

Globale und lokale Optimierungsverfahren für dreidimensionale Anordnungsprobleme

Dissertation

zur

Erlangung des Doktorgrades (Dr. rer. nat.)

der

Mathematisch-Naturwissenschaftlichen Fakultät

der

Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von

Mike Schäfer

aus

Bonn-Beuel

Bonn, Juli 2002

Inhaltsverzeichnis

1	Einleitung	1
1.1	Problemklassen und Literaturhinweise	5
1.1.1	Problemklassen; Operations Research	6
1.1.2	Ingenieurwissenschaftliche Ansätze	8
2	Anordnung mittels globaler Optimierung	13
2.1	Grundidee ILP-Formulierung	14
2.2	Definition des Anordnungsproblems als ILP	16
2.3	Iso-orientierte, rektagonale Objekte	18
2.3.1	Untere Schranken; Bin-Packing	20
2.4	Beliebige Flächenwinkel; polyedrische Objekte	26
3	Kompaktierung mittels lokaler Optimierung	31
3.1	Nichtlineare Formulierung	34
3.2	Linearisierung der Formulierung	36
3.3	Konvergenzbeweis des lokalen Optimierungsverfahrens	38
3.4	Schrittweitenregulierung	42
3.5	Nichtlineare Optimierung: Verfahren von Zoutendijk	43
4	Verbindung von globaler und lokaler Optimierung	47
4.1	Äquivalenzrelation auf Distanzpolyedern	48
4.2	Herleitung der Separationsungleichung	50
4.3	Alternative Optimierungsmethoden	53
4.3.1	Constraint Programming	54
4.3.2	Simulated Annealing	57
5	Polyederseparation	59
5.1	Definition polyedrischer Objekte	60
5.2	Distanzpolyeder und Minkowski-Summe	62
5.2.1	Distanzpolyeder: algebraische Definition	63
5.2.2	Distanzpolyeder: geometrische Motivation	66
5.3	Randspur polyedrischer Objekte (\mathcal{P} -Spur, Polyhedral Tracing)	68
5.3.1	Faltungsoperation auf \mathcal{P} -Spuren	70
5.3.2	\mathcal{P} -Spuren und Schnitttest	77
5.3.3	Algebraische Definition der Faltungsoperation	78

5.4	GJK-Algorithmus: Separationsebenen	80
5.4.1	Algebraische Bestimmung des Simplex S_k	83
6	Darstellung räumlicher Drehungen	85
6.1	Algebraische Definition und Matrixdarstellung	86
6.2	Drehachse und -winkel	87
6.3	Eulersche Winkel	88
6.4	Quaternionen (Cayley-Kleinsche Parameter, \mathbb{H})	90
6.4.1	2×2 -Matrixdarstellung über \mathbb{C}	90
6.4.2	\mathbb{R}^4 -Vektordarstellung von \mathbb{H}	91
6.4.3	Implementierung der Quaternionenalgebra \mathbb{H}	94
7	Infinitesimale Drehungen	95
7.1	Hüllkörpereigenschaft infinitesimaler Rotationen	97
8	Anwendung auf industrielle Problemstellungen	117
8.1	Modellierung	118
8.2	E-Modulboxen mit Kabelraumabschätzung	119
8.2.1	Definition des Verdrahtungsgraph	121
8.2.2	Definition des Linearen Programms	123
8.2.3	Ergebnisse für E-Modul-Box SG129	125
8.3	Bauteilanordnung im PKW-Kofferraum	128
8.4	Weitere Anwendungen	130
9	Zusammenfassung und Ausblick	133
A	Definitionen und Bezeichnungen	137
B	Abschnittsübersicht nach Problemklassen	140
C	Softwarekomponenten	141

Kapitel 1

Einleitung

Thema dieser Arbeit sind automatisierte Verfahren zur Anordnung dreidimensionaler, polygonaler Objekte. Zielkriterium ist dabei eine möglichst dichte Packung der Objekte unter Berücksichtigung von vorgegebenen, problemspezifischen Randbedingungen.

Die vorliegende Dissertation ist das Ergebnis von Arbeiten, die an der GMD - Forschungszentrum Informationstechnik GmbH und am Institut für Informatik der Universität Bonn in der Forschungsgruppe von Prof. Dr. Thomas Lengauer, Ph.D. durchgeführt wurden. Der an der Universität Bonn angesiedelte Teil der Projektlaufzeit wurde vom Bundesministerium für Bildung, Forschung, Wirtschaft und Technologie im Rahmen des Programms „Mathematische Methoden in Industrie und Wirtschaft“ gefördert¹. Industrieller Kooperationspartner während der gesamten Projektlaufzeit war die DaimlerChrysler AG Stuttgart. Allen an dieser Arbeit fachlich und organisatorisch beteiligten Personen sei an dieser Stelle herzlich gedankt.

In dieser Arbeit werden zwei Klassen von Optimierungsmethoden betrachtet: *Globale Optimierungsverfahren*, mit denen die relative Anordnung von Objekten festgelegt wird (z. B. Objekt A liegt rechts/links von Objekt B; siehe Kapitel 2), und *lokale Verfahren*, mit denen die Kompaktierung einer gegebenen Ausgangsanordnung durchgeführt werden kann, wobei die relative Lage der Objekte weitgehend erhalten bleibt (siehe Kapitel 3). Für die globalen Optimierungsverfahren wird jeweils eine *diskrete Drehwinkelmenge* vorgegeben, wobei wegen des exponentiellen kombinatorischen Wachstums der Anordnungsmöglichkeiten in der praktischen Anwendung nur kleine Winkelmengen betrachtet werden (z. B. Vielfache von 90°). In den lokalen Optimierungsverfahren werden hingegen *kontinuierliche Objektdrehungen* berücksichtigt. Hierzu wurde ein von V. Milenkovic für zweidimensionale Problemstellungen vorgestelltes Verfahren auf den dreidimensionalen Bereich erweitert. In dieser Arbeit wird in Kapitel 4 zudem eine neuartige, integrierte Formulierung von globaler Optimierung mit lokal optimaler Drehwinkelanpassung hergeleitet. Der Implementierungsschritt hin zu einem praxisnahen Gesamtverfahren steht für die integrierte Formulie-

¹Die Förderung erfolgte unter dem BMBF-Förderkennzeichen 03-LE7BO1-4.

rung jedoch noch aus. Gemeinsame Basis aller Verfahren bildet die *Lineare Programmierung* zur Problemformulierung und Optimierung, so daß ausgereifte und numerisch stabile Lösungsalgorithmen zur Verfügung stehen.

Bei der algorithmischen Behandlung von Anordnungsproblemen spielen neben den gewählten Optimierungsmethoden geometrische Fragestellungen und Objektrepräsentationen eine entscheidende Rolle. Optimierung und Verfahren zur Objektseparation, d. h. zur Sicherstellung der Überschneidungsfreiheit der Objekte, sind dabei eng gekoppelt. Grundsätzlich können *primale* und *duale Verfahren zur Objektseparation* unterschieden werden (im folgenden wird auch geometrisch-primal zur Unterscheidung von primal/dual im Optimierungssinne verwendet). Bei primalen Verfahren wird Überschneidungsfreiheit a priori durch geeignete Charakterisierungen der Objektlagen gewährleistet. Duale Verfahren lassen hingegen Überschneidungen in Zwischenlösungen zu; sie benötigen daher Verfahren für den Schnittest von Objekten, um Unzulässigkeiten zu erkennen und zusätzliche Informationen zu Separationsrichtungen abzuleiten, was im allgemeinen mit einem erheblichen Rechenzeitaufwand verbunden ist. Die in dieser Arbeit beschriebenen globalen und lokalen Optimierungsverfahren verfolgen daher geometrisch-primale Strategien, bei denen Restriktionen zur Objektseparation den Kern der Problemformulierung bilden.

In Kapitel 5 werden die verwendeten geometrischen Strukturen und Verfahren zur Objektseparation eingeführt. Grundlage bilden die in Abschnitt 5.2 behandelten *Distanzpolyeder*. Bei fester Orientierung der Objekte kann Überschneidungsfreiheit damit auf die Lage eines Punktes außerhalb eines polyedrisch begrenzten Gebietes zurückgeführt werden (im 2D-Fall werden Distanzpolyeder auch als Hodograph oder No-Fit-Polygon bezeichnet). In Abschnitt 5.3 wird mit *\mathcal{P} -Spuren* ein alternatives Berechnungskonzept zu Distanzpolyedern vorgestellt. *\mathcal{P} -Spuren* weisen neben einer niedrigeren Worst-Case Beschreibungskomplexität bei nicht konvexen Polyedern auch algorithmische Vorteile für robuste und numerisch stabile Implementierungen auf. Die Verbindung von *\mathcal{P} -Spuren* und Optimierungsverfahren ist hierbei neu und ermöglicht die Formulierung von globaler Optimierung mit lokal optimaler Drehwinkelanpassung (wesentlich sind hierbei die zur Definition von *\mathcal{P} -Spuren* eingeführten Winkelbereiche von Ecken, Kanten und Facetten polyedrischer Objekte). Speziell für die lokale Optimierung mit kontinuierlichen Objektdrehungen ist der in Abschnitt 5.4 vorgestellte *GJK-Algorithmus* geeignet. Das vollständige Distanzpolyeder wird hierbei nur durch einen Simplex approximiert (die separierende Facette ist jeweils enthalten), der nach jedem Optimierungsschritt effizient an die neuen Objektorientierungen angepaßt werden kann.

