

Institut für Mathematische Maschinen und Datenverarbeitung  
Friedrich-Alexander-Universität Erlangen-Nürnberg

Lehrstuhl für Informatik IX  
(Graphische Datenverarbeitung)  
Prof. Dr. H.-P. Seidel

**Modellbasierte Veränderungen von B-Spline-  
Oberflächen zur Simulation von Verformungen der  
Gesichtshaut**



**Dirk Bartz**

Studienarbeit 1994/1995



# **Modellbasierte Veränderungen von B-Spline-Oberflächen zur Simulation von Verformungen der Gesichtshaut**

**Studienarbeit im Fach Informatik**

vorgelegt von

*Dirk Bartz*

geb. am 22.07.1967 in Simmern/Hunsrück

Betreuende Hochschullehrer:

*Prof. Dr. Hans-Peter Seidel (IMMD IX)*

*Prof. Dr.-Ing. Bernd Girod (LNT)*

Betreuer: *Erwin Keeve (LNT)*

Beginn der Arbeit: 01.03.1994

Abgabe der Arbeit: 24.03.1995

Ich versichere, daß ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und daß die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Erlangen, den 20.03.1995 \_\_\_\_\_

---

# Inhaltsverzeichnis

1	Einleitung	1
2	Methoden der Flächenmodellierung	3
2.1	Polygone versus Splineflächen – ein Überblick	3
2.1.1	Polygonale Flächen	3
2.1.2	Bivariate Splineflächen	4
2.1.2.1	Tensorprodukt-B-Splineflächen	4
2.1.2.2	Splines über Dreiecken	5
2.2	Mathematische Grundlagen der DMS-Splines	7
2.2.1	Halboffene konvexe Hülle	7
2.2.2	Baryzentrische Koordinaten	8
2.2.3	Berechnungsschema der DMS-Splines	9
2.3	Implementierung der DMS-Splines	11
2.3.1	Bestimmung der Parametrisierung	12
2.3.2	Konstruktion des Kontrollnetzes	14
2.3.3	Bestimmung der Auswertungspunkte	16
2.3.4	Auswertung der DMS-Splinebasisfunktionen	16
3	Medizinische Grundlagen	19
3.1	Anatomie des menschlichen Schädels	19
3.2	Mechanische Eigenschaften des Gesichtsgewebes	20
4	Ein physikalisches Modell	23
4.1	Das dynamische Gewebemodell	23
4.1.1	Konstruktion des Federnetzwerkes	23
4.1.2	Konditionierung des Federnetzwerkes	27
4.2	Numerische Simulation des Gewebemodells	27
5	Ergebnisse	31
5.1	Eine Craniofaciale Korrekturoperation	31
5.2	Ergebnisse der Implementierung	33
5.2.1	Modellgestützte Simulation	33
5.2.2	DMS-Splineoberfläche des Gesichts	37
6	Zusammenfassung und Ausblick	41
<b>Anhang</b>		
A	Anmerkungen zur Komplexität	45

---

A.1	Komplexität der Initialisierung	45
A.2	Komplexität der Iterationsschritte	47
A.3	Zusammenfassung	47
B	Veränderung einer Oberfläche – ein Leitfaden	48
C	Implementierungsübersicht	49
C.1	Übersicht über die entwickelten Programme	49
C.2	Klassenbeschreibung zu MOBIS	50
D	Literaturverzeichnis	53

---

# 1 Einleitung

Nonverbales Kommunikationsverhalten ist im zwischenmenschlichen Bereich von großer Bedeutung [Tram90] und zeigt sich – neben Körperhaltung und -bewegung – vor allem durch die Mimik. Fehlbildungen des Gesichts können die nonverbale Kommunikation behindern, weshalb deren Korrektur durch medizinische Eingriffe eine besondere Bedeutung zukommt.

1977 entwickelten Ekman und Friesen das Facial Action Coding System (FACS) [EF77]. Mit diesem System klassifizierten sie menschliche Emotionen, wie sie sich in der Mimik zeigen, in 46 Aktionseinheiten (action units). Jede dieser Aktionseinheiten wird einem oder mehreren Muskeln zugeordnet. So beschreibt zB. die Aktionseinheit AU10 das Hochziehen der oberen Lippe (upper lip raiser) durch die beteiligten Muskeln<sup>1</sup>.

Anfang der achtziger Jahre fand das FACS Anwendung in der Computergraphik [Park91]: Platt und Badler verwendeten dieses System in [PB81] zur parametrisierten Animation von Gesichtern. Seitdem wurde dieses System in einer Vielzahl von weiteren Arbeiten verwendet ([Wate87], [WT90], [Hoch92]).

In [VMW83] wurde ein System vorgestellt, daß mit Hilfe von Aufnahmen der Computertomographie (CT) und Methoden des Computer Aided Geometric Design (CAGD) die Planung von craniofacialen Korrekturoperationen vereinfachte. Pieper stellte in [Piep92] ein Teilmodell vor, das mit Hilfe von CT-Aufnahmen, Abtastungen des Körpers mit einem 3D-Laserscannern und Finite Elemente Methoden (FEM) die Planung und Visualisierung von Eingriffen der plastischen Chirurgie unterstützt.

Die vorliegende Studienarbeit ist im Rahmen eines DFG-Projektes zur interaktiven Operationsplanung [GKG95] entstanden. Basierend auf einer Methode zur Modellierung des Gesichts aus [WT90]/[Wate92] wurde ein Gewebemodell implementiert, mit dem Veränderungen der Gesichtsoberfläche bei craniofacialen Korrekturoperationen simuliert werden sollen.

Mit dem in [WT90] vorgestellten und in [Wate92] erweiterten Modell werden Veränderungen der Gesichtsoberfläche aufgrund von virtuellen Muskelaktoren nach dem FACS simuliert. Im Gegensatz dazu wird in diesem Ansatz auf ein Muskelmodell verzichtet und ein möglicher statischer Zustand des Gesichtsgewebes nur durch ein Fünfschichtenmodell des Gesichtsgewebes simuliert. Veränderungen der Gesichtsoberfläche durch eine Korrekturoperation werden hier durch Veränderungen der untersten Schicht, die den Gesichtsschädel repräsentieren soll, simuliert.

In den einigen Modellen werden polygonale Darstellungen zur Visualisierung gewählt ([PB81], [WT90]). In [Hoch92] wird im Gegensatz dazu eine splinebasierte Darstellung verwendet. Allerdings ist die dort verwendete Splinemethode auf ein *rectilineares Gitter* von Datenpunkten angewiesen.

---

1. Levator Labii Superioris und Caput Infraorbitalis

---

In dieser Arbeit wird eine splinebasierte Darstellung, die auf der Basis einer Triangulierung in Form eines unstrukturierten Gitters die Gesichtsoberfläche bestimmt [DMS92], verwendet. Mit dieser Methode ist eine Qualitätssteigerung der Darstellung zu erreichen, die jedoch mit einem erhöhten Rechen- und Zeitaufwand einher geht.

In dieser Arbeit wird – nach einem kleinen Überblick über Methoden der Flächenmodellierung – dieses Verfahren vorgestellt. Der dritte Teil beschreibt die verwendeten medizinischen Grundlagen zur Anatomie und Physiologie des menschlichen Körpers. In Kapitel vier werden das verwendete Gewebemodell und dessen numerische Simulation, wie es in [WT90] vorgestellt wurde, beschrieben. Kapitel fünf stellt die Ergebnisse vor, die mit der Implementierung dieser splinebasierten Modellierungsmethode erzielt wurden. Den Abschluß bilden einige Überlegungen zur Verbesserung des vorgestellten Modells und dessen effizienterer Realisierung.

---

## 2 Methoden der Flächenmodellierung

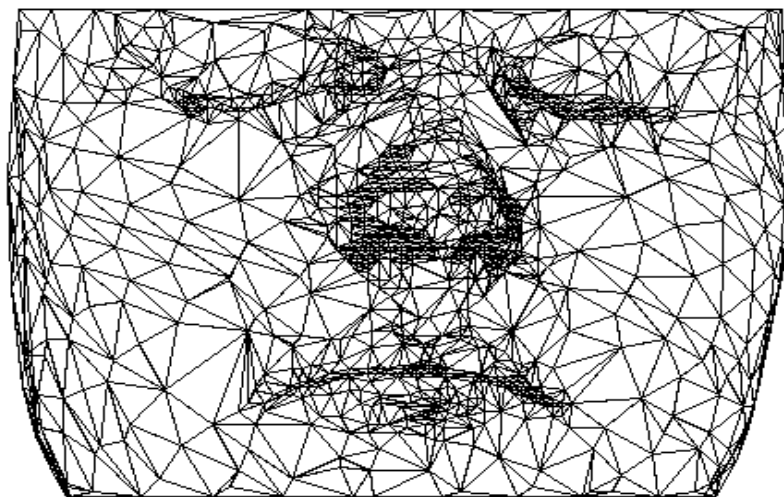
Im ersten Teil dieses Kapitels werden einige Methoden zur Modellierung von Flächen vorgestellt. Die beiden folgenden Teile beschreiben die mathematischen Grundlagen der *DMS-Splines* und ihre Implementierung.

### 2.1 Polygone versus Splineflächen – ein Überblick

Zur Beschreibung von Flächen im dreidimensionalen Raum bieten sich grundsätzlich zwei verschiedene Methoden an: eine diskrete und eine kontinuierliche Darstellung. Ein typischer Vertreter für die diskrete Methode ist die polygonale Beschreibung. Für die kontinuierliche Methode werden im folgenden splinebasierte Darstellungen betrachtet.

#### 2.1.1 Polygonale Flächen

Mit einem 3D-Scanner ist es möglich, ein Gesicht innerhalb kurzer Zeit zu vermessen. Aus diesen Daten kann eine polygonale Darstellung in Form einer *Triangulierung* gewonnen werden. Mit Hilfe dieser Triangulierung, die Position und Nachbarschaftsbeziehungen der einzelnen Dreiecke beschreibt, können die Daten eines Gesichts mit konventionellen Methoden der Computergraphik visualisiert werden. Die an den Dreiecksgrenzen entstandenen Kanten können mit Hilfe eines Schattierungsverfahren wie dem Gouraud- oder Phong-Shading *optisch geglättet*<sup>1</sup> werden.



**Abb. 2.1: Triangulierung eines Gesichts**

---

1. Optisch glatt: Für die Betrachterin erscheint es glatt, mathematisch ist es aber nicht zwangsläufig glatt.

Entsprechend der Anzahl der Abtastpunkte des Scanners – und damit der Anzahl der Dreiecke in der Oberflächentriangulierung – steigt die Qualität der polygonalen Darstellung. Mit einem Cyberware 3D-Laserscanner wird ein Gesicht auf einem äquidistanten strukturiertem Gitter in Zylinderkoordinaten diskret abgetastet. Bereiche mit einer detaillierteren Oberflächenstruktur, wie z.B. die Mundpartie, werden mit der gleichen Genauigkeit vermessen, wie Bereiche mit einer detailläreren Struktur, z.B. die Wangenregion.

Mit Hilfe von Komprimierungsverfahren [SZL92] kann die Triangulierung entsprechend einer vorgegebenen Genauigkeit angepaßt werden, d.h. große, detailarme Flächen können mit weniger Dreiecken dargestellt werden.

## 2.1.2 Bivariate Splineflächen

Gegenüber einer polygonalen Beschreibung mit einer rein optischen Glattheit besitzen die betrachteten bivariaten Splineflächen eine “natürliche” Glattheit, d.h. sie sind an jeder Stelle  $(n-1)$ -mal stetig differenzierbar, wobei  $n$  der Grad der Splinefläche ist. So sind also quadratische Splineflächen einmal stetig differenzierbar.

Im folgenden werden kurz zwei verschiedene Arten von bivariaten Splineflächen vorgestellt: Tensorprodukt-B-Splineflächen und Splineflächen über Dreiecken.

### 2.1.2.1 Tensorprodukt-B-Splineflächen

Die *Tensorprodukt-B-Splineflächen* sind eine Erweiterung der univariaten B-Splinekurven. Sie sind die wohl am weitesten verbreitete Splineflächenmethode und stellen, mit ihrer Variante der Tensorproduct-Non-Uniform-Rational-B-Splines (TP-NURBS), in herkömmlichen Programmen für den Computer-Aided-Geometric-Design (CAGD) den Stand der Technik dar.

Definiert sind die Tensorprodukt-B-Splineflächen als Tensorprodukt zweier B-Splinekurven. Der Einfachheit halber beschränkt sich diese Arbeit auf *uniforme Tensorprodukt-B-Splines*, deren allgemeine Form in Gleichung 2.1 angegeben ist.

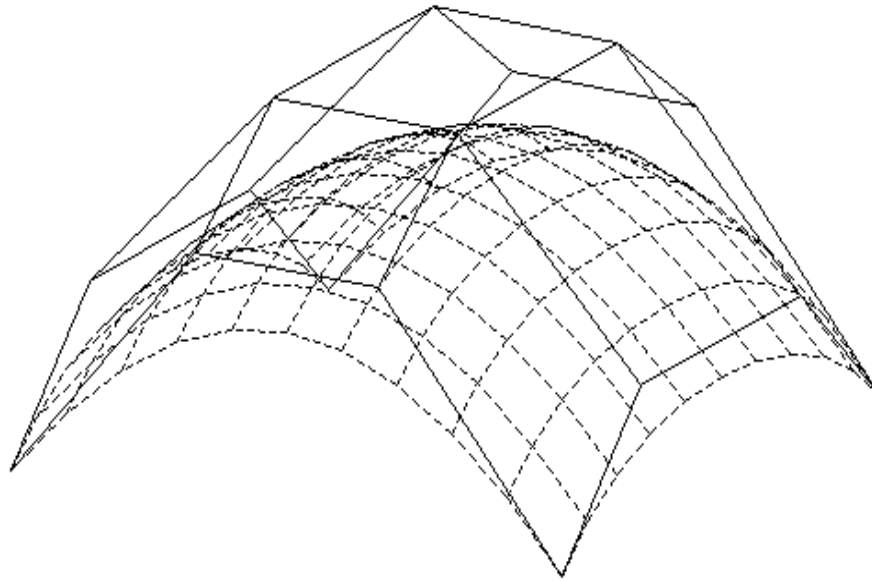
$$F(u, v) = \sum_{i=0}^k \sum_{j=0}^m N_i^n(u) N_j^n(v) c_{ij} \quad (2.1)$$

$$\text{mit } u, v \in \mathbb{R}^2 \wedge c_{ij} \in \mathbb{R}^3$$

Der Funktionswert der Fläche bestimmt sich aus der Doppelsumme über die B-Splinebasisfunktionen  $N_i(u)$  und  $N_j(v)$  vom Grad  $n$  und den deBoor-Punkten (Kontrollpunkten)  $c_{ij}$  aus dem Kontrollnetz der Tensorproduktfläche  $F$ .

Tensorproduktflächen sind über rechteckigen Parameterflächen definiert. Diese Topologie spiegelt sich in ihrer Gestalt wieder; die Fläche wird auch von einer mehr oder weniger deutlichen rechteckigen Form sein.

Ein weiteres Problem der Tensorproduktflächen besteht darin, daß Verfeinerungen lokal nicht möglich sind. Um die rechteckige Topologie zu wahren, muß für jeden einzufügenden Knoten der Parametrisierung eine komplette Knotenlinie ergänzt werden [Seid94]. Zwar läßt sich mit einer hierarchischen Methode eine lokale Verfeinerung erreichen [FB88], jedoch ändert dies nichts an der rechteckigen Topologie der Tensorproduktflächen.



**Abb. 2.2: Bikubische Tensorprodukt-B-Splinefläche mit Kontrollnetz**

Im Allgemeinen, insbesondere bei medizinischen Anwendungen, kann nicht von dieser Topologie ausgegangen werden. So liefert z.B. der Marching-Cubes-Algorithmus [LC87], angewandt auf die rekonstruierten CT-Daten eines Schädels, eine Triangulierung der Oberfläche. Da aus verschiedenen Gründen eine Anpassung des rechteckigen Parametergebietes unbefriedigend bleibt [HL89] – Verlust von Glattheit, ungünstige Degenerationen der Flächen –, wird eine Flächenbeschreibung benötigt, die auf dreieckigen Parametergebieten basiert.

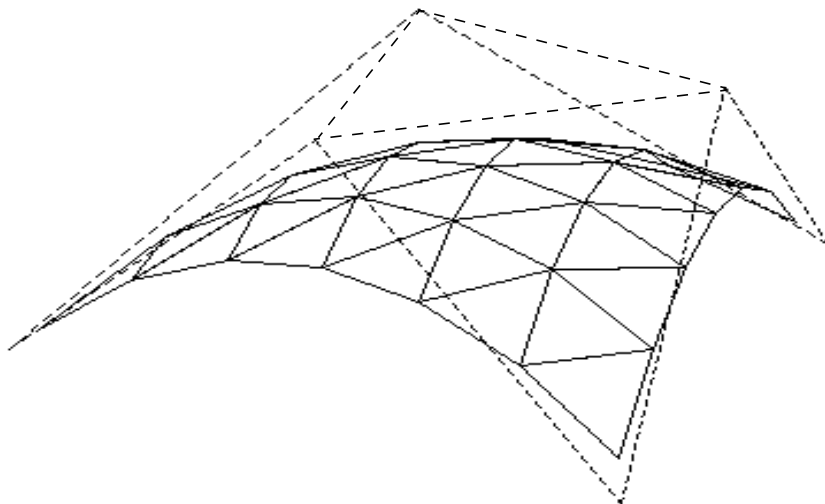
### 2.1.2.2 Splines über Dreiecken

Eine der verschiedenen Flächenmethoden über Dreiecken sind die *Simplex Splines*. Im bivariaten Fall bestimmt sich ein Simplex Spline  $M(u|V)$  vom Grad  $n$  über einer Knotenmenge  $V = \{t_0, t_1, \dots, t_{n+2}\}$  aus der Parameterebene  $\mathbb{R}^2$ , einem beliebigen Punkt  $u \in \mathbb{R}^2$  und drei Simplex Splines vom Grad  $n-1$ , die über einer  $n-1$  elementigen Teilmenge von  $V$  definiert sind [Fong92][PS94a]. Der Simplex Spline vom Grad 0 bestimmt sich schließlich über der *halboffenen konvexen Hülle*<sup>1</sup> über drei Knoten  $t_{i_0}, t_{i_1}, t_{i_2}$  aus  $V$ , die ein Dreieck bilden. Simplex Splines vom Grad  $n$  verfügen über einige günstige Eigenschaften [PS94a]:

- Simplex Spline sind *stückweise polynomial* vom Grad  $n$ .
- *Lokalität*: Außerhalb der halboffenen konvexen Hülle von  $\{t_0, \dots, t_{n+2}\}$  liefert der Simplex Spline  $M(u|t_0, \dots, t_{n+2})$  keinen Beitrag.
- *Positivität*:  $M(u|t_0, \dots, t_{n+2}) \geq 0$ .
- *Optimale Glattheit*: Für alle  $t_i$  aus  $\{t_0, t_1, \dots, t_{n+2}\}$  in allgemeiner Lage<sup>1</sup> ergibt sich für die Fläche eine  $C^{n-1}$ -Stetigkeit.

