}p^2 + \underbrace{(r^2s_2^2 - C^2)}_{b}q^2 + \underbrace{(2r^2s_1s_2 + 2BC)}_{c}pq + \underbrace{(-2r^2s_3s_1 - 2rs_3As_1)}_{d}p$$
$$+\underbrace{(-2r^2s_3s_2 - 2rs_3As_2)}_{e}q + \underbrace{(2rs_3^2A + s_3^2A^2 + r^2s_3^2 + B^2E + BCF + C^2D)}_{f} = 0.$$

Q.E.D

Kapitel 3

Lösung des Constraints

Im folgenden werden die drei verwendeten Lösungsverfahren vorgestellt. Die beiden ersten Ansätze resultieren aus einer geometrischen Betrachtungsweise. Über die Berechnung eines Schnittpunkts wird ein Polynom 4. Grades hergeleitet, welches in *geschlossener Form* angegeben werden kann.

Beim zweiten Ansatz wird beschrieben, wie dieses Polynom durch ein numerisches *Iterationsverfahren* gelöst werden kann. Im letzten Verfahren wird anstelle der vorangegangenen analytischen Vorgehensweise versucht, die Lösung durch *Minimierung* eines Funktionals mit der Methode der kleinsten Quadrate zu erzielen.

3.1 Schnittpunktberechnung

Das Polynom des Beleuchtungs-Constraints in p und q mit den bekannten Koeffizienten $a, b, \ldots, f$ kann man als Kurve zweiter Ordnung betrachten (siehe Anhang) und mit einer Schnittpunktberechnung in der Ebene den Lösungsraum auf maximal vier Lösungen einschränken. Für eine eindeutige Lösung sind mindestens drei verschiedene Gleichungen nötig. Verwendet man mehr, so ist durch das überbestimmte Gleichungssystem eine Ausgleichsrechnung erforderlich.

Wendet man die geschlossene Form für eine quadratische Gleichung auf (2.6) an, so erhält man die beiden Lösungen

$$p = \frac{-q c_1 - d_1 \pm \sqrt{g}}{2\,a_1}, \tag{3.1}$$

mit den Abkürzungen

$$\begin{aligned} g &= G\,q^2 + H\,q + J, \\ G &= {c_1}^2 - 4\,a_1\,b_1, \\ H &= 2\,c_1\,d_1 - 4\,a_1\,e_1, \\ J &= {d_1}^2 - 4\,a_1\,f_1. \end{aligned}$$

Durch Verwendung eines zweiten Polynoms mit unterschiedlichen Koeffizienten aber gleichen p, q

$$a_2\,p^2 + b_2\,q^2 + c_2\,pq + d_2\,p + e_2\,q + f_2 = 0.$$

kann die Menge der Schnittpunkte wie folgt bestimmt werden. Durch Einsetzen der Lösungen des ersten Polynoms (3.1) in das zweite Polynom, erhält man zwei Terme, die nur noch von q abhängig sind

$$\begin{aligned}
\frac{a_2\,\left(-q\,c_1-d_1+\sqrt{g}\right)^2}{4\,{a_1}^2}+b_2\,q^2+\frac{c_2\,\left(-q\,c_1-d_1+\sqrt{g}\right)q}{2\,a_1}+\frac{d_2\,\left(-q\,c_1-d_1+\sqrt{g}\right)}{2\,a_1}+e_2\,q+f_2 &= 0,\\
\frac{a_2\,\left(-q\,c_1-d_1-\sqrt{g}\right)^2}{4\,{a_1}^2}+b_2\,q^2+\frac{c_2\,\left(-q\,c_1-d_1-\sqrt{g}\right)q}{2\,a_1}+\frac{d_2\,\left(-q\,c_1-d_1-\sqrt{g}\right)}{2\,a_1}+e_2\,q+f_2 &= 0
\end{aligned}$$

und durch Expandieren der Zähler folgt

$$\begin{aligned}
&\frac{q^2a_2\,{c_1}^2}{4\,{a_1}^2}+\frac{q\,a_2\,c_1\,d_1}{2\,{a_1}^2}-\frac{q\,a_2\,\sqrt{g}\,c_1}{2\,{a_1}^2}+\frac{a_2\,{d_1}^2}{4\,{a_1}^2}-\frac{a_2\,\sqrt{g}\,d_1}{2\,{a_1}^2}+\frac{q^2a_2\,G}{4\,{a_1}^2}+\frac{q\,a_2\,H}{4\,{a_1}^2}+\frac{a_2\,J}{4\,{a_1}^2}\\
&\qquad+q^2\,b_2-\frac{q^2c_2\,c_1}{2\,a_1}-\frac{q\,c_2\,d_1}{2\,a_1}+\frac{q\,c_2\,\sqrt{g}}{2\,a_1}-\frac{q\,d_2\,c_1}{2\,a_1}-\frac{d_2\,d_1}{2\,a_1}+\frac{d_2\,\sqrt{g}}{2\,a_1}+e_2\,q+f_2 = 0,\\
&\frac{q^2a_2\,{c_1}^2}{4\,{a_1}^2}+\frac{q\,a_2\,c_1\,d_1}{2\,{a_1}^2}+\frac{q\,a_2\,\sqrt{g}\,c_1}{2\,{a_1}^2}+\frac{a_2\,{d_1}^2}{4\,{a_1}^2}+\frac{a_2\,\sqrt{g}\,d_1}{2\,{a_1}^2}+\frac{q^2a_2\,G}{4\,{a_1}^2}+\frac{q\,a_2\,H}{4\,{a_1}^2}+\frac{a_2\,J}{4\,{a_1}^2}\\
&\qquad+q^2\,b_2-\frac{q^2c_2\,c_1}{2\,a_1}-\frac{p\,c_2\,d_1}{2\,a_1}-\frac{q\,c_2\,\sqrt{g}}{2\,a_1}-\frac{q\,d_2\,c_1}{2\,a_1}-\frac{d_2\,d_1}{2\,a_1}-\frac{d_2\,\sqrt{g}}{2\,a_1}+e_2\,q+f_2 = 0.
\end{aligned}$$

Das nachfolgende Suchen eines Hauptnenners ergibt die Darstellung

$$\begin{aligned}
&\frac{q^2{c_1}^2a_2-2\,q^2c_1\,a_1\,c_2+4\,q^2{a_1}^2b_2+q^2G\,a_2+2\,q\,c_1\,d_1\,a_2-2\,q\,c_1\,a_1\,d_2}{4\,{a_1}^2}\\
&+\frac{-2\,q\,d_1\,a_1\,c_2+4\,q\,{a_1}^2e_2-2\,q\,\sqrt{g}\,c_1\,a_2+q\,H\,a_2+2\,q\,\sqrt{g}\,a_1\,c_2+{d_1}^2a_2}{4\,{a_1}^2}\\
&\qquad+\frac{-2\,d_1\,a_1\,d_2+4\,{a_1}^2f_2-2\,\sqrt{g}\,d_1\,a_2+J\,a_2+2\,\sqrt{g}\,a_1\,d_2}{4\,{a_1}^2} = 0,\\
&\frac{q^2{c_1}^2a_2-2\,q^2c_1\,a_1\,c_2+4\,q^2{a_1}^2b_2+q^2G\,a_2+2\,qc_1\,d_1\,a_2-2\,q\,c_1\,a_1\,d_2}{4\,{a_1}^2}\\
&+\frac{-2\,q\,d_1\,a_1\,c_2+4\,q\,{a_1}^2e_2+2\,q\,\sqrt{g}\,c_1\,a_2+q\,H\,a_2-2\,q\,\sqrt{g}\,a_1\,c_2+{d_1}^2a_2}{4\,{a_1}^2}\\
&\qquad+\frac{-2\,d_1\,a_1\,d_2+4\,{a_1}^2f_2+2\,\sqrt{g}\,d_1\,a_2+J\,a_2-2\,\sqrt{g}\,a_1\,d_2}{4\,{a_1}^2} = 0
\end{aligned}$$

und durch Sortieren nach q mit anschließendem Zusammenfassen folgt

$$h_1\,\left(h_2\,q^2+h_3\,\sqrt{g}\,q+h_4\,\sqrt{g}+h_5\,q+h_6\right)=0$$

mit den Koeffizienten h_i

$$\begin{aligned}
h_1 &= \frac{1}{4\,{a_1}^2},\\
h_2 &= a_2\left({c_1}^2+G\right)+4\,b_2\,{a_1}^2-2\,c_2\,c_1\,a_1,\\
h_3 &= 2\left(c_2\,a_1-a_2\,c_1\right), \qquad \text{bzw.} \qquad h_3=2\left(a_2\,c_1-c_2\,a_1\right),\\
h_4 &= 2\left(d_2\,a_1-a_2\,d_1\right), \qquad \text{bzw.} \qquad h_4=2\left(a_2\,d_1-d_2\,a_1\right),\\
h_5 &= a_2\left(H+2\,d_1\,c_1\right)+2\,a_1\left(2\,e_2\,a_1-d_2\,c_1-c_2\,d_1\right),\\
h_6 &= a_2\left({d_1}^2+J\right)-2\,d_2\,d_1\,a_1+4\,f_2\,{a_1}^2.
\end{aligned}$$

Nach dem Beseitigen von h_1 und Isolieren der Wurzel erhält man folgende *Wurzelgleichungen* für zwei unterschiedliche Paare von h_3 und h_4

$$(h_3\,q + h_4)\,\sqrt{g} = -h_2\,q^2 - h_5\,q - h_6. \tag{3.2}$$

Um sie weiter vereinfachen zu können, ist die Wurzel durch Quadrieren zu beseitigen. Dabei muß man beachten, daß dieser Umformungsschritt keine Äquivalenzumformung ist, denn die errechnete Lösungsmenge ist umfangreicher. Eine Probe durch anschließendes Einsetzen der verschiedenen Lösungen in die Wurzelgleichung (3.2) ist also unerläßlich.

$$(h_3\,q + h_4)^2\,g = \left(h_2\,q^2 + h_5\,q + h_6\right)^2.$$

Bei dieser Umformung wird eine nützliche Eigenschaft der verschiedenen h_3 und h_4 ausgenutzt. Durch das Quadrieren fallen die unterschiedlichen Vorzeichen weg und von dieser Stelle an braucht man nur noch mit einer Gleichung fortzufahren. Durch das Zerlegen der Potenzen erhält man folgenden Term

$$\begin{aligned} &{h_3}^2 G\,q^4 + 2\,h_3\,h_4\,G\,q^3 + {h_3}^2 H\,q^3 + 2\,h_3\,h_4\,H\,q^2 + {h_4}^2 G\,q^2 + {h_3}^2 J\,q^2 + 2\,h_3\,h_4\,J\,q \\ &\quad + {h_4}^2 H\,q + {h_4}^2 J = {h_2}^2 q^4 + 2\,h_2\,h_5\,q^3 + 2\,h_2\,h_6\,q^2 + {h_5}^2 q^2 + 2\,h_5\,h_6\,q + {h_6}^2. \end{aligned}$$

Nachdem die rechte Seite der Gleichung auf die linke Seite gebracht und der Term nach den Potenzen von q sortiert worden ist, erhält man ein Polynom 4. Grades in q

$$\boxed{k_1\,q^4 + k_2\,q^3 + k_3\,q^2 + k_4\,q + k_5 = 0} \tag{3.3}$$

mit den Koeffizienten k_i

$$\begin{aligned} k_1 &= G\,{h_3}^2 - {h_2}^2, \\ k_2 &= 2\,G\,h_3\,h_4 + H\,{h_3}^2 - 2\,h_2\,h_5, \\ k_3 &= G\,{h_4}^2 + 2\,H\,h_3\,h_4 + J\,{h_3}^2 - 2\,h_2\,h_6 - {h_5}^2, \\ k_4 &= H\,{h_4}^2 + 2\,J\,h_3\,h_4 - 2\,h_5\,h_6, \\ k_5 &= J\,{h_4}^2 - {h_6}^2. \end{aligned}$$

3.1.1 Geschlossene Form

Eine algebraische Gleichung vom Grad n besitzt nach dem Fundamentalsatz der Algebra genau n Lösungen, die i.a. komplexe Zahlen und nicht notwendig voneinander verschieden sind. Diese Lösungen können normalerweise bis zum Grad $n = 4$ in geschlossener Form durch Koeffizienten ausgedrückt werden. Die vier Lösungen von (3.3) lauten

$$\boxed{\begin{aligned} q &= \frac{-3\,k2\,\sqrt[4]{N} + \sqrt{3}\left(N^{3/4} + \sqrt{2}\sqrt{O-P}\right)}{12\,k1\,\sqrt[4]{N}}, \\ q &= \frac{-3\,k2\,\sqrt[4]{N} + \sqrt{3}\left(N^{3/4} - \sqrt{2}\sqrt{O-P}\right)}{12\,k1\,\sqrt[4]{N}}, \\ q &= \frac{-3\,k2\,\sqrt[4]{N} - \sqrt{3}\left(N^{3/4} - \sqrt{2}\sqrt{O+P}\right)}{12\,k1\,\sqrt[4]{N}}, \\ q &= \frac{-3\,k2\,\sqrt[4]{N} - \sqrt{3}\left(N^{3/4} + \sqrt{2}\sqrt{O+P}\right)}{12\,k1\,\sqrt[4]{N}}. \end{aligned}} \tag{3.4}$$

mit folgenden Abkürzungen

$$\begin{aligned}
K &= {k_4}^2\left(4\,k_1\,{k_3}^3-{k_2}^2{k_3}^2\right)+4\left({k_2}^3{k_4}^3+k_5\,{k_2}^2{k_3}^3\right)+k_1\,k_5\left(6\,{k_2}^2{k_4}^2-16\,{k_3}^4\right)\\
&\quad -18\,k_3\left(k_2\,{k_4}^3k_1+{k_2}^3k_4\,k_5\right)+27\left({k_4}^4{k_1}^2+{k_5}^2{k_2}^4\right)+80\,k_2\,k_4\,k_5\,k_1\,{k_3}^2\\
&\quad +{k_5}^2{k_1}^2\left(128\,{k_3}^2+192\,k_2\,k_4\right)-144\,k_3\left(k_5\,{k_1}^2{k_4}^2+\,{k_5}^2k_1\,{k_2}^2\right)-256\,{k_5}^3{k_1}^3,\\
L &= 2\,{k_3}^3-9\,k_3\left(k_2\,k_4+8\,k_5\,k_1\right)+27\left({k_4}^2k_1+k_5\,{k_2}^2\right),\\
M &= k_1\sqrt[3]{4\left(L+3\right)\sqrt{3}\,\sqrt{K}}+\sqrt[3]{4\left(L-3\right)\sqrt{3}\,\sqrt{K}},\\
N &= \sqrt{3\,{k_2}^2-8\,k_1\,k_3+2\;M},\\
O &= \sqrt{N}\left(3\,{k_2}^2-8\,k_3\,k_1-M\right),\\
P &= 3\sqrt{3}\left({k_2}^3-4\,k_2\,k_1\,k_3+8\,{k_1}^2k_4\right).
\end{aligned}$$