Objektorientierungen bzw. räumliche Drehungen müssen in einer für die Koordinatentransformation und Parameteroptimierung geeigneten Weise dargestellt werden; Kapitel 6 gibt einen Überblick der verschiedenen Darstellungsformen. Die in Abschnitt 6.4 vorgestellte *Quaternionendarstellung*

erlaubt z. B. eine sehr effiziente Implementierung der Koordinatentransformation und wird daher im Bereich Computergraphik/Animation vielfach eingesetzt. Die Parameterabhängigkeiten innerhalb von Rotationsabbildungen sind jedoch quadratischer Natur und somit nicht unmittelbar für die Methoden der Linearen Optimierung zugänglich. In Kapitel 7 wird mit *infinitesimalen Rotationen* eine Linearisierung für die Optimierung der Rotationsparameter eingeführt (dieses Konzept wird auch in der Physik, insbesondere der klassischen Mechanik, verwendet). In Abschnitt 7.1 wird für diese Linearisierung eine Hüllkörpereigenschaft hergeleitet, was aus theoretischer Sicht das Hauptresultat dieser Arbeit bildet: Sind die durch die Linearisierung verzerrten Hüllkörper überschneidungsfrei angeordnet, so sind es auch die darin eingebetteten Ursprungsobjekte. Die Hüllkörpereigenschaft geht auch in den Konvergenzbeweis für das lokale Optimierungsverfahren in Abschnitt 3.3 ein, da eine explizite Schrittweitensteuerung bzgl. der Rotationssterme entfallen kann.

Der Aufbau dieser Arbeit folgt einer thematischen Gliederung in Optimierungsmethoden (Kapitel 2, 3, 4), geometrische Grundoperationen (Kapitel 5, 6, 7) und Anwendung der Verfahren auf Testinstanzen (Kapitel 8). Dies ermöglicht eine geschlossene Darstellung der einzelnen Gebiete; beim ersten Lesen ergibt sich jedoch eine Vorwärtsreferenz von Kapitel 4 auf 5.3 (die Lesereihenfolge sollte ggf. entsprechend angepaßt werden). Alternativ hierzu können die geometrischen Problemstellungen schwerpunktmäßig einem der Optimierungsverfahren zugeordnet werden:

- Für die globale Optimierung in Kapitel 2 wird das Konzept des Distanzpolyeders aus Kapitel 5 zur Diskretisierung des Anordnungsraumes benötigt. Eine Alternative hierzu bilden die in Abschnitt 5.3 behandelten \mathcal{P} -Spuren polyedrischer Objekte.
- Für die lokale Optimierung in Kapitel 3 bilden infinitesimale Drehungen aus Kapitel 7 die Basis für eine Linearisierung der Problemformulierung. Für die Anpassung von Separationsebenen wird der GJK-Algorithmus aus Abschnitt 5.4 verwendet.
- Die in Kapitel 4 benutzte Einteilung von Distanzpolyedern in topologische Äquivalenzklassen stützt sich auf das Konzept von Winkelbereichen, die in Abschnitt 5.3 mit der Definition von \mathcal{P} -Spuren eingeführt werden.

Diese Querbezüge zwischen den Kapiteln sind in Abbildung 1.1 auf Seite 4 zusammengefaßt. Im Zentrum der Abbildung sind auf der Vertikalen die Kapitel mit Optimierungsschwerpunkt aufgetragen (globale und lokale Optimierung, sowie kombinierte Formulierung); orthogonal dazu die geometrischen Schwerpunkte (Polyederseparation und Infinitesimale Drehungen). Die oben aufgeführten Bezüge sind mit Pfeilen dargestellt.

Oberhalb und unterhalb des Zentrums sind zudem die wesentlichen methodischen Unterscheidungen innerhalb der globalen bzw. lokalen Optimierung dargestellt.

Für die globale Optimierung bilden rektogonale Objekte mit achsenparallelen Kanten und Flächen einen Spezialfall, der mittels Anordnungsgraphen und deren Erweiterung zu Anordnungs-klassen gelöst werden kann. Beliebige polyedrische Objekte bilden die zweite Klasse, wobei hier hierarchische Objektdarstellungen verwendet werden.

Bei der lokalen Optimierung kann die Problemformulierung entweder direkt mit Verfahren der nichtlinearen Optimierung behandelt werden (siehe 3.5, Verfahren von Zoutendijk), oder zunächst eine von V. Milenkovic für den zweidimensionalen Fall eingeführte linearisierte Formulierung betrachtet werden (siehe 3.2). Die Linearisierung wird in dieser Arbeit auf den dreidimensionalen Fall übertragen und in Abschnitt 3.3 wird ein Konvergenzbeweis für dieses Verfahren angegeben. Für die Schrittweitenregulierung ergeben sich dabei die dargestellten Bezüge zu den Abschnitten 3.5 (insbesondere Def. 6: verbessernde, zulässige Richtung) und 7.1 (Hüllkörper, daher keine Schrittweitensteuerung notwendig).

Den Anwendungshintergrund bilden Problemstellungen, die im Konstruktionsprozeß von Kraftfahrzeugen gelöst werden müssen. Diese ergeben sich zum einen bei der Neukonstruktion, zum anderen jedoch auch im Rahmen der Modellpflege innerhalb des Produktzyklus einer Fahrzeugserie. Für die Lösung der Anordnungsprobleme bedeutet dies eine Unterscheidung zwischen frei zu gestaltenden und vorgegebenen Bauräumen, was die Wahl der eingesetzten mathematischen Optimierungsmethoden beeinflusst.

Für die in Abschnitt 8.2 betrachtete Anordnung von Modulen der Fahrzeugelektronik mit Verkabelung muß zum Beispiel zunächst eine relative Lage der einzelnen Module zueinander festgelegt werden, bevor eine Kompaktierung erfolgen kann. Hierfür werden die Methoden der *globalen Optimierung* eingesetzt und um eine Kabelraumabschätzung erweitert.

Ein Anwendungsszenario für das in dieser Arbeit entwickelte *lokale Optimierungsverfahren* ist das Einpassen eines neuen Bauteils in eine bestehende Umgebung, wobei kontinuierliche Objektdrehungen betrachtet werden. Hierbei ist eine Ausgangsanordnung der Objekte vorgegeben, die durch eine Folge von Lageänderungen schrittweise verbessert wird.

Für die verschiedenen Problemklassen sind in Anhang B die zugehörigen Abschnitte der Arbeit aufgeführt, in denen die Grundlagen der Optimierungsverfahren eingeführt werden. Eine Übersicht der eingesetzten Softwarekomponenten findet sich in Anhang C.

1.1 Problemklassen und Literaturhinweise

In diesem Abschnitt soll eine Einordnung und Abgrenzung der in dieser Arbeit behandelten Optimierungsverfahren zu bereits aus der Literatur be-

kannten Verfahren erfolgen. Anordnungs- und Packungsprobleme sind dort in verschiedenen Anwendungszusammenhängen bereits intensiv untersucht worden.

Zunächst wird ein kurzer Überblick zu den verschiedenen Problemklassen und charakteristischen Unterscheidungsmerkmalen gegeben, wobei die Klassifizierung den im Operations Research üblichen Bezeichnungen folgt. Die verfügbare Literatur auf diesem Gebiet ist allerdings so umfangreich, daß zudem auf die Bibliographien [SP92, DF92, BW95] und Übersichtsartikel [DD92, DD95] verwiesen wird.

In Absatz 1.1.2 werden dann Arbeiten aus dem Bereich der Ingenieurwissenschaften, hier insbesondere der mechanischen Konstruktion, näher betrachtet. Sie stellen alternative oder ergänzende Ansätze zu den in dieser Arbeit beschriebenen Methoden dar, wobei der industrielle Einsatz in allen Fällen noch aussteht.

1.1.1 Problemklassen; Operations Research

Im Fachgebiet des Operations Research ist zum Themengebiet der Verschnitt- und Packungsprobleme umfangreiche Literatur vorhanden. Ein wesentliches Klassifizierungsmerkmal der Problemstellung ist durch die *räumliche* oder auch höherdimensionale *abstrakte Dimension* gegeben.

Eine *eindimensionale* Problemstellung ist das klassische *Bin-Packing* Problem, bei dem Objekte unterschiedlicher Höhe h_i derart auf Behälter einer vorgegebenen maximalen Füllhöhe H verteilt werden sollen, so daß für jeden Behälter $\sum h_i \leq H$ gilt und die Anzahl benötigter Behälter minimiert wird. Eine untere Schranke für das Bin-Packing Problem wird im Zusammenhang mit dem globalen Optimierungsverfahren für orthogonale Objekte in Abschnitt 2.3.1 behandelt.

Sind bei einem Bin Packing Problem *viele gleichartige Objekte* gegeben, so werden diese Probleme auch als *Cutting Stock* Probleme bezeichnet (zur Typologie von Anordnungsproblemen siehe auch Dyckhoff, Finke [DF92]). Die Vorstellung geht hier von einem Lagerbestand aus (z. B. Röhren), aus dem ein Auftragsvolumen zu fertigen ist. Aus algorithmischer Sicht ist dies von Bedeutung, da in diesen Fällen sich mehrfach wiederholende Teilanordnungen gebildet werden können. In dieser Arbeit werden demgegenüber Problemstellungen mit einer geringen Objektanzahl, jedoch mit unterschiedlichen, komplex geformten Einzelobjekten betrachtet.

Sogenannte *Knapsackprobleme* ergeben sich, wenn den Objekten neben der Größe h_i (in diesem Fall üblicherweise mit v_i für Volumen bezeichnet), zusätzliche Nutzenwerte c_i zugeordnet sind. Ziel ist es, den Gesamtwert der Objekte zu maximieren, die in genau einen Behälter/Rucksack mit vorgegebenem Volumen V passen, d. h. $\max \sum c_i$ unter der Randbedingung $\sum v_i \leq V$. Es ist hierbei noch keine räumliche Verschachtelung der Objekte zu beachten; Bin Packing und Knapsack Probleme können jedoch auf

mehrdimensionale Behälter und Objekte erweitert werden.