Leider gibt es i.d.R. deutlich mehr als nur drei Simplex Splines vom Grad  $n-1$ , da es mehr als drei mögliche Tripel von Knoten  $t_i$  aus  $V$  gibt. Jedes dieser Tripel führt zu einem gültigen Ergebnis; welches jedoch zu einer effizienten Bestimmung führt, ist im Kontext der Simplex Splines unklar.

In [DMS92] wird eine Methode vorgestellt, die Simplex Splines und B-Patches [Seid91] zu den sogenannten *Dreieckssplines*, oder auch *DMS-Splines*, verbindet [Fong92]. Diese Methode definiert ein Schema, nach dem die gesuchten Simplex Splines vom Grad  $n-1$  bestimmt werden können.



**Abb. 2.3: Quadratische DMS-Splinefläche mit Kontrollnetz**

Die DMS-Splines haben zusätzlich zu denen der Simplex Splines folgende Eigenschaften:

- *Affine Invarianz*. Daraus ergibt sich die Eigenschaft der Partition der Eins, d.h. die Summe aller (normierten) DMS-Splines an einem Punkt ergibt eins.
- *Konvexe Hülle*: Die DMS-Splinefläche liegt in der konvexen Hülle ihres Kontrollnetzes.
- *Lokale Kontrolle*: Veränderungen der Position eines Kontrollpunktes verändern die

1. Siehe Definition halboffene konvexe Hülle in Kapitel 2.2.1

1. Allgemeine Lage bedeutet hier, daß alle Punkte affin-unabhängig angeordnet sind.

Fläche nur in dessen lokalen Einflußbereich.

Angeichts der Tatsache, daß die Parametrisierung auf einem unstrukturierten Gitter in Form einer Triangulierung beruht, stellen die DMS-Splines ein gut geeignetes Werkzeug zur Visualisierung von Gesichtsstrukturen dar. Methoden und Probleme der DMS-Splines werden in Abschnitt 2.2 und 2.3 näher erläutert.

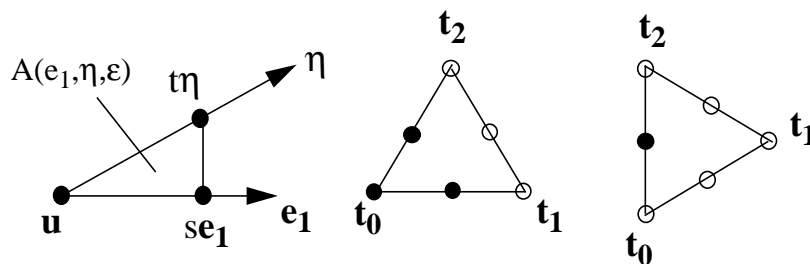
## 2.2 Mathematische Grundlagen der DMS-Splines

In den ersten beiden Teilen dieses Abschnitts werden die Begriffe der *halboffenen konvexen Hülle* und der *baryzentrischen Koordinaten* definiert. Der dritte Teil behandelt das Definitionsschema der Simplex Splines, und damit der DMS-Splines.

### 2.2.1 Halboffene konvexe Hülle

Jeder Punkt  $u$  der Parameterebene gehört genau zu einem Dreieck der Triangulierung über  $V = \{t_0, t_1, \dots, t_{n+2}\}$ ,  $V \subset \mathbb{R}^2$ . Deshalb müssen Punkte, die auf der gemeinsamen Kante der Dreiecke  $\Delta_i(t_{i0}, t_{i1}, t_{i2})$  und  $\Delta_j(t_{j0}, t_{j1}, t_{j2})$ , mit  $i \neq j$ , liegen, genau einem dieser Dreiecke zugeordnet werden, da dieser Punkt  $u$  sonst einen zu großen Beitrag zur Fläche liefert. Um dies zu gewährleisten, wurden die Simplex Splines über der halboffenen konvexen Hülle  $[t_0, t_1, t_2)$  eines Dreiecks definiert [Seid91] [Fong92].

Abbildung 2.4 illustriert die Zugehörigkeit eines Punktes  $u$  zur halboffenen konvexen Hülle. Genau dann, wenn die Fläche  $A(e_1, \eta, \epsilon)$  komplett im Inneren<sup>1</sup>  $\Delta^0$  des Dreiecks  $\Delta(t_0, t_1, t_2)$  liegt, gehört der Punkt  $u$  zur halboffenen konvexen Hülle von  $\{t_0, t_1, t_2\}$ , wobei  $e_1$  der Einheitsvektor der horizontalen Achse,  $\eta$  ein beliebiger Vektor mit positiver Steigung,  $s$  und  $t$  zwei positive reelle Zahlen, begrenzt durch  $\epsilon$ , sind.



**Abb. 2.4:** Die schwarzen Punkte gehören zur halboffenen konvexen Hülle, die weißen gehören nicht zu ihr [Fong92].

Formal beschreibt sich diese Beziehung durch [Seid91]:

1. Interior =  $\text{int}[t_0, t_1, t_2]$

$$A_{(e_1, \eta, \varepsilon)}(u) = \{s \cdot e_1 + t \cdot \eta \mid 0 < s, t < s + t < \varepsilon; s, t, \varepsilon \in \mathbb{R}\} \quad (2.2)$$

$$u \in [t_0, t_1, t_2) \Leftrightarrow A_{(e_1, \eta, \varepsilon)}(u) \subseteq \Delta^0$$

mit

$$\Delta^0 = \text{int}[t_0, t_1, t_2] \subset [t_0, t_1, t_2) \subset \Delta = [t_0, t_1, t_2] \quad (2.3)$$

### 2.2.2 Baryzentrische Koordinaten

Sei  $T = \{t_0, t_1, t_2\} \subset \mathbb{R}^2$  ein nichtdegeneriertes Dreieck und  $u \in \mathbb{R}^2$ , dann werden die Koordinaten von  $u$  bezüglich des Dreiecks  $T$  baryzentrische Koordinaten genannt [Fong92].

Es werden folgende Determinanten definiert:

$$d(T) = \begin{vmatrix} 1 & 1 & 1 \\ t_0 & t_1 & t_2 \end{vmatrix}, \quad d_0(u|T) = \begin{vmatrix} 1 & 1 & 1 \\ u & t_1 & t_2 \end{vmatrix},$$

$$d_1(u|T) = \begin{vmatrix} 1 & 1 & 1 \\ t_0 & u & t_2 \end{vmatrix}, \quad d_2(u|T) = \begin{vmatrix} 1 & 1 & 1 \\ t_0 & t_1 & u \end{vmatrix}.$$

Die baryzentrischen Koordinaten  $\lambda_j$  von  $u$  bezüglich  $T$  ergeben sich dann zu

$$\lambda_j(u|T) = \frac{d_j(u|T)}{d(u|T)}, \quad (2.4)$$

wobei folgende Eigenschaften gelten:

$$\sum_{j=0}^2 \lambda_j(u|T) = 1 \quad \text{und} \quad u = \sum_{j=0}^2 \lambda_j(u|T) \cdot t_j. \quad (2.5)$$

Abbildung 2.5 zeigt das baryzentrische Koordinatensystem bezüglich des Dreiecks  $T$ . Die mit  $\lambda_j(u|T)$  bezeichneten Linien geben die Positionen wieder, auf denen ein Punkt  $w$  die gleiche baryzentrische Koordinate  $\lambda_j$  besitzt wie der Punkt  $u$ .

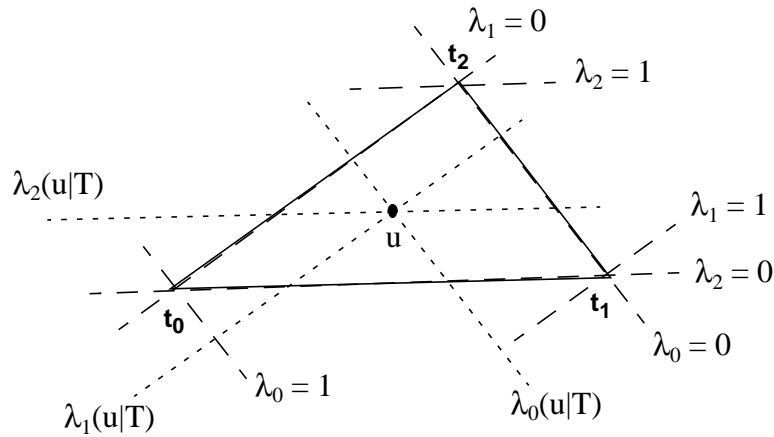


Abb. 2.5: Baryzentrisches System bezüglich eines Dreiecks T

### 2.2.3 Berechnungsschema der DMS-Splines

Gegeben sei eine Triangulierung  $T = \{\Delta_i = [t_{i0}, t_{i1}, t_{i2}] \mid (i0, i1, i2 \in \mathbb{N}_0 \wedge t_i \in \mathbb{R}^2)\}$  mit  $T \subset \mathbb{R}^2$  und folgenden Eigenschaften [PS94a]:

- Alle Dreiecke  $\Delta_i \in T$  sind *positiv orientiert*, dh.

$$\begin{vmatrix} 1 & 1 & 1 \\ t_{i0} & t_{i1} & t_{i2} \end{vmatrix} > 0, \quad t_{i0}, t_{i1}, t_{i2} \in \mathbb{R}^2 .$$

- Zwei Dreiecke  $\Delta_i, \Delta_j \in T$  schneiden sich
  - entweder in genau einem Knoten  $t_i$ ,
  - oder einer Kante  $(t_i, t_j)$ , mit  $i \neq j$ ,
  - oder überhaupt nicht.

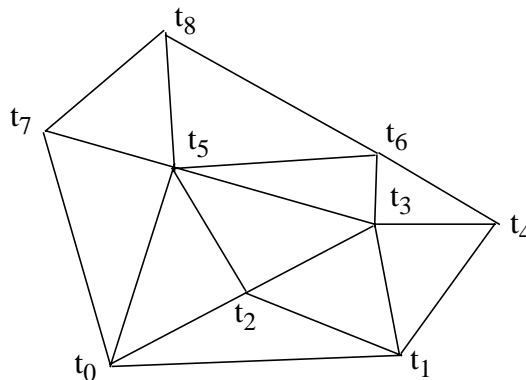


Abb. 2.6: Triangulierung  $T \subset \mathbb{R}^2$

Unter diesen Voraussetzungen bestimmt sich der Simplex Spline M an der Stelle  $u \in \mathbb{R}^2$  rekursiv aus folgender Beziehung:

$$M(u|t_0, t_1, t_2) = \frac{\chi_{[t_0, t_1, t_2]}(u)}{2 \cdot |\Delta(t_0, t_1, t_2)|} = \frac{\chi_{[t_0, t_1, t_2]}(u)}{d(t_0, t_1, t_2)}, \tag{2.6}$$

$$M(u|t_0, \dots, t_{n+2}) = \sum_{j=0}^2 \lambda_j(u|t_{w0}, t_{w1}, t_{w2})M(u|T \setminus \{t_{wj}\}),$$

wobei  $\chi_{[t_0, t_1, t_2]}(u)$  die *charakteristische Funktion* auf der halboffenen konvexen Hülle  $[t_0, t_1, t_2)$ ,  $|\Delta(t_0, t_1, t_2)|$  der Betrag der Fläche des Dreiecks  $\Delta = [t_0, t_1, t_2]$  und  $\lambda_j(u|\Delta_w)$  die  $j$ -te baryzentrische Koordinate des Punktes  $u$  bezüglich des Dreiecks  $\Delta_w = [t_{w0}, t_{w1}, t_{w2}]$  ist.

Um aus der Menge der möglichen Simplex Splines geeignete auszuwählen – und damit die DMS-Splines zu erzeugen – werden nun zu jedem Knoten  $t_i$  weitere Knoten  $t_{i,1} \dots t_{i,n}$  konstruiert, die dann – zusammen mit  $t_{i,0} = t_i$  – die *Knotenwolke von  $t_i$*  bilden. Bei quadratischen DMS-Splines werden also zwei zusätzliche Punkte konstruiert, so daß die Knotenwolke von  $t_i$  aus den Werte  $t_{i,0}, t_{i,1}, t_{i,2}$  besteht. Die Konstruktion der zusätzlichen Knoten muß so erfolgen, daß jede Kombination von Punkten aus den Knotenwolken der Eckpunkte eines Dreiecks  $\Delta_i$  affin unabhängig bleibt [Fong92][Seid91].

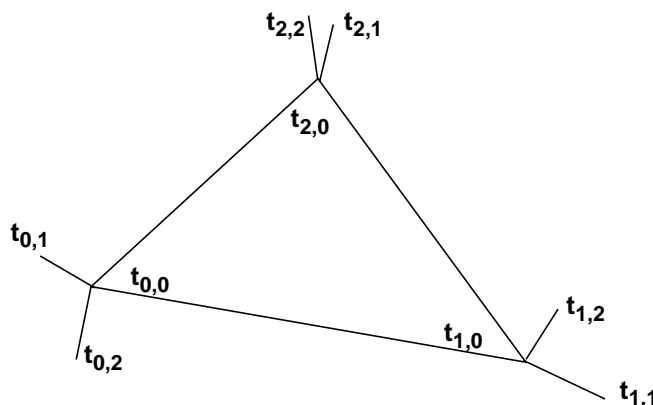


Abb. 2.7: Knotenwolken für einen quadratischen DMS-Spline

Aus den Knotenwolken eines Dreiecks wird für jeden Simplex Spline  $M(u|V_{ijk}^{\Delta_i})$  über dem Dreieck  $\Delta_i$  mit  $i + j + k = n$  (im quadratischen Fall  $n = 2$ ), die Menge:

$$V_{ijk}^{\Delta_i} = \{t_{0,0}, \dots, t_{0,i}, t_{1,0}, \dots, t_{1,j}, t_{2,0}, \dots, t_{2,k}\} \tag{2.7}$$

gebildet. Des weiteren wird

$$\Omega_{ijk} = \bigcap_{(i', j', k') \leq (i, j, k)} \Delta_{i'j'k'}^1 \tag{2.8}$$

und

---

1. Mit:  $(i', j', k') \leq (i, j, k) \Leftrightarrow i' \leq i \wedge j' \leq j \wedge k' \leq k$

$$\Omega_n = \text{int} \left( \bigcap_{i+j+k=n} \Omega_{ijk} \right) \quad (2.9)$$

definiert.

Zur Sicherstellung der gleichen Orientierung aller Dreiecke aus Kombinationen von Eckknoten aus  $V_{ijk}^{\Delta_i}$  darf  $\Omega_n$  nicht leer werden. Sinnvoll ist es außerdem,  $\Omega_n$  möglichst groß zu wählen, da die Eigenschaft der Partition der Eins nur innerhalb von  $\Omega_n$  gilt. Im Inneren einer Parametertriangulierung ist dies nicht von Bedeutung, da sich die verschiedenen  $\Omega_n$  immer zum gesamten Bereich ergänzen. Am Rand kann dies jedoch zu Problemen führen. Eine nähere Diskussion dieses Problems wird in Abschnitt 2.3.1 vorgenommen.

Der *normierte DMS-Spline*  $N_{ijk}^{\Delta_i}(u)$  bestimmt sich aus dem DMS-Spline  $M(u|V_{ijk}^{\Delta_i})$ , gewichtet mit der doppelten Fläche des Dreiecks  $\Delta(t_{0,i}, t_{1,j}, t_{2,k})$ , also mit der Determinante der drei zueinander positiv orientierten Eckpunkte  $d(t_{0,i}, t_{1,j}, t_{2,k})^1$ . Damit wird der normalisierte DMS-Spline  $ijk$  über dem Dreieck  $\Delta_i$  aus

$$N_{ijk}^{\Delta_i}(u) = d(t_{0,i}, t_{1,j}, t_{2,k}) M(u|V_{ijk}^{\Delta_i}) \quad (2.10)$$

erhalten. Die Oberfläche  $F$  vom Grad  $n$  über einem *Knotennetz*  $K$  aller Knotenwolken der Triangulierung  $T$  an der Stelle  $u$  ergibt sich aus

$$F(u) = \sum_{\Delta_i \in T} \sum_{i+j+k=n} N_{ijk}^{\Delta_i}(u) \cdot c_{ijk}^{\Delta_i}, \quad (2.11)$$

wobei  $c_{ijk}^{\Delta_i} \in \mathbb{R}^3$  ein Kontrollpunkt aus dem Kontrollnetz der Fläche  $F$  ist (siehe Abb. 2.10) [PS94a].

## 2.3 Implementierung der DMS-Splines

Im vorhergehenden Abschnitt wurde das Berechnungsschema der DMS-Splines dargelegt. Aus diesem Schema ergeben sich vier Kernfragen, die in einer Implementierung gelöst werden müssen:

- Bestimmung der Parametrisierung durch die Konstruktion geeigneter Knotenwolken für alle Knoten  $t_i \in T$ .
- Konstruktion eines geeigneten Kontrollnetzes.
- Bestimmung ausgewählter Auswertungspunkte  $u$  innerhalb der triangulierten Fläche im  $\mathbb{R}^2$ .

---

1.  $d(t_{0,i}, t_{1,j}, t_{2,k}) = \begin{vmatrix} 1 & 1 & 1 \\ t_{0i} & t_{1j} & t_{2k} \end{vmatrix}$

- Auswertung der DMS-Splinebasisfunktionen an den Auswertungspunkten  $u$ .

### 2.3.1 Bestimmung der Parametrisierung

Um eine geeignete Parametrisierung zu bestimmen muß zunächst im  $\mathbb{R}^2$  eine Triangulierung konstruiert werden, die die Oberflächentopologie wiedergibt. Dazu wird eine Projektion  $\varphi$  definiert, die die Oberflächentriangulierung der Eingangsdaten des verwendeten Cyberware Scanners aus dem  $\mathbb{R}^3$  in den  $\mathbb{R}^2$  projiziert.

Bei einem Abtastvorgang fährt der Scanner auf einer Kreisbahn um das zu vermessene Objekt und mißt dabei Höhe und Entfernung vom Scanner. Nach diesem Abtastschritt liegen die Daten in einem Zylinderkoordinatensystem vor. Die Scannersoftware konvertiert diese Daten anschließend in das kartesische Koordinatensystem. Im letzten Bearbeitungsschritt werden die Oberflächenregionen des Kopfes, die nicht zum Gesicht gehören, geclippt, so daß nur das Gesicht übrig bleibt.