3.1.2 Numerisches Iterationsverfahren

Die Lösung des Polynoms (3.3) kann aber auch durch ein numerisches Iterationsverfahren erfolgen. Da dieses umfangreiche Thema den Rahmen sprengen würde, wird auf Herleitungen und Beweise verzichtet, jedoch ist für das Verständnis dieses Programmteils im Anhang ein kleiner Überblick für das bei der Implementation verwendete *Muller-Verfahren* gegeben, welches in [ENG88, MUL56] genauer beschrieben ist. Das Verfahren von Muller liefert ohne Kenntnis von Startwerten in kürzester Zeit sämtliche reellen und konjugiert komplexen Nullstellen eines Polynoms P_n mit reellen Koeffizienten.

$$\begin{aligned}
P_n(x) &= a_0+a_1x+a_2x^2+\ldots+a_nx^n\\
&= \sum_{j=0}^{n}a_jx^j \qquad \text{für } a_j\in R, \qquad a_n\neq 0.
\end{aligned} \tag{3.5}$$

Im allgemeinen muß ein Algorithmus, der eine Lösung ξ der Gleichung $P_n(x)=0$ sucht, zunächst eine Umformung in eine äquivalente Gleichung der Gestalt

$$x=\varphi(x) \tag{3.6}$$

durchführen. Dabei sei φ eine in einem abgeschlossenen Intervall I stetige und reellwertige Funktion. Dann konstruiert man mit Hilfe eines *Startwertes* $x^0\in I$ eine Zahlenfolge $\{x^\nu\}$ nach der *Iterationsvorschrift*

$$x^{\nu+1}:=\varphi(x^\nu), \qquad \nu=0,1,2,\ldots. \tag{3.7}$$

Wenn die *Iterationsfolge* $\{x^\nu\},\nu=1,2,\ldots$ konvergiert, dann ist ξ eine Lösung der Gleichung.

$$\xi=\lim_{\nu\to\infty}x^\nu=\lim_{\nu\to\infty}x^{\nu+1}=\lim_{\nu\to\infty}\varphi(x^\nu)=\varphi(\lim_{\nu\to\infty}x^\nu)=\varphi(\xi). \tag{3.8}$$

Die Iteration wird solange fortgesetzt, bis von einem $\nu=N$ die Differenz $|x^{\nu+1}-x^\nu|$ kleiner als ein vorgegebener Schwellwert ϵ liegt.

3.2 Methode der kleinsten Quadrate

Dieses Verfahren setzt direkt auf dem Beleuchtungs-Contraint auf. Bei jeder Gleichung (2.6) kann für bestimmte p und q die Differenz von 0 als Fehler definiert werden

$$\text{error}_i(p,q) = ap^2 + bq^2 + cpq + dp + eq + f, \tag{3.9}$$

und für n Gleichungen läßt sich mit der quadratischen Abweichung folgendes Funktional bestimmen

$$\boxed{F(p,q) = \frac{1}{n}\sum_{i=1}^{n} \text{error}_i(p,q)^2.} \tag{3.10}$$

Die gesuchte Lösung (p', q') wird durch Minimierung dieses Funktionals bestimmt. Liegt das ermittelte Minimum unter einem konstanten Schwellwert ϵ, so werden (p', q') als Gradient übernommen.

Kapitel 4

Realisierung und Implementation

Nach den theoretischen Grundlagen sollen in diesem Kapitel die Probleme und gesammelten Erfahrungen bei der praktischen Umsetzung des Beleuchtungs-Constraints erläutert werden. Hier können nicht alle Aspekte behandelt werden, dafür wurde aber versucht, den Quelltext entsprechend gut zu kommentieren. Die formalen Variablen des Beleuchtungs-Constraints müssen bei der Implementierung mit aktuellen Parametern belegt werden. Wie man diese erhält und was dabei zu beachten ist, wird im folgenden beschrieben.

4.1 Problemanalyse

4.1.1 Das initiale Bild

Für das Aufstellen einer einzigen Gleichung (2.6) werden bereits zwei Bilder benötigt, zwischen denen die Verschiebungsvektoren ermittelt werden können. Um eine eindeutige Lösung bei der Berechnung der Gestalt zu erhalten, sind drei Gleichungen erforderlich (siehe Kapitel 3.1), d.h es müssen mindestens vier Bilder vom Objekt aufgenommen werden. Damit gewährleistet wird, daß die Gleichungen dieselben Gradienten (p, q) enthalten, sind die Verschiebungsvektoren immer von einem *initialen* Referenzbild aus zu generieren.

4.1.2 Bewegung durch Transformation

Die Bewegung spielt bei diesem Verfahren eine wichtige Rolle. Sie wird bei der synthetischen Generierung durch eine Objekttransformation simuliert. Bei genauer Betrachtung kann die Anzahl der relevanten Transformationen erheblich eingeschränkt werden.

Aufgrund der Annahme fester Szenenobjekte kann eine *Skalierung* nicht vorkommen und durch die Parallelprojektion werden Objekte auch nicht proportional zu ihrer Entfernung verkleinert, wie das bei der perspektivischen Zentralprojektion der Fall wäre.

Da in dem Beleuchtungs-Constraint nur die Differenzen der Verschiebungsvektoren eingehen, entfallen auch mögliche *Translationen*. Am Rand treten zwar Unterschiede auf, jedoch sind innerhalb des Objektes u_x, u_y, v_x und v_y gleich 0, was keinen Informationsgewinn bei der Gestaltserkennung bedeutet.

Abbildung 4.1: Verdeckungsproblem bei der Bestimmung von Verschiebungsvektoren.

Bei der synthetischen Generierung wird für alle im Bild f abgebildeten Objektpunkte die Rotation (4.1) durchgeführt und die *Sichtbarkeit* der Punktes an der neuen Position überprüft. Zeigt die Oberflächennormale im transformierten Punkt in Richtung der negativen Z-Achse, so ist dieser prinzipiell sichtbar. Die Vektoren werden dann durch Projektion der Verschiebung in die xy-Ebene subpixelgenau bestimmt.

Hier liegt ein kleiner Nachteil des implementierten Verfahrens, das zu viele Vektoren bestimmt. Denn es wird auch ein Verschiebungsvektor von einem Punkt generiert, der nach der Rotation durch teilweise Überdeckung nicht sichtbar wäre. Der Verschiebungsvektor in der Abbildung zwischen den Punkten P_2 und P_2' wäre in realen Grauwertbildern nicht zu ermitteln.

Die Implementation einer *Sichtstahlverfolgung* beseitigte zwar diesen Nachteil, erwies sich aber aufgrund der aufwendigen Schnittpunktberechnung als nicht praktikabel. Außerdem tritt dieses Problem nur bei extrem *konkaven Objekten* auf und beide Verfahren zeigten in den getesteten Beispielen keine wesentlichen qualitativen Unterschiede.

4.2 Modellierung

Um die Implementierung etwas transparenter zu gestalten und den Datenfluß kontrollieren zu können, wurden Teilaufgaben auf verschiedene Programme verteilt.

- `Simul` als 3D-Modellierer zur Simulation der Aufnahmesituation.
- `Solve` zur Berechnung der Gestalt nach dem Beleuchtungs-Constraint.
- `Visual` für die graphische 2D/3D-Darstellung der Ergebnisse.
- `Verify` zur Bewertung der Güte und Darstellung diverser Fehlermaße.

4.2.1 Verwendete Schnittstellen

Die für den Informationsaustausch notwendig gewordenen Dateien können grundsätzlich in zwei verschiedenen Aufzeichnungsformaten erzeugt und gelesen werden, die auch automatisch identifiziert werden. Zum besseren Verständnis und zur Verifikation existiert ein lesbares *ASCII-Textformat* und aus Gründen der Effizienz wurde zusätzlich ein *Binärformat* eingeführt.

Die Grauwertbilder werden in dem verbreiteten *Portable GrayMap (PGM)* Format geschrieben, die dann z.B. mit *PBM+* oder *xv* weiterverarbeitet werden können. Der Aufbau der eigenen Dateiformate ist im nächsten Kapitel beschrieben.

Für die Generierung und Visualisierung ist ein Grafikbildschirm notwendig. Zur Zeit wird ein 8-bit Farbdisplay unter *X-Windows* unterstützt und die entsprechende Grafikschnittstelle für den AMIGA oder den 24-bit DA-Server ist auf Anfrage erhältlich.

4.2.2 Die PixMap als Zeichenpuffer

Um möglichst unabhängig von den Hardwarevorraussetzungen zu sein, ist ein interner Zeichenpuffer, eingeführt worden, der im folgenden als *PixMap* bezeichnet wird. Dieser ermöglicht das Berechnen von

Bildern in 256 Grauwertstufen auf einem System mit monochromen Bildschirm und vereinfacht das Abspeichern der berechneten Bilder.

Die PixMap kann unabhängig vom Bildschirm skaliert und positioniert werden. Neben dem Zeichnen von Punkten in der PixMap ist für die Darstellung der Nadelkarten und des optischen Flußes eine Routine zum Zeichnen von Linien bereitgestellt worden.

4.3 Konkretisierung

4.3.1 Probleme der direkten Methode

Die mathematisch hergeleitete Lösung bei der direkten Methode durch Angabe einer geschlossenen Form bereitet einige Probleme bei der Realisierung. Durch einen endlichen Wertebereich entstehen Ungenauigkeiten, die in diesem Fall zu entscheidenden Beeinträchtigungen führen.

Häufig auftretende negative Terme unter einer Wurzel erfordern eine besondere Behandlung. Da im Ergebis keine komplexen Werte zu erwarten sind, können diese Zwischenwerte durch Rechenungenauigkeiten entstanden sein oder sie verschwinden während der Berechnung wieder. Unter der ersten Annahme ist es möglich, die Wurzel durch 0 zu ersetzen.

Leider ist der Term M aus Gleichung (3.4) ein kritischer Ausdruck. Die Werte unter der Wurzel durchlaufen den gesamten darstellbaren Zahlenbereich und ein Beseitigen führt dazu, daß keine Lösungen existieren. Daher wird an dieser Stelle die Wurzel aus dem Betrag gebildet und das Vorzeichen entsprechend herausgezogen. Angesichts dieser Entscheidung ist das direkte Verfahren bereits fragwürdig.

4.3.2 Eindeutigkeit durch Ausgleichsrechnung

Für eine beliebige Anzahl n von Bildern ist durch das überbestimmte Gleichungssystem normalerweise eine Ausgleichsrechnung erforderlich. Durch die Suche nach einem Minimum kann je nach der verwendeten Metrik ein n-Tupel gefunden werden, welches die an der Lösung dichtesten Werte enthält. So kann bei der euklidischen Metrik durch komponentenweises Bilden der quadratischen Differenzen der einzelnen Lösungsvektoren ein eindeutiges Ergebnis erzeugt werden.

Im Vergleich hat es sich gezeigt, daß anscheinend jeweils der betragskleinste Wert die Lösung darstellte. Daher wird nach einer Sortierung der Beträge das arithmetische Mittel gebildet und die jeweils ersten Werte im Lösungsvektor erzeugen das beste Ergebnis.

4.3.3 Normierte Gradienten

Bei der Entwicklung hat sich herausgestellt, daß bei den verwendeten Datenstrukturen die normierte Darstellung des Gradienten (p, q) durch einen Normalenvektor $\mathbf{n}$ sinnvoller ist, wobei die redundante Z-Komponente weggelassen werden kann. Unter der Annahme, daß die sichtbaren Oberflächennormalen in negative Z-Richtung zeigen, lauten die Umformungen in eine Normale

$$\mathbf{n} = (n_x, n_y, n_z) = \frac{(p, q, -1)}{\sqrt{p^2 + q^2 + 1}} \tag{4.2}$$

bzw. in einen Gradienten

$$p = -\frac{n_x}{n_z} \qquad q = -\frac{n_y}{n_z} \qquad \text{für} \qquad n_z \neq 0. \tag{4.3}$$

Abbildung 4.2: Bestimmung der Gradienten durch orthogonale Winkel.

Die im folgenden mit Gradient bezeichneten Daten liegen daher intern als Normale ohne Z-Komponente vor. Für den Benutzer ist dies lediglich beim Aufbau der Gradientendatei wichtig.