Zweidimensionale Problemstellungen ergeben sich bei der Verschnittminimierung auf platten- oder bahnförmigem Material oder Anordnungsproblemen wie dem *Pallet Loading*, bei welchem gleichartige Objekte mit rechteckiger Grundfläche (Kartons) auf eine Palette mit vorgegebenen Abmessungen platziert werden sollen. Unter methodischen Gesichtspunkten ergibt sich hier eine grundsätzliche Unterscheidung zwischen *orthogonalen* und beliebigen *polygonalen Objektumrissen*. Eine weitere Untergliederung bei orthogonalen Packungsproblemen ergibt sich aus der Art der zulässigen Anordnungsmuster. In vielen industriellen Fertigungsprozessen sind nur sogenannte *Guillotine Schnitte* möglich, bei denen durchgehende Schnitte von einer Materialkante bis zur gegenüberliegenden Kante gefordert werden (z. B. beim Brechen von Glasplatten). Algorithmisch führt dies zu rekursiven Divide-and-Conquer Verfahren bzw. Ansätzen der dynamischen Programmierung, da Teilanordnungen entlang der Schnittkanten zusammengefügt werden können. In dieser Arbeit werden für orthogonale Problemstellungen beliebige Verschachtelungen zugelassen (*Nested Patterns*, Abschnitt 2.3); der Schwerpunkt liegt jedoch auf polyedrischen Objekten (Abschnitt 2.4, sowie Kapitel 3 und 4).

Im *dreidimensionalen* Fall wird in der Literatur bisher vorwiegend das *Container Loading* Problem betrachtet, bei dem eine Menge orthogonaler Objekte mit unterschiedlichen Abmessungen in einen ebenfalls orthogonalen, vorgegebenen Container verladen werden muß. Die Ausrichtung der Objekte ist dabei meist in einer Dimension, der Höhe („This side up“ Bedingung), vorgegeben. Die *Orientierungsmöglichkeit* der Objekte bilden ein wesentliches Unterscheidungsmerkmal von Packungsproblemen, welches die algorithmische Komplexität stark beeinflusst. Im orthogonalen Fall werden fast ausschließlich achsenparallele Ausrichtungen der Objekte betrachtet. Die in dieser Arbeit betrachteten Problemstellungen aus dem Bereich der Konstruktion enthalten hingegen Objekte mit komplexen Formen und kontinuierlichen Drehwinkelbereichen, so daß hier ein spezielles lokales Optimierungsverfahren entwickelt wurde. Spezielle Eigenschaften orthogonaler Problemstellungen können in diesem Anwendungsbereich zudem wegen der zahlreichen Randbedingungen, wie zum Beispiel der Integration von Verkabelungen (siehe Abschnitt 8.2), oft nicht direkt ausgenutzt werden.

Als Spezialfälle zwei- und dreidimensionaler Packungsprobleme sind Untersuchungen auf analytischem Wege zu *regulären Strukturen* wie Tessellierungen der Ebene oder dichteste Kugelpackungen zu sehen. Diese haben jedoch keinen direkten Bezug zur vorliegenden Arbeit.

Höherdimensionale Problemstellungen ergeben sich aus der Betrachtung *abstrakter Dimensionen*. Klassische Problemstellungen des Operations Research, wie z. B. Scheduling Probleme, lassen sich als Anordnungs- bzw. orthogonale Packungsprobleme modellieren. Die einzelnen Dimensionen entsprechen beschränkten Ressourcen, die zu jedem Zeitpunkt exklusiv, d. h.

überschneidungsfrei, einzelnen Aufgaben zugeordnet werden müssen. Zur Lösung derartiger Problemstellungen werden spezielle Verfahren verwendet, die oft heuristischer Natur sind. Im allgemeinen werden zunächst zulässige Lösungen gesucht, bevor Optimalität betrachtet wird. Neben Methoden der ganzzahligen Linearen Programmierung werden hier auch Constraint Programming Verfahren erfolgreich eingesetzt. Die in dieser Arbeit beschriebenen Methoden sind nicht speziell hierfür ausgelegt und fokussieren auf die Behandlung dreidimensionaler, geometrischer Problemstellungen.

Die oben aufgeführten Anordnungsprobleme fallen in die Komplexitätsklasse der NP-schweren Optimierungsprobleme. Für Probleme in dieser Klasse sind keine deterministischen Algorithmen bekannt, die für alle Probleminstanzen eine polynomielle Laufzeit zur Berechnung einer Optimallösung garantieren können. Die in dieser Arbeit betrachtete Variante der volumenminimalen Anordnung polyedrischer Objekte fällt ebenfalls in die Komplexitätsklasse NP und ist NP-vollständig, da die vorgestellten Varianten hierauf reduziert werden können (z. B. Bin-Packing als ebenfalls NP-vollständiges Problem). Zu nichtdeterministischen Berechnungsmodellen, der Definition der Komplexitätsklassen und des Begriffs der Reduktion siehe z. B. Garey, Johnson [GJ79].

Zusammenfassend erfolgt die Abgrenzung der in dieser Arbeit behandelten Anordnungsprobleme wie folgt:

Betrachtet werden schwerpunktmäßig polyedrische Objekte mit variabler Orientierung, wobei für die globalen Optimierungsverfahren diskrete, beschränkte Drehwinkelmengen, für die lokalen Optimierungsverfahren jedoch kontinuierliche Drehungen betrachtet werden. Die Verfahren sind für unterscheidbare, komplex geformte Objekte ausgelegt; reguläre Strukturen, Symmetrien oder wiederholbare Teilanordnungen werden nicht explizit berücksichtigt. Die betrachteten Objektanzahlen liegen für die globalen Verfahren in der Größenordnung von 25 Objekten; für die lokalen Optimierungsmethoden sind wesentlich höhere Objektanzahlen möglich (> 100).

Die vorgestellten Methoden sind nicht für spezielle orthogonale Packungsprobleme, wie etwa Guillotine Schnitte, und abstrakten Dimensionen, z. B. Scheduling Anwendungen, entwickelt. Für den Spezialfall orthogonaler Geometrien wird vielmehr die Integration einer Raumabschätzung für Verbindungselemente (z. B. Verkabelung, Klimaschächte) in ein globales Optimierungsverfahren aufgezeigt, wobei beliebige Verschachtelungen (Nested Patterns) möglich sind.

1.1.2 Ingenieurwissenschaftliche Ansätze

Zum Einsatz computergestützter Methoden in der Konstruktion sind auch im ingenieurwissenschaftlichen Bereich verschiedene Ansätze betrachtet worden.

Eine Folge von Arbeiten ist von der Gruppe von J. Cagen et al vorgestellt worden, die Varianten von Simulated Annealing als Optimierungsparadigma nutzen. Erste Ansätze, siehe [SC93, Cag94], bauen dabei auf sogenannten „Shape Grammars“ auf, die überschneidungsfreie Anordnungsvarianten charakterisieren. Die Zielfunktionswerte weisen bei Verwendung dieser Grammatiken jedoch auch bei niedrigen Temperaturparametern starke Sprünge auf, was dem Konzept von Simulated Annealing widerspricht. Daher wurde in folgenden Arbeiten [SC94, CAC97] ein Ansatz gewählt, bei dem sich die Objekte zunächst beliebig überschneiden dürfen. Mit der Verringerung des Temperaturparameters werden diese Überschneidungen dann kontinuierlich verringert, was der Vorgehensweise von [HL95] im 2D Fall entspricht. Ein erheblicher Anteil der Rechenzeiten wird dann jedoch für die Entfernung der Objektüberschneidungen aufgewendet. In [SSS97] wird daher versucht, vielversprechende Regionen des Konfigurationsraumes zu erkennen, was unter dem Begriff „Detection of Productive Search“ diskutiert wird.

Ein weiterer Ansatz zur Verbesserung der Rechenzeiten sind hierarchische Objektmodelle, die mit verschiedenen Auflösungsstufen arbeiten. In [KCR96] werden Octrees zusammen mit einem Simulated Annealing Ansatz betrachtet, wobei die Grundidee der Modellierung auch schon in [SD88] für den 2D Fall (Quadrees) benutzt wird. Eine Verfeinerung dieser Technik wird in [CDY98] vorgestellt, wobei die ursprünglich würfelförmige Raumaufteilung von Octrees durch quaderförmige Unterteilungen mit beliebigem Seitenverhältnis ersetzt wird. Ein hierarchisches Objektmodell für beliebige polyedrische Objekte wird in dieser Arbeit in Abschnitt 2.4 vorgestellt, wobei Hierarchien von Distanzpolyedern betrachtet werden, die Überschneidungsfreiheit zusichern.

Im Bereich der Konstruktion ist die Raumminimierung an mechanische Randbedingungen gekoppelt, wie z. B. das korrekte Zusammengreifen von Zahnrädern in Getrieben. Die bisher noch nicht zufriedenstellend gelöste Hauptaufgabe besteht in einer konsistenten und vollständigen Beschreibung dieser Bedingungen. Im Zusammenhang mit der Implementierung von Expertensystemen sind hier erste Ansätze zu finden [WC90, LMR86, Rau84]. [SC97] beschränkt sich zunächst auf einfache räumliche Randbedingungen, die durch eine geeignete Grammatik und eine damit verbundene Definitionssprache ausgedrückt werden können. In [SC96a] wird dies auf funktionale Zusammenhänge erweitert und unter dem Begriff „Configuration Design“ diskutiert; siehe hierzu auch [SSS97]. In diesem Zusammenhang ist auch ein Projekt des amerikanischen National Institute of Standards (NIST) zu sehen, welches den Austausch funktionaler Beschreibungen zwischen CAD-Systemen ermöglichen soll. Hier werden erweiterbare Definitionssprachen wie STEP oder XML diskutiert [SSBR99].