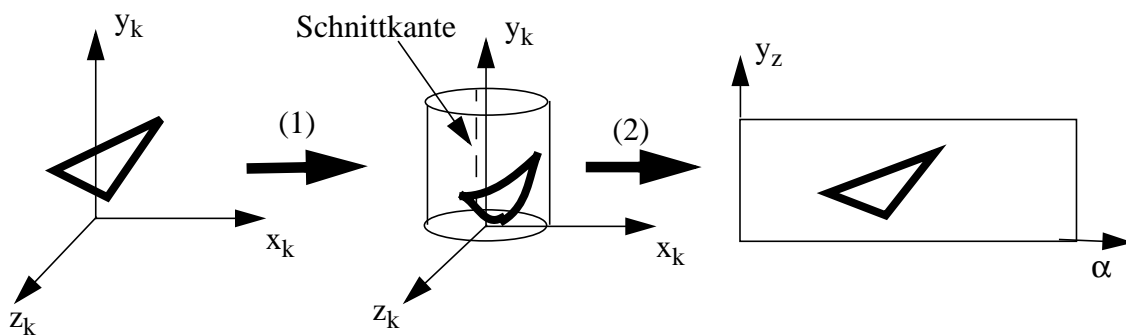


Abb. 2.8: Die Projektion  $\varphi$

Für die Projektion  $\varphi$  wird die Oberflächentriangulierung wieder in Zylinderkoordinaten transformiert und die Punkte der Triangulierung auf den Zylinder projiziert (Abb. 2.8, (1)), d.h. die Entfernung zur Vertikalachse wird ausgeblendet. Anschließend erfolgt eine horizontale Entfaltung des Zylinders entlang einer vertikalen Schnittkante (Abb. 2.8, (2)). Beim letzten Schritt muß darauf geachtet werden, daß bei der Entfaltung diese Schnittkante nicht durch das Gesicht verläuft, da sonst die Topologie an dieser Stelle zerstört würde.

Formal ergibt sich  $\varphi$  aus:

$$\begin{aligned} \varphi : \mathbb{R}^3 &\longrightarrow \mathbb{R}^2 \\ \varphi : (x_k, y_k, z_k) &\longrightarrow (\alpha, y_z) \\ y_z &:= y_k \\ \alpha &:= \operatorname{atan}\left(\frac{x_k}{z_k}\right) \end{aligned} \tag{2.12}$$

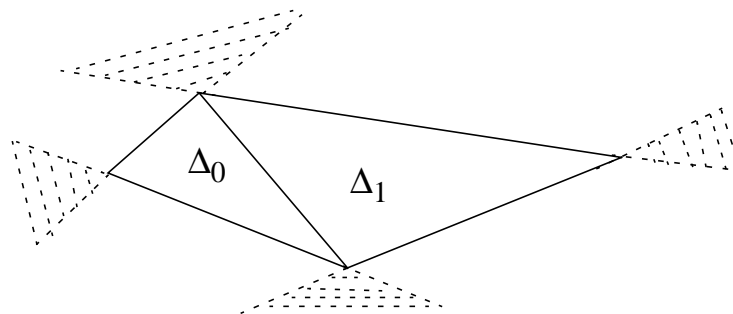
Nachdem die Oberflächentriangulierung in die Ebene projiziert wurde, müssen die zusätzlichen Knoten zu den einzelnen Eckknoten der Dreiecke konstruiert werden. Bei dieser Konstruktion müssen zwei Fälle von einander unterschieden werden:

- Knoten im Inneren der Triangulierung
- Knoten auf dem Rand der Triangulierung

Die zusätzlichen Knoten im Inneren müssen stets affin unabhängig zu den Knoten der Nachbarecken der Parametertriangulierung liegen, d.h. die Beträge der Determinanten von allen  $(n-1) \cdot 3^3$  möglichen Kombinationen<sup>1</sup> müssen größer als eine Genauigkeitschranke  $\varepsilon$  sein.

Zu diesem Zweck werden sämtliche Möglichkeiten<sup>2</sup> in der Umgebung des Eckknotens, zu dem neue Knoten assoziiert werden sollen, bestimmt und die qualitativ beste<sup>3</sup> ausgewählt.

Für die Knoten auf dem Rand müssen zusätzliche Vorkehrungen getroffen werden, da hier ein Problem behandelt werden muß, das bereits in Abschnitt 2.2.3 angedeutet wurde. Um die Eigenschaft der Partition der Eins über den kompletten Parameterdreiecken am Rand zu gewährleisten, müssen die assoziierten Knoten so gewählt werden, daß  $\Omega_n$  des entsprechenden Parameterdreiecks  $\Delta_i \in T$  maximal ist. Abbildung 2.9 zeigt die Bereiche (schraffiert) einer Triangulierung aus zwei Dreiecken, in denen die assoziierten Knoten eingefügt werden können, damit  $\Omega_n$  maximal wird, dh. sich über die gesamte Dreiecksfläche erstreckt.



**Abb. 2.9:** Bereiche zur Wahl möglicher assoziierter Knoten für maximales  $\Omega_n$

1.  $n-1$  ist die Anzahl der Nachbarecken; der Exponent drei ergibt sich aus drei Ecken pro Dreieck; die Basis drei ergibt sich aus dem Grad der DMS-Splines. Quadratisch: ein Eckknoten plus zwei assoziierte Knoten.
2. Wir rotieren mit einem konstanten Winkel  $\Theta$  um den Eckknoten  $t_{i,0}$ ; als Länge nehmen wir  $1/5$  der Entfernung zum nächsten Knoten. Mit dieser Länge wird sichergestellt, daß sich assoziierte Knoten nie überschneiden können. Um zu garantieren, daß diese Knoten immer innerhalb der Dreiecke liegen, die in  $t_{i,0}$  einen Eckpunkt haben, muß außerdem darauf geachtet werden, daß die gegenüberliegenden Dreiecksseiten nicht geschnitten werden.
3. Es wird der Knoten mit der größten normierten Determinante gewählt.

Hieraus ergibt sich folgender Algorithmus:

```

forall  $t_{i,0} \in T$  do
  if  $t_{i,0}$  on_rim_of_T then
    estimate maximal_region m
    estimate neighborhood of  $t_{i,0}$ 
    select knot within m with best quality
      concerning neighborhood
  else
    estimate neighborhood of  $t_{i,0}$ 
    estimate knot with best quality
      concerning neighborhood
  fi
od

```

### Algorithmus für die Parametrisierung

#### 2.3.2 Konstruktion des Kontrollnetzes

Bei der Konstruktion des Kontrollnetzes handelt es sich um ein Approximationsproblem, wobei die Kontrollpunkte die veränderlichen Parameter der Approximation der DMS-Splinesfläche an die Oberflächentriangulierung darstellen. Zur Lösung dieses Problems bieten sich verschiedene Verfahren mit unterschiedlicher Qualität an:

1. Einfache Approximation durch Punktidentifikation: Bei diesem Verfahren werden die Kontrollpunkte  $C_{200}^{\Delta_i}$ ,  $C_{020}^{\Delta_i}$  und  $C_{002}^{\Delta_i}$  der quadratischen DMS-Splineoberfläche – zum Parameterdreieck  $\Delta_i$  – mit den Eckpunkten des zugehörigen Dreiecks  $i$  der Oberflächentriangulierung gleichgesetzt. Die Kontrollpunkte  $C_{110}^{\Delta_i}$ ,  $C_{011}^{\Delta_i}$  und  $C_{101}^{\Delta_i}$  werden durch lineare Interpolation der benachbarten Kontrollpunkte mit der Summe der Multiindizes von zwei bestimmt. Diese Approximationsmethode liefert mäßig gute Ergebnisse, ist dafür aber sehr schnell, einfach zu implementieren und numerisch sehr stabil.

Folgende Formeln beschreiben die Bestimmung der Kontrollpunkte  $C_{ijk}^{\Delta_i}$  aus den Eckpunkten  $p_j^{\Delta_i}$  des Dreiecks  $\Delta_i$  der Oberflächentriangulierung:

$$\begin{aligned}
 C_{200}^{\Delta_i} &= p_0^{\Delta_i}, & C_{020}^{\Delta_i} &= p_1^{\Delta_i}, & C_{002}^{\Delta_i} &= p_2^{\Delta_i} \\
 C_{110}^{\Delta_i} &= \frac{1}{2}(C_{200}^{\Delta_i} + C_{020}^{\Delta_i}) \\
 C_{011}^{\Delta_i} &= \frac{1}{2}(C_{020}^{\Delta_i} + C_{002}^{\Delta_i}) \\
 C_{101}^{\Delta_i} &= \frac{1}{2}(C_{200}^{\Delta_i} + C_{002}^{\Delta_i})
 \end{aligned} \tag{2.13}$$

2. Least-Square-Approximation: Die Kontrollpunkte bestimmen sich so, daß das Quadrat der Abstände zwischen Approximation und zu approximierender Oberfläche minimal ist. Dieses Problem läßt sich wie folgt formulieren:

$$\min \left\{ \sum_{i \in I} \left| F(u_i) - p_{ijk}^{\Delta_i} \right| \right\}, \quad (2.14)$$

wobei  $I$  die Indexmenge der Auswertungspunkte  $u_i$  bezüglich der Kontrollpunkte  $C_{ijk}^{\Delta_i}$  und  $p_{ijk}^{\Delta_i}$  der entsprechende Punkt in der Oberflächentriangulierung ist. Da ein Punkt einer Dreiecksecke oder einer Kante der Triangulierung i.d.R. von mehreren Dreiecken benutzt wird, erzeugt diese Formulierung nur ein unterbestimmtes Gleichungssystem. Deshalb müssen für die Punkte der Oberflächentriangulierung leicht veränderte Punkte der Triangulierung konstruiert werden, damit die Matrix der Basisfunktionen (Designmatrix des Least-Square-Problems) linear unabhängig wird<sup>1</sup>. Werden mehr als sechs Punkte pro Dreieck approximiert, so ergibt sich daraus ein echtes Least-Square-Problem mit einem überbestimmten Gleichungssystem<sup>2</sup>. Nach weiteren Umformungen ergibt sich aus diesem Problem ein lineares Gleichungssystem, welches sich mit Standardmethoden der Linearen Algebra lösen läßt [PS94b]:

$$(E^T \cdot E) \cdot C = E^T \cdot Y \Leftrightarrow C = (E^T \cdot E)^{-1} \cdot E^T \cdot Y \quad (2.15)$$

mit der Designmatrix  $E$ , dem Vektor  $Y$  der zu approximierenden Oberflächenpunkte  $p_i^{\Delta_i}$  und dem Vektor  $C$  der Kontrollpunkte.

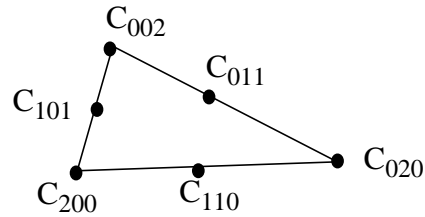
In bestimmten Fällen der Interpolation von verteilten Daten (Scattered Data) führt ein Least-Square-Verfahren nur zu unbefriedigenden Ergebnissen [PS94b]. Da bei dieser Problemstellung jedoch bereits eine Triangulierung vorliegt, die die verteilten Daten exakt wiedergibt, sind diese Probleme hier nicht zu erwarten<sup>3</sup>.

Im Gegensatz zur ersten Methode ist die Lösung eines Least-Square-Problems ein Problem mit der Zeitkomplexität  $O(n^3)$ . Deshalb ist bei der Lösung dieses Problems ein sehr hoher Zeit- und Platzaufwand nötig. So ergibt sich bereits bei einem sehr kleinen Datensatz mit ca. 500 Abtastpunkten und ca. 1000 Dreiecken eine Designmatrix mit ca. 6000 Spalten- und Zeilenvektoren.

3. Least-Square-Approximation mit Fairing: Um die Glattheit von Oberflächen, deren Kontrollnetz durch Least-Square-Approximationen erzeugt wurde, zu verbessern, kann ein zusätzlicher Fairingschritt vorgenommen werden. In diesem in [PS94b] vorgeschlagenen Verfahren werden Störungen in der Oberfläche reduziert, indem mit einem Energiefunktional ihre Krümmung minimiert wird.

---

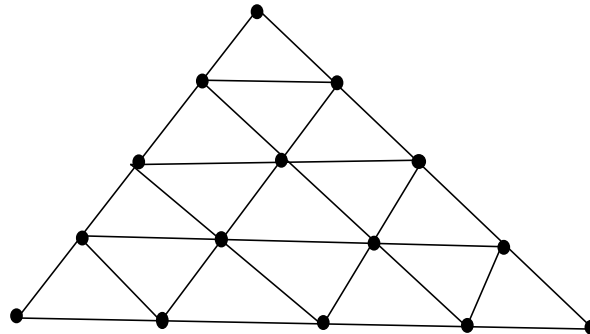
1. Eine direkte Lösung dieses Problems wird im Ausblick, Absatz (7) beschrieben.  
 2. Bei genau sechs Punkten genügt es, daß wir das Least-Square-Problem als "normales" lineares Gleichungssystem betrachten.  
 3. Die Ursache der Probleme liegt darin, daß die Triangulierung die verteilten Daten nicht exakt wiedergibt, sondern nur an sie angepaßt ist.



**Abb. 2.10: Kontrollpunkte eines Dreiecks zu einer DMS-Splinefläche mit dem Grad 2**

### 2.3.3 Bestimmung der Auswertungspunkte

Abhängig von der vorgegebenen Genauigkeit  $n$  werden die Kanten eines Parameterdreiecks in  $n-1$  Abschnitte unterteilt, so daß auf jeder Kante  $n$  auszuwertende Randpunkte liegen. Anschließend werden die Randpunkte der einen Kante mit den entsprechenden Randpunkten der anderen Kanten verbunden, so daß insgesamt innerhalb des Parameterdreiecks  $\frac{n \cdot (n+1)}{2}$  Auswertungspunkte bestimmt werden (siehe Abbildung 2.11). Um zu vermeiden, daß die Auswertungspunkte auf den Kanten mehrfach bestimmt werden, orientiert sich die Auswertung nicht an den Dreiecken, sondern an den Auswertungspunkten. Allerdings bleibt die entsprechende Zugehörigkeitsbeziehung innerhalb der Datenstrukturen erhalten, so daß beim Rendering aus diesen Auswertungspunkten  $n^2 + 2 \cdot n + 1$  Dreiecke konstruiert werden können.



**Abb. 2.11: Auswertungspunkte bei Genauigkeit  $n = 5$**

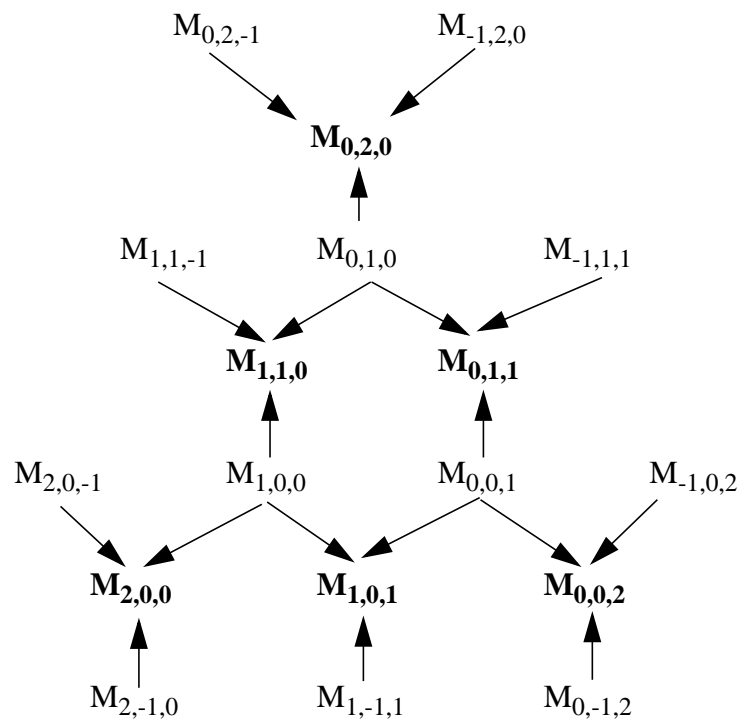
### 2.3.4 Auswertung der DMS-Splinebasisfunktionen

Zur Auswertung der quadratischen DMS-Splinebasisfunktionen wurde ein Algorithmus aus [PS94a] ausgewählt, welcher auf einer Variation des deBoor-Algorithmus zur Auswertung von B-Splinebasisfunktionen beruht. Kernpunkt des verwendeten Algorithmus ist die Idee der Wiederverwertbarkeit von bereits berechneten Werten. In [PS94a] wird ein Schema vorgestellt, welches diese Idee mit einem Auswertungsnetzwerk realisiert.

Ein DMS-Spline vom Grad  $n$  bestimmt sich aus drei DMS-Splines vom Grad  $n-1$ , mit dem entsprechend um eins reduzierten Index (siehe Abb. 2.12, Abb. 2.13). Positive Indizes werden auf die Knotenwolken der zugehörigen Eckknoten  $t_{i,0}$  abgebildet, so z.B. der Multiindex 200 auf die Knoten  $\{r_0, r_1, r_2, s_0, t_0\}$ .

Negative Indizes werden mit Hilfe von drei Regeln auf ein entsprechendes Tripel von Eckknoten abgebildet:

1. Multiindizes, die -1 enthalten: Aus der entsprechenden Knotenwolke wird kein Knoten ausgewählt. Beispielsweise ergibt  $V_{1,1,-1}$  die Knotenmenge  $\{r_0, r_1, s_0, s_1\}$ , es wurde aus der Knotenwolke des Eckknoten  $t$  kein Knoten ausgewählt. In allen anderen Fällen mit einem oder zwei Indizes gleich -1 wird genauso verfahren.