4.3.4 Repräsentation des Gradienten durch Winkel

Die Abstände im Gradientenraum entsprechen nicht den Orientierungsänderungen. Bei großen p oder q können große Punktabstände geringen Orientierungsänderungen entsprechen. Aus diesem Grund werden die Gradienten bei der *Minimierung* nicht linear eingesetzt, sondern aus den orthogonalen Winkeln α, β der Vektoren $\mathbf{v} = (p, q, -1)$ berechnet, die in ihrer Richtung den sichtbaren Oberflächennormalen entsprechen.

$$p = \frac{1}{\tan(\alpha)}, \qquad q = \frac{1}{\tan(\beta)} \qquad \text{für} \qquad 0 < \alpha, \beta < \pi. \tag{4.4}$$

In der Praxis werden die Winkel von 1 bis 179 Grad mit einer ganzzahligen Schrittweite durchlaufen, was einen quadratischen Aufwand bedeutet, und deshalb eine feinere Unterteilung nicht sinnvoll erscheint. Die Werte des Gradienten sind somit auf den Wertebereich von $-58 < p, q < 58$ begrenzt und es werden am Objektrand weniger Lösungen erzeugt als bei den analytischen Verfahren.

4.3.5 Erstellen einer Tiefenkarte

Durch die Annahme von Parallelprojektion kann die Tiefe bis auf einen konstanten Summanden angenähert berechnet werden. Dabei wird für jeden Punkt der Gradientenmatrix ein Tiefenwert ermittelt. Da ein Gradient (p, q) die Steigung einer Tangentialebene im Punkt beschreibt, ist bei der Bestimmung eine Verschiebung des Rasters und eine Schnittpunktberechnung von senkrechten Ebenen nötig, die durch Oberfächennormalen am Objektrand entstehen.

Bei der Verwendung von Sekanten zur Ebenenbestimmung kann direkt das Raster verwendet werden und alle Flächen besitzen eine endliche Steigung. Die Berechnung erfolgt durch lineare Interpolation benachbarter Oberflächennormalen.

Abbildung 4.4: Interpolation der Normalen und Bestimmung der Tiefe.

- Die Oberflächenberechnung sollte nicht von der gewählten Entwicklungsrichtung abhängen und daher das Integral

$$z(x,y) = z(x_0,y_0) + \int_{(x_0,y_0)}^{(x,y)} (pdx + qdy) \tag{4.8}$$

  bei jedem geschlossenen Weg gleich 0 sein.

Für die Erfüllung dieser Bedingungen existieren aufwendige Minimierungsverfahren, wie z.B. bei [HOR86] beschrieben. In dieser Studienarbeit wurde darauf verzichtet und durch Verwendung von unterschiedlichen Entwicklungsrichtungen recht gute und besonders schnelle Ergebnisse erzielt. Die mit der Kombination aus horizontaler und vertikaler sowie zeilen- und spaltenweiser Entwicklung gewonnenen Resulate werden anschließend arithmetisch gemittelt.

Steigungen, die parallel zur Entwicklungsrichtung verlaufen, werden besonders gut erkannt, aber die schlechte Bestimmung der seitlichen Orientierung verursacht eine Verzerrung, die im gleichförmigen Hintergrund durch eine Nachbearbeitung beseitigt werden muß. Unter der Annahme, daß ein Randpunkt zum Hintergrund gehört, werden alle benachbarten Punkte mit der gleichen Orientierung in der Tiefe angeglichen.

Kapitel 5

Bedienungsanleitung

5.1 Programmsteuerung

Die Steuerung der Programme erfolgt durch Angabe von *Optionen*, die mit einem Spiegelstrich '-' eingeleitet werden. Der direkt nachfolgende Buchstabe ist entscheidend, egal ob er groß oder klein geschrieben wird. Ebenso kann dieser beliebig erweitert werden, d.h. die Option '-l' ist identisch mit '-light' oder '-LICHT'.

Erwartet eine Option mehrere Parameter, so sind sie durch Leerzeichen voneinander zu trennen. Die Reihenfolge ist unwesentlich und bei sich ausschließenden Optionen wird nur die letzte Angabe berücksichtigt. Gibt man beim Programmaufruf ein '?' oder '-h' an, so erscheint ein *Benutzungshinweis* mit der Auflistung der unterstützten Optionen.

5.2 Datengenerierung mit SIMUL

Das Programm `simul` soll die Aufnahme einer digitalen Grauwertbildfolge unter Laborbedingungen simulieren und unterstützt die speziell für das Beleuchtungs-Constraint benötigten Daten. Dabei wird ein 3D-Objekt erzeugt und mit *Parallelprojektion* abgebildet. Die Grauwerte werden durch *Lambert-Reflektion* mit einer unendlich entfernten *Punktlichtquelle* ermittelt. Die Objektoberfläche kann durch Verändern der *Reflektanzkonstante* beeinflußt werden oder bei polyedrischen Objekten eine *Glättung* durch Interpolation der Oberflächennormalen durchgeführt werden.

Eine Folge von Bildern wird durch Transformationsanweisungen auf das initiale Objekt erzeugt. Da Translationen keinen Einfluß auf das verwendete Verfahren haben und eine Skalierung nicht zulässig ist, werden die Verschiebungsvektoren und das Grauwertverhältnis zum initialen Objekt nur durch *Rotation* bestimmt. Es kann sowohl eine Objekt- als auch eine Kamerarotation durchgeführt werden, wobei sich der Ursprung des linkshändigen Koordinatensystems stets im Objektmittelpunkt befindet.

Die so erzeugten Daten können optional als spezielle *Displacement-Datei* gespeichert werden, welche die Differenzen der Verschiebungsvektoren und das entsprechende Grauwertverhältnis enthält. Für die Qualitätskontrolle der später berechneten Werte können zusätzlich die optimalen Oberflächennormalen des initialen Objektes in einer *Gradientendatei* gesichert werden. Schließlich ist auch das Ausgeben des *Grauwertbildes* von dem generierten Objekt möglich.

Die Objektstruktur und die benötigten Parameter und Optionen können in einer *Konfigurationsdatei* beschrieben werden. Das Experimentieren mit den einfachen Beispieldateien (z. B. `'sphere.cfg'` oder `'cube.cfg'`) sollte die Funktionsweise der Programmsteuerung ausreichend illustrieren.

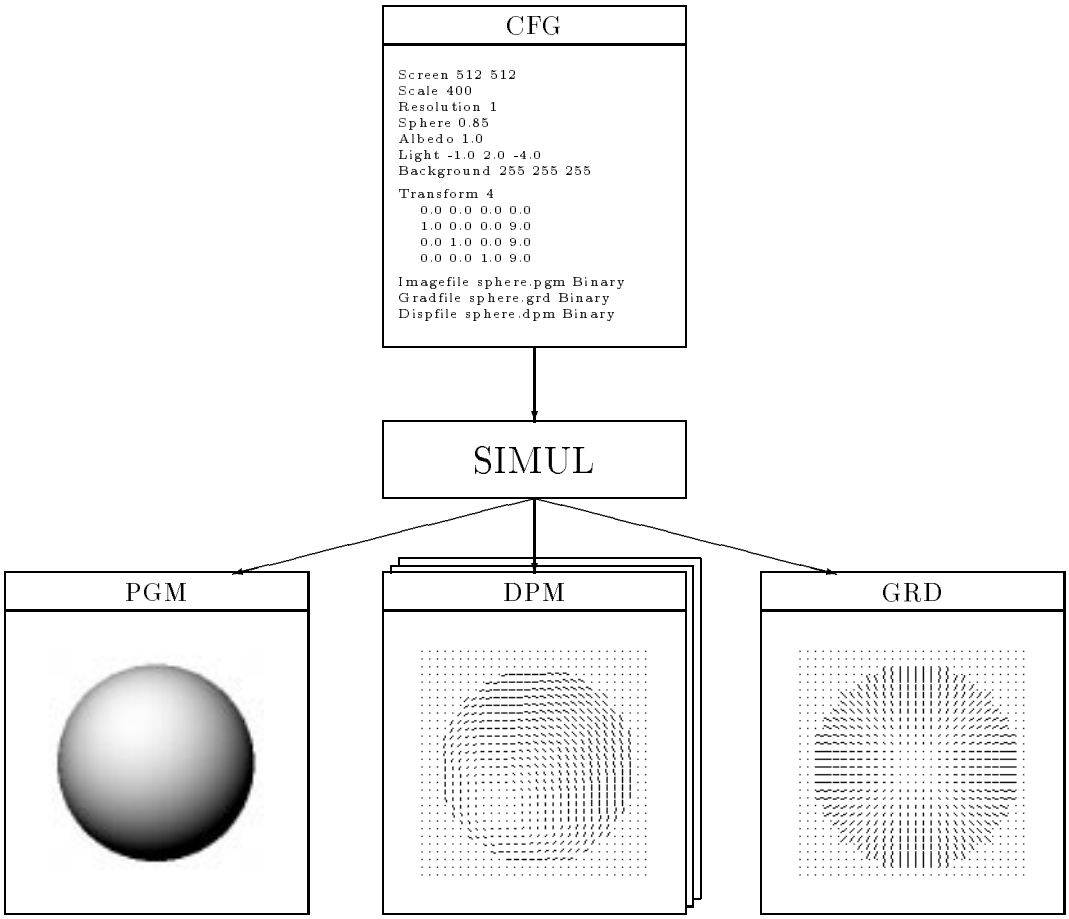

Abbildung 5.1: Erzeugung von Grauwert-, Gradienten- und Displacement-Dateien mit `simul` anhand einer Konfigurationsdatei.

```
> simul

SIMUL Version 1.0  © Copyright 1993  Volker Rodehorst, TU-Berlin

AUFRUF   :  simul <Dateiname.cfg>
```

5.3 Lösung durch SOLVE

Dieses Programm berechnet *Oberflächennormalen* aus den oben beschriebenen Daten der *Differenzen von Verschiebungsvektoren* und dem zugehörigen *Grauwertverhältnis*, welche anschließend in einer Gradientendatei abgelegt werden. Die Lösung des Beleuchtungs-Constraints kann algebraisch sowohl *direkt* als auch durch ein numerisches *Iterationsverfahren* berechnet werden.

Bei der Durchführung einer *Minimierung* ist die Schrittweite n in Grad anzugeben, $0 < n < 180$. Ab 10 Grad erhält man schon zufriedenstellende Werte und bei exakteren Schritten von z.B 3 Grad ist der quadratische Anstieg der Rechenzeit nicht zu unterschätzen.

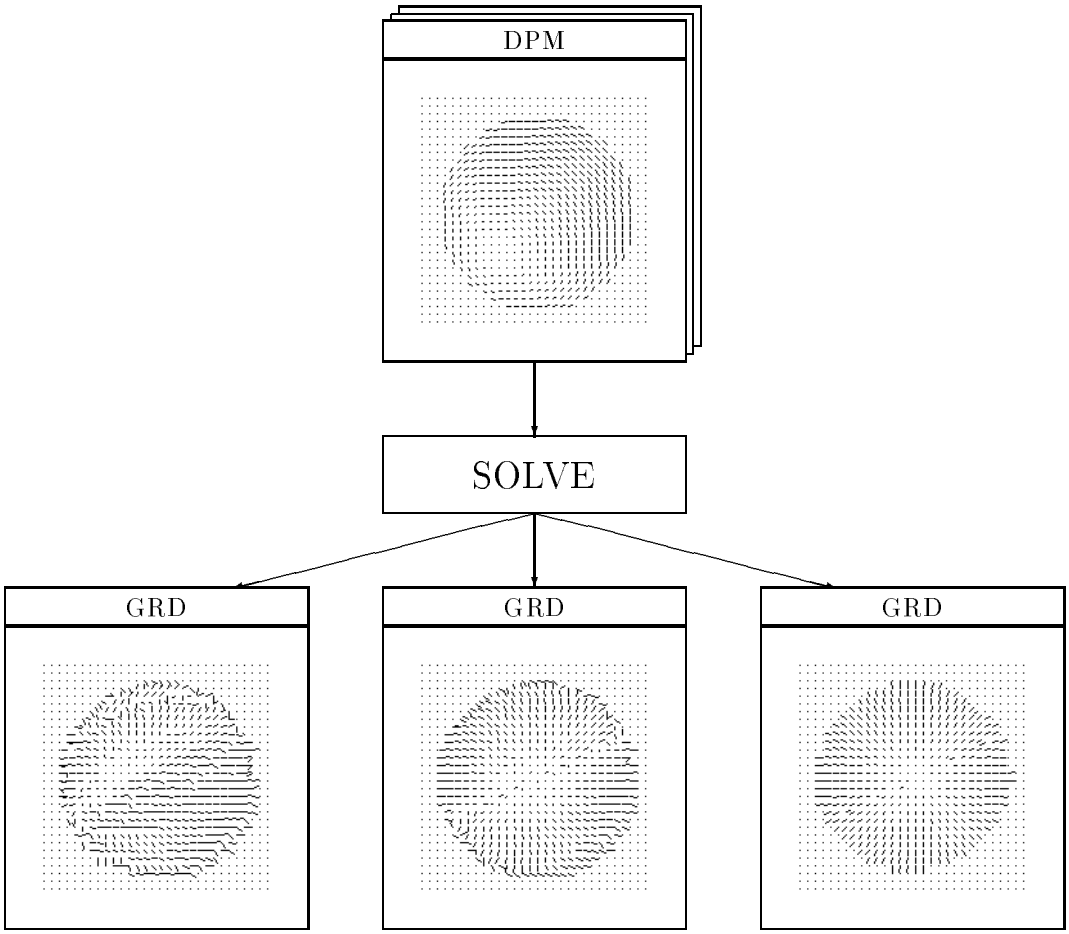

Abbildung 5.2: Berechnung der Gradientendatei mit `solve` aus Displacement-Dateien durch analytische Verfahren oder Minimierung.