Ein anderer Ansatz zu dieser Thematik sind interaktive Systeme, in denen Konstrukteure durch geeignete graphische Benutzeroberflächen funktionale Zusammenhänge spezifizieren können, wie dies in [KS98] vorgestellt

wird.

Neben funktionalen Zusammenhängen sind häufig auch Verbindungen zwischen den anzuordnenden Objekten, z. B. Kabel oder Rohrleitungen, zu betrachten. In [SC96b] wird ein Simulated Annealing Ansatz für die Wegesuche vorgestellt, der in [SCW98] mit der Anordnungsoptimierung verbunden wird. In [CCD⁺96] wird ein wissensbasierter Ansatz, der vordefinierte Klassen von Verlegungsvarianten enthält, in ein Simulated Annealing Verfahren zur Anordnungsoptimierung integriert. Eine Kombination von Anordnungs- und Verbindungsoptimierung ist ebenfalls in [YC00] zu finden, wo unter dem Begriff „Pattern Search“ eine Anordnungsoptimierung vorgestellt wird, die wechselweise jeweils Translationen und Rotationen anpaßt und zudem einen Positionstausch von Objekten erlaubt, wenn sonst keine Verbesserung mehr erzielt werden kann. In dieser Arbeit wird in Abschnitt 8.2 ein Konzept zur Integration von Raumabschätzungen für Verbindungen in die auf Linearer Programmierung beruhende globale Anordnungsoptimierung vorgestellt. Innerhalb des in der Anordnungsphase reservierten Raumes wird dann in einer zweiten Phase der detaillierte Verlauf der Verbindungswege festgelegt. Durch diese Unterteilung in zwei Phasen können wesentlich mehr Anordnungsvarianten betrachtet werden, da der Rechenaufwand für die Raumabschätzung beträchtlich geringer als für die vollständige Optimierung der Verbindungswege ist.

Eine Anwendung aus dem Bereich der Architektur wird von U. Flemming und R. Woodbury in [FW95] betrachtet. Hier wird für die Raumaufteilung in Gebäudekomplexen ein auf Branch & Bound beruhender Ansatz für die Anordnung zweidimensionaler, rechteckiger Raumgeometrien verfolgt. Die Diskretisierung des Anordnungsraumes mit Hilfe von Entscheidungsvariablen für den Spezialfall rechteckiger Geometrien, erfolgt ähnlich zu den in dieser Arbeit diskutierten Methoden zur globalen Anordnungsoptimierung. Diese Modellierung wird auch in G. Scheithauer und J. Terno [ST93] beschrieben. Aus dem Architekturumfeld entstammen zudem zwei auf Expertensystemen aufbauende Ansätze zur Verlegung von Rohrleitungen und Klimaschächten [ZL91, MKW⁺86], die auch im weiteren Sinne eine Anordnungsoptimierung verfolgen.

Da globale Optimierungsverfahren wegen der hohen Komplexität der Problemstellungen oft nicht eingesetzt werden können, sind lokale Optimierungsverfahren im Bereich der Design Optimization üblich. Zielkriterium sind hier häufig Stabilitätseigenschaften von Konstruktionselementen. In Landon, Balling [UBBL88] und Sandgren, Dworak [SD88] werden zwei derartige gradientenbasierte Verfahren vorgestellt, wobei der Schwerpunkt in [SD88] auf der geometrischen Objektdarstellung mittels Quadrees liegt.

Der verstärkte Einsatz von Optimierungsmethoden im Bereich der mechanischen Konstruktion wird wesentlich von der Integration der dort auftretenden funktionalen Randbedingungen abhängen. Die Fragestellung, ob diese hinreichend gut von globalen Optimierungsmethoden abgebildet wer-

den können, erscheint noch offen. Hier sind zunächst prototypische Implementierungen für häufig auftretende Randbedingungen notwendig. Lokale Optimierungsmethoden, wie das in dieser Arbeit vorgestellte Kompaktierungsverfahren, erscheinen hier schneller umsetzbar, da wesentliche Zusammenhänge schon mit der Ausgangsanordnung vorgegeben werden können.

Kapitel 2

Anordnung mittels globaler Optimierung

In diesem Kapitel werden globale Optimierungsverfahren zur Anordnung polyedrischer Objekte vorgestellt. Zunächst werden quaderförmige und iso-orientierte, rektagonale Objekte betrachtet um das grundlegende Verfahren darzustellen. Dieses wird dann auf beliebige polyedrische Objekte erweitert, wobei das Konzept der in Kapitel 5, Abschnitt 5.2 eingeführten Distanzpolyeder von zentraler Bedeutung ist.

Als Zielfunktion wird das Volumen eines Hüllkörpers betrachtet, welches minimiert werden soll. Es können jedoch auch andere Maße optimiert werden, die sich aus den Positionen der Objekte ergeben. In Abschnitt 8.2 betrachten wir z. B. als sekundäres Zielkriterium die Länge von Kabelverbindungen zwischen elektronischen Bauelementen.

In dem hier vorgestellten globalen Optimierungsansatz werden Translationen und diskrete Objektrotationen behandelt. Wegen der exponentiellen Komplexitätssteigerung muß die Menge der zulässigen Drehwinkel jedoch auf kleine Mengen beschränkt bleiben, wie z. B. die 6 (bei unterscheidbaren Flächen 24) möglichen achsenparallelen Orientierungen von Quadern.

Kontinuierliche Objektdrehungen werden von dem in Kapitel 3 behandelten lokalen Optimierungsverfahren berücksichtigt. Eine Verbindung der beiden Verfahren wird in Kapitel 4 für den zweidimensionalen Fall vorgestellt.

Wesentliches Element der in diesem Kapitel behandelten Methodik ist die Charakterisierung überschneidungsfreier Anordnungen durch Distanzpolyeder. Überschneidungsfreiheit von Objekten mit fester Orientierung wird dabei auf die Lage eines Referenzpunktes relativ zu einem polyedrisch beschränkten Gebiet zurückgeführt, wodurch sich eine Diskretisierung des Anordnungsraumes ergibt. Dies kann durch lineare Restriktionen in Verbindung mit ganzzahligen Entscheidungsvariablen modelliert werden. Hieraus ergibt sich eine Problemformulierung als gemischt ganzzahliges Lineares Programm (ILP, für Integer Linear Program), welches mittels eines Branch & Bound Ansatzes gelöst wird.

Dieses Kapitel ist im einzelnen wie folgt gegliedert: In Abschnitt 2.1 wird das Grundprinzip eingeführt. Die formale Definition des Anordnungsproblems als ILP erfolgt in Abschnitt 2.2. In den folgenden Abschnitten 2.3 und 2.4 werden dann iso-orientierte und beliebige polyedrische Objekte getrennt behandelt, um spezielle Eigenschaften dieser Formulierungen aufzuzeigen. So zerfällt das Kompaktierungsproblem bei iso-orientierten Objekten in unabhängige Teilprobleme für die drei Koordinatenrichtungen und es können untere Schranken für Bin-Packing Probleme übertragen werden (siehe Abschnitt 2.3.1). Für polyedrische Objekte sind hingegen geeignete Objekthierarchien zu betrachten, um die exponentielle Komplexitätssteigerung des Verfahrens für praxisrelevante Datensätze handhabbar zu machen.

2.1 Grundidee ILP-Formulierung

Um die Grundidee des Verfahrens vorzustellen, betrachten wir zunächst ein einfaches zweidimensionales Beispiel. Mit dem Konzept von Minkowski-Summen bzw. Hodographen, der daraus abgeleiteten Diskretisierung des Anordnungsraumes und der Modellierung von Fallunterscheidungen durch Entscheidungsvariablen vertraute Leser können diesen Abschnitt überspringen. Die gewählten Bezeichnungen sind zu Beginn von Abschnitt 2.2 nochmals zusammengefaßt.

Es sei im folgenden eine Menge von n konvexen Polyedern $\mathcal{P} = \{P_1, \dots, P_n\}$ gegeben, wobei $i, j \in \{1, \dots, n\}$ jeweils für die Indizierung benutzt werden. Als Beispielobjekte sind in Abb. 2.1(a) ein Rechteck P_i und ein Quadrat P_j gewählt und wir stellen die Frage, wann diese Objekte überschneidungsfrei angeordnet sind. Das Quadrat P_j wird hierzu berührend um das dargestellte Rechteck P_i verschoben, wobei die Bahn eines beliebig aber fest gewählten Referenzpunktes r_j von P_j festgehalten wird. In der Abbildung ist die linke, untere Ecke des Quadrates als Referenzpunkt gewählt und die Bahn entspricht der gestrichelt dargestellten Linie. P_i, P_j sind genau dann überschneidungsfrei angeordnet, wenn der Referenzpunkt r_j außerhalb dieser Grenzlinie liegt. Die Grenzlinie entspricht dem Rand des in Abschnitt 5.2.1 definierten Distanzpolyeders D_{P_i, P_j} und ist für polyedrische Ausgangsobjekte ebenfalls ein polyedrisch beschränktes Gebiet.