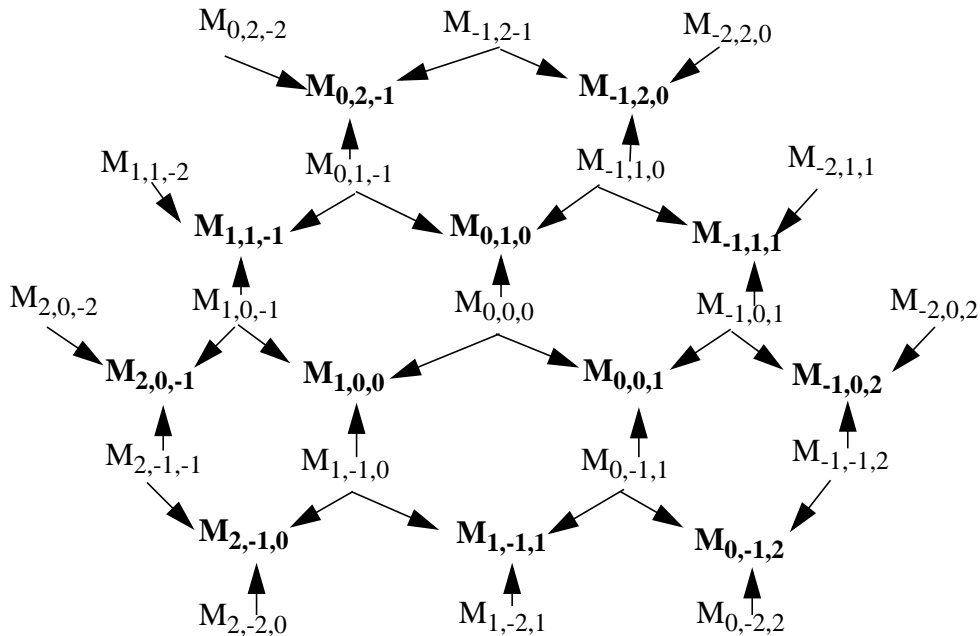


**Abb. 2.12: Auswertungsnetzwerk für quadratische DMS-Splines aus linearen DMS-Splines aus [PS94a]**

2. Multiindizes analog zu 1,1,-2: Der zu  $M_{1,1,-2}$  gehörende Vorgänger  $M_{1,1,-1}$  wurde aus der Knotenmenge  $\{r_0, r_1, s_0, s_1\}$  bestimmt. In diesem Fall muß aus dieser Menge ein weiterer Knoten entfernt werden, wobei die Wahl auf den vorhergehenden Knoten des Eckknoten mit dem negativen Index fällt. Der vorhergehende Knoten bestimmt sich durch die Orientierung des Dreiecks, so daß z.B.  $r$  vor  $s$ ,  $s$  vor  $t$  und  $t$  vor  $r$  liegt. Damit ergibt sich für diesen Fall  $\{r_0, r_1, s_1\}$ .
3. Multiindizes analog zu 2,-2,0: Abhängig von dem entsprechenden Vorgänger 2,-1,0, wird sich für die Streichung eines Knotens aus der ersten Knotenwolke entschieden,

da diese noch vollständig in der Knotenmenge enthalten ist. Damit wird  $\{r_1, r_2, t_0\}$  oder  $\{r_0, r_2, t_0\}$  erhalten, wobei die Wahl auf  $\{r_1, r_2, t_0\}$  fällt<sup>1</sup>.

Durch die Randbedingungen des Algorithmus ist eine Erweiterung zur Auswertung von



**Abb. 2.13: Auswertungsnetzwerk für lineare DMS-Splines aus konstanten DMS-Splines aus [PS94a]**

DMS-Splineoberflächen vom Grad drei oder höher noch nicht geklärt [PS94a]. Da aber bereits quadratische DMS-Splines bei affin unabhängigen Knotenwolken  $C^1$ -stetig sind, reichen diese für die in dieser Arbeit vorgestellte Verwendung völlig aus. Nichtsdestoweniger kann es passieren, daß baryzentrische Koordinaten negativ werden. Bei Verlust der affinen Unabhängigkeit durch kollineare Knoten – z.B. bei multiplen Knoten – erzielt der Algorithmus keine adäquaten Ergebnisse. Deshalb muß sichergestellt sein, daß in der Parametrisierung keine affinabhängigen Knoten enthalten sind.

1. Die Wahl von  $\{r_0, r_2, t_0\}$  wäre auch konsistent gewesen.

### 3 Medizinische Grundlagen

In diesem Kapitel werden die verwendeten medizinischen Grundlagen vorgestellt. Es beginnt mit einer kurzen Einführung in die Anatomie des Schädels, speziell des Gesichtsskeletts. Der zweite Teil gibt eine Zusammenfassung der mechanischen Eigenschaften des Gesichtsgewebes.

#### 3.1 Anatomie des menschlichen Schädels

Das Kopfskelett lässt sich nach [Rohe90] in drei Abschnitte unterteilen:

- das *Neurocranium* (Hirnschädel), das Augenhöhlen, Labyrinth und die Teile des Hirnschädels, die das Gehirn umfassen, enthält,
- das *Splanchnocranium* (Gesichtsskelett) mit Nasen- und Mundhöhlen und einigen anderen kleineren Knochen und
- das *Kiemen- oder Branchialskelett* mit den knöchernen Teilen des Kau-, Hör- und Sprachapparates.

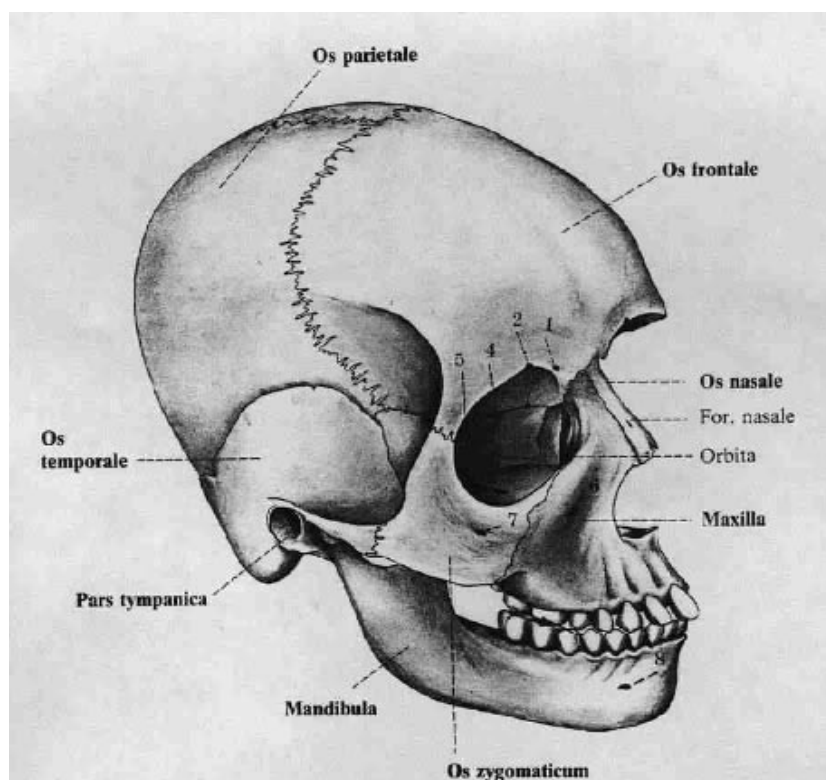


Abb. 3.1: Splanchnocranium und Neurocranium, aus [Rohe90]

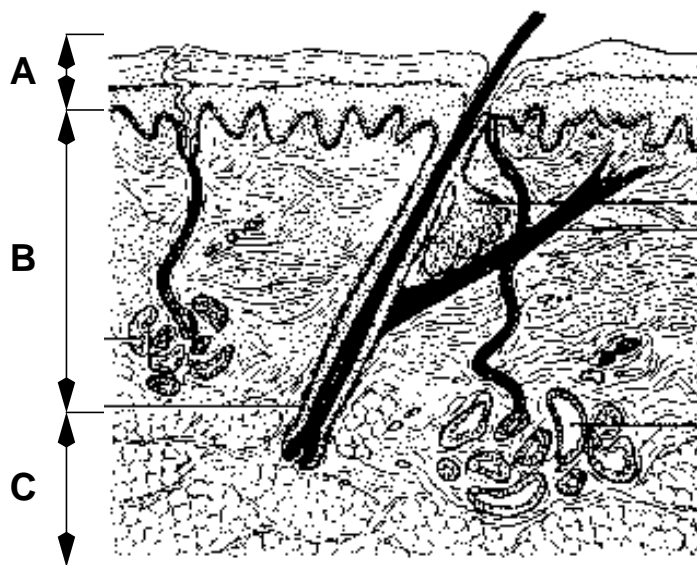
Im folgenden werden kurz *Maxilla* (Oberkiefer) und *Mandibula* (Unterkiefer) beschrieben, da der Eingriff bei der Patientin, deren Operation in Abschnitt 5.1 beschrieben wird, sich auf diese Bestandteile des Splanchnocranium beschränkt hat.

Die paarig angelegte Maxilla bildet das tragende Element des Gesichtsschädels. Sie besteht aus einem *paarigen Körper* (*Corpus maxillae*) und jeweils vier *Fortsätzen* [Rohe90].

Die Mandibula ist ein unpaariger, einheitlicher Knochen und schließt den Gesichtsschädel nach unten ab. Sie besteht aus dem *gebogenen Unterkieferkörper* (*Corpus mandibulae*) und den beiden *Unterkieferästen* (*Ramus mandibulae*) [Rohe90].

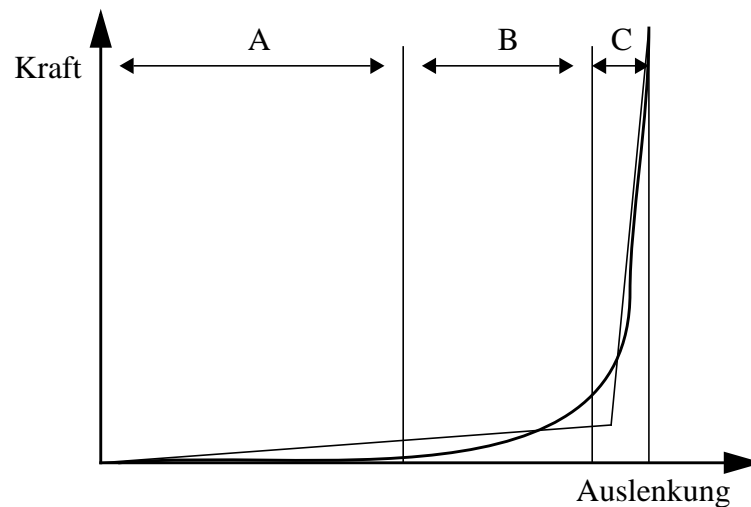
### 3.2 Mechanische Eigenschaften des Gesichtsgewebes

Die Haut des Menschen ist eine vielschichtige Struktur, die aus *Epidermis*, *Korium* und dem *subkutanen Fettgewebe* (*Subkutis*) besteht (Abbildung 3.2). Durch ihren netzwerkartigen Aufbau bildet sie ein inhomogenes, ungerichtetes Verhalten aus [Wate92].



**Abb. 3.2: Die Haut [PZ90]:**  
A: Epidermis, B: Korium, A u. B: Kutis, C: Subkutis

Der größte Teil des Hautgewebes wird vom Korium eingenommen, das auch im wesentlichen die *mechanischen Eigenschaften der Haut* bestimmt. Es besteht aus einer Mischung aus kollagenen (ca. 72%), elastischen (ca. 4%) und gelatineartigen (ca. 20%) Fasern, die bei geringer Krafteinwirkung (z.B. Druck) – während sich die kollagenen Fasern entrollen – einen geringen Widerstand und bei stärkerer Krafteinwirkung – wenn sich die kollagenen Fasern entrollt haben – einen großen Widerstand entwickeln. Läßt die Krafteinwirkung nach, so rollen sich die kollagenen Fasern aufgrund der federartigen Eigenschaften der elastischen Fasern wieder ein und restaurieren auf diese Weise ihre alte Position.



**Abb. 3.3: Verformung des Gesichtsgewebes unter Druck und ihre biphasige Approximation [WT90]**

Abbildung 3.3 zeigt den Widerstand des Hautgewebes bei steigender Krafteinwirkung. In Bereich A zeigt sich nur der schwache Widerstand der elastischen Fasern. Bereich B zeigt den zunehmenden Widerstand der kollagenen Fasern, der sich im Bereich C schließlich voll entwickelt hat [WT90]. Zusammengefaßt ergeben sich für das Gewebe folgende Eigenschaften:

- Epidermis: Mehrschichtige Oberhaut, bis zu 4 mm dick, bestehend aus verhorntem *Plattenepithel*<sup>1</sup>, das ihr eine moderate Steifheit verleihen.
- Korium: Zweischichtige Lederhaut, bestehend aus einer feinfaserigen, elastischen Schicht und aus einer kräftigen, verfilzten Schicht aus Kollagenfaserbündeln. Die Steifheit wird in Abbildung 3.3 wiedergegeben.
- Subkutis: Unterhautzellgewebe, das nahezu den ganzen Fettgehalt der Haut enthält. Darüber hinaus sind in der Subkutis die Blutgefäße und Nerven eingelagert. Insgesamt ist die Subkutis ein hochflexibler Bereich des Gewebes.
- Muskelschicht (Faszie): Aktive Schicht zur Manipulation der Oberfläche [Wate87], [WT90], siehe auch Kapitel 6. Dilatierte Muskeln zeigen eine relativ große Steifheit, kontrahierte Muskeln eine sehr große Steifheit.
- Knochenschicht: Einer der festesten Bestandteile des Körpers, deshalb von außerordentlicher Steifheit (Druck- und Zugfestigkeit: 10 – 15 kg/mm<sup>2</sup>).

1. "Geschlossener ein- oder mehrschichtiger Zellverband, der innere oder äußere Körperflächen bedeckt." [PZ90]



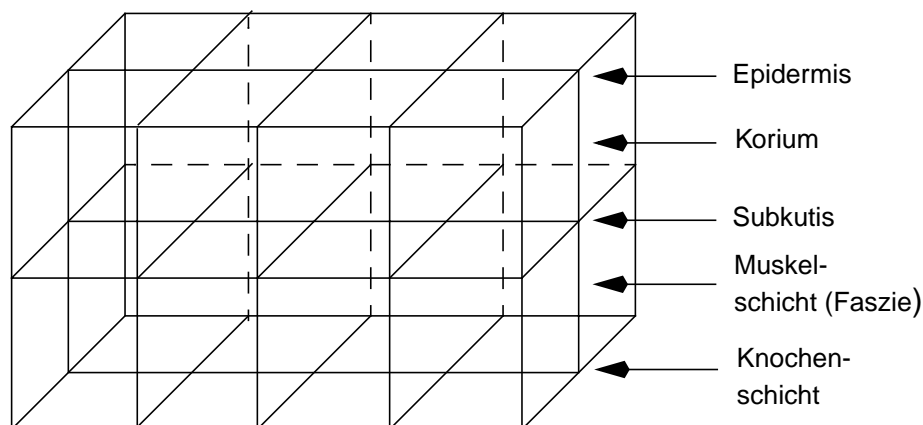
---

## 4 Ein physikalisches Modell

In diesem Kapitel wird ein Modell vorgestellt, mit dem versucht wird, das Gesichtsgewebe des menschlichen Körpers zu simulieren [WT90]. Es beginnt mit der Vorstellung des Gewebemodells. Im zweiten Teil wird die numerische Simulation des Gewebemodells beschrieben.

### 4.1 Das dynamische Gewebemodell

Die Fasern der Epidermis und des Koriums zeigen ein stark federähnliches Verhalten. So liegt es auch nahe dieses Verhalten mit einem *Federmodell* zu simulieren. Zu diesem Zweck wird ein *Federnetzwerk* konstruiert, das aus drei Haupt- und zwei Zwischenschichten besteht, und anschließend in geeigneter Weise konditioniert (siehe Abschnitt 4.1.2).

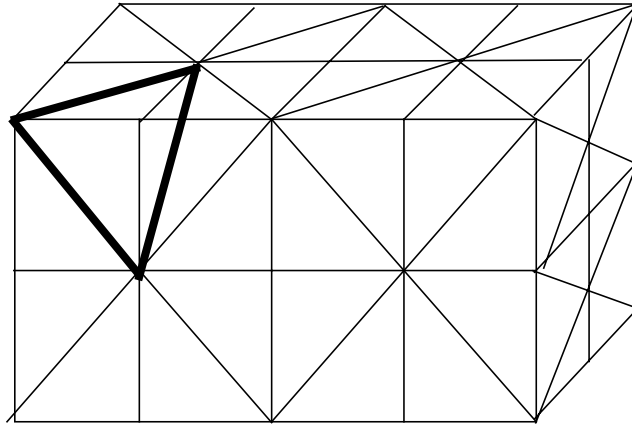


**Abb. 4.1: Federmodell des Gesichtsgewebes**

#### 4.1.1 Konstruktion des Federnetzwerkes

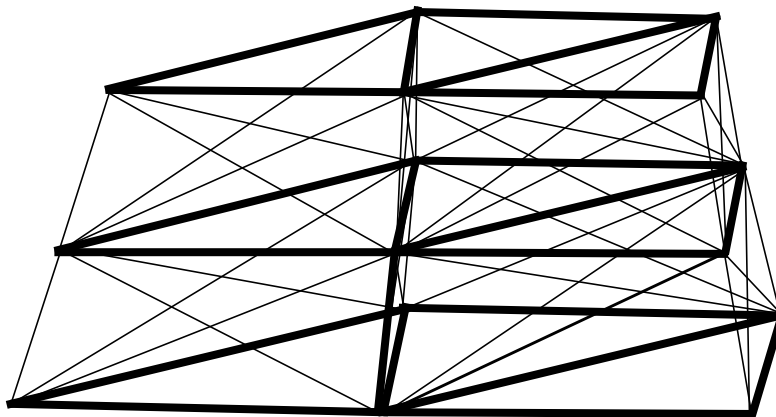
Ausgehend von den Hauptschichten in Form eines regulären Gitters wird das Federnetzwerk entwickelt. Die Zwischenschichten stellen dabei die direkten Verbindungen zwischen den Hauptschichten dar (Abbildung 4.1). Diese Netztopologie kann senkrechten und waagerechten Krafteinwirkungen gut widerstehen; Scherungen und Verwindungen vermag es jedoch kaum Widerstand zu leisten [WT90]. Um ein strukturell stabiles Modell aus Tetra-

ederelementen (in Abbildung 4.2 fett eingezeichnet) zu erhalten, müssen deshalb flächendiagonale Federelemente eingefügt werden.



**Abb. 4.2: Netztopologie mit Diagonalelementen, innere Elemente zur Übersicht nicht dargestellt**

Die Oberflächendaten des Gesichts liegen als reduzierte Triangulierung vor. Deshalb wird in dem Federmodell auch nicht von einem Federnetz in Form eines *strukturierten Gitters* ausgegangen, sondern von der Form eines *unstrukturierten Gitters*<sup>1</sup>.