Für eine eindeutige Lösung sind mindestens 3 Eingabedateien notwendig, die durch eine fortlaufende *Nummerierung* gekennzeichnet sind und beim Dateinamen nicht mit angegeben werden. Die Dateien können in unterschiedlichen Simulationen entstanden sein und dürfen verschiedene Aufzeichnungsformate enthalten. Das berechnete Objekt und die Dimension der Dateien sollten jedoch übereinstimmen. Damit

ist es durch Änderung der Nummerierung möglich, verschiedene Bildfolgen zusammenzustellen, ohne die Daten in einer zusammenhängenden Folge neu generieren zu müssen.

```
> solve

SOLVE Version 1.0  © Copyright 1993  Volker Rodehorst, TU-Berlin

AUFRUF   :  solve [?] [Optionen] <Dateiname.dpm> <Dateiname.grd>

OPTIONEN : -b[inary]          Aufzeichnungsformat der Loesung.
           -d[irect]          Direkte algebraische Loesung.
           -h[elp]            Ausgabe dieses Benutzungshinweises.
           -m[inimize]  n     Minimierung in n-Grad Schritten.
           -q[uiet]           Keine Statusmeldungen anzeigen.
```

5.4 Darstellung mit VISUAL

Das Programm dient zur Visualisierung und optischen Bewertung der berechneten *Oberflächengradienten*. Neben der Darstellung als Nadelkarte ist auch eine anschaulichere 3D-Repräsentation durch Berechnung einer Tiefenkarte möglich.

Die Mindestgröße einer Nadel beträgt 4 Pixel, um sie noch als solche erkennen zu können. Durch Verändern der Fenstergröße oder Reduzierung der Gradientenanzahl kann man die *Auflösung* bestimmen. Durch die automatische Skalierung werden nicht darstellbare Gradientendateien um einen entsprechenden Faktor verkleinert.

Bei der 3D-Darstellung ist die *Blickrichtung* von oben auf das Objekt voreingestellt und eine perspektivische Seitenansicht läßt sich durch Rotation erzeugen. Die Wahl einer beliebigen Rotationsachse, wie bei `simul`, ist für eine konkrete Ansicht schwer zu ermitteln, so daß eine beliebige Kombination von *orthogonalen Rotationen* angegeben werden kann (z.B. -z 45 -x 60).

Bei der Wahl einer *Kamerarotation* wird die Punktlichtquelle mittransformiert, deren Position auch expizit angegeben werden kann. Die Z-Komponente sollte dabei negativ sein, um eine gute Ausleuchtung zu erhalten.

Für eine schnelle Vorschau oder bei einer flächenhaften Darstellung ist es ebenfalls sinnvoll, die Anzahl der benutzten Gradienten zu reduzieren. Schließlich ist auch das Ausgeben des *Grauwertbildes* von dem generierten Objekt möglich.

```
> visual

VISUAL Version 1.0  © Copyright 1993  Volker Rodehorst, TU-Berlin

AUFRUF   :  visual [?] [Optionen] <Dateiname.grd>

OPTIONEN : -b[inary]          Aufzeichnungsformat der Grafik.
           -c[amera]          Durchfuehrung einer Kamerarotation.
           -d[imension] n     Groesse des quadratischen Fensters.
           -f[ace]            3D-Gitternetz mit verdeckten Flaechen.
```

```
-h[elp]            Ausgabe dieses Benutzungshinweises.
-i[nterpol]        Schattierte 3D-Darstellung mit Glaettung.
-l[ight]     x y z Richtungsvektor zur Punktlichtquelle.
-p[ixmap]    name  Grafik der PixMap unter 'name' speichern.
-q[uiet]           Keine Statusmeldung ausgeben.
-r[educe]    n     Reduzieren der Gradienten um den Faktor n.
-s[hading]         Schattierte 3D-Flaechen mit Lambertmodell.
-w[ireframe]       3D-Drahtgitterliniennetz.
-x[rotation] n     Rotation in n Grad um die X-Achse.
-y[rotation] n     Rotation in n Grad um die Y-Achse.
-z[rotation] n     Rotation in n Grad um die Z-Achse.
```

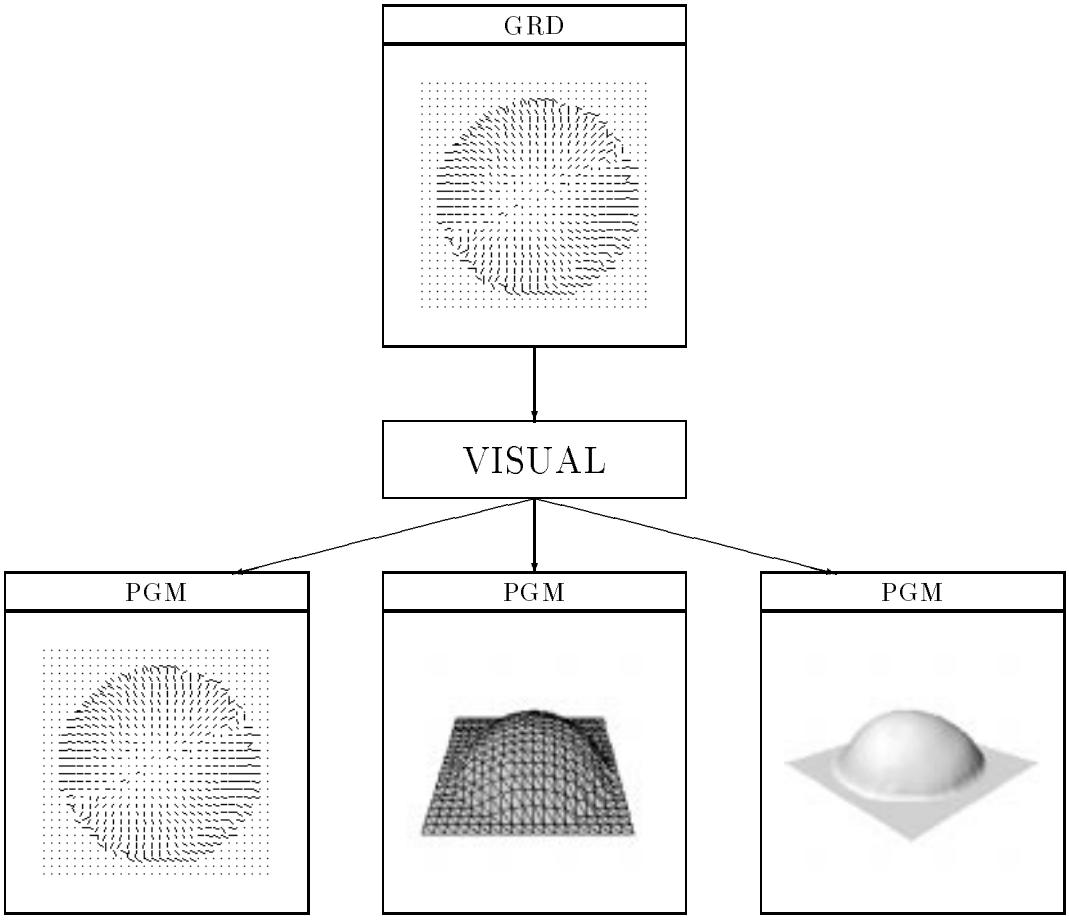

Abbildung 5.3: Graphische Darstellung von Gradientendateien mit `visual`.

5.5 Bewertung durch VERIFY

Das Programm ermittelt diverse Fehlermaße für die berechnete Gradientendatei. Als Referenz wird die mit `simul` erstellte optimale Gradientendatei übergeben. Die Zahlen sind als Durchschnittswerte über das gesamte Bild gemittelt.

Weiter besteht die Möglichkeit, die Differenzen der Oberflächennormalen als Verschiebungsvektoren darzustellen und die mittlere Abweichung des Gradienten (p, q) als Fehler in Abhängigkeit vom Winkel anzuzeigen. Eine genauere Interpretation der Graphen ist bei den Versuchsergebnissen zu finden.

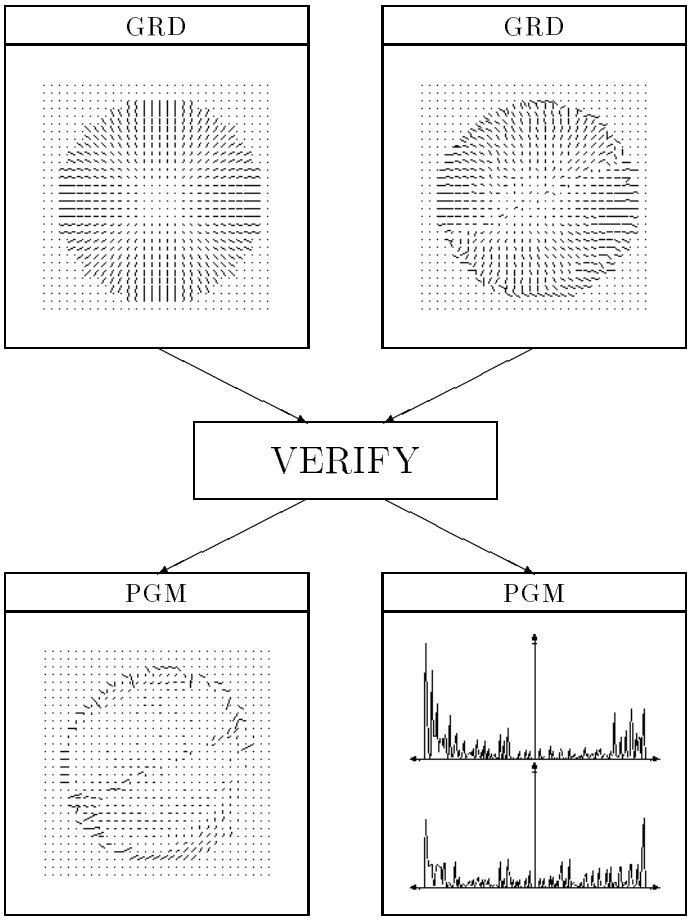

Abbildung 5.4: Bewertung der berechneten Gradienten durch `verify`.

```
> verify

VERIFY Version 1.0  © Copyright 1993  Volker Rodehorst, TU-Berlin

AUFRUF   :  verify [?] [Optionen] <Dateiname.orig> <Dateiname.calc>
```

```
OPTIONEN : -b[inary]          Aufzeichnungsformat der Grafik.
           -d[imension] n     Groesse des quadratischen Fensters.
           -e[rror]           Fehler in abhaengigkeit der Winkel.
           -h[elp]            Ausgabe dieses Benutzungshinweises.
           -p[ixmap]    name  Grafik der PixMap unter 'name' speichern.
           -q[uiet]           Keine Statusmeldungen anzeigen.
           -r[educe]    n     Reduzieren der Gradienten um den Faktor n.
           -v[ector]          Darstellung der Differenzvektoren.
```

5.6 Aufbau der Konfigurationsdatei

Die Konfigurationsdatei ist ein ASCII-Text, der mit einem üblichen Editor verändert werden kann. Bei der Erstellung einer neuen Datei ist zur Kennzeichnung das Anhängen des Suffixes '.cfg' empfehlenswert.

5.6.1 Reservierte Schlüsselworte

Die folgenden Schlüsselwörter sind in beliebiger Reihenfolge zu verwenden, wobei zwischen Groß- und Kleinschreibung nicht unterschieden wird.

`Albedo`	`Background`	`Binary`	`Camera`	`Dispfile`	`Gradfile`
`Image`	`Imagefile`	`Light`	`Points`	`Polygons`	`Quiet`
`Resolution`	`Scale`	`Screen`	`Smooth`	`Sphere`	`Transform`

Ergänzende Kommentare, die in der C-Syntax mit $/*\ldots*/$ gekennzeichnet sind, können an belieber Stelle im Text zur Erklaŕung verwendet werden. In der folgenden Befehlsbeschreibung werden Schlüsselworte in `Schreibmaschinenschrift` und Terminalsymbole, wie z.B. Konstanten oder Zeichenketten für Dateinamen, in *Schrägschrift* gedruckt. Optionale Parameter werden durch Klammerung [...] angegeben.

5.6.2 Bildschirm

Für die Darstellung der Grafiken ist die Grösse des Fensters in Pixeln mit

`Screen` *xsize ysize*

zu definieren. Die maximale Anzahl ist dabei auf 1024 Punkte begrenzt. Um die Dimension der stets quadratischen Bilder und erzeugten Karten verändern zu können ist

`Scale` *pixel*

zu verwenden.

5.6.3 Objektdefinition

Zu Zeit können zwei Klassen von Objekten definiert werden. Mit

`Sphere` *radius*

wird eine *Kugel* erzeugt und durch einen Parameter kann man die Größe beeinflussen. Der Radius kann Werte zwischen 0 und 1 enthalten, wobei mit 1 der Bereich optimal ausgenutzt wird.

Als zweites kann ein *Polyeder* erzeugt werden, der durch Angabe von Punkten und Polygone definiert wird. Zu Beginn der *Punktliste* wird die Anzahl n der nachfolgenden Punkte benötigt

`Points` n $x_1 y_1 z_1$ $x_2 y_2 z_2$ ... $x_n y_n z_n$

Die Koordinaten eines Punktes sollten normiert sein und zwischen -1 und 1 liegen, so daß der Ursprung des Koordinatensystems im Objektmittelpunkt liegt. Der Begriff Polygon wird in diesem Fall sehr speziell verwandt, denn es handelt sich im Prinzip jeweils nur um Dreiecke. In der *Polygonliste*, die durch die Polygonanzahl m angeführt wird, werden die Punktindizes $(1, 2, \ldots, n)$ von den drei Begrenzungspunkten definiert

`Polygons` m $p1_1 p2_1 p3_1$ $p1_2 p2_2 p3_2$ ... $p1_m p2_m p3_m$

Für eine korrekte Oberlächennormale ist die Reihenfolge der Punkte gegen den Uhrzeigersinn zu konstruieren.

5.6.4 Beleuchtung

Der Vektor in Richtung der unendlich entfernten Punktlichtquelle kann mit

`Light` *xpos ypos zpos*

angegeben werden. Die Koordinaten brauchen nicht normiert zu sein, so daß auch die Position der Lichtquelle angegeben werden kann.