Der zulässige Bereich für den Referenzpunkt r_j soll nun durch die Restriktionen eines Linearen Programms definiert werden. Da hiermit nur die Beschreibung konvexer Gebiete möglich ist, wird der Bereich außerhalb des Distanzpolyeders durch konvexe Teilgebiete überdeckt; für konvexe Objekte erzeugt dabei jede Facette des Distanzpolyeders D_{P_i, P_j} ein solches Teilgebiet. Um Mehrfachaufzählungen zu vermeiden und das Laufzeitverhalten zu verbessern, sollten die Teilgebiete disjunkt gewählt werden (dies ist jedoch keine notwendige Bedingung). In Abb. 2.1(b) ist dies für die vier relativen Lagen „oberhalb“, „rechts von“, „unterhalb“ und „links von“ dargestellt,

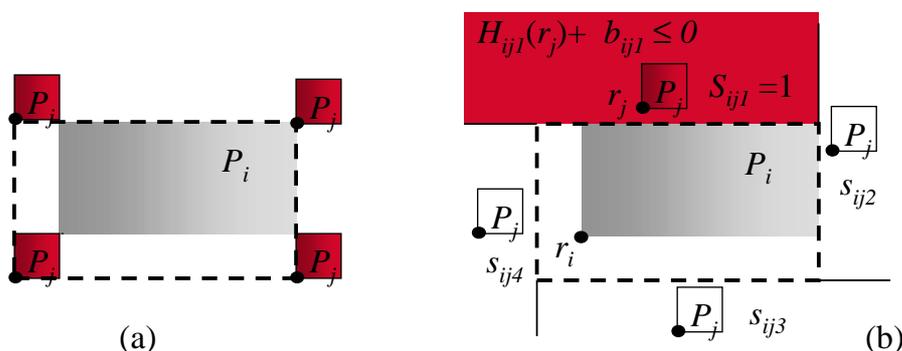


Abbildung 2.1: Überschneidungsfreie Anordnung eines Rechtecks und eines Quadrates

die mit s_{ij*} markiert sind (Separationsvariablen).

Für jedes Objektpaar P_i, P_j erhalten wir eine endliche Menge solcher Teilgebiete, deren Anzahl mit m_{ij} bezeichnet wird. Wir verwenden $k \in \{1, \dots, m_{ij}\}$ als Index. Die Auswahl eines Teilgebietes erfolgt über Entscheidungsvariablen $s_{ijk} \in \{0, 1\}$, die wie folgt definiert sind:

$$s_{ijk} = \begin{cases} 1, & \text{falls der Referenzpunkt } r_j \text{ von Objekt } P_j \text{ in Teilgebiet} \\ & k \text{ des Außengebietes um } D_{P_i, P_j} \text{ liegt} \\ 0, & \text{sonst} \end{cases} \quad (2.1)$$

Um genau eine relative Lage für jedes Polyederpaar auszuwählen, wird zudem

$$\sum_{k=1}^{m_{ij}} s_{ijk} = 1$$

gefordert. Für $s_{ijk} \in \{0, 1\}$ ist damit eine eindeutige Wahl sichergestellt.

Das konvexe Teilgebiet k zum Distanzpolyeder D_{P_i, P_j} ist durch eine Menge von Halbraumungleichungen definiert; d_{ijk} sei deren Anzahl. Daraus ergibt sich eine Restriktionsmenge $H_{ijk}x + b_{ijk} \leq \mathbf{0}$, wobei H_{ijk} eine Koeffizientenmatrix der Dimension $d_{ijk} \times 3$ und b_{ijk} der zugehörige Vektor absoluter Glieder der Dimension d_{ijk} ist. Diese Ungleichungen werden aus der Ursprungslage der Polyeder abgeleitet. Bei Translationen muß entsprechend Eigenschaft 10 des Distanzpolyeders (siehe Seite 65) die Lage des relativen Ursprungs $r_j - r_i$ betrachtet werden. Es ergeben sich damit die Ungleichungen

$$H_{ijk}(r_j - r_i) + b_{ijk} \leq \mathbf{0} . \quad (2.2)$$

In der Formulierung des Linearen Programms sollen jedoch nur diejenigen Restriktionen bindend sein, für die $s_{ijk} = 1$ gilt. Sei \mathbf{M} ein Vektor der

Dimension d_{ijk} , wobei alle Komponenten mit dem Wert einer hinreichend groß gewählten Konstante M_0 belegt seien. Die Ungleichungen (2.2) werden wie folgt erweitert:

$$H_{ijk}(r_j - r_i) + b_{ijk} \leq (1 - s_{ijk})\mathbf{M} \quad (2.3)$$

Für $s_{ijk} = 1$ ergibt sich damit 0 als rechte Seite einer Ungleichung; für $s_{ijk} = 0$ jedoch M_0 , so daß die Restriktion in diesem Fall nicht bindend werden kann.

2.2 Definition des Anordnungsproblems als ILP

In diesem Abschnitt wird nun die formale Definition des Anordnungsproblems für eine Menge \mathcal{P} konvexer Polyeder gegeben. Dieses Grundmodell wird im folgenden dann auf die Modellierung nicht konvexer Objekte erweitert.

Zunächst werden die verwendeten Bezeichnungen nochmals zusammengefaßt:

Objekte und Konstanten:

n	Anzahl konvexer Polyeder
$\mathcal{P} = \{P_1, \dots, P_n\}$	Menge konvexer Polyeder
D_{P_i, P_j}	Distanzpolyeder/Hodograph von P_i, P_j
m_{ij}	Anzahl konvexer Teilgebiete um D_{P_i, P_j}
d_{ijk}	Anzahl Begrenzungsebenen von Teilgebiet k
$H_{ijk} \in \mathbb{R}^{d_{ijk} \times 3}$	Koeffizientenmatrix zu Teilgebiet k
$b_{ijk} \in \mathbb{R}^{d_{ijk}}$	Absolute Glieder zu H_{ijk}
\mathbf{M}	Vektor mit hinreichend großer Konstante M_0

Indizes:

$1 \leq i < j \leq n$	Indizes für Polyeder(-paare)
$1 \leq k \leq m_{ij}$	Index für Teilgebiet/relative Objektlage
$d \in \{x, y, z\}$	Index für Koordinatenrichtung

Variablen:

$r_i \in \mathbb{R}^3$	Referenzpunkt von P_i
$s_{ijk} \in \{0, 1\}$	Entscheidungsvariable für Teilgebiet k von D_{P_i, P_j}
l_d	Länge der Anordnung in Richtung d

Die Formulierung des Anordnungsproblems als gemischt ganzzahliges Lineares Programm mit kontinuierlichen Vektorvariablen $r_i \in \mathbb{R}^3$ und ganzzahligen 0/1-Variablen s_{ijk} für eine Menge von n konvexen Polyedern ist damit

wie folgt gegeben ($\forall i, j, k$ mit $1 \leq i < j \leq n, 1 \leq k \leq m_{ij}$):

$$\min f(r_i) \quad (2.4)$$

mit Nebenbedingungen:

$$H_{ijk}(r_j - r_i) + b_{ijk} \leq (1 - s_{ijk})M \quad (2.5)$$

$$\sum_{k=1}^{m_{ij}} s_{ijk} = 1 \quad (2.6)$$

$$r_i \geq 0, s_{ijk} \in \{0, 1\} \quad (2.7)$$

Ein allgemeines Lösungsschema für diese Problemformulierung sind Branch & Bound Verfahren, welche in jedem Verzweigungsschritt einer Entscheidungsvariablen s_{ij} feste Werte zuordnen, die in allen Lösungen des nachfolgenden Teilbaumes enthalten sind (*Branching*). Genauer: Es wird jeweils ein Polyederpaar P_i, P_j gewählt, zu dem m_{ij} Teilbäume erzeugt werden, in denen aus der Menge der Separationsvariablen s_{ijk} ($1 \leq k \leq m_{ij}$) jeweils genau eine Variable mit 1 und alle anderen mit 0 belegt werden. Für jeden so erzeugten Knoten wird dann die LP-Relaxierung des Problems gelöst. Dies bedeutet, daß die Ganzzahligkeitsbedingung $s_{ijk} \in \{0, 1\}$ für alle noch nicht festgelegten Variablen s_{ij*} entfällt, wodurch eine untere Schranke für den in diesem Teilbaum erreichbaren Zielfunktionswert gegeben ist (*Bound*).

Für die Länge der Anordnung in eine bestimmte Koordinatenrichtung können Variablen l_d mit folgenden zusätzlichen Restriktionen eingeführt werden ($\forall i, d$ mit $1 \leq i \leq n, d \in \{x, y, z\}$):

$$(r_i)_d \leq l_d \quad (2.8)$$

Als Approximation für das Volumen des umhüllenden Quaders kann dann die lineare Zielfunktion $f(l_d) = l_x + l_y + l_z$ minimiert werden. Im folgenden Abschnitt wird zudem gezeigt, daß das Lineare Programm für iso-orientierte Objekte in drei unabhängige Teilprobleme für die einzelnen Koordinatenrichtungen zerfällt. In diesem Fall wird mit $l_x + l_y + l_z$ gleichzeitig auch das tatsächliche Volumen $l_x l_y l_z$ minimiert.