**Abb. 4.3: Netztopologie eines unstrukturierten Gitters, bestehend aus drei Dreiecken pro Schicht**

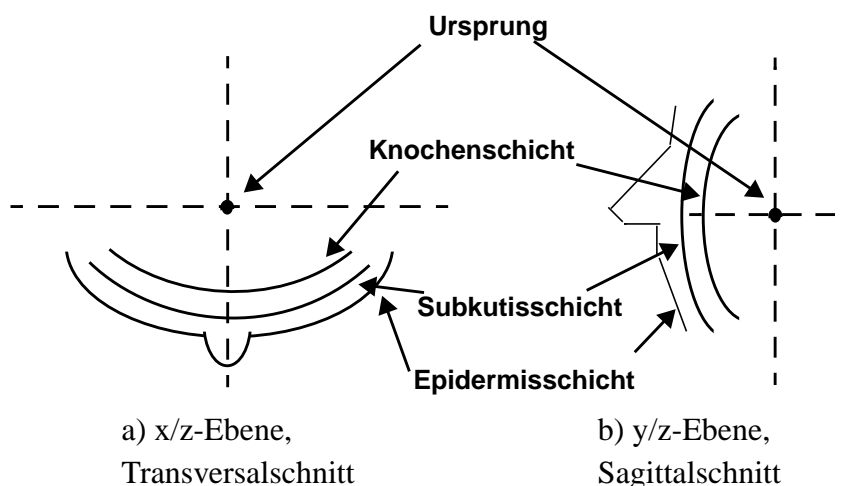
Ausgehend von einer Triangulierung  $T_2$ , die die Epidermis repräsentiert, und einer Triangulierung  $T_0$ , die die Knochenschicht repräsentiert<sup>2</sup>, wird das Netzmodell konstruiert.

1. Die ursprüngliche, von dem 3D-Laserscanner erzeugte, unreduzierte Triangulierung war ein strukturiertes (sogar äquidistantes) Gitter. Durch die Reduktion geht diese Eigenschaft verloren, was aber in diesem Zusammenhang keine weiteren Probleme aufwirft.
2. Im Idealfall sollte dies eine Triangulierung eines aus einer Computer-Tomographie segmentierten Gesichtsschädels sein, zZ. ist es jedoch nur eine Kopie der Gesichtstriangulierung.

Um die mittlere Hauptschicht und die Verbindungen zwischen den Hauptschichten zu konstruieren, wird von folgenden Voraussetzungen ausgegangen:

1. Beide Triangulierungen sind in den Einheitswürfel abgebildet, wobei der Ursprung in mittlerer Höhe hinter den Triangulierungen liegt (Abbildung 4.4a und 4.4b).
2. Die Triangulierungen haben die gleiche Ausrichtung, d.h., daß z.B. die Nasenpartie der Knochenschichttriangulierung im gleichen Winkel liegt wie die Nasenpartie der Epidermistriangulierung (siehe Abbildung 4.4a).
3. Die Triangulierung des Gesichtsschädels (Knochenschicht) bzw. des Gesichtsschädelersatzes ist mit einem adäquaten Faktor<sup>1</sup> skaliert, so daß sie in einem bestimmten Abstand hinter der Triangulierung der Epidermis liegt.

Zunächst wird die Schicht, die die Subkutis repräsentieren soll, aus der Epidermisschicht erzeugt, indem deren Triangulierung erst kopiert und dann skaliert<sup>2</sup> wird. Damit wird gewährleistet, daß diese Schicht zwischen der Epidermis- und der Knochenschicht liegt.



**Abb. 4.4: Anordnung von Epidermis- und Knochenschicht mit Beispielausrichtung**

Die Verbindungen zwischen Epidermis- und Subkutisschicht gestalten sich einfach, da beide eine identische Topologie besitzen. So werden die sich entsprechenden Punkte der Triangulierungen miteinander verbunden. Zusätzlich wird jeweils eine Verbindung zwischen den Nachbarn eines Punktes  $p_i$  in der Epidermisschicht und dem Punkt  $p_i$  der Subkutisschicht geschaffen. Diese Verbindungen stellen die *Flächendiagonalen* des Netzwerkes dar.

Die Konstruktion der Verbindungen zwischen der Subkutisschicht und der Knochenschicht gestaltet sich deutlich schwieriger, da ihre Triangulierungen prinzipiell unterschiedlich sein können. Um in dieser Situation Verbindungen zwischen den beiden Schichten zu konstruieren,

1. Der Faktor beträgt ca. 0,7.
2. Für die Skalierung der Subkutisschicht beträgt der Faktor ca. 0,8.

ren, wird – ausgehend von den oben geschaffenen Voraussetzungen – ein Kriterium definiert, mit dem die verschiedenen Verbindungsmöglichkeiten bewertet werden:

$$q' = scale \cdot q_i,$$

$$\min_{p_i \in T_0} \left\{ \| p_i - q' \|_2 \right\}, \quad (4.1)$$

$$\text{mit } q_i \in T_1 \wedge scale \in \mathbb{R}.$$

Dabei ist  $T_1$  die Menge der Punkte der Subkutistriangulierung,  $T_0$  die Menge der Punkte der Knochenschichttriangulierung und  $scale$  der Projektionsfaktor in Richtung Ursprung.



**Abb. 4.5: Federnetzwerk eines Gesichts nach Medianschnitt**

Mit diesem Kriterium kann eine Nachbarschaftsbeziehung zwischen einem Punkt  $q_i$  der Subkutisschicht und einem Punkt  $p_i$  der Knochenschicht aus der Vielzahl von Möglichkeiten ausgewählt werden. Nachdem der Punkt  $q_i$  noch mit den Nachbarpunkten von  $p_i$  innerhalb der Subkutisschicht und  $p_i$  mit den Nachbarpunkten innerhalb der Knochenschicht verbunden wurde, ist die Konstruktion des Federnetzwerkes vollständig.

### 4.1.2 Konditionierung des Federnetzwerkes

Um eine möglichst realistische Simulation des Gewebes zu gewährleisten, müssen die Federelemente des Netzwerkes die in Abschnitt 3.2 beschriebenen mechanischen Eigenschaften nachbilden.

Entsprechend diesen Eigenschaften werden die Federkonstanten der Federelemente der Epidermis, der Subkutis und der Muskelschicht gewählt. Um das Verhalten des Koriums zu simulieren, wird eine biphasige Approximation verwendet, wie sie in Abbildung 3.2 dargestellt ist und in [WT90] vorgeschlagen wurde. Zusätzlich wird die Steifheit der Knochen-schicht mit einem *Constraint*  $B_i$  bestimmt, der die Punkte dieser Schicht als absolut unbeweglich modelliert [WT90].

Tabelle 1 beschreibt die im Federnetzwerk verwendeten Federelemente.

<i>Schicht</i>	<i>Steifheit</i>
Epidermis	Mittel
Korium	Biphasige Approximation: mittel / hoch
Subkutis	Gering
Muskel	Hoch
Knochen	Völlig steif

**Tabelle 1: Steifheit der Federelemente**

## 4.2 Numerische Simulation des Gewebemodells

Um das dynamische Federnetzwerk mathematisch zu beschreiben, wird die *diskrete Lagrange'sche Bewegungsgleichung* [WT90] verwendet:

$$f_i = m_i \cdot \frac{d^2}{dt^2} x_i(t) + \Upsilon_i \cdot \frac{d}{dt} x_i(t) + g_i \quad i \in \{0 \dots N\} \quad (4.2)$$

oder, mit  $v_i(t) = \frac{d}{dt} x_i(t)$  und  $a_i(t) = \frac{d}{dt} v_i(t)$  :

$$f_i = m_i \cdot a_i(t) + v_i(t) + g_i \quad i \in \{0 \dots N\}. \quad (4.3)$$

Diese Gleichung beschreibt ein System aus (N+1) Knoten, die mit jedem ihrer Nachbarn durch ein Federelement verbunden sind. Jeder Knoten  $i$  hat eine Masse  $m_i$  und eine Position  $x_i(t)$  im Raum. Seine aktuelle Geschwindigkeit beträgt  $v_i(t)$  und er wird mit  $a_i(t)$  beschleunigt. Gedämpft wird seine Geschwindigkeit durch den Faktor  $\Upsilon_i$  [WT90].

Die Größe  $g_i$  beschreibt die Kraft, mit der die Nachbarknoten von  $i$  auf diesen einwirken. Sie setzt sich zusammen aus der jeweiligen Federkonstante  $c_{ij}$  des Federelementes zum Nachbarknoten  $j$ , der Entfernung  $r_{ij}$  zum Nachbarknoten und der natürlichen Länge  $l_{ij}$  des Federelementes  $ij$ , dh. der Länge, bei der das Federelement völlig entspannt ist. Damit ergibt sich  $g_i$  aus:

$$g_i = \sum_{j \in N_i} \frac{c_{ij} \cdot (\|r_{ij}\| - l_{ij})}{\|r_{ij}\|} \cdot r_{ij} \quad , \quad (4.4)$$

wobei  $N_i$  die Menge der Nachbarknoten von  $i$  beschreibt [WT90].

Die Lagrange'sche Bewegungsgleichung beschreibt ein Federnetzwerk, welches sich durch ein Gleichgewicht der Kräfte der Federelemente im Netzwerk und der Kraft  $f_i$ , die von außen auf den Knoten  $i$  des Netzwerkes einwirkt, auszeichnet. Im Normalfall wird die Kraft  $f_i$  Null sein, dh. das Federnetzwerk ist vollkommen ausbalanciert, wobei die Geschwindigkeit  $v_i(t)$  und die Beschleunigung  $a_i(t)$  der einzelnen Knoten Null sind. Dieser Fall stellt außerdem die Anfangssituation des Modells dar.

Wird dieses Gleichgewicht durch das Verschieben eines Knoten gestört, so pflanzt sich diese Wirkung durch das Netzwerk fort, bis dieses zu einem neuen Gleichgewicht gefunden hat [Wate92].

Um aus einer solchen Störung die Bewegung der einzelnen Punkte abzuleiten, muß aus der Gleichung 4.3 die Beschleunigung  $a_i(t)$  des Knoten  $i$  bestimmt werden. Durch Umformung der Gleichung 4.3 und durch Hinzufügen des in Abschnitt 4.1.2 beschriebenen Constraints  $B_i$  ergibt sich mit

$$a_i(t) = (f_i - \Upsilon_i \cdot v_i(t) - g_i) \cdot \frac{B_i}{m_i} \quad (4.5)$$

bzw.

$$\frac{d^2}{dt^2} x_i(t) = \left( f_i - \Upsilon_i \cdot \frac{d}{dt} x_i(t) - g_i \right) \cdot \frac{B_i}{m_i} \quad (4.6)$$

ein Differentialgleichungssystem 2. Ordnung. Aus diesem Gleichungssystem lassen sich mit einem *zweistufigen numerischen Integrationsverfahren* die neue Position  $x_{i+1}(t)$  des Knoten  $i$  und seine neue Geschwindigkeit  $v_{i+1}(t)$  ermitteln.

Eine einfache Integrationsmethode stellt das *Verfahren von Euler*, auch Polygonzugmethode genannt, dar [WT90][PTVF92]:

$$\begin{aligned} K_1 &= h(y_n, t_n) = a(v_n, x_n) \\ v_{n+1} &= v_n + \Delta t \cdot K_1 \\ x_{n+1} &= x_n + \Delta t \cdot v_{n+1} \end{aligned} \quad (4.7)$$

Bei geeigneter Wahl von  $\Delta t$  sind mit diesem Verfahren schnell und numerisch stabil gute Ergebnisse zu erzielen. Muß eine noch bessere numerische Stabilität sichergestellt werden, so bieten sich die Verfahren von Heun, Runge-Kutta oder sogar Mehrschrittverfahren wie Adams-Moulton an [Schw86]. Allerdings sei darauf hingewiesen, daß deren bessere Stabilität mit einem Mehraufwand zu bezahlen ist, der zu Lasten der Interaktivität geht [WT90].

Das Federnetzwerk erreicht seinen neuen Gleichgewichtszustand, wenn die Beschleunigung und die Geschwindigkeit der Knoten zu keinen weiteren Positionsänderungen führen. Da dieser Zustand nur näherungsweise erreicht wird, muß ein Schwellwert gewählt werden, um in der Praxis ausreichend gute Resultate in akzeptabler Zeit zu erreichen.

Wird ein Netzknoten – relativ zur Größe des Netzwerkes – sehr weit ausgelenkt, so zeigt das Federnetzwerk kein konvergentes, sondern ein divergentes Verhalten<sup>1</sup>. Dieses Verhalten entspricht dem Modell, da es als Überdehnung der Federelemente gedeutet werden muß, die zum Riß der Federelemente – respektive des Gewebes – führt. Da in dem Gewebemodell nur die Elemente der Knochenschicht ausgelenkt werden (siehe Anhang B), sind nur kleine Auslenkungen sinnvoll, weshalb dieses Problem hier nicht zu erwarten ist.

Ein weiteres – den Federnetzwerken inhärentes – Problem, stellt das Oszillieren eines oder mehrerer einzelner Knoten dar. Dieses Verhalten tritt vor allem bei kleinen Netzwerken auf, weshalb dieses Problem bei den zu behandelnden Problemgrößen kaum auftritt.

Bei der numerischen Simulation wird davon ausgegangen, daß das Federnetzwerk wieder ausbalanciert ist, wenn eines der beiden folgenden Kriterien erfüllt ist:

1. Die Geschwindigkeit  $v_i(t)$  des Knoten  $i$  mit der höchsten Geschwindigkeit ist kleiner als der gewählte Schwellwert  $\epsilon$ . Damit ist die gewählte numerische Genauigkeit erreicht.
2. Nach hundert Iterationen wurden keine deutlichen Veränderungen der Positionen der Knoten im Federnetzwerk mehr festgestellt; obwohl es Knoten im Netzwerk gibt, deren Geschwindigkeit über dem Schwellwert  $\epsilon$  liegt, bewirken sie keine Änderungen im Netzwerk. In diesem Fall wird vermutet, daß die Punkte nur um ihre eigene Position oszillieren.

Zum Schluß sei noch einmal klargestellt, daß die gewählten Konvergenzkriterien nur einen Kompromiß zwischen theoretischer Genauigkeit, Interaktivität und realem Konvergenzverhalten darstellen können.

---

1. Diese Aussage beschränkt sich auf den beobachteten Zeitraum von ca. 30000 Iterationen. Auch nach dieser Anzahl von Iterationen zeigte das Netzmodell ein divergentes Verhalten.



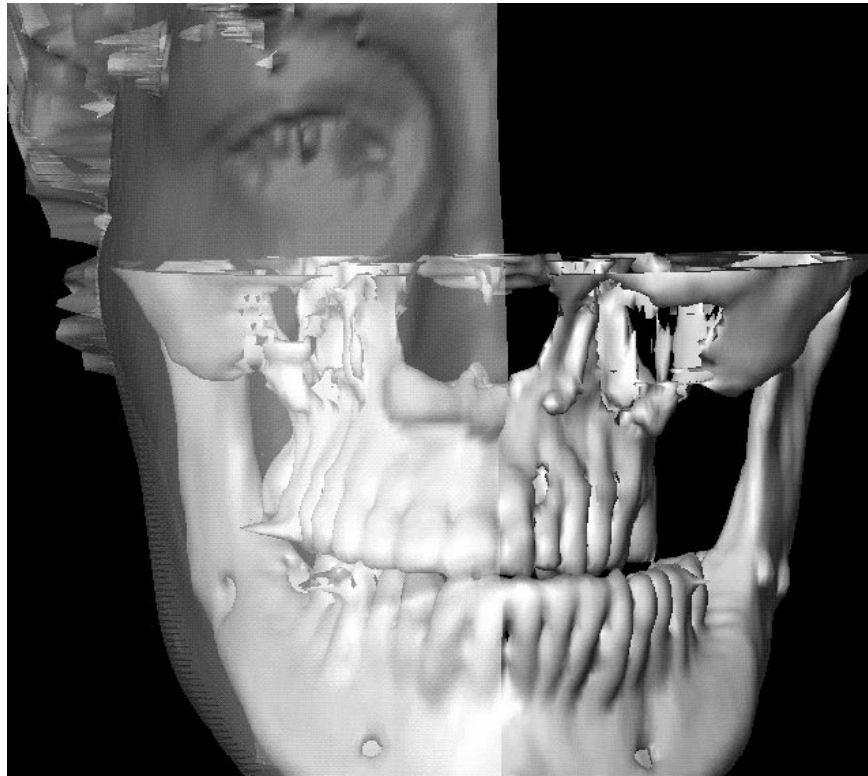
---

## 5 Ergebnisse

In diesem Kapitel werden die Ergebnisse der Modellbildung vorgestellt. Zu diesem Zweck wird im ersten Abschnitt kurz die durchgeführte Operation beschrieben. Daran schließt im zweiten Abschnitt die Darstellung der Ergebnisse der modellgestützten Simulation und – im letzten Abschnitt – der Visualisierung durch DMS-Splines an.

### 5.1 Eine Craniofaciale Korrekturoperation

Der Oberkiefer der Patientin war nach rechts<sup>1</sup> und nach vorne verschoben. Zusätzlich war er nach rechtsoben geneigt. Der Unterkiefer war nach links verschoben (siehe Abbildung 5.1).



**Abb. 5.1: Frontalsicht auf das Gesichtsskelett der Patientin vor der Operation, aus [GKG95]**

---

1. Orientierungsangaben werden immer aus Sicht der Patientin angegeben.

Der chirurgische Eingriff bei der Patientin bestand im wesentlichen aus zwei Teileingriffen:

- (1) Durchtrennung des Corpus maxillae<sup>1</sup> im Bereich der Facies anterior, d.h. unterhalb der Nasenhöhle, in einen unteren Teil – mit dem Zahnfortsatz (Proc. alveolaris) – und in einen oberen Teil, der am verbleibenden Rest des Splanchnocraniums befestigt ist.
- (2) Durchtrennung des linken Bügels der Mandibula am Übergang zwischen dem hufeisenförmigen Corpus mandibulae und dem linken Unterkieferast.

Abbildung 5.2 illustriert die Schnittebenen der Durchtrennungen.