5.6.5 Transformationen

Eine Folge von n Bildern wird durch Rotation des initialen Objekts simuliert. Diese kann mit Angabe einer beliebigen Achse, gefolgt von einen Drehwinkel in Grad, beschrieben werden. Der erste Listeneintrag legt die Transformation des initialen Referenzobjektes fest, gefolgt von beliebig vielen Einträgen.

`Transform` n $x_1 y_1 z_1 w_1$ $x_2 y_2 y_2 w_2$ ... $x_n y_n z_n w_n$ `[Camera]`

Ein positiver Winkel entspricht dabei einer Drehung im Uhrzeigersinn, wenn man von einer positiven Achse zum Ursprung schaut. Durch den Zusatz `Camera` wird eine Kamerarotation durchgeführt, die neben dem Objekt auch die Beleuchtung transformiert.

5.6.6 Weitere Einstellungen

Neben den bisher notwendigen Anweisungen sind auch diverse optionale Befehle vorhanden. Die *Hintergrundfarbe* läßt sich mit

`Background` *red green blue*

individuell auf die persönlichen Bedürfnisse einstellen. Für eine Ausgabe auf einem Drucker ist es manchmal sinnvoll, den normalerweise schwarzen Hintergrund auf weiß zu wechseln. Der Wertebereich der einzelnen Farbkomponenten variiert von 0 bis 255. Um erweiterte *Statusmeldungen* bei der Berechnung zu unterdrücken, kann

`Quiet`

verwendet werden.

5.6.7 Objektoberfläche

Die in der Lambert-Reflektion verwendete *Reflektanzkonstante* ist mit

`Albedo` *c*

zu verändern, welche mit 1 voreingestellt ist. Bei der Berechnung eines Polyeders kann mit

`Smooth`

eine *Glättung* der Polygonkanten durch lineare Interpolation der Oberflächennormalen erzielt werden.

5.6.8 Beispiel: Kugel

```
Screen     200 200      /* Groesse des Grafikbildschirms */
Sphere     0.85                  /* Eine Kugel erzeugen */
Scale      20                   /* Groesse des Objektes */
Light     -1.0  2.0 -4.0    /* Punktfoermige Lichtquelle */

Transform  4                     /* 4 Bilder generieren */
  0.0  0.0  0.0   0.0                 /* Initiales Bild */
  1.0  0.0  0.0   2.0  /* Rotation 2 Grad um die x-Achse */
  0.0  1.0  0.0   4.0  /* Rotation 4 Grad um die y-Achse */
  0.0  0.0  1.0   8.0  /* Rotation 8 Grad um die z-Achse */

Dispfile sphere.dpm          /* Verschiebungsdifferenzen */
```

5.6.9 Beispiel: Würfel

Das folgende Beispiel soll die Verwendung von polyedrischen Objekten veranschaulichen.

```
Screen     512 512      /* Groesse des Grafikbildschirms */
Background 255 255 255         /* Hintergrundfarbe weiss */
Scale      400                  /* Groesse des Polyeders */
Resolution 10                /* Aufloesung des Polyeders */
Albedo     1.0                    /* Reflektanzkonstante */
Light      2.0  1.0 -4.0    /* Punktfoermige Lichtquelle */

Transform 7                        /* 7 Bilder generieren */
  4.0  3.0  2.0  55.0   /* Initiales Bild transformieren */
  1.0  0.0  0.0   3.0  /* Rotation 3 Grad um die x-Achse */
  0.0  1.0  0.0   6.0  /* Rotation 6 Grad um die y-Achse */
  0.0  0.0  1.0   9.0  /* Rotation 9 Grad um die z-Achse */
  1.0  0.0  0.0  -9.0 /* Rotation -9 Grad um die x-Achse */
  0.0  1.0  0.0  -6.0 /* Rotation -6 Grad um die y-Achse */
  0.0  0.0  1.0  -3.0 /* Rotation -3 Grad um die z-Achse */

Imagefile cube.pgm Binary  /* Grauwertbild des Polyeders */
Gradfile  cube.grd Binary    /* Nadelkarte des Polyeders */
Dispfile  cube.dpm Binary    /* Verschiebungsdifferenzen */

Points 8
/* 1 */    -0.45   0.45   0.45     /*  Y      1-------2  */
/* 2 */     0.45   0.45   0.45     /*  |     /|      /|  */
/* 3 */     0.45  -0.45   0.45     /*  |    5-+-----6 |  */
/* 4 */    -0.45  -0.45   0.45     /*  |    | |     | |  */
/* 5 */    -0.45   0.45  -0.45     /*  |  Z | 4-----+-3  */
/* 6 */     0.45   0.45  -0.45     /*  | /  |/      |/   */
/* 7 */     0.45  -0.45  -0.45     /*  |/   8-------7    */
/* 8 */    -0.45  -0.45  -0.45     /*  +------------- X  */

Polygons 12
/*  1 */    6  1  2                    /* Flaeche oben */
/*  2 */    1  6  5
/*  3 */    4  7  3                   /* Flaeche unten */
/*  4 */    7  4  8
/*  5 */    1  8  4                   /* Flaeche links */
/*  6 */    8  1  5
/*  7 */    3  6  2                  /* Flaeche rechts */
/*  8 */    6  3  7
/*  9 */    4  2  1                  /* Flaeche hinten */
/* 10 */    2  4  3
/* 11 */    6  8  5                   /* Flaeche vorne */
/* 12 */    8  6  7
```

5.6.10 Dateierzeugung

Zur Verbindung zwischen den Programmen sind verschiedene Dateien eingeführt worden, die jeweils optional angelegt werden können.

`Imagefile`	*name.pgm*	[`Binary`]	
`Gradfile`	*name.grd*	[`Binary`]	[`Image`]
`Dispfile`	*name.dpm*	[`Binary`]	[`Image`]

Mit `Dispfile` werden bei n Transformationen $(n-1)$ Dateien erzeugt, deren Unterscheidung durch Anhängen einer fortlaufenden Nummerierung gewährleistet wird. Durch den Zusatz `Image` werden nicht die Daten, sondern die dargestellte Grafik (Nadelkarte bzw. Optischer Fluß) als Grauwertbild gespeichert.

5.7 Dateiformate

Bei der Verwendung der nachfolgenden Dateiformate ist möglichst auf einen korrekten Aufbau zu achten. Im Textformat sind zwar beliebig viele Leerzeichen, Tabulatoren und Zeilenvorschübe erlaubt, jedoch ist eine zusätzliche Kommentierung z.Z. nicht möglich.

5.7.1 Aufbau der Gradientendatei .grd

Die Datei beinhaltet die normierten Oberflächengradienten p, q und hat folgenden Aufbau:

id	
n	
p_1	q_1
p_2	q_2
$\vdots$	$\vdots$
p_{n^2}	q_{n^2}

Die Identifikation *id* lautet im Textformat `'#GRD'` und im Binärformat `0x47524144 = 'GRAD'` Der *int* n bezeichnet dabei eine Dimension der quadratischen Gradientenmatrix gefolgt von den *double* p und q, die aufgrund der Normierung einen Wertebereich von 0 bis 1 haben (siehe Kapitel Implementierung und Realisierung). Im Textformat werden sechs Nachkommastellen beachtet, welches einer Genauigkeit von einem *float* entspricht.

5.7.2 Aufbau der Displacement-Datei .dpm

Die Datei enthält die Differenzen der Verschiebungsvektoren u, v und das entsprechende Grauwertverhältnis r. Ein nicht definiertes Verhältnis mit einem Grauwert von 0 in Bild f_1 wird durch den Wert 256.0 gekennzeichnet. Sie besitzt folgende Struktur:

id				
s_1	s_2	s_3		
n				
r_1	ux_1	uy_1	vx_1	vy_1
r_2	ux_2	uy_2	vx_2	vy_2
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
r_{n^2}	ux_{n^2}	uy_{n^2}	vx_{n^2}	vy_{n^2}

Die Kennung *id* einer Displacement-Datei wird im Textformat mit `'#DPM'` und im Binärformat mit `0x44495350 = 'DISP'` durchgeführt. Die ersten drei *double* bezeichnen die Richtung s zur Lichtquelle und der *int* n definiert die Dimension der folgenden quadratischen Matrix. Alle anderen Werte sind wie bei der Gradientendatei je nach Aufzeichnungsformat von Typ *double* bzw. *float*.

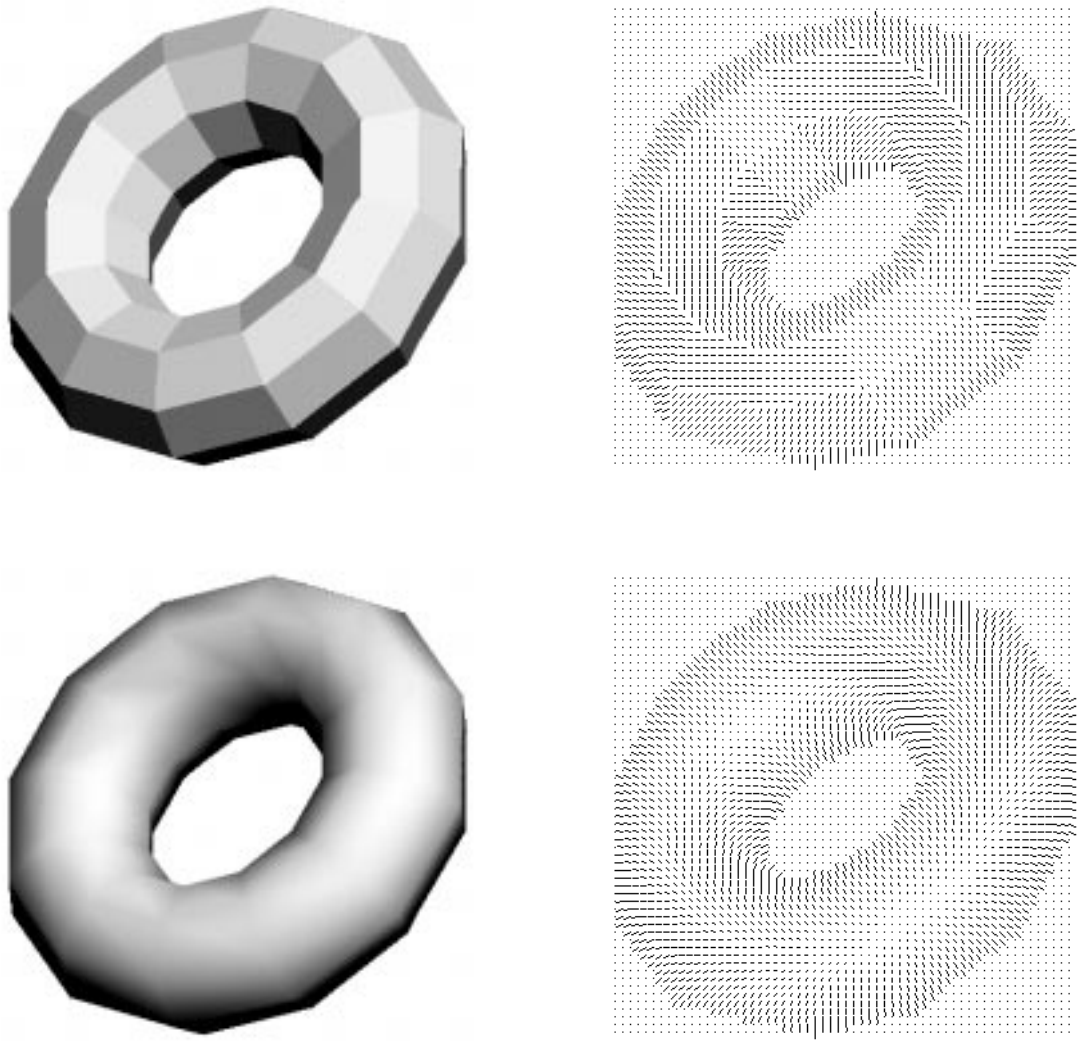

Abbildung 5.5: Glättung durch lineare Interpolation der Normalen

Kapitel 6

Versuchsergebnisse

6.1 Vergleich der verschiedenen Lösungsverfahren

Die folgenden graphischen Darstellungen der Fehler, die bei der Berechnung mit den einzelnen Lösungsverfahren entstehen, wurden mit `verify` erzeugt. Die jeweils linke Abbildung zeigt die Differenzen der berechneten Oberflächennormalen zu den idealen Referenzwerten. Diese Unterschiede werden als Verschiebungsvektoren dargestellt und sind nicht mit einer Nadelkarte zu verwechseln.

Bei der jeweils benachbarten Abbildung wird der Fehler in Abhängigkeit vom Winkel der Oberflächennormalen betrachtet. Die Winkel α und β beschreiben die Neigung der Normalen bezüglich des orthogonalen Koordinatensystems in X bzw. Y-Richtung (siehe Abb. 4.2). Auf der δ-Achse ist dann die mittlere Abweichung vom Referenzvektor als Winkel aufgetragen.

6.1.1 Direkte Methode

- Die direkte Methode stellt das weitaus schnellste Verfahren dar.
- Es handelt sich um eine sehr instabile Rechnung, denn durch z. T. echte Subtraktionen entstehen so kleine Werte, die im Rechner nicht darstellbar sind.
- Häufig auftretende negative Terme unter einer Wurzel werden approximiert und konnten nicht zufriedenstellend behandelt werden.
- Nur Oberflächennormalen $\mathbf{n}$, die mit dem Beleuchtungsvektor $\mathbf{s}$ einen Winkel $\alpha < \frac{\pi}{2}$ bilden, werden einigermaßen erkannt.
- Das direkte Verfahren ist in dieser Form auch für ein integratives Verfahren unbrauchbar.