Die Modellierung nicht konvexer Objekte kann durch konvexe Überdeckungen bzw. disjunkte konvexe Zerlegungen erfolgen. Sei hierzu ein nicht konvexes Polyeder P_i durch n_i konvexe Teilpolyeder P_{i*} überdeckt. Für diese kann ein gemeinsamer Referenzpunkt r_i gewählt werden. Alternativ dazu können Referenzpunkte r_{i*} gegeben sein, deren relative Lage zueinander durch konstante Vektoren c_{i*} festgelegt ist. Diese Lage wird dann durch zusätzliche Restriktionen der folgenden Form fixiert ($\forall a, b$ mit $1 \leq a < b \leq n_i$):

$$r_{ia} = r_{ib} + c_{iab} \quad (2.9)$$

2.3 Iso-orientierte, rektagonale Objekte

In diesem Abschnitt wird die Struktur der Koeffizientenmatrix H_{ijk} ausgehend von der allgemeinen Definition des Anordnungsproblems für den speziellen Fall von iso-orientierten, rektagonalen Objekten genauer betrachtet. Es zeigt sich, daß das Lineare Programm in unabhängige Teilprobleme für jede Koordinatenrichtung zerfällt. Für vorgegebene Entscheidungsvariablen s_{ijk} können diese Teilprobleme dann durch einen Anordnungs- oder Constraint-Graph (siehe hierzu auch [Len90, KW84, Wat84, SLW83, WKC89]) dargestellt werden und die Lösung des Linearen Programms kann auf ein Kürzeste-Wege-Problem in diesem Graph zurückgeführt werden. Da die relativen Objektlagen durch die Entscheidungsvariablen s_{ijk} bereits vorgegeben sind, sprechen wir in diesem Fall nicht von einem Anordnungs- sondern von einem *Kompaktierungsproblem*.

Wir betrachten hierzu nochmals das Teilgebiet s_{ij1} in Abb. 2.1. Sei l_i die Länge von P_i . Das Gebiet ist durch die folgenden beiden Ungleichungen beschrieben:

$$(r_j)_x \leq (r_i)_x + (l_i)_x \quad (2.10)$$

$$(r_i)_y + (l_i)_y \leq (r_j)_y \quad (2.11)$$

Dies entspricht der Matrixschreibweise $H_{ij1}(r_j - r_i) + b_{ij1} \leq 0$ mit

$$H_{ij1} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \text{ und } b_{ij1} = (-(l_i)_x, (l_i)_y) . \quad (2.12)$$

Wesentlich sind die folgenden Beobachtungen: Da jede Zeile von H_{ijk} dem Normalenvektor einer Separationsebene und damit bei iso-orientierten Objekten einem der kanonischen Basisvektoren e_i oder $-e_i$ entspricht, enthält sie genau einen Koeffizienten $+1$ oder -1 . Die Ungleichungen sind also alle von der Form (2.10) bzw. (2.11), d. h. in einer Ungleichung tritt jeweils nur genau eine der Koordinaten x, y oder z auf. Die Ungleichungssysteme für die einzelnen Koordinatenachsen können daher unabhängig voneinander gelöst werden.

Im folgenden wird die Äquivalenz dieser richtungsgebundenen, eindimensionalen Kompaktierungsprobleme zu Kürzeste-Wege Problemen in gerichteten, kantengewichteten Graphen gezeigt. Mit $G_d = (V, A_d)$ sei der zu definierende Anordnungsgraph zur Koordinatenrichtung d bezeichnet. Die Knotenmenge $V = \{s, r_1, \dots, r_n, t\}$ entspricht den Referenzpunkten r_i der Quader, einem zusätzlichen Knoten s , der den linken Rand der Anordnung markiert und einem Knoten t , der dem rechten Rand zugeordnet ist (siehe Abb. 2.2).

Für jede Restriktion $(r_i)_d + c \leq (r_j)_d$ des Linearen Programms ist eine gerichtete Kante $e = (r_i, r_j)$ mit Kantengewicht $c_{ij} = c$ in A_d enthalten:

$$(r_i)_d + c \leq (r_j)_d \quad \Rightarrow \quad (r_i, r_j) \in A_d \text{ mit } c_{ij} = c \quad (2.13)$$

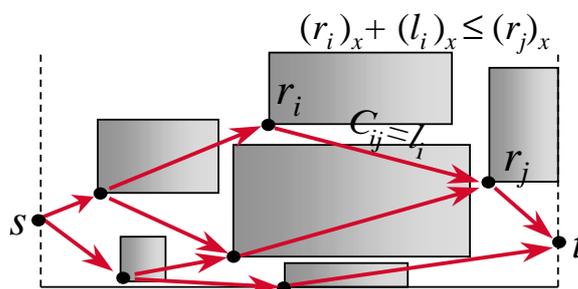


Abbildung 2.2: Anordnungsgraph $G_x = (V, A_x)$ mit Kante (r_i, r_j) induziert von Ungleichung $r_i + c \leq r_j$

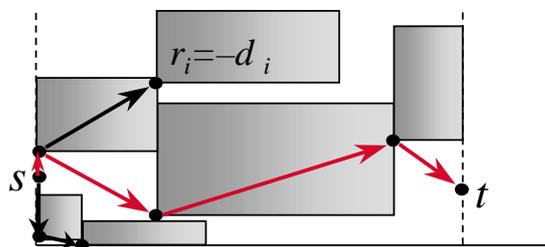


Abbildung 2.3: Minimale Koordinate $(r_i)_x$ und längste Pfade in $G_x = (V, A_x)$

Gilt $c = (l_i)_d$, so entspricht dies anschaulich der Restriktion „ P_j liegt hinter P_i “. Restriktionen in der Form $(r_j)_d \leq (r_i)_d + (l_i)_d$ (siehe z. B. Ungleichung (2.10)) werden in die Normalform $(r_j)_d - (l_i)_d \leq (r_i)_d$ überführt und erzeugen Kanten $e = (r_j, r_i)$ mit Kantengewicht $-(l_i)_d$.

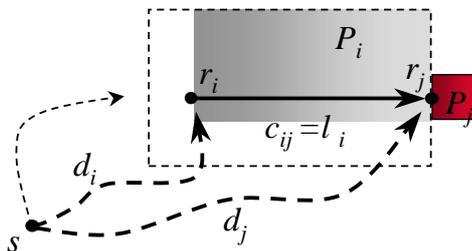
Zudem werden Kanten $e = (s, r_i)$ mit Kantengewicht $c_{si} = 0$ und Kanten $e = (r_i, t)$ mit Kantengewicht $c_{it} = (l_i)_d$ für alle r_i eingefügt. In Abb. 2.2 ist ein Anordnungsgraph beispielhaft dargestellt (die Kanten (s, r_i) und (r_i, t) sind nicht vollständig enthalten).

Die kleinstmögliche Koordinate $(r_i)_d$ entspricht dem längsten Pfad von Knoten s zu Knoten r_i und die Gesamtlänge der kürzestmöglichen Anordnung entspricht dem längsten Pfad von s nach t in G_d (in Abb. 2.3 rot hervorgehoben).

In der Praxis werden Kürzeste-Wege Algorithmen zur Suche des längsten Pfades verwendet, wobei statt der oben definierten Gewichte die negativen Kantengewichte betrachtet werden. Durch Multiplikation mit -1 ergibt sich aus (2.13) $r_i + c \leq r_j$ die folgende Ungleichung:

$$-(r_j) \leq -(r_i) - c \quad (2.14)$$

Man betrachte nun Abb. 2.4. Sei d_i die Länge eines kürzesten Weges von s zu r_i und d_j entsprechend zu r_j . Für kürzeste Wege gilt die folgende

Abbildung 2.4: Dreiecksungleichung $r_i + c_{ij} \leq r_j$

Dreiecksungleichung:

$$d_j \leq d_i + c_{ij} \quad (2.15)$$

Denn wäre $d_i + c_{ij} < d_j$, so wäre der Pfad $s \rightarrow r_i$ gefolgt von der Kante (r_i, r_j) mit Gewicht c_{ij} kürzer als d_j (im Widerspruch zur Annahme d_j kürzester Weg). Die Ungleichungssysteme (2.14) und (2.15) sind nach Substitution von $-(r_i)_d = d_i$ und $-c = c_{ij}$ offensichtlich äquivalent und mit $(r_i)_d = -d_i$ ist eine zulässige Lösung zu den ursprünglichen Restriktionen des Linearen Programms gegeben, wobei die Koordinaten $(r_i)_d$ jeweils minimal sind.

Je nach Wahl der Entscheidungsvariablen s_{ijk} in der ILP-Formulierung können im Anordnungsgraph $G_d = (V, A_d)$ gerichtete negative Kreise auftreten. Dies entspricht Variablenbelegungen, zu denen keine topologisch korrekte Anordnung der Quader existiert und das Lineare Programm unlösbar ist. Die verwendeten Kürzeste-Wege-Algorithmen müssen diese Situation erkennen.

Da die Separationsvariablen s_{ijk} inkrementell gesetzt werden, d. h. immer nur eine neue Kante in den Graph eingefügt wird, und die bisherigen Distanzwerte d_i zur Verfügung stehen, sind Bellman-Ford-artige Labeling Algorithmen günstig. Die Dreiecksungleichung muß zunächst nur für die neue Kante überprüft werden und ggf. werden Distanzaktualisierungen für die Nachfolgeknoten notwendig.

2.3.1 Untere Schranken; Bin-Packing

Im vorigen Abschnitt wurde gezeigt, wie bei vorgegebenen Entscheidungsvariablen s_{ijk} ein Anordnungsgraph aus dem Linearen Programm abgeleitet werden kann. In diesem Abschnitt werden nun Bedingungen für die Belegung der Variablen s_{ijk} abgeleitet, damit diese einer legalen, überschneidungsfreien Anordnung entsprechen. Die maximale Länge der Anordnung sei hierzu mit L_d für die einzelnen Koordinatenrichtungen vorgegeben. Es können dann untere Schranken für das eindimensionale Bin-Packing Problem übertragen werden. Die gewählte Einführung der Methodik in diesem Abschnitt orientiert sich an Fekete, Schepers [FS97b].