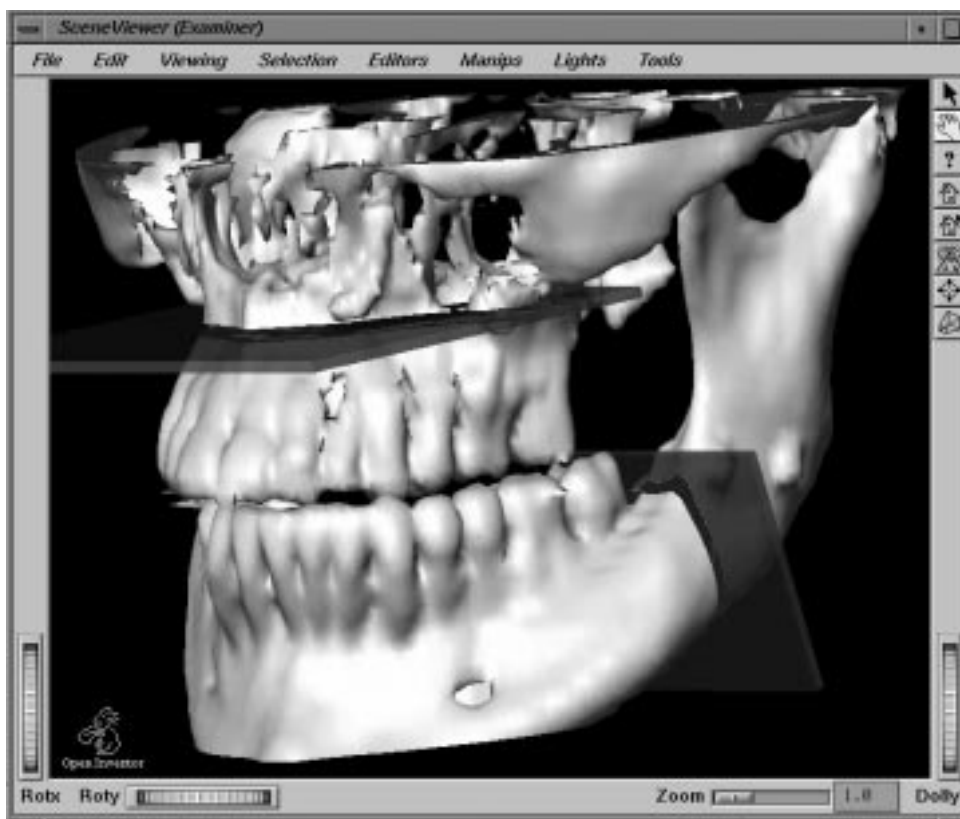


Abb. 5.2: Schnittebenen zur craniofacialen Korrekturoperation, aus [GKG95]

Nach der Durchtrennung wurden die gelösten Knochenteile neu ausgerichtet und fixiert.

---

1. Körper der Maxilla

## 5.2 Ergebnisse der Implementierung

Im ersten Teil dieses Abschnitts werden die Ergebnisse einer modellgestützten Simulation vorgestellt, mit der versucht wird, ausgehend von einem Datensatz, der die präoperative Gesichtsstruktur einer Patientin repräsentiert, deren postoperative Gesichtsstruktur zu bestimmen. Im zweiten Teil wird die Verbesserung der Darstellungsqualität mit Hilfe der DMS-Splines demonstriert.

### 5.2.1 Modellgestützte Simulation

Der folgende Algorithmus beschreibt das Vorgehen, mit dem eine Simulation einer cranio-facialen Korrekturoperation durchgeführt wird:

```
determine processingsteps and the
  dedicated regions in bonelayer
forall processingsteps do
  Select all points in region
  do
    apply translationvector to points
    while points have not reached targetpositions
  od
```

Vorgehensweise zur Durchführung der Simulation

Im ersten Schritt erfolgt die Planung der Schrittreihenfolge und der zugehörigen Regionen der Knochenschicht des Modells. In jedem dieser Bearbeitungsschritte werden die zu der Bearbeitungsregion zugehörigen Punkte bestimmt und mit einem Translationsvektor in Richtung ihrer Zielposition bewegt.

Die folgende Tabelle beschreibt die Bearbeitungsschritte dieser Simulation. Die rechte Spalte beschreibt mit der ersten Zahl die Bearbeitungshäufigkeit, d.h. die Anzahl, mit der die Bearbeitungsschritte durchgeführt wurden. Die zweite Zahl gibt die Anzahl der Iterationen an, die für jeden Bearbeitungsschritt von dem Federnetzwerk durchgeführt wurden (in den beschriebenen Fällen immer nur eine Iteration):

<i>Bearbeitungsschritt<sup>a</sup></i>	<i>ungefähre Anzahl der bewegten Punkte</i>	<i>ungefähre Anzahl der Iterationen</i>
Punkte zwischen Nase und Oberlippe (entspricht dem unteren Teil der Maxilla) werden nach vorne links bewegt.	60	10 x 1

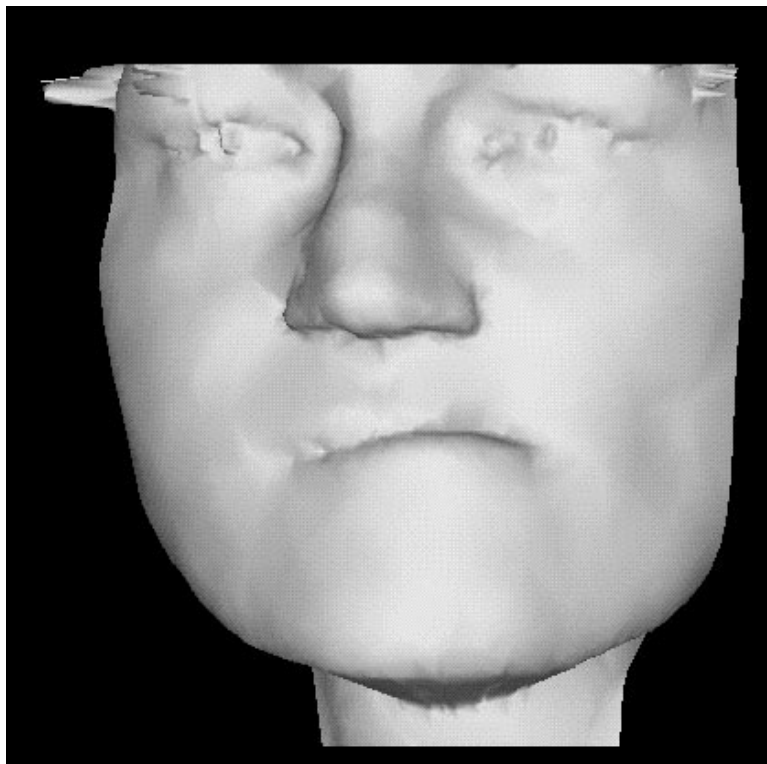
**Tabelle 2: Überblick über die durchgeführte Simulation**

<i>Bearbeitungsschritt<sup>a</sup></i>	<i>ungefähre Anzahl der bewegten Punkte</i>	<i>ungefähre Anzahl der Iterationen</i>
Punkte, die mit dem unteren Teil der Maxilla assoziiert werden, werden nach links bewegt.	60	10 x 1
Punkte des unteren Gesichts, die ungefähr der Mandibula entsprechen, werden nach rechts bewegt.	150	8 x 1
Punkte des linken Teils der Mandibula (vergleich oberer Schritt) werden nach rechts bewegt.	70	5 x 1

**Tabelle 2: Überblick über die durchgeführte Simulation**

- a. Orientierung immer aus Sicht der Patientin

Das Ergebnis der Simulation ist sehr stark abhängig von der Wahl der Punkte und des Translationsvektors, mit dem sie in eine neue Position gebracht werden. Dies erschwert die Planung und Reproduktion von Ergebnissen.

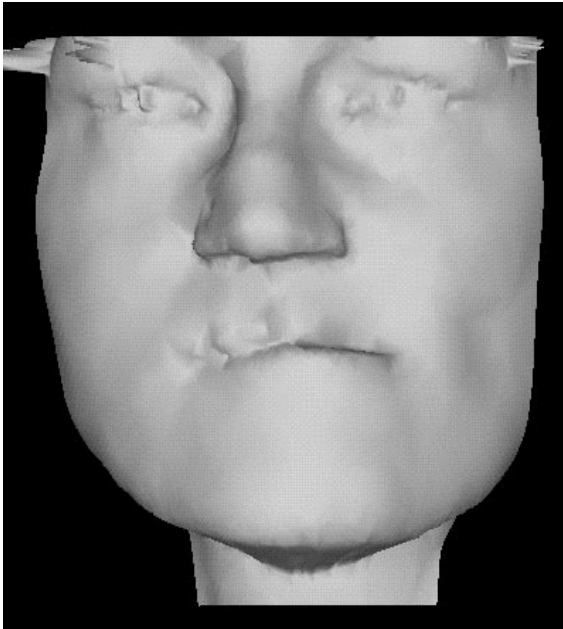


**Abb. 5.3: Präoperative Aufnahme**

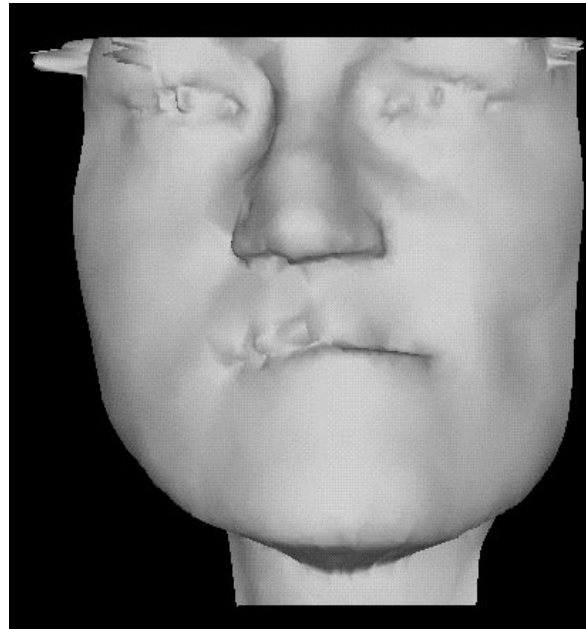
Die geringe Anzahl der Netziterationen (bei entsprechend erhöhter Bearbeitungshäufigkeit) zeigt, daß das Konvergenzverhalten der Federnetzwerke in der Praxis mit großen Netzwer-

ken bei relativ kleinen Auslenkungen kein Probleme darstellt. Während der Simulation griff immer das in Abschnitt 4.2 erwähnte erste Kriterium<sup>1</sup>.

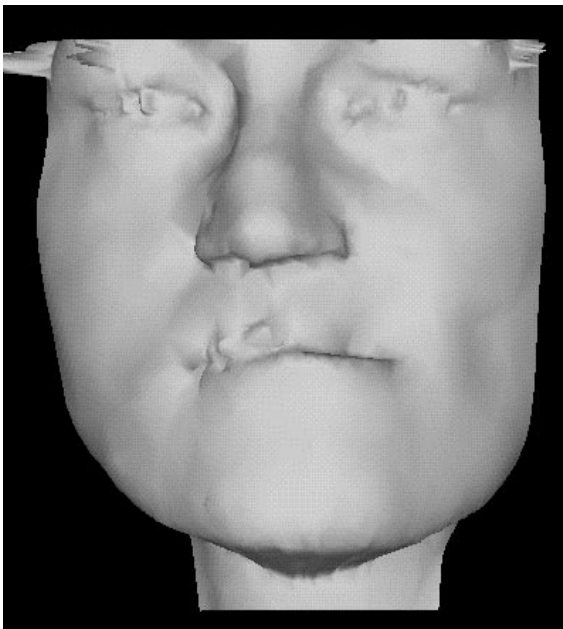
Die folgenden Bilder geben die Konstellationen der Simulation nach den einzelnen Bearbeitungsschritten, die in Tabelle 2 beschrieben sind, wieder:



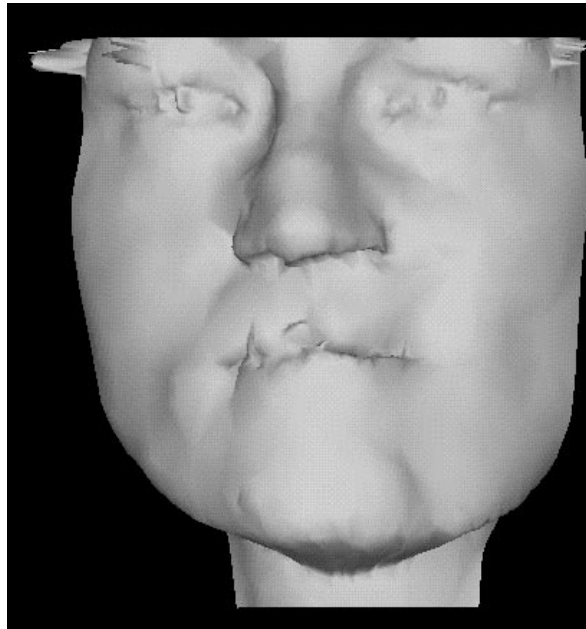
**Abb. 5.4: Simulation nach dem ersten Bearbeitungsschritt**



**Abb. 5.4a: Simulation nach dem zweiten Bearbeitungsschritt**



**Abb. 5.5: Simulation nach dem dritten Bearbeitungsschritt**



**Abb. 5.5a: Simulation nach dem vierten Bearbeitungsschritt**

---

1. Zur Erinnerung: Die Bewegungen im Netzwerk unterschritten einen Schwellenwert  $\epsilon$ .

Die beste Annäherung an das Original wird in Abbildung 5.5a erreicht. Leider wird der optische Eindruck stark durch Verwerfungen in der Ober- und Unterlippe gestört. Die Verwerfungen entstehen durch die einheitliche Modellierung von “normalem” Gesichtsgewebe und den Lippen. Durch ein separates Detailmodell der Lippen (z.B. ein Ringmuskel, wie in [Wate87] vorgeschlagen) könnte dieser Effekt behoben werden.

### 5.2.2 DMS-Splineoberfläche des Gesichts

Bei der Implementierung des Gewebemodells wird die Epidermisschicht, durch eine DMS-Splineoberfläche visualisiert. Mit dieser Methode wird die Qualität der Darstellung erhöht, da zusätzliche Punkte durch die Splineinterpolation auf der Oberfläche bestimmt werden, die in der Triangulierung der Gesichtsoberfläche nicht enthalten sind.

Tabelle 3 zeigt das Zeitverhalten bei der Parametrisierung und Bestimmung der DMS-Splineoberfläche<sup>1</sup> und die Anzahl der daraus resultierenden Punkte und Dreiecke bei den angegebenen Genauigkeitsparametern  $n$ .

Datensatz #Punkte/ #Dreiecke	P/[s]	$n=3$		$n=5$		$n=7$	
		A/[s]	#Punkte <sup>a</sup> #Dreiecke	A/[s]	#Punkte #Dreiecke	A/[s]	#Punkte #Dreiecke
A: geringe Auflösung 552/1012	31	22	2115 4048	94	8277 16192	344	18487 36432
B: hohe Auflösung 1157/2273	68	60	4586 9092	392	18263 36368	1966	41032 81828
C: präoperativ 1502/2976	102	193	5979 11904	780	23861 47616	3423	107136 53647
D: postoperativ (fast un- reduzierter Datensatz) 8641/17001	584	2517	34282 47616	25156	136567 272016	150000 <sup>b</sup>	1000000 <sup>b</sup> 623036

**Tabelle 3: Zeitmessung und geometrische Daten von Parametrisierung P, Bestimmung der Auswertungspunkte, Auswertung und Oberflächenkonstruktion A**

- Punkte, die auf einer Dreieckskante oder einer -ecke liegen, werden nur einmal angegeben. Deshalb ergibt sich deren Anzahl nicht aus der Anzahl der Dreiecke mal Anzahl der Punkte pro Dreieck.
- Bei dieser Problemgröße würde das Swappen – aus Speicherplatzmangel – die Messung verfälschen. Aus diesem Grund kann hier nur eine grobe Schätzung angegeben werden, deren Werte mit Hilfe von Komplexitätsbetrachtungen (siehe Anhang A) extrapoliert wurden.

Die Meßwerte wurden auf einer Silicon Graphics Onyx - Reality Engine 2 mit zwei Prozessoren erzeugt. Die spezielle Graphikhardware des Rechners wurde nicht ausgenutzt.

Die folgenden Bilder zeigen nur die Ergebnisse der Visualisierung des Datensatzes B. Dieser Datensatz stellte eine 3D-Laserscannerabtastung einer Frau dar, die nicht identisch zur Patentin aus den Abschnitten 5.1 und 5.2.1 ist.

1. Dazu gehört: Bestimmung der Auswertungspunkte, Auswertung der Splinebasiswerten, Approximation der Kontrollpunkte (hier nur mit der einfachen Approximation) und die Oberflächenkonstruktion.

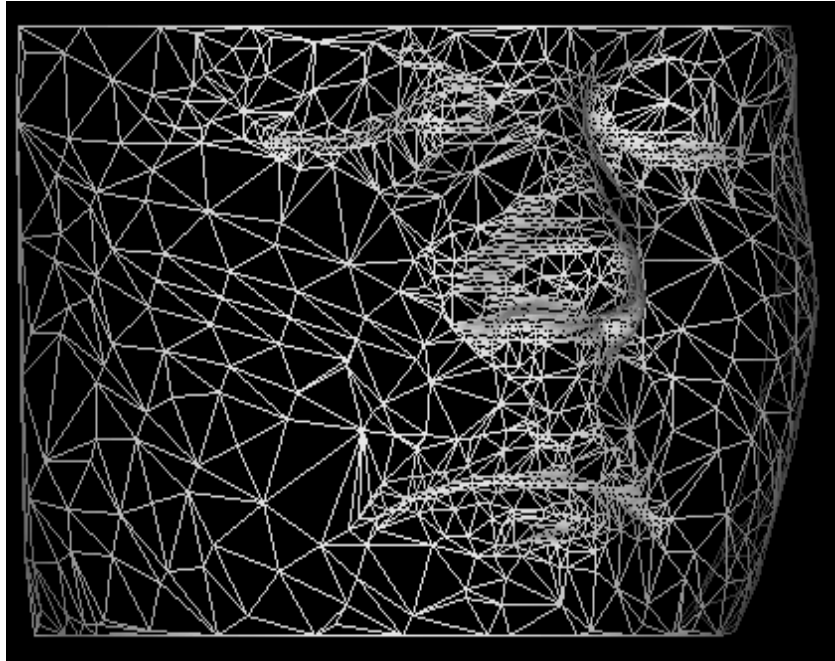


**Abb. 5.6: Polygonale Fläche des Datensatzes B**

In dem hier dargestellten Datensatz B ist der Zugewinn an Detailstruktur klar zu erkennen. Die scheinbar rauhere Oberfläche der Splinedarstellung ist eine Folge der Kontrollpunktapproximation: hier wurde die einfache Approximation gewählt. Die Auflösungsgenauigkeit beträgt  $n=5$ , was zu sechzehnmal soviel Dreiecken wie bei der polygonalen Darstellung führt (siehe Abschnitt 2.3.3).