6.1.2 Numerisches Iterationsverfahren

- Das Iterationsverfahren liefert im gesamten Bild wesentlich stabilere Ergebnisse, was die Anzahl der Lösungen und ihre Qualität betrifft.
- Stärkere Ausfälle am Objektrand lassen sich durch Einführen eines Schwellwertes leicht beseitigen.

- Während der Berechnung wird auch bei komplexwertigen Termen 'korrekt' fortgefahren und nicht approximiert.
- Das Iterationsverfahren zeigte bei den getesteten Beispielen eine sehr schnelle Konvergenz.
- Das Verfahren ist sowohl integrativ als auch als autonomes Verfahren einsetzbar.
- Eine sichtbare Verbesserung der Werte anhand einer Nachiteration mit dem Newton-Verfahren, durch Verwendung der Nullstellen des Muller-Verfahrens als Startwerte, konnte nicht erzielt werden.

6.1.3 Methode der kleinsten Quadrate

- Die Minimierung mit der Methode der kleinsten Quadrate lieferte in den Experimenten durchweg die besten Resultate.
- Durch wählbare Auflösung bei der Minimierung kann die Qualität der Berechnung den zeitlichen Erfordernissen angepaß werden.
- Auch das Einsetzen großer Winkel (z.B. 10 Grad) erzeugte verwendbare Nadelkarten.
- Der entscheidende Nachteil liegt in dem quadratischen Aufwand der Minimierung.
- Durch die diskrete Beschränkung der Oberflächennormalen auf 1..179 Grad fehlen Lösungen im Randbereich der Objekte.

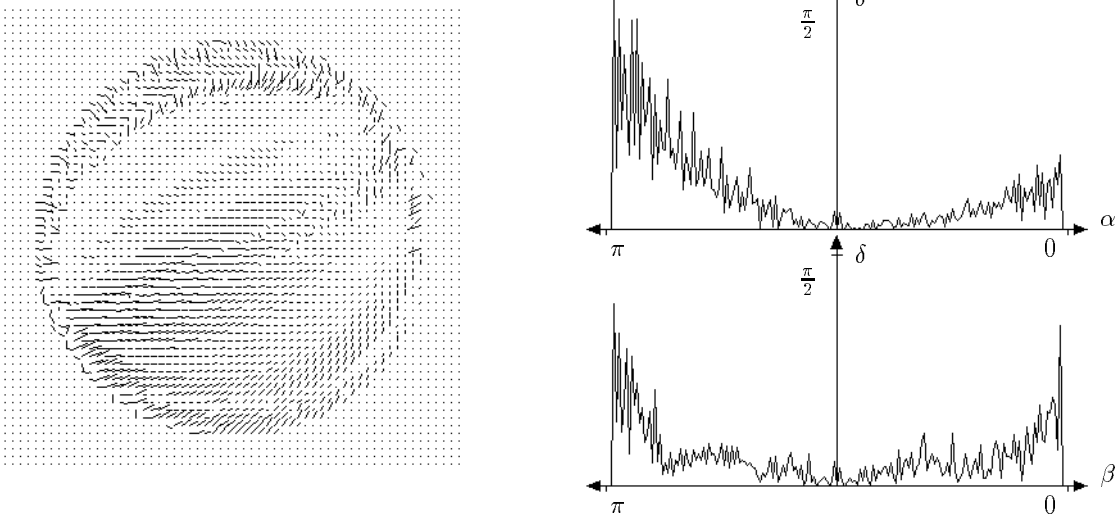

Abbildung 6.1: Differenzen und mittlere Winkelabweichung beim direkten Verfahren.

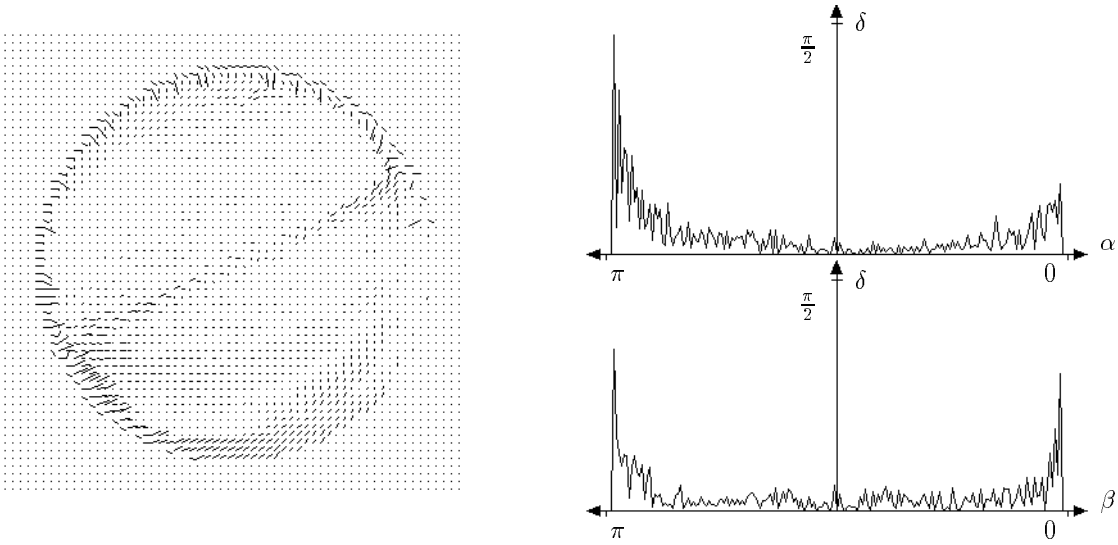

Abbildung 6.2: Differenzen und durchschnittliche Winkelfehler beim iterativen Verfahren.

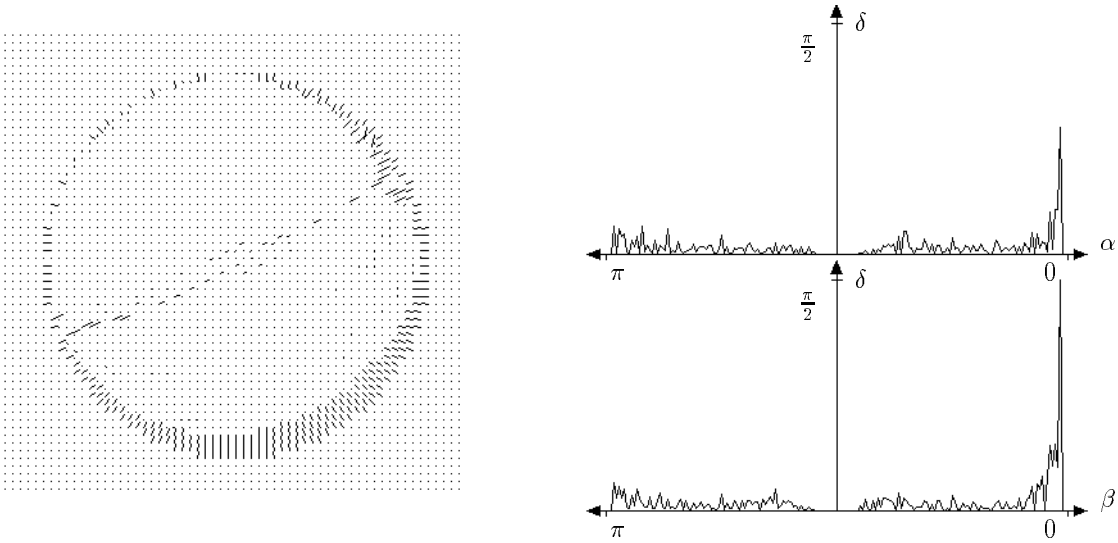

Abbildung 6.3: Differenzen und mittlere Winkelabweichung bei der Methode der kleinsten Quadrate.

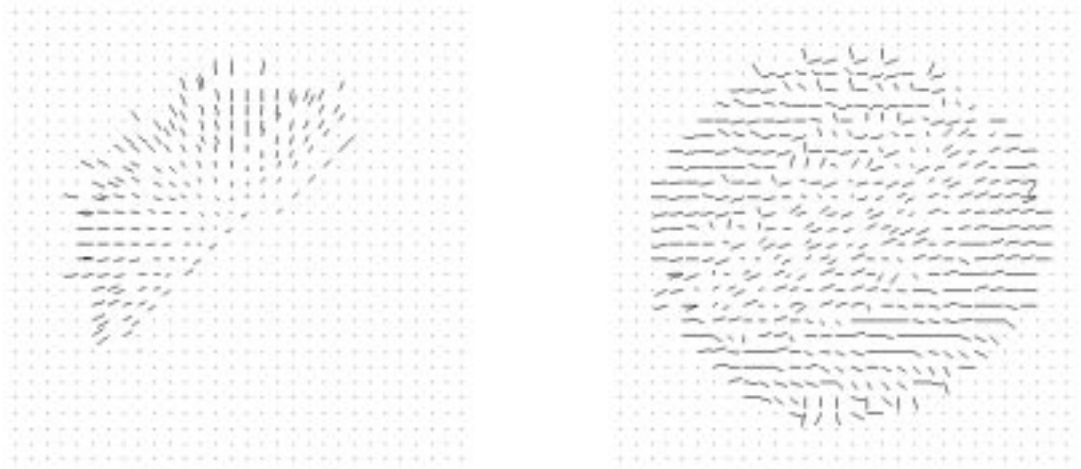

Abbildung 6.4: Ausfälle in der Nadelkarte durch schlechte Ausleuchtung $\mathbf{s}$ = (-100, 100, -1) und einseitige Bewegung (Rotation nur um die X-Achse).

6.2 Ergebnisse

- Zu warnen ist vor einer Beleuchtungsrichtung entlang der optischen Achse. Da beim Beleuchtungs-Constraint in allen Koeffizienten des Polynoms s_1 und s_2 als Faktoren vorhanden sind, werden $a, \ldots, f$ zu 0.
- Die Bewegung sollte keine Vorzugsrichtung besitzen, sondern möglichst Rotationsanteile um alle drei kartesischen Koordinatenachsen enthalten. Oberflächennormalen, die senkrecht zur Bewegungsrichtung verlaufen, werden besonders gut detektiert.
- Durch kleine Bewegungen in 1^o Schritten erhält man dichte Nadelkarten und bei großen Rotationen um 30^o verbessert sich die Qualität. Die Anzahl der Nadeln ist dann aber gering, weil bei der Verschiebungsvektorbestimmung viele Punkte hinter dem Objekt verschwinden bzw. vorher nicht sichtbar waren.
- Es kann bei Oberflächennormalen mit einer bestimmten Richtung zur Lichtquelle zu geringen Ausfällen kommen. Bei der Kugel handelt es sich um eine Diagonale, die senkrecht zum Beleuchtungsvektor verläuft. Diese entsteht dadurch, daß die Koeffizienten d, e und f unterhalb der Rechengenauigkeit nahe bei 0 liegen.
- Alle vorgestellten Verfahren sind massiv parallelisierbar. Selbst in sequentieller Form sind sie von der benötigen Rechenleistung her auf einem üblichen PC ausführbar, wenn keine Echtzeitanforderungen gestellt werden.

- Das Iterationsverfahren und die Minimierung sind in der Kombination mit anderen Verfahren uneingeschränkt zu empfehlen.
- Bei der exakten Bestimmbarkeit der lokalen Verschiebungsvektoren und der Lichtquellenrichtung sind sie auch für den Einsatz als eigenständiges Verfahren geeignet.

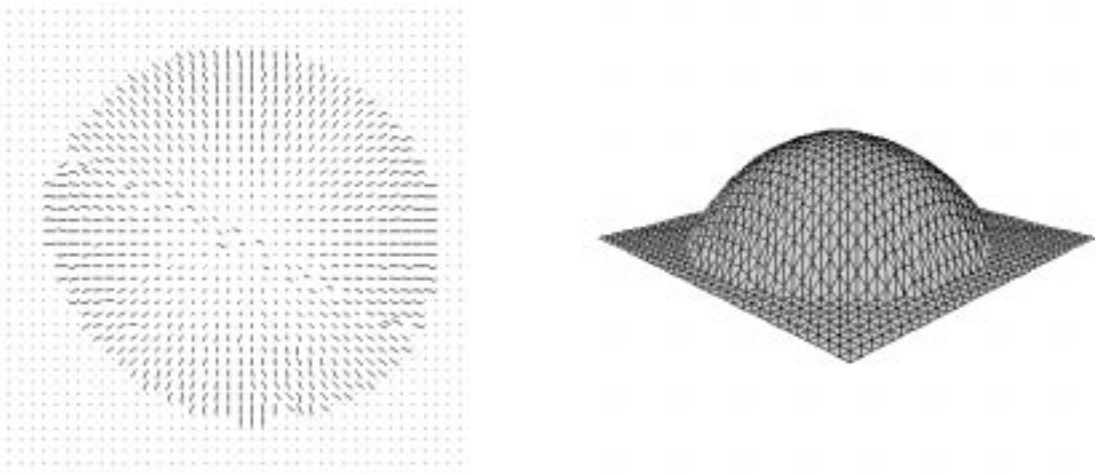

Abbildung 6.5: Erkannte Kugelhälfte als Nadelkarte und Flächenmodell.

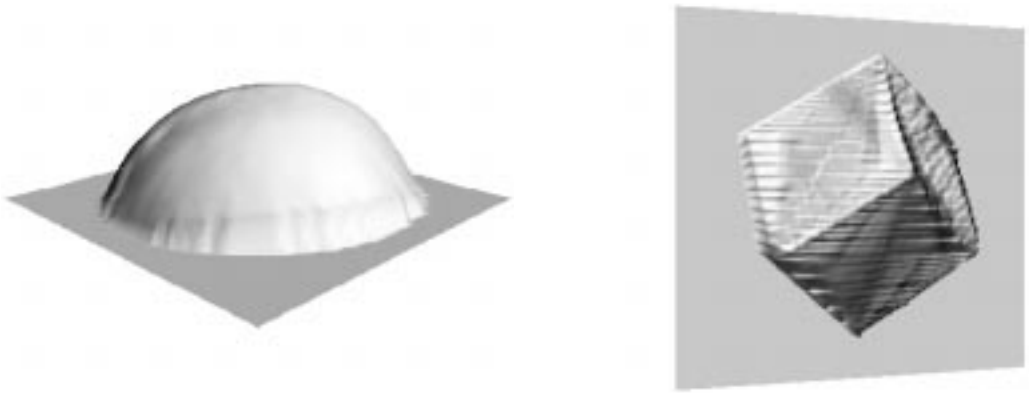

Abbildung 6.6: Rekonstruierte Kugel und Würfel als schattiertes Modell.