Wir führen zunächst ungerichtete Anordnungsclassen ein, aus denen im allgemeinen mehrere, verschieden orientierte Anordnungsgraphen abgeleitet werden können. Eine Anordnungsclassen G^C besteht aus einem ungerichteten Graph $G_d^C = (V^C, E_d^C)$ für jede Koordinatenrichtung d mit Knotenmenge $V^C = \{r_1, \dots, r_n\}$. Zu einer gegebenen Anordnung erhält man die zugehörige Anordnungsclassen wie folgt: G_d^C enthält eine Kante $e = r_i, r_j$ genau dann, wenn sich die von den Quadern P_i, P_j in der d -Koordinate überdeckten Intervalle überschneiden:

$$[(r_i)_d, (r_i + l_i)_d] \cap [(r_j)_d, (r_j + l_j)_d] \neq \emptyset \quad \Rightarrow \quad r_i, r_j \in E_d^C \quad (2.16)$$

G_d^C ist ein Intervallgraph, da die Projektion der Quader auf die d -Achse einer Konfiguration von Intervallen entspricht und eine Kante $e = r_i, r_j$ genau dann in E_d^C enthalten ist, wenn sich die Intervalle überschneiden.

Wir betrachten Abb. 2.5 mit einem zweidimensionalen Beispiel. Die Anordnungsclassen G^C ist innerhalb der Quader dargestellt, wobei die Knotenmarkierung aus Gründen der Übersichtlichkeit mit P_i erfolgt; Kanten von G_x^C sind schwarz, Kanten von G_y^C sind rot markiert. Zudem sind die entsprechenden Anordnungsgraphen G_d dargestellt, jedoch ohne die Knoten s, t . Es gilt:

$$r_i, r_j \notin G_d^C \quad \Leftrightarrow \quad (r_i, r_j) \text{ oder } (r_j, r_i) \in G_d \quad , \quad (2.17)$$

d. h. die Anordnungsgraphen G_d sind komplementär zur Anordnungsclassen G_d^C . Durch die Anordnungsgraphen wird eine transitive Orientierung der Kanten festgelegt, wobei verschiedene Orientierungen zu einer gegebenen Anordnungsclassen möglich sind. Die Abbildungen 2.6 und 2.7 zeigen jeweils alternative Orientierungen von G_x und G_y zu der in Abb. 2.5 dargestellten Classen G^C . Es werden nun die definierenden Eigenschaften der Graphen G_d^C für Anordnungsclassen eingeführt. In [FS97b] wird gezeigt, daß damit für jede transitive Orientierung der zugeordneten Anordnungsgraphen eine legale, d. h. überschneidungsfreie Anordnung existiert.

Eigenschaft 1 (Anordnungsclassen G_d^C [FS97b])

E1: G_d^C ist ein Intervallgraph

E2: $\bigcap_{d \in \{x, y, z\}} E_d^C = \emptyset$

E3: für jede stabile Menge $\{r_s \mid s \in S\} \subseteq V^C$ von G_d^C gilt: $\sum_{s \in S} (l_s)_d \leq L_d$

Eigenschaft E1 folgt direkt aus der Definition von G_d^C . Eigenschaft E2 stellt sicher, daß Quader P_i, P_j in mindestens einer Koordinatenrichtung separiert sind; für diese Richtung ist die Kante (r_i, r_j) nicht in G_d^C enthalten. Eigenschaft E3 ergibt sich aus folgender Beobachtung: Da S die Indexmenge einer stabilen Knotenmenge in G_d^C ist, dürfen sich die d -Koordinatenintervalle der

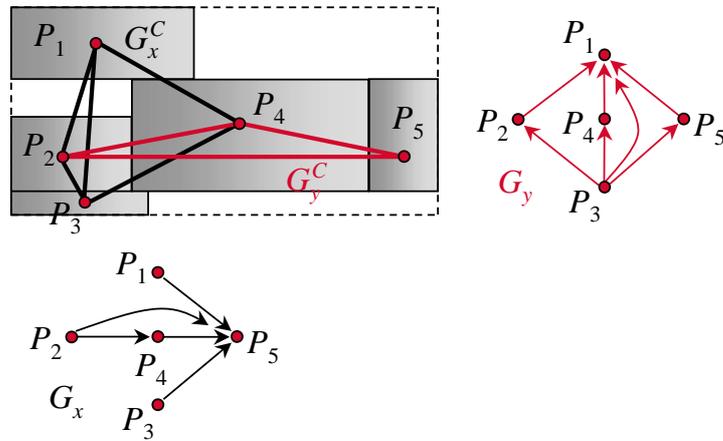


Abbildung 2.5: Anordnungsclassen $G_d^C = (V^C, E_d^C)$ und Anordnungsgraphen $G_d = (V, A_d)$; da G_d^C und G_d komplementär sind, ergeben alle schwarzen (bzw. alle roten) Kanten zusammen einen vollständigen Graph

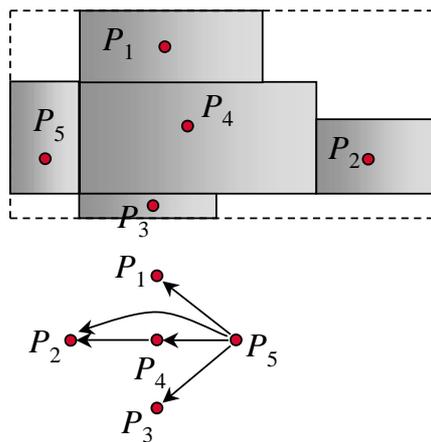


Abbildung 2.6: Alternative Orientierung von G_x mit entsprechender Anordnung

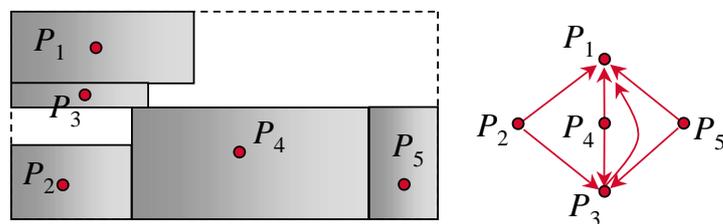


Abbildung 2.7: Alternative Orientierung von G_y mit entsprechender Anordnung

Quader $P_s (s \in S)$ nicht überschneiden, müssen also hintereinander angeordnet werden. Die Summe der einzelnen Quaderlängen $(l_s)_d$ muß daher kleiner als die vorgegebene maximale Länge L_d sein.

Hieraus läßt sich eine weitere Eigenschaft ableiten, die Grundlage für die Berechnung von unteren Schranken ist. Sei G^C eine Anordnungs-klassse und sei $\{P_s \mid s \in S\} \subseteq \mathcal{P}$ eine Teilmenge der Quader mit $\sum_{s \in S} (l_s)_d > kL_d$, d. h. einer Länge, die mehr als das k -fache der Maximallänge L_d beträgt. Dann gibt es eine zur Koordinatenrichtung d orthogonale Ebene, die mindestens $k + 1$ Quader schneidet. Die Koordinatenintervalle dieser Quader in d -Richtung überschneiden sich also und wir erhalten damit folgendes Lemma:

Lemma 2 (k-Clique in G_d^C [FS97b])

Sei G^C eine Anordnungs-klassse, $S \subseteq \{1, \dots, n\}$ eine Teilmenge der Quader mit Längen l_s , $d \in \{x, y, z\}$ eine Koordinatenrichtung und L_d die maximal zulässige Länge einer Anordnung in Richtung d . Dann gilt:

$$G_d^C \text{ enthält eine Clique der Größe } k = \left\lceil \frac{\sum_{s \in S} (l_s)_d}{L_d} \right\rceil. \quad (2.18)$$

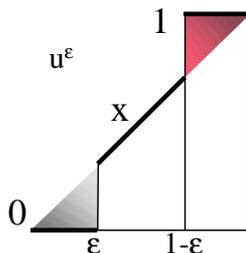
Dieses Lemma kann mit Hilfe von unteren Schranken für eindimensionale Bin-Packing Probleme verschärft werden. Es sei im folgenden die Maximallänge L_d der Anordnung auf 1 normiert und die Quaderlängen l_i seien mit dem entsprechenden Faktor skaliert. Wir betrachten die folgende Problemstellung:

Definition 1 (Bin-Packing Problem)

Gegeben seien Strecken der Länge 1, diese werden als Bins bezeichnet, und n Intervalle der Längen l_1, \dots, l_n mit $l_i \in [0, 1]$. Gesucht ist eine überschneidungsfreie Zuordnung der Intervalle zu Bins, so daß die Anzahl benötigter Bins minimal ist und die Summe der Intervalllängen je Bin jeweils kleiner gleich 1 ist.

Das Bin-Packing Problem ist NP-schwer im starken Sinne, so daß eine exakte Lösung nicht in Frage kommt. Als eines der klassischen NP-schweren Optimierungsprobleme ist es jedoch bereits intensiv untersucht worden (siehe z. B. [CGJ84]) und im folgenden wird eine effizient zu berechnende Klasse unterer Schranken für dieses Problem vorgestellt. Untere Schranken liefern eine Aussage der Form „Es werden mindestens k Bins zur überschneidungsfreien Anordnung der Intervalle benötigt“. Für die Anordnungs-klasssen G^C kann daraus dann wie in Lemma 2 eine Clique der Größe k abgeleitet werden.

Als Grundlage für die Berechnung unterer Schranken des Bin-Packing Problems betrachten wir die folgende Klasse *dual zulässiger Funktionen*:

Abbildung 2.8: Gewinn- und Verlustzonen für u^ε **Definition 2 (Dual zulässige Funktion/Dual Feasible Function)**

Sei $L := (l_1, \dots, l_n)$ eine Instanz des Bin-Packing Problems. Eine Funktion $u : [0, 1] \rightarrow [0, 1]$ heißt dual zulässige Funktion zu L , wenn für Teilmengen S der Intervalle l_i gilt:

$$\sum_{s \in S} l_s \leq 1 \quad \Rightarrow \quad \sum_{s \in S} u(l_s) \leq 1 . \quad (2.19)$$

Die Menge der dual zulässigen Funktionen ist abgeschlossen unter Konvexkombination und Verkettung. Insbesondere gilt jedoch, daß zulässige Intervallzuordnungen eines Bin-Packing Problems auch nach Transformation der Längen durch eine dual zulässige Funktion $u(l_i)$ weiterhin zulässige Anordnungen darstellen, womit sich folgendes Lemma ergibt:

Lemma 3

Sei $L := (l_1, \dots, l_n)$ eine Instanz des Bin-Packing Problems und u eine dual zulässige Funktion. Jede untere Schranke für die transformierte Problem Instanz $u(L) := (u(l_1), \dots, u(l_n))$ ist ebenfalls eine untere Schranke für L .

Wir betrachten zunächst ein einfaches Beispiel für eine dual zulässige Funktion u . Bei dieser Funktion werden alle Intervalle $l_i \leq \varepsilon$ vernachlässigt, Intervalle länger als $(1-\varepsilon)$ werden im Gegenzug jedoch auf die Maximallänge 1 vergrößert.

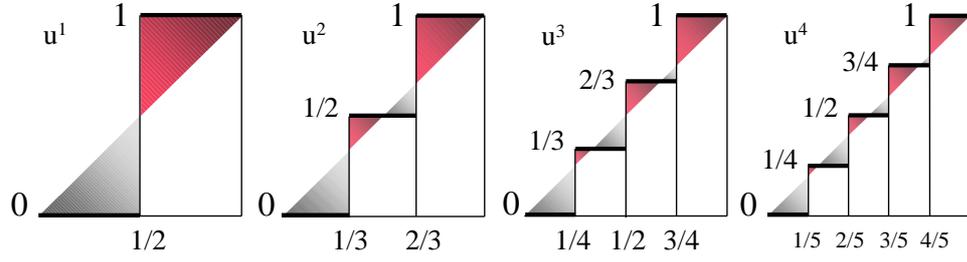
Definition 3 (Dual zulässige Funktion u^ε)

Sei $\varepsilon \in [0, \frac{1}{2}[$. Die Abbildung u^ε ist wie folgt definiert:

$$u^\varepsilon : [0, 1] \rightarrow [0, 1]$$

$$x \mapsto \begin{cases} 0, & \text{für } x \leq \varepsilon \\ x, & \text{für } \varepsilon < x < 1 - \varepsilon \\ 1, & \text{für } 1 - \varepsilon \leq x \end{cases}$$

Abbildung 2.8 verdeutlicht die Bereiche, in denen die Objekte vergrößert (roter Bereich; *Gewinnzone*, da günstig für untere Schranken) bzw. verkleinert werden (grauer Bereich; *Verlustzone*).

Abbildung 2.9: Gewinn- und Verlustzonen von u^k (für u^1, \dots, u^4)**Lemma 4 (u^ε dual zulässig)**

Die Abbildung u^ε ist eine dual zulässige Funktion.

Beweis:

Sei $S \subseteq \{1, \dots, n\}$ mit $\sum_{s \in S} l_s \leq 1$. Falls $l_s < (1 - \varepsilon) \quad \forall s \in S$, so gilt

$$\sum_{s \in S} u^\varepsilon(l_s) \leq \sum_{s \in S} l_s \leq 1.$$

Sei also $s_0 \in S$ mit $l_{s_0} \geq (1 - \varepsilon)$. Damit muß aber für alle $s \in S \setminus \{s_0\}$ gelten $l_s \leq \varepsilon$, woraus $u^\varepsilon(l_s) = 0$ folgt. Insgesamt also

$$\sum_{s \in S} u(l_s) = u(l_{s_0}) + \sum_{s \in S \setminus \{s_0\}} u(l_s) = 1 + \sum_{s \in S \setminus \{s_0\}} 0 = 1. \quad (2.20)$$

□

Ziel bei der Berechnung unterer Schranken ist es, daß möglichst viele der gegebenen Objekte innerhalb der Gewinnzonen liegen. Wir betrachten nun eine Klasse dual zulässiger Funktionen, die verschiedenen Verteilungen der Objektgrößen angepaßt werden kann und in [FS97a] eingeführt wurde; siehe dort auch für den Beweis der dualen Zulässigkeit. Sie beruht auf einer Rundungstechnik und Beispiele für die resultierende Treppenfunktion sind mit entsprechenden Gewinn- und Verlustzonen in Abb. 2.9 dargestellt.

Definition 4 (Dual zulässige Funktion u^k)

Sei $k \in \mathbb{N}$. Die Abbildung u^k ist wie folgt definiert:

$$u^k : [0, 1] \rightarrow [0, 1] \quad (2.21)$$

$$x \mapsto \begin{cases} x, & \text{für } (k+1)x \in \mathbb{N} \\ \lfloor (k+1)x \rfloor \frac{1}{k}, & \text{sonst} \end{cases} \quad (2.22)$$

Sei k_{opt} die minimale Anzahl benötigter Bins für eine Instanz $L = (l_1, \dots, l_n)$ des Bin-Packing Problems. Die triviale untere Schranke für das Bin-Packing Problem ist durch die Gesamtlänge der Intervalle l_i gegeben:

$$S_{triv}(l_i) := \left\lceil \sum_{i=1}^n l_i \right\rceil \leq k_{opt}. \quad (2.23)$$

Diese untere Schranke kann mit Hilfe der dual zulässigen Funktionen u^ε verbessert werden:

$$S_{eps}(l_i) := \max_{\varepsilon \in [0, 1/2]} \left[\sum_{i=1}^n u^\varepsilon(l_i) \right] \quad (2.24)$$

Die Schranke S_{eps} kann in einer Laufzeit von $O(n \log n)$ bestimmt werden, da für ε nur ein diskreter Wertebereich betrachtet werden muß, der sich aus Summen der Intervalllängen l_i ergibt. Durch Kombination von $S_{eps}(l_i)$ mit der dual zulässigen Funktion u^k erhalten wir:

$$S_{eps}^k(l_i) := \max_{\varepsilon \in [0, 1/2]} \left[\sum_{i=1}^n u^k(u^\varepsilon(l_i)) \right] \quad (2.25)$$

Ferner sei $S_{eps}^0(l_i) := S_{eps}(l_i)$ und damit

$$S_{eps}^K(l_i) := \max_{k \in \{0, \dots, K\}} \left[\sum_{i=1}^n u^k(u^\varepsilon(l_i)) \right]. \quad (2.26)$$

Für festes K kann $S_{eps}^K(l_i)$ ebenfalls in einer Gesamtlaufzeit von $O(n \log n)$ berechnet werden, bei vorsortierten Längen l_i sogar in Laufzeit $O(n)$.

Mit diesen Schranken kann die Menge der zu betrachtenden Anordnungs-klassen und zugehörigen Anordnungsgraphen entsprechend Lemma 2 eingeschränkt werden.

2.4 Beliebige Flächenwinkel; polyedrische Objekte

Die unteren Schranken für orthogonale Problemstellungen sind daran gebunden, daß das Lineare Programm in unabhängige eindimensionale Unterprobleme für die Koordinatenrichtungen zerfällt. Auf polyedrische Objekte mit beliebig geneigten Flächen sind diese unteren Schranken nicht übertragbar. In diesem Abschnitt wird ein alternatives Konzept eingeführt, welches die Anzahl der verschiedenen relativen Objektlagen reduziert. Die Grundidee ist, jeweils benachbarte Teilgebiete s_{ijk} zusammenzufassen, wobei die konvexe Hülle dieser Gebiete gebildet werden muß. Abb. 2.10 verdeutlicht dies für einen einfachen zweidimensionalen Fall.

Bevor wir die Zusammenfassung von Teilgebieten näher betrachten, muß zunächst eine Zerlegung des Anordnungsraumes für den dreidimensionalen Fall eingeführt werden. Im zweidimensionalen Fall ist die disjunkte Zerlegung in kanonischer Weise gegeben: Es wird jeweils die Winkelhalbierende zwischen zwei Facetten als Begrenzungslinie der Teilgebiete gewählt. Die naheliegende Erweiterung auf den dreidimensionalen Fall wäre wie folgt: Zu

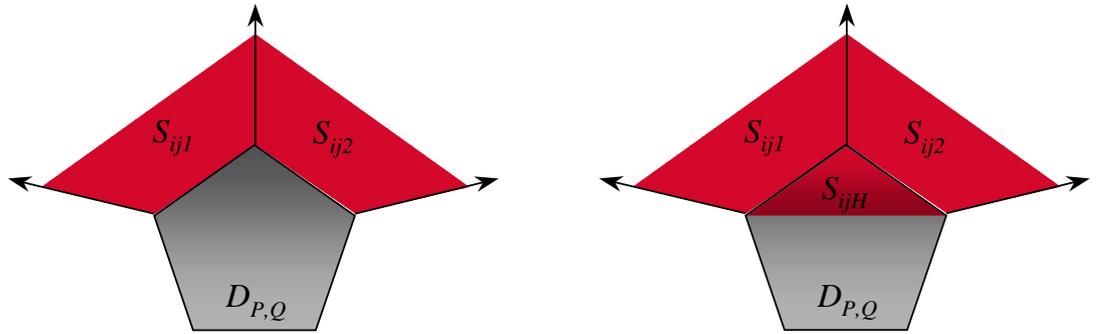


Abbildung 2.10: Zusammenfassung von Teilgebieten s_{ijk} und konvexe Hülle