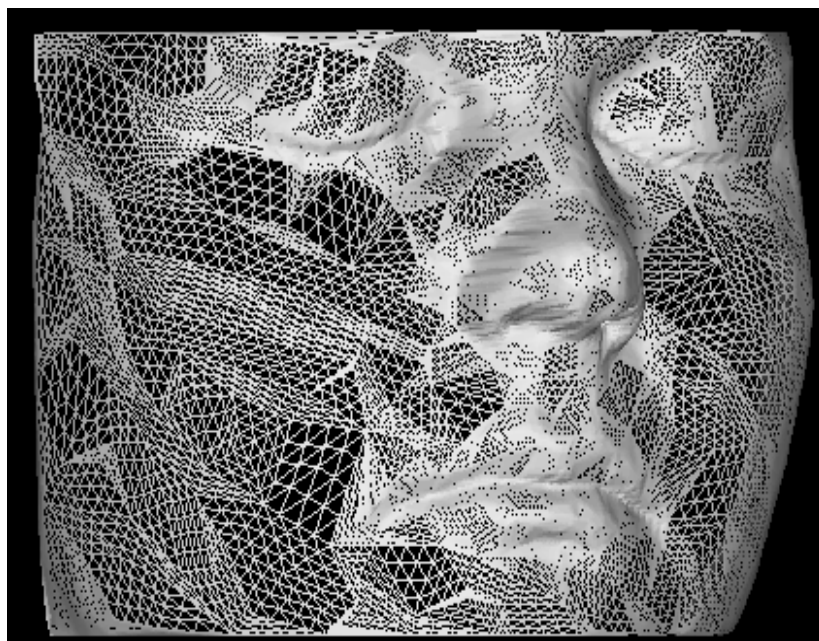


**Abb. 5.7: DMS-Splinefläche zum Datensatz B**



**Abb. 5.8: Polygonnetz des Datensatz B**

In dieser Darstellung des Datensatzes B wird die Verfeinerung des Polygonnetzes deutlich: Regionen, die bereits eine feine Struktur besitzen, werden durch die DMS-Splinedarstellung weiter verfeinert (Nase, Mund, Augen).



**Abb. 5.9: Polygonnetz der DMS-Splinedarstellung zu dem Datensatz B**



---

## 6 Zusammenfassung und Ausblick

In dieser Arbeit ist ein Modell vorgestellt worden, das zur Simulation von Gesichtsgewebe verwendet wurde. Ausgehend von einer Triangulierung der Gesichtsoberfläche, die mit einem 3D-Laserscanner vermessen und anschließend reduziert wurde, ist dieses Modell aus einem Federnetzwerk konstruiert worden.

Wenn die Ergebnisse der Simulation mit den Ergebnissen der tatsächlich durchgeführten Operation verglichen werden, so fällt auf, daß die Simulation ähnliche, aber keinesfalls gleiche oder gar identische Gesichter produziert. Die Gründe für dieses Verhalten sind vielfältig: In dem Gewebemodell werden eine Reihe von Annahmen vorgenommen, die die Modellbildung wesentlich vereinfachen und damit handhabbar machen. Dies führt jedoch zu einem Verlust an Realitätstreue. So ist z.B. die Wahl gleicher Federkonstanten der gleichen Schicht eine große Vereinfachung bei der Anpassung des Gewebemodells und gibt deshalb nicht die inhomogene Struktur des natürlichen Gewebes wieder.

Eine Planung von craniofacialen Eingriffen wird augenblicklich im klinischen Alltag anhand von Gipsmodellen, Röntgenaufnahmen und Fotografien vorgenommen [GKG95]. Mit diesen Methoden ist jedoch keine Simulation möglich, da hier nur eine Wiedergabe von optischen Informationen erfolgt, die nur wenige Aussagen über die physikalischen Eigenschaften der entsprechenden Körperregion zulassen. Gegenüber diesen Methoden werden durch das vorgestellte Modell neue Aspekte gewonnen.

Um die Gesichtsstruktur zu visualisieren wurde eine splinebasierte Methode – die DMS-Splines – gewählt. Mit dieser Methode ist es möglich, einen besseren plastischen Eindruck, eine höhere Genauigkeit<sup>1</sup> und eine mathematisch gut handhabbare Oberflächenbeschreibung zu gewinnen.

Durch die vorgenommene Reduzierung in der Triangulierung werden Regionen mit einfacher Oberflächenstruktur mit wenigen Dreiecken, Regionen mit großem Detailreichtum mit einer höheren Anzahl von Dreiecken dargestellt. Diese Strukturierung wird von den DMS-Splines ausgenutzt und verbessert dadurch – besonders in Bereichen mit einer komplexeren Struktur – die Auflösung. Prinzipiell ist es durch die Splineinterpolation sogar möglich, die Darstellungsgenauigkeit über die Abtastgenauigkeit des verwendeten Laserscanners hinaus zu vergrößern<sup>1</sup>.

Um die Qualität der Simulation gegenüber der durchgeführten Operation zu verbessern, sind eine Reihe von Erweiterungen des bestehenden Modells möglich:

- (1) Individualisierung der Federelemente: Um der inhomogenen Struktur des Hautgewebes gerecht zu werden, müßten die Federelemente einer Netzschicht individuell angepaßt werden, im Gegensatz zu der implementierten uniformen

---

1. Diese höhere Genauigkeit beruht auf der Interpolation der Datenpunkte durch eine Splinefläche. Aus diesem Grund erhöht sich die Genauigkeit nur mathematisch. Tatsächlich kann es passieren, daß sich im Detail die Splinefläche von der wirklichen Gesichtsoberfläche unterscheidet.

---

Konditionierung.

- (2) Eine realitätsorientierte Simulationen des Gesichts bleibt ohne eine Muskelkomponente unvollständig. In [Wate87] wird ein parametrisiertes Muskelmodell vorgestellt, das mit Hilfe eines Systems zur Klassifizierung von Emotionen [EF77] kontrolliert wird. Dieses Modell wird in [WT90] in das Gewebemodell, das auch in dieser Arbeit verwendet wurde, integriert.
- (3) In bestimmten Situationen durchstößt die Knochenschicht die höheren Netzschichten. Eine Integration eines Algorithmus zur Überprüfung von 3D-Kollisionen erscheint hier als möglicher Weg, um diesem Phänomen zu begegnen. Sollte eine solche Kollision festgestellt werden, so könnten die Netzschichten einfach weiter geschoben werden.
- (4) Anpassung von Computertomographiedaten (CT-Daten) des Schädels an das Modell: Im verwendeten Modell werden die Elemente der Knochenschicht durch eine Instanz der Epidermisschicht approximiert. Die Verwendung von tatsächlichen Schädeln würde die Approximation verbessern<sup>1</sup> und zusätzlich Fragen, die in (6) vorgestellt werden, teilweise lösen. Die Verwendung von Schädeln würden allerdings auch einige Frage aufwerfen, die zum Teil in Abschnitt 4.1.1 als Voraussetzung zur Konstruktion der Verbindungselemente zwischen Subkutis- und Knochenschicht aufgeführt wurden. Im folgenden werden die wichtigsten Probleme kurz diskutiert:
  - Bestimmung einer geeigneten Skalierung der aus CT-Daten gewonnenen Schädeltriangulierung, so daß die Knochenschicht räumlich hinter der Subkutisschicht liegt.
  - Um eine gleiche Ausrichtung von Epidermis-, Subkutis- und Knochen-schicht zu erreichen (siehe Abbildung 4.4), könnten anhand von Gesichts-merkmalen (z.B. Nasenbein/-spitze, Unterkiefer/Kinn, Augenhöhlen, etc.) die aktuelle Ausrichtung der Gesichts- und Schädeln bestimmt und in einander überführt werden.
  - Lösung des Verbindungsproblems bei Verbindungen zwischen Subkutis und Schädel: Bei der Konstruktion von Verbindungselementen zwischen Subkutis- und Knochenschicht werden die "nächsten" Nachbarpunkte der jeweils anderen Schicht gesucht. Wird aufgrund einer Schädelhöhle (z.B. Augenhöhle) kein direkter Nachbar gefunden, muß darauf geachtet werden, daß nur geeignete "Ersatznachbarn"<sup>2</sup> ausgewählt werden.
- (5) Bei der Darstellung der Ergebnisse fällt auf, daß die Modellierung der Lippen bei einem einheitlichen Gewebemodell sehr problematisch ist. In zukünftigen Arbeiten muß überdacht werden, ob in solchen Fällen (Lippen, Ohren, etc.) nicht zusätzliche Detailmodelle konstruiert werden müssen.

---

1. Eine Lösung dieses Problems wird in [Wate92] angedeutet.

2. Der "Ersatznachbar" sollte – entsprechend der Voraussetzungen zur Konstruktion des Verbindungsnetzwerkes in Abschnitt 4.1.1 – möglichst in Richtung zum Ursprung liegen.

- 
- (6) Die Manipulation der Triangulierungspunkte der Knochenschicht erfolgt durch Auswahl der Punkte und einer anschließenden Positionsänderung durch einen Verschiebungsvektor. Dieses Verfahren entspricht in keiner Weise der Manipulation, die ein/e ChirurgIn an dem entsprechenden Körperteil vornimmt, und ist zusätzlich nur durch Austesten der Wirkung bestimmbar. Es muß also eine Methode entwickelt werden, die mittels geeigneter “chirurgischer Primitive” die zugehörigen Punkte und Verschiebungsvektoren bestimmt. Des weiteren erscheint in diesem Zusammenhang die Zusammenfassung von Punkten zu größeren “anatomischen Objekten” – z.B. Nase, Mund, Auge, etc. – sinnvoll. Eine allgemeine automatisierte Methode, die diese Zuordnung vornehmen könnte, ist jedoch vermutlich nur mit mehr oder weniger großen Abstrichen möglich.
- (7) Eine Verbesserung der Approximationsmethode des Kontrollnetzes würde die Qualität der Oberflächendarstellung weiter erhöhen:  
Die in Abschnitt 2.3.2 vorgestellte Methode zur Konstruktion der Designmatrix birgt in der Konstruktion von zusätzlichen, linear unabhängigen Punkten Ungenauigkeiten, da diese Punkte nicht auf den Meßpunkten des 3D-Laserscanners liegen, aber trotzdem deren Koordinaten annehmen. Da die entsprechenden Kontrollpunkte benachbarter Dreiecke identisch sind, bietet sich eine bessere Methode an: Indem die identischen Kontrollpunkte zu einem veränderlichen Parameter der Approximation zusammengefaßt werden, wird das Gleichungssystem umgeformt. Diese Formulierung ermöglicht die Verwendung der Meßpunkte als Approximationspunkte und reduziert damit den Umfang der Designmatrix und damit des Problems.  
Die in [PS94b] vorgeschlagene Methode des zusätzlichen Glättens durch ein Fairing-Funktional wird allerdings vermutlich zu keiner Verbesserung führen<sup>1</sup>, da in diesem Fall nicht wirklich von verteilten Daten im Sinne der Scattered Data Interpolation gesprochen werden kann.
- (8) Die Qualität der DMS-Splineoberfläche steht und fällt mit der Qualität ihrer Parametrisierung. Sobald deren Qualität sinkt, folgt daraus eine Verschlechterung der Qualität der Oberfläche. Bei dem Vermessen der Oberfläche eines Gesichts mit Hilfe eines Laserscanners können numerische Grenzfälle auftreten, die bei der Erzeugung der Parametrisierung zu degenerierten Dreiecken führen. Ein ähnliches Problem kann bei der Reduktion der Polygonoberfläche durch [SZL92] auftreten. Bei vereinzelt auftretenden Degenerationen von einzelnen Dreiecken kann der Parametrisierungsalgorithmus diese Degenerationen korrigieren, treten diese Probleme jedoch gehäuft auf, so muß hier eine neue Lösung gefunden werden.
- (9) Verbesserungen in numerischer Hinsicht können durch spezielle Lösungsverfahren der Numerik für lineare Gleichungssysteme erreicht werden, wenn die Bandcharakteristik der Matrix des Least-Square-Problems bei der Approximation des Kontrollnetzes ausgenutzt wird.

---

1. siehe auch Abschnitt 2.3.2

---

(10) Die implementierten Algorithmen weisen weitestgehend ein beträchtliches Parallelisierungspotential von fast beliebiger Skalierbarkeit auf. Nach einer Abbildung der Algorithmen auf eine Multiprozessorarchitektur kann – bei vernünftiger Problemaufteilung und ausreichender Problemgröße – mit einem hohen Speedup gerechnet werden. Das Problem läßt sich gut auf

- Shared-Memory-Multiprozessorarchitekturen, wie der SGI Onyx oder der Convex SPP,
- oder eine Message-passing-Bibliothek wie PVM (Parallel Virtual Machine) – die auch für SGI-Workstations existiert –

abbilden.

Ein Abbildung der Algorithmen auf eine massiv parallele oder SIMD Architektur wie dem MASP-1, wird wegen der hohen Komplexität der Einzelprobleme vermutlich zu keinen Verbesserungen führen.

---

# Anhang

## A Anmerkungen zur Komplexität

Eine vollständige und tiefgehende Untersuchung zur Komplexität ist nicht Gegenstand dieser Arbeit und würde deren Rahmen sprengen. Trotzdem erscheint es sinnvoll, eine Zeitkomplexitätsbetrachtung zumindest zu skizzieren (Speicherplatzkomplexitäten können nie größer als Zeitkomplexitäten werden, weshalb auf deren Betrachtung verzichtet wird.).

In dieser Untersuchungsskizze zur Komplexität der implementierten Algorithmen werden zwei Bereiche unterschieden:

- Komplexität der Initialisierung
- Komplexität der Iterationsschritte

Beiden Bereichen gemeinsam sind die Größen, aus denen sich deren Komplexitätsklassen bestimmt. Es werden hier folgende Größen unterschieden:

- $p$  für die Anzahl der (Eck-)Punkte der Triangulierung,
- $t$  für die Anzahl der Dreiecke der Triangulierung,
- $n$  für die gewählte Auflösungsgenauigkeit der DMS-Splineoberfläche, d.h. die Anzahl der Auswertungspunkte pro Dreiecksseite.

### A.1 Komplexität der Initialisierung

Bei der Initialisierung werden eine Reihe von Einzeloperationen durchgeführt, die im folgenden betrachtet werden.

- (1) Konstruktion der Nachbarschaftsbeziehungen in der Triangulierung.  
Bei diesem Schritt werden zu jedem Punkt die Dreiecke bestimmt, die diesen Punkt als Eckpunkt verwenden. Damit ergibt sich für diesen Schritt eine Komplexität  $O(p \cdot t)$ . Die Nachbarpunkte werden aus den Nachbardreiecken bestimmt. Die Komplexität ist im Extremfall so groß wie die der Bestimmung der Nachbardreiecke zu einem Punkt. Damit ist es möglich diese Operation nach oben abzuschätzen und erhalten so einen konstanten Faktor, der vernachlässigt werden kann.
- (2) Konstruktion der Oberflächen aus der Triangulierungsinformation.  
Für jedes Dreieck der Triangulierung muß ein Normalenvektor bestimmt werden, damit die Beleuchtungsinformation der fertigen Oberfläche konsistent durchgeführt werden kann. Abhängig von der Nachbarschaftsstruktur der Triangulierung können unterschiedlich viele Dreiecke Nachbardreiecke eines Punk-

---

tes sein. Die Anzahl der Nachbardreiecke kann von keinem – falls die Triangulierung nur aus einem Dreieck besteht – bis zu allen Dreiecken im Extremfall reichen. In der Praxis der Triangulierung von Gesichtsoberflächen wird im Allgemeinen eine obere Grenze angegeben werden können, so daß in einer vereinfachten Annahme diese mit einer ausreichend großen Konstante begrenzt werden kann. Diese Praxis wird hier und im folgenden als Regelfall oder Bestcase bezeichnet, da der Regelfall auch gleichzeitig den bestmöglichen Fall darstellt. Die Komplexität dieses Teilproblems liegt bei  $O(t)$  bzw. im beschriebenen Extremfall (Worstcase) bei  $O(t^2)$ .

(3) Konstruktion der Splineoberfläche aus der Triangulierung.

Wie in Abschnitt 2.3 beschrieben, teilt sich dieses Problem in vier Teilprobleme:

- Konstruktion der Parametrisierung:  
Bei der Parametrisierung werden zu jedem Punkt der Triangulierung zwei weitere Punkte konstruiert. Dieser Vorgang ist – analog zu (2) – sehr stark von der Nachbarschaftsstruktur der Triangulierung abhängig. Damit ergibt sich eine Komplexität, die im Regelfall mit  $O(p)$ , im Extremfall mit  $O(p^2)$  angegeben werden kann.
- Konstruktion der Auswertungspunkte:  
Abhängig von der angegebenen Genauigkeit  $n$  wird die Anzahl der Auswertungspunkte pro Dreieck der Triangulierung angegeben. Die genaue Anzahl berechnet sich aus  $\frac{n \cdot (n+1)}{2}$  und ergibt somit eine quadratische Komplexität  $O(n^2)$ .
- Auswertung der DMS-Splinebasisfunktionen:  
Es werden bei quadratischen DMS-Splines sechs Splinebasisfunktionen pro Auswertungspunkt bestimmt. Alle Dreiecke in der Nachbarschaft<sup>1</sup> zu diesem Auswertungspunkt können einen Beitrag liefern. Dies führt zur gleichen Problematik wie bereits bei der Oberflächenkonstruktion und bei der Konstruktion der Parametrisierung und ergibt damit eine Komplexität, die i.d.R.  $O(n^2 \cdot t)$ , im Extremfall bei  $O(n^2 \cdot t^2)$  beträgt.
- Konstruktion des Kontrollnetzes:  
Bei der Konstruktion des Kontrollnetzes werden eine Matrixmultiplikation zur Bestimmung der Normalengleichung bei einer Least-Square-Approximation, und die Lösung eines linearen Gleichungssystems durchgeführt. Seit Strassen 1969 bewiesen hat, daß die Matrixmultiplikation nicht zwangsläufig die Komplexität  $O(t^3)$  besitzen muß [CLR90], ist es zu einer Entwicklung gekommen, an deren aktuellem Stand Coppersmith und Winograd 1987 gezeigt haben, daß sich die Komplexität dieses Problems in bestimmten Fällen auf  $O(t^{2.376})$  reduzieren läßt. Trotzdem soll im folgenden diese Art von Problemen weiterhin als Probleme von kubischer Komplexität betrachten

---

1. In diesem Fall beschränkt sich der Nachbarschaftsbegriff nicht auf die unmittelbaren Nachbarn.

---

werden. In dem behandelten Fall ergibt sich ein lineares Gleichungssystem, dessen Größe von der Anzahl  $t$  der Dreiecke der Triangulierung abhängt. Somit ergibt sich eine Komplexität von  $O(t^3)$ .

Für die einfache Approximation durch Punktidentifikation ergibt sich nur eine Komplexität von  $O(p)$  und wird deshalb in folgenden vernachlässigt.