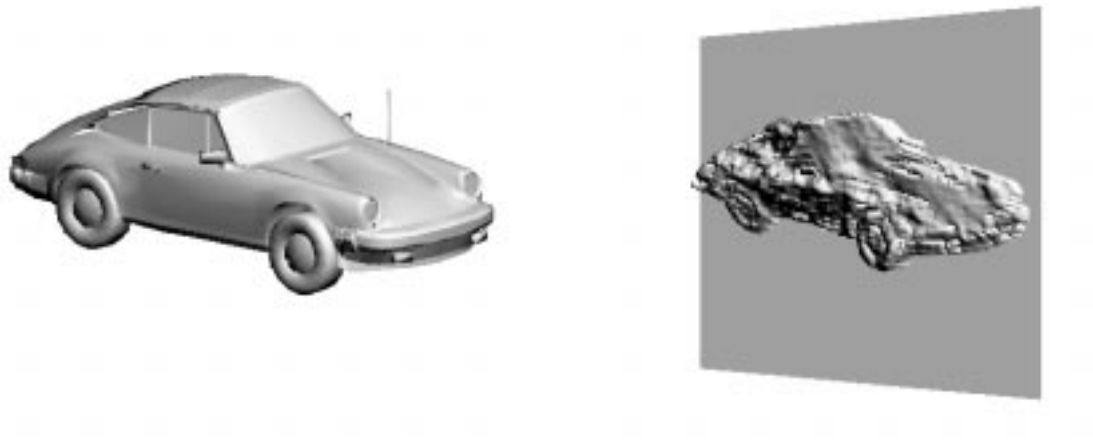

Abbildung 6.7: Erkennungsversuche eines komplexen Sportwagens.

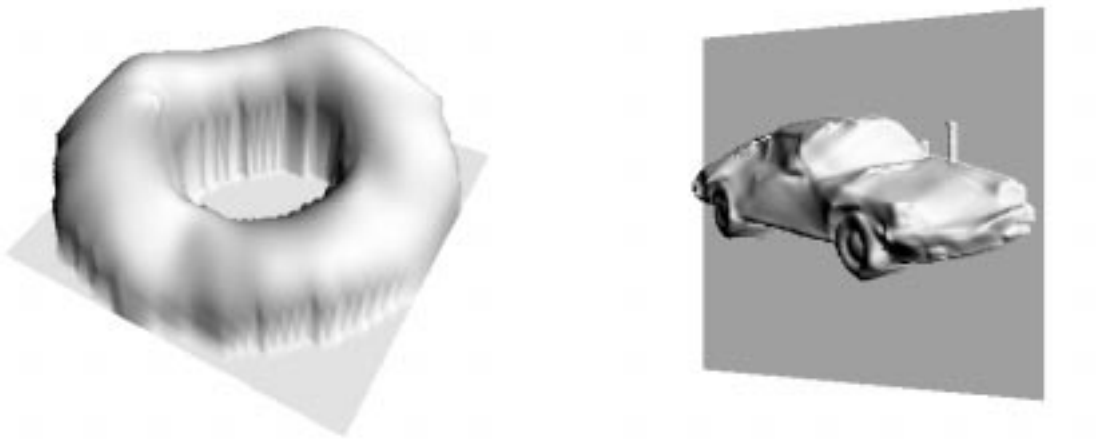

Abbildung 6.8: Leistungsfähigkeit von `visual` bei optimalen Referenzdaten.

Anhang A

Kurven zweiter Ordnung

Dieser anfangs vielversprechende Ausflug in die lineare Algebra wurde leider aus Gründen, die am Ende des folgenden Abschnitts zu finden sind, nicht für das Beleuchtungs-Constraints verwandt. Jedoch finde ich den generellen Ansatz, durch Transformation das gemischte und die linearen Glieder der Gleichung (2.6) zu beseitigen, recht interessant. Eine Implementierung dieser Vereinfachung des Polynoms bestätigte auch die allgemeine Verwendbarkeit des folgenden Verfahrens, so daß ich diesen Exkurs mit aufgenommen habe.

A.1 Die Matrizenschreibweise

Bei der Beschäftigung mit dem Polynom stellte ich fest, daß man üblicherweise diese Gleichungsform in einer anderen Darstellung behandelt. Punktmengen einer Gleichung, deren Koordinaten folgender Form genügen

$$ax^2 + 2bxy + cy^2 + 2dx + 2ey + f = 0$$

werden als *Kurven zweiter Ordnung* der Ebene bezeichnet. Eine bessere Darstellung erhält man in Matrizenschreibweise:

$$\mathbf{xAx^T} + \mathbf{2ax^T} + f = 0, \qquad \mathbf{A} = \begin{pmatrix} a & b \\ b & c \end{pmatrix}, \qquad \mathbf{a} = (d, e), \qquad \mathbf{x} = (x, y).$$

Somit hat das umgeformte Beleuchtungs-Constaint (2.6) in dieser Matrizendarstellung die Parameter

$$\mathbf{A} = \begin{pmatrix} a & \frac{c}{2} \\ \frac{c}{2} & b \end{pmatrix}, \qquad \mathbf{a} = (\frac{d}{2}, \frac{e}{2}), \qquad \mathbf{x} = (p, q). \tag{A.1}$$

Weil die Kurven zweiter Ordnung der Ebene als Schnitt mit einem geraden Kreiskegel dargestellt werden können, werden sie oft auch *Kegelschnitte* genannt. Wenn die Schnittebene nicht durch die Kegelspitze verläuft, so entstehen eine Hyperbel, Parabel oder Ellipse, ansonsten zerfallene Kegelschnitte. Parallele Geraden entstehen, wenn die Kegelspitze ins Unendliche rückt und der Kegel in einen Zylinder ausartet.

A.2 Klassifizierung der Kurve

Nach der Transformation der Kurve auf die Normalform, kann das Gebilde klassifiziert werden. Anstelle der Determinanten von A mit $ab - \frac{c^2}{4} = \frac{1}{4}(4ab - c^2)$, betrachtet man dazu normalerweise wegen der Invarianzeigenschaft die *Diskriminante* mit $-4 \det A = c^2 - 4ab$.

Die Klassifizierung ist in folgender Tabelle zusammengefaßt:

1. Kurven mit Symmetriepunkten, $c^2 - 4ab > 0$

Rang A	Normalform	Name der Kurve
3	$\frac{p^2}{a^2} - \frac{q^2}{c^2} = 1$	Hyperbel
2	$\frac{p^2}{a^2} - \frac{q^2}{c^2} = 0$	Paar sich schneidende Geraden
3	$\frac{p^2}{a^2} + \frac{q^2}{c^2} = 1$	(reelle) Ellipse
3	$\frac{p^2}{a^2} + \frac{q^2}{c^2} = -1$	Im Reellen keine Lösung (imaginäre Ellipse)
2	$\frac{p^2}{a^2} + \frac{q^2}{c^2} = 0$	Nullpunkt (entartete Punkt-Ellipse)

2. Kurven ohne Symmetriepunkte, $c^2 - 4ab = 0$

Rang A	Normalform	Name der Kurve
3	$p^2 = 4xq$	Parabel
2	$p^2 = a^2$	Parallele Geraden
2	$p^2 = -a^2$	Nicht reelle Geraden
1	$p^2 = 0$	Gerade

A.3 Verknüpfung mit Transformationsmatrizen

Ziel dieser Transformation ist es, das gemischte und die linearen Glieder der Gleichung durch Translation und Rotation (*Hauptachsentransformation)* zu beseitigen. Durch Verwendung homogener Koordinaten können die Transformationsgleichungen in einer einheitlichen Matrixform dargestellt werden. Ein Punkt (p, q) wird in homogener Koordinatenschreibweise durch das Tripel (p_h, q_h, w) dargestellt, wobei w beliebig ungleich 0 ist, und

$$p_h = p \cdot w \qquad \text{und} \qquad q_h = q \cdot w$$

gilt. Für die Transformation einer Kurve auf die Normalform ist also eine erweiterte Matrixdarstellung zweckmäßig, die durch Verwendung von homogenen Koordinaten mit $w = 1$ entsteht

$$ap^2 + bq^2 + cpq + dp + eq + f = (p, q, 1) \cdot P \cdot \begin{pmatrix} p \\ q \\ 1 \end{pmatrix} = 0, \qquad P = \begin{pmatrix} a & \frac{c}{2} & \frac{d}{2} \\ \frac{c}{2} & b & \frac{e}{2} \\ \frac{d}{2} & \frac{e}{2} & f \end{pmatrix}.$$

Die geforderte Translation $(p', q') = (p + h, q + k)$ läßt sich nun in Matrixschreibweise

$$\begin{pmatrix} p' \\ q' \\ 1 \end{pmatrix} = T \cdot \begin{pmatrix} p \\ q \\ 1 \end{pmatrix}, \qquad T = \begin{pmatrix} 1 & 0 & h \\ 0 & 1 & k \\ 0 & 0 & 1 \end{pmatrix},$$

mit den Translationsparametern

$$h = \frac{P_{13}}{P_{11}} = \frac{d}{2a}, \qquad k = \frac{P_{23}}{P_{22}} = \frac{e}{2b}$$

ausdrücken. Die Rotation mit der Rotationsmatrix M kann als

$$\begin{pmatrix} p' \\ q' \\ 1 \end{pmatrix} = R \cdot \begin{pmatrix} p \\ q \\ 1 \end{pmatrix}, \qquad R = \begin{pmatrix} M_{11} & M_{12} & 0 \\ M_{21} & M_{22} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

dargestellt werden. Die Komponenten von M erhält man durch die Berechnung der Eigenvektoren $(E_{11}, E_{21})^T$ und $(E_{12}, E_{22})^T$ aus der Matrix A in Gleichung (A.1). An dieser Stelle ist anzumerken, daß die Eigenvektorbestimmung allgemein eine der schwereren Aufgaben darstellt. Doch bei dieser geringen Matrixgröße muß man nicht unbedingt auf ein aufwendiges Verfahren zurückgreifen. Damit für die Determinante $\det M = 1$ gilt, ist anschließend eine Normalisierung notwendig

$$M = \begin{pmatrix} \frac{E_{11}}{n} & \frac{E_{12}}{n} \\ \frac{E_{21}}{n} & \frac{E_{22}}{n} \end{pmatrix}, \qquad n = \sqrt{E_{11}E_{22} - E_{21}E_{12}}.$$

Die Rotation mit dieser Matrix M erzeugt eine neue Matrix, in der nur die Diagonalelemente belegt und die restlichen Komponenten gleich 0 sind. In M_{11} und M_{22} befinden sich dann die Eigenwerte von A und die neuen Koordinaten p', q' korrespondieren zu der Basis $B = \{(M_{11}, M_{21}), (M_{12}, M_{22})\}$

A.4 Durchführung der Transformation

Zuerst wird die Kurve mit R rotiert, um die gemischte Komponente der Gleichung zu beseitigen, und anschließend der Symmetriepunkt der Kurve mit T in den Koordinatenursprung transliert, wobei die linearen Anteile entfallen. Durch folgende Kombination der Matrizen wird die Kurve transformiert:

$$P' = T^T \cdot (R^T \cdot P \cdot R) \cdot T = (R \cdot T)^T \cdot P \cdot (R \cdot T).$$

Als Ergebnis dieser Transformation liegen die Komponenten a', b' und f' der Normalform auf der Diagonalen von P' und die restlichen Komponenten sind gleich 0. Somit hat sich die Kurve auf die einfache Form

$$a'p^2 + b'q^2 + f' = 0$$

reduziert. In dem neuen Bezugssystem läßt es sich also leichter rechnen und eine Rücktransformation von Punkten in das ursprüngliche Koordinatensystem stellt auch kein Problem dar.

Nach dieser vorteilhaften Vereinfachung des Polynoms ist aber das neue Bezugssystem das Problem bei der Implementierung des Beleuchtungs-Constraints. Für die eindeutige Lösung von p und q ist der Schnitt von mindestens drei Kurven erforderlich. Da jede Kurve in ihrer Normalform ein unterschiedliches Bezugssystem hat, ist der Schnitt in einem Koordinatensystem nicht möglich.

Weiter hat sich herausgestellt, daß der Aufwand für die Eigenvektorberechnung von nur einer Kurve in keinem Verhältnis zu den minimalen Vorteilen stehen, wenn die anderen Kurven in ihrer konventionellen Gestalt belassen werden müssen.

Anhang B

Das Muller-Verfahren

Dieses Kapitel soll das verwendete Muller-Verfahren zum besseren Verständnis des Quelltextes grob umschreiben, aber eignet sich weniger als Einführung in die numerische Mathematik. Für einen fundierten Zugang ist die hier verwendete Literatur [ENG88, MUL56] heranzuziehen.

B.1 Prinzip des Verfahrens

Gesucht sind sämtliche reellen und konjugiert komplexen Nullstellen eines Polynoms P_n mit reellen Koeffizienten ohne Kenntnis von Startwerten.

$$\begin{aligned} P_n(x) &= a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n \\ &= \sum_{j=0}^{n} a_j x^j \qquad \text{für } a_j \in IR, \qquad a_n \neq 0. \end{aligned} \tag{B.1}$$

Bei der *iterativen Bestimmung von Nullstellen* könnten sich die dividierten Polynome immer weiter von den Nullstellen des Ausgangspolynoms P_n entfernen, so daß die Genauigkeit immer mehr abnimmt.

Deshalb sollte das Abdividieren von Nullstellen grundsätzlich mit der betragskleinsten Nullstelle beginnen, d.h. mit einer Methode zu arbeiten die für das jeweilige Polynom eine Anfangsnäherung so auswählt, daß die Iteration gegen die *betragskleinste Nullstelle* konvergiert.

Zunächst wird durch Muller-Iteration ein hinreichend guter Näherungswert x_1^N für die betragskleinste Nullstelle x_1 von P_n bestimmt. Nach Division von P_n durch $x - x_1^N$ mit Horner und Vernachlässigung des Restes erhält man ein Polynom P_{n-1} vom Grad n-1, welches ungefähr gleich dem Deflationspolynom $\frac{P_n(x)}{x-x_1}$ ist.

Von P_{n-1} wird wiederum durch Muller-Iteration ein Näherungswert x_2^N für die betragskleinste Nullstelle x_2 bestimmt. Dabei ist mit $f \equiv P_{n-1}$ statt $f \equiv P_n$ zu arbeiten und x_1 durch x_2 zu ersetzen. Mit x_2^N wird analog verfahren.

Man erhält so nacheinander Näherungswerte für sämtliche Nullstellen von P_n ungefähr dem Betrage nach geordnet, und schließt aus, eine Nullstelle zweimal zu berechnen. Möglicherweise erhält man z.B. die im Betrag zweitkleinste Nullstelle zuerst, aber in den meisten Testbeispielen ergab sich die Anordnung

$$|x_1| \leq |x_2| \leq \ldots \leq |x_n|. \tag{B.2}$$

B.2 Die Muller-Iteration

Abkürzend wird im folgenden $f_k := f(x^k)$ benannt. Für die drei Wertepaare $(x^k, f_k), k = \nu-2, \nu-1, \nu$ wird das entsprechende quadratische Interpolationspolynom ϕ und dessen Nullstellen bestimmt. Als neue Näherung $x^{\nu+1}$ für die gesuchte betragskleinste Nullstelle x_1 von P_n wird eine der Nullstellen gewählt. Man erhält

$$x^{\nu+1} = x^\nu + h_\nu \cdot q_{\nu+1}, \qquad \nu = 2, 3, \ldots \tag{B.3}$$

mit

$$q_{\nu+1} = \frac{-2C_\nu}{B_\nu \pm \sqrt{B_\nu^2 - 4A_\nu C_\nu}} \tag{B.4}$$

wobei folgende Beziehungen gelten

$$\begin{aligned} h_\nu &= x^\nu - x^{\nu-1}, \qquad q_\nu = \frac{h_\nu}{h_{\nu-1}}, \\ A_\nu &= q_\nu f_\nu - q_\nu(1+q_\nu) f_{\nu-1} + q_\nu^2 f_{\nu-2}, \\ B_\nu &= (2q_\nu + 1) f_\nu - (1+q_\nu)^2 f_{\nu-1} + q_\nu^2 f_{\nu-2}, \\ C_\nu &= (1+q_\nu) f_\nu. \end{aligned} \tag{B.5}$$

Als Vorzeichen der Wurzel im Nenner von $q_{\nu+1}$ ist diejenige der Zahlen zu wählen, die den größeren Betrag besitzt, d.h. die Nullstelle von ϕ, welche näher an x^ν liegt. Wenn $f(x^\nu) = f(x^{\nu-1}) = f(x^{\nu-2})$ gilt, so verschwindet der Nenner. Für den Fall schlägt Muller vor, mit $q_{\nu+1} = 1$ weiterzurechnen.

B.3 Automatischer Startprozeß

Wenn hinreichend genaue Anfangsnäherungen für die Nullstelle eines Polynoms vorliegen, kann man mit Iterationsverfahren Folgen von Näherungswerten konstruieren, die gegen die Nullstelle konvergieren. Als Startwerte für die Iteration werden fest vorgegeben

$$x^0 = -1, \qquad x^1 = 1, \qquad x^2 = 0. \tag{B.6}$$

An den Stellen x^0, x^1, x^2 werden als Funktionswerte nicht die des jeweiligen Polynoms f genommen, sondern die Werte

$$\begin{aligned} a_0 - a_1 + a_2 \quad &\text{für} \quad f_0 = f(x^0), \\ a_0 + a_1 + a_2 \quad &\text{für} \quad f_1 = f(x^1), \\ a_0 \quad &\text{für} \quad f_2 = f(x^2). \end{aligned} \tag{B.7}$$

Die Verwendung dieser künstlichen Startwerte wurde von Muller selbst empfohlen. Man kann aber ebenso an den Startstellen $-1, 0, +1$ die wirklichen Polynomwerte benutzen.

B.4 Abbruchbedingung

Die Iteration wird abgebrochen, falls zu vorgegebenem $\epsilon > 0$ die Abfrage

$$\frac{|x^{\nu+1} - x^\nu|}{|x^{\nu+1}|} < \epsilon \tag{B.8}$$

erfüllt ist. Ist dies für ein $\nu = N - 1$ der Fall, so ist $x^N = x_1^N$ der gesuchte Näherungswert für x_1.

B.5 Auftreten konjugiert komplexer Nullstellen.

Falls der Radikant der Wurzel in (B.4) negativ ausfällt, so kann dies zwei Ursachen haben:

1. Eine reelle Lösung der Gleichnung $f(x) \equiv P_n(x) = 0$ wird durch eine Folge konjugiert komplexer Zahlen approximiert. Die Imaginärteile der Folge $\{x^\nu\}$ sowie die Imaginärteile der zugehörigen Polynomwerte streben dann gegen Null.

2. x_1 ist eine komplexe Nullstelle. Mit x_1 ist dann auch $\overline{x_1}$ Nullstelle von P_n. In diesem Fall liefert die Division $\frac{P_n}{(x-x_1^N)(x-\overline{x_1}^N)}$ unter Vernachlässigung des Restes eine Polynom P_{n-2} vom Grad $n-2$, für das wiederum die Muller- Iteration eine Näherung für die i.a. betragskleinste Nullstelle liefert.

B.6 Konvergenz des Verfahrens

Die Kovergenz im Großen konnte nicht nachgewiesen werden. Es konnte aber gezeigt werden, daß Konvergenz eintritt, wenn der Prozess hinreichend nahe an einer einfachen bzw. doppelten Nullstelle beginnt. Jedoch erreichte Muller mit einer Modifikation Konvergenz in allen getesteten Fällen zu dem angegebenen Startprozeß.

B.7 Konvergenzordnung

Bei Iterationsverfahren, deren Operationszahl nicht im vorraus bestimmbar ist, kann die Konvergenzordnung als Maßstab für den erforderlichen Rechenaufwand eines Verfahrens dienen. Die Iterationsfolge $\{x^\nu\}$ konvergiert von mindestens p-ter Ordnung gegen ξ, wenn eine Konstante $0 \leq M < \infty$ existiert, so daß für $p \in IN$ gilt

$$\lim_{\nu \to \infty} \frac{|x^{\nu+1} - \xi|}{|x^\nu - \xi|^q} = M. \tag{B.9}$$

Das Iterationsverfahren $x^{\nu+1} = \varphi(x^\nu)$ heißt dann ein Verfahren von mindestens p-ter Ordnung. Es besitzt genau die Ordnung p, wenn $M \neq 0$ ist. Die Konvergenzgeschwindigkeit wächst mit der Konvergenzordnung.

Bei $p = 1$ spricht man von linearer Konvergenz, bei $p = 2$ von quadratischer Konvergenz und allgemein bei $p > 1$ von superlinearer Konvergenz. Die Konvergenzordnung des Muller-Verfahrens wird für den Fall einfacher Nullstellen mit $p = 1,84$ und für den Fall doppelter Nullstellen mit $p = 1,23$ angegeben.

Anhang C

Quelltext

C.1 Entwicklungsumgebung

Die Programme wurde auf einem AMIGA mit dem MANX-Aztec *C Compiler* geschrieben und auf eine SUN-SPARC Architektur mit dem GNU-C Compiler portiert. Der Quelltext wurde möglichst nahe am *ANSI-Standard* entwickelt. Für die Syntaxkontrolle einer Konfigurationsdatei werden der lexikalische Analysator *lex* und der Parser-Generator *yacc*, bzw. deren nicht kommerzielle Version *flex* und *bison*, benötigt.

C.2 Verzeichnisstruktur

In den folgenden Tabellen sind für jedes Verzeichnis die enthaltenen Dateien aufgelistet und es wird die entspechende Funktionalität beschrieben.

MLib	
Dateiname	Funktionalität
draw.c	Zeichenfunktionen.
file.c	Dateiein- und ausgabeoperationen.
poly.c	Behandlung von Polyedern.
trans.c	3D-Transformationen.
interpol.c	Schattierungsfunktionen.
util.c	Allgemeine Hilfsoperationen.
pixmap.c	Interner Zeichenpuffer.
clip.c	Kappalgorithmen für die Grafikausgabe.
vector.c	Operationen auf Vektoren.
vector.h	Definitionen für Vektorrechnung.
defs.h	Allgemeine Definitionen.
types.h	Typendeklarationen.
const.h	Konstantendefinitionen.
mlib.proto	Prototypen der definierten Funktionen.

Display	
Dateiname	Funktionalität
xlib.c	X-Windows Bildschirmtreiber.
xlib.proto	Prototypen der definierten Funktionen.

Simul	
Dateiname	Funktionalität
main.c	Hauptprogramm zur Simulation einer Aufnahmesituation.
calc.c	Berechnungsfunktionen.
yacc.y	Syntaxbeschreibung der Eingabedatei.
lex.l	Lexikalische Analyse.
makefile	Automatisches Übersetzen und Binden.
simul.proto	Prototypen der definierten Funktionen.

Solve	
Dateiname	Funktionalität
main.c	Hautprogramm der Gestaltsberechnung.
solve.c	Allgemeine Lösungsstrategie.
iterat.c	Iterationsverfahren.
mini.c	Minimierung.
direct.c	Direktes Loesungsverfahren.
makefile	Automatisches Übersetzen und Binden.
solve.proto	Prototypen der definierten Funktionen.

Visual	
Dateiname	Funktionalität
main.c	Hauptprogramm der graphischen Darstellung.
makefile	Automatisches Übersetzen und Binden.
visual.proto	Prototypen der definierten Funktionen.

Verify	
Dateiname	Funktionalität
main.c	Hauptprogramm zur Bewertung der Güte.
makefile	Automatisches Übersetzen und Binden.
verify.proto	Prototypen der definierten Funktionen.

Config	
Dateiname	Funktionalität
sphere.cfg	Konfigurationsdatei einer Kugel.
cube.cfg	Beispiel eines Wuerfels.
poly.cfg	Kleiner Polyeder.
torus.cfg	Grosser Torus.
porsche.cfg	Detaillierter Sportwagen.

C.3 Ausblicke und Erweiterungen

Für die Zukunft wäre es sinnvoll, über eine schrittweise Beseitigung der starken Einschränkungen, wie z.B. der Parallelprojektion oder Lambert-Reflektion, nachzudenken. Als weniger aufwendige Erweiterungen könnte ich mir folgende Themen vorstellen:

- Zufällige Erzeugung von Rauscheinflüssen bei der Datengenerierung.
- Möglichkeit zur flexiblen Glättung einer berechneten Nadelkarte.
- Veränderbare Anzahl der zu berechnenden Grauwerte.
- Kompressionsverfahren für die '.dpm' und '.grd' Dateien.
- Grafische Benutzeroberfläche.

Diese Liste ließe sich noch beliebig erweitern, und ich würde mich auch über neue konstruktive Vorschläge freuen, die dann vielleicht in einer späteren Version Berücksichtigung finden werden. Für die Mitteilung eventueller Fehler bin ich jederzeit dankbar.

C.4 Wichtiger Hinweis

Die Angaben und Programme wurden mit großer Sorgfalt erarbeitet bzw. zusammengestellt. Trotzdem sind Fehler nicht ganz auszuschließen. Ich möchte deshalb darauf hinweisen, daß weder eine Garantie noch irgendeine Haftung für Folgen, die auf fehlerhafte Angaben zurückgehen, übernommen werden kann.

Abbildungsverzeichnis

Literaturverzeichnis

[ALO89] **Aloimonos, J.** und **Shulman, D.**: *Integration of Visual Modules*, Academic Press, (1989), S.29-39.

[BRO87] **Bronstein, I.N.** und **Semendjajew, K.A.**: *Taschenbuch der Mathematik*, Verlag Harri Deutsch, (1987)

[ENG88] **Engeln-Müllges, G.** und **Reuter, F.**: *Formelsammlung zur Numerischen Mathematik mit MODULA 2-Programmen*, BI-Wiss.-Verl., (1988), S.36-40.

[FEL88] **Fellner W.D.**: *Computer Grafik*, BI-Wiss.-Verl., (1988)

[FOL90] **Foley J.D., van Dam A. u.a.**: *Computer Graphics Principles and Practice*, Addison-Wesley, (1990)

[HOR86] **Horn, B.K.P.**: *Robot vision*, MIT Press, (1986)

[KLE92] **Klette, R.**: Skripte zur Vorlesung *Computer Vision III* (1991) und *Computer Vision II* (1992)

[KOE75] **Köhler, J. u.a.**: *Analytische Geometrie und Abbildungsgeometrie in vektorieller Darstellung*, Diesterweg, (1975)

[MUL56] **Muller, D.E.**: *A Method for Solving Algebraic Equations using an Automatic Computer*, Math. Tables Aids Comp. 10, (1956), S.208-215.

[SHI87] **Shirai, Y.**: *Three-Dimensional Computer Vision*, Springer Verlag, (1987).