## A.2 Komplexität der Iterationsschritte

Ein Iterationsschritt besteht aus der Auswertung der diskreten Lagrange'schen Bewegungsgleichung (siehe Abschnitt 4.2). Dabei werden alle Punkte der Schichten des Federnetzwerkes neu bestimmt. Abhängig von der oben bereits beschriebenen Problematik der Nachbarschaftsstruktur in der Triangulierung werden die Kräfte, mit denen die Punkte aufeinander einwirken, bestimmt. Damit ergibt sich hier eine Komplexität, die im Regelfall bei  $O(p)$ , im Extremfall bei  $O(p^2)$  liegt.

Die Anzahl der Iterationsschritte bestimmt sich aus der Größe des Federnetzes und der Auslenkung der einzelnen Punkte. Eine genaue Angabe läßt sich hier nicht machen. Es sei aber darauf hingewiesen, daß die Anzahl der Schritte mit der Größe der Auslenkung steigt, jedoch mit zunehmender Größe des Federnetzes sinkt. Bei diesen Betrachtungen wird die sehr stark vereinfachende Annahme benutzt, daß jeweils nur eine konstante Anzahl von Iterationen durchgeführt wird<sup>1</sup>.

An eine Simulation schließt sich immer eine Neukonstruktion der Polygon- und der Splineoberflächen an. Zur Konstruktion der Splinefläche ist es erforderlich erneut das Kontrollnetz zu approximieren. Damit ergibt sich hier wieder die Komplexität von  $O(t^3)$ .

Bei der Splineoberfläche ist die Anzahl der Dreiecke  $t' = t \cdot n^2$  deutlich höher, als bei der Polygonoberfläche. Aus diesem Grund wird nur die Komplexität der Splineoberfläche, die zwischen  $O(t \cdot n^2)$  und  $O(t^2 \cdot n^4)$  liegt, betrachtet.

## A.3 Zusammenfassung

Tabelle 4 gibt einen Überblick über die Komplexitätsbetrachtungen, die in diesem Kapitel unternommen wurden.

---

1. Tatsächlich ist es mindestens ein Iterationsschritt. Bei den beobachteten konvergierenden Simulationen ließ sich zudem für diese Anzahl immer eine obere Schranke angeben, so daß diese Annahme nicht völlig willkürlich erscheint.

Alle Einzeloperationen wurden in den beiden betrachteten Bereichen zusammenfügt und –

	Initialisierung	Iterationsschritt
Regelfall/Bestcase	$O(t \cdot (p + n^2 + t^2))$	$O(p^2 + t^3 + t \cdot n^2)$
Extremfall/Worstcase	$O(t \cdot (t + p + n^2 \cdot t + t^3) + p^2)$	$O(p^2 + t^3 + t^2 \cdot n^4)$

Tabelle 4: Überblick Komplexität der Algorithmen

entsprechend der Komplexitätsarithmetik [CLR90] – vereinfacht.

## B Veränderung einer Oberfläche – ein Leitfaden

Um eine Oberfläche zu verformen, muß in dem vorgestellten Modell eine Reihe von Eckpunkten der Knochenschicht verschoben werden (siehe auch Abschnitt 5.2.1). In diesem Abschnitt wird an einem kleinen Beispiel das Vorgehen demonstriert.

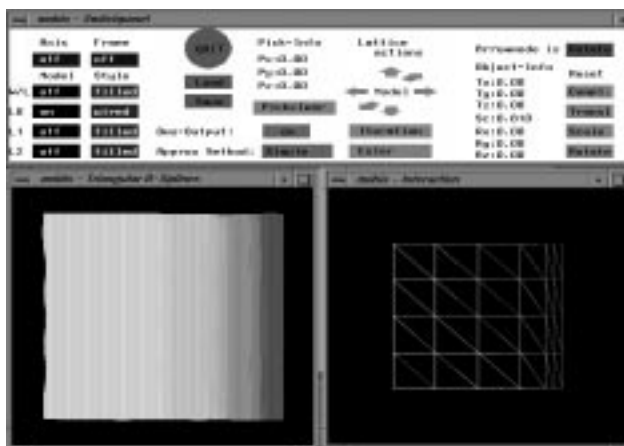


Abb. A: Die Benutzeroberfläche

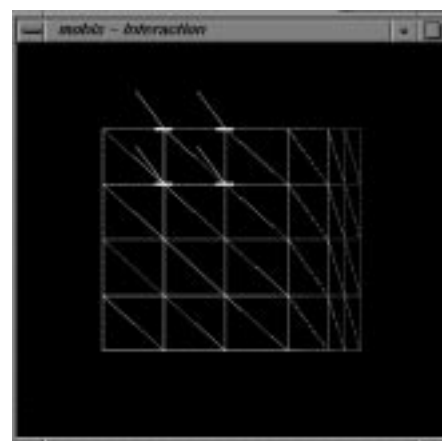


Abb. B: Auslenkungsvektoren

Als Beispiel wurde ein Halbzylinder gewählt (Abbildung A). Im ersten Schritt werden in der Knochenschicht vier Punkte ausgewählt und deren Translationsvektor bestimmt (Abbildung B). Abbildung C zeigt die veränderte Oberfläche nach vier Bearbeitungsschritten, während derer der gleiche Translationsvektor auf die vier Punkte angewendet wurde. Rechts ist die veränderte Knochenschicht abgebildet – nur die vier Punkte sind verschoben – und links die daraus erzeugte DMS-Splineoberfläche – die Punkte der linken Hälfte sind verändert.

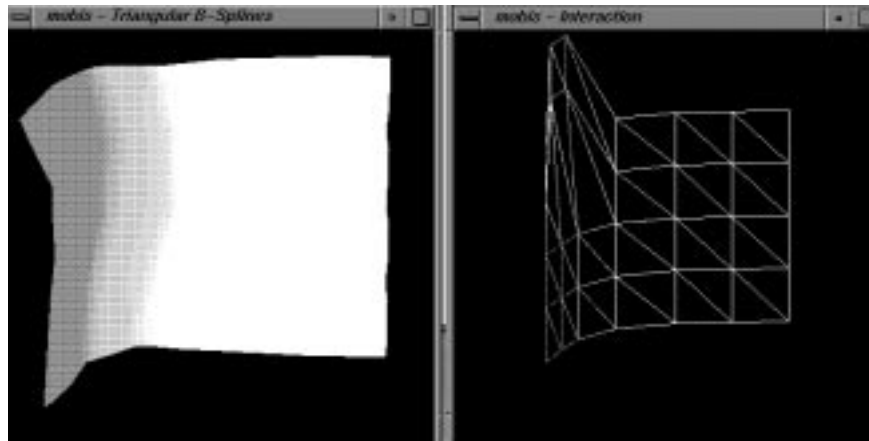


Abb. C: Veränderte Oberfläche (rotiert)

## C Implementierungsübersicht

Im Rahmen der Studienarbeit wurde eine Reihe von Programmen entwickelt. Im folgenden werden die einzelnen Programme kurz vorgestellt. Im zweiten Teil wird die Klassenbeschreibung des Programms zur Gewebemodellierung und -visualisierung vorgestellt.

### C.1 Übersicht über die entwickelten Programme

- ilm: Programm zur Visualisierung von Oberflächen. Im Unterschied zu dem MovieByu-Format wird zu jeder Punktkoordinate noch der Normalenvektor angegeben, da bei großen Dateien viel Zeit auf die Bestimmung von Nachbarschaftsbeziehungen verwendet wird, die zur Normalenbestimmung nötig sind.
- limagic: Programm zur Visualisierung von Oberflächen, die in einem GNUPLOT-Format vorliegen.
- ivbyu: Programm zur Konvertierung von Daten im Inventorformat in das MovieByu-Format. Nichtverwendete Punkte werden ausgefiltert.
- rtopol: Programm zur Korrektur von Topologien. Wenn zwei verschiedene Punkte die gleiche Koordinate besitzen, so ist diese Singularität ein Hinweis auf Fehler in der Topologie. Diese Phänomene werden von RTOPOL korrigiert.
- mobis: Programm zur Gesichtsgewebemodellierung und -visualisierung durch DMS-Splines. Das Programm verarbeitet Triangulierungen, die im MovieByu-Format vorliegen. Als Ausgabe produziert das Programm eine Datenmenge, deren an das MovieByu-Format anlehntes Format mit ILM verarbeitet werden kann.
- readtst: Programm zur Gesichtsgewebemodellierung und -visualisierung durch eine

---

polygonale Darstellung. Aus- und Eingabedateien analog zu MOBIS.

## C.2 Klassenbeschreibung zu MOBIS

Die implementierten Module teilen sich im wesentlichen in drei Sorten auf:

- Datentypen
- Funktionsbibliotheken
- Kommunizierende Objekte

### 1. Datentypen:

- vector: Vektoren des  $IR^3$ , mit einigen Operationen auf diesem Datentyp (Addition, Subtraktion, Skalarmultiplikation, etc.)
- ulist: Auswertungspunkte der DMS-Splineoberfläche mit den zugehörigen abhängigen Dreiecken. Die Funktionalität entspricht der einer Reihung, mit zusätzlichen Funktionen einer Liste.
- tlist: Liste der abhängigen Dreiecke und echter Teildatentyp von ULIST. Die Funktionalität entspricht der einer Liste.
- idx\_list: Reihung für Integerzahlen, enthält auch Teile der Funktionalität einer Liste.
- liste: Liste von Integerzahlen mit erweiterter Funktionalität.

### 2. Funktionsbibliotheken:

- GL: Funktionen zum Öffnen von Fenstern, etc.
- body: Funktionen zur Verwaltung von Ereignissen, des Interaktionsfensters, der Beleuchtung, etc. Diese Bibliothek wird von MOBIS und READTST verwendet.
- lsapprox: Funktionen zur Lösung eines Least-Square-Problems. Die Funktionen sind eine C++-Anpassung von Funktionen aus [PTVF92].
- toolbox: Eine kleine Sammlung nützlicher Funktionen zur Fehlerbehandlung, Speicherverwaltung, etc.

### 3. Kommunizierende Objekte:

Entsprechend einem objektorientierten Programmierparadigma wurde hier das Konzept der kommunizierenden Objekte verwendet. In diesem Konzept wird von einer Menge von Programmobjekten ausgegangen, die ihre Informationen mit Hilfe von Nachrichten – Aufrufen von öffentlichen Methoden – austauschen. In diesem Fall umfaßt das Konzept eine Reihe von Hauptaktivitätsträgern (nicht zu verwechseln mit dem Begriff des Aktivitätsträgers in der Betriebsprogrammierung) und untergeordneten Objekten.

---

Als Hauptaktivitätsträger wurden implementiert:

- triangle: Verwaltung der Dreiecks- und (Eck-)Punktstrukturen der polygonalen und der DMS-Splinedarstellung. Dadurch umfaßt dieses Objekt auch die Funktionalität zur numerischen Simulation des Federnetzmodells.
- DMSspline: Objekt zur Bestimmung der DMS-Splinefläche an den Auswertungspunkten und ihrer Visualisierung; die Flächenkonstruktion findet in TRIANGLE statt. Dieses Objekt umfaßt darüber hinaus folgende untergeordneten Objekte:
  - parame: Bestimmung der Parametrisierung aus den polygonalen Oberflächendaten.
  - dms: Auswertung der DMS-Splinebasisfunktionen
  - Cijk: Konstruktion des Kontrollnetzes
- panel: BenutzerInneninterface
- statistics: Objekt zur zentralen Verwaltung der verschiedenen Statistiken.

---

---

## D Literaturverzeichnis

- BHS94 W. Boehm, J. Hoschek, H.-P. Seidel, "Mathematical aspects of computer aided geometric design", In: M. Artin, H. Kraft, R. Remmert (Hrsg.): , "Duration and Change - Fifty Years at Oberwolfach", S. 106 - 138, Springer Verlag, Berlin 1994.
- BS89 I. N. Bronstein, K. A. Semendjajev, "Taschenbuch der Mathematik"<sup>24</sup>, Teubner Verlagsgesellschaft, Leipzig, 1989.
- CLR90 T. H. Cormen, C. E. Leiserson, R. L. Rivest, "Introduction to Algorithms", MIT Press, Cambridge, 1990.
- DMS92 W. Dahmen, C. A. Micchelli, H.-P. Seidel, "Blossoming begets B-spline bases built better by B-patches", In: Mathematics of Computation, 1(1), S. 97-115, Juli 1992.
- EF77 P. Ekman, W. V. Friesen, "Facial Action Coding System", Consulting Psychology Press, Palo Alto, 1977.
- Fari93 G. E. Farin, "Curves and Surfaces for Computer Aided Geometric Design"<sup>3</sup>, Academic Press, Boston, 1993.
- FDFH92 J. D. Foley, A. van Dam, S. K. Feiner, J. F. Hughes, "Computer Graphics, Principles and Practice"<sup>2</sup>, Addison-Wesley, Reading, 1992.
- FB88 D. R. Forshey, R. H. Bartels, "Hierarchical B-spline refinement", In: Computer Graphics (SIGGRAPH'88), 22(4), S. 205-212, 1988.
- Fong92 P. Fong, "Shape control for multivariate B-spline surfaces over arbitrary triangulations", Master Thesis, University of Waterloo, 1992.
- FS93 P. Fong, H.-P. Seidel, "An implementation of triangular B-spline surfaces over arbitrary triangulations", In: Computer Aided Geometric Design, 10, S. 267-275, 1993.
- GKG95 S. Girod, E. Keeve, B. Girod, "Advances in interactive craniofacial surgery planning by 3D-simulation and visualization", In: Int. J. Oral Maxillofac. Surg., Vol. 24, No. 1, S. 120-123, 1995.
- Gold80 H. Goldstein, "Classical Mechanics", Addison-Wesley, Reading, 1980.
- GS93a G. Greiner, H.-P. Seidel, "Modeling with triangular B-spline", In: J. Turner, G. Allen, J. Rossignac (Hrsg.): Proc. ACM/IEEE Solid Modeling Symposium'93, S. 211-220, 1993.

- 
- GS93b G. Greiner, H.-P. Seidel, "Splines in computer graphics: Polar forms and triangular B-spline surfaces", In: Eurographics '93, State-of-the-Art-Report, 1993.
- HL89 J. Hoschek, D. Lasser, "Grundlagen der geometrischen Datenverarbeitung", Teubner Verlag, Stuttgart, 1989.
- Hoch92 M. Hoch, "Modellbasierte Animation von Gesichtsausdrücken", Diplomarbeit, Universität Erlangen IMMD IV, 1992.
- LC87 W. E. Lorensen, H. E. Cline, "Marching Cubes: a high resolution 3D surface construction algorithm", In: Computer Graphics (SIGGRAPH'87), 21(4), S. 38-44, 1987
- MTT91 N. Magnenat-Thalmann, D. Thalmann, "Computer Animation'91", Springer Verlag, Tokyo, 1991
- Park91 F. I. Parke, "Control parametrization for facial animation", In: [MTT91].
- PB81 S. M. Platt, N. I. Badler, "Animating facial expression", In: Computer Graphics (SIGGRAPH'81), 15(3), S. 245-252, 1981.
- Piep92 S. Pieper, "CAPS: Computer Aided Plastic Surgery", Dissertation MIT, Cambridge, 1992.
- PS94a R. Pfeifle, H.-P. Seidel, "Faster evaluation of quadric bivariate DMS-spline surfaces", In: Proc. of Graphics Interface'94, S. 182-189, 1994.
- PS94b R. Pfeifle, H.-P. Seidel, "Fitting triangular B-splines to functional scattered data", Technical Report No. 5, Universität Erlangen IMMD IX, 1994.
- PTVF92 W. H. Press, S. Teukolsky, W. T. Vetterling, B. P. Flannery, "Numerical Recipes in C"<sup>3</sup>, Cambridge University Press, Cambridge NY, 1992.
- PZ90 W. Pschyrembel (Begr.), C. Zink (Bearb.), "Pschyrembel Klinisches Wörterbuch"<sup>256</sup>, Walter de Gruyter Verlag, Berlin, 1990.
- Rohe90 J. W. Rohen, "Funktionelle Anatomie des Menschen"<sup>6</sup>, Schattauer Verlag, Stuttgart, 1990.
- SZL92 W. J. Schroeder, J. A. Zarge, W.E. Lorensen, "Decimation of triangle meshes", In: Computer Graphics (SIGGRAPH'92), 29(2), S. 65-70, 1992
- Schw86 H. R. Schwarz, "Numerische Mathematik", Teubner Verlag, Stuttgart, 1986.

- 
- Seid91 H.-P. Seidel, "Polar forms and triangular B-Spline surfaces", Blossoming: The New Polar-Form Approach to Spline Curves and Surfaces, SIGGRAPH'91 Course Note #26, 1991.
- Seid94 H.-P. Seidel, "Geometrische Grundlagen des Computer Aided Geometric Design", In: O. Giering, J. Hoschek (Hrsg.): "Geometrie und ihre Anwendungen", S. 201-246, Hanser Verlag, München, 1994.
- ST90 R. F. Schmidt, G. Thewes, "Physiologie des Menschen"<sup>24</sup>, Springer Verlag, Berlin, 1990.
- Stro93 B. Stroustrup, "The C++ programming language"<sup>2</sup>, Addison-Wesley, Reading, 1993.
- TF88a D. Terzopoulos, K. Fleischer, "Deformable models", In: The Visual Computer, 4(6), S. 306-331, 1988.
- TF88b D. Terzopoulos, K. Fleischer, "Viscoelasticity, plasticity and fracture", In: Computer Graphics (SIGGRAPH'88), 21(4), S. 269-278, 1988.
- TW91 D. Terzopoulos, K. Waters, "Techniques for realistic facial modeling and animation", In: [MTT91] S. 59-74, 1991.
- Tram90 C. Doermer-Tramitz, "... auf den ersten Blick", Westdeutscher Verlag, Opladen, 1990.
- VMW83 M. W. Vannier, J. L. Marsh, J. O. Warren, "Three dimensional computer graphics for craniofacial surgical planning and evaluation", Computer Graphics (SIGGRAPH'83), 17(3), S. 263-273, Juli 1983
- Wate87 K. Waters, "A muscle model for animating 3D-facial expression", In: Computer Graphics (SIGGRAPH'87), 21(4), S. 17-24, Juli 1987.
- Wate92 K. Waters, "A physical model of facial tissue and muscle articulation derived from computer tomography data", In: SPIE Vol. 1808 Visualization in Biomedical Computing, S. 574 - 582, 1992.
- WT90 K. Waters, D. Terzopoulos, "A physical model of facial tissue and muscle articulation", In: Proc. of the First Conference on Visualization in Biomedical Computing, S. 77-82, Mai 1990.
- WT91 K. Waters, D. Terzopoulos, "Modelling and animating faces using scanned data", In: Journal of Visualization and Animation, 2(4), S. 123-128, Dezember 1991.

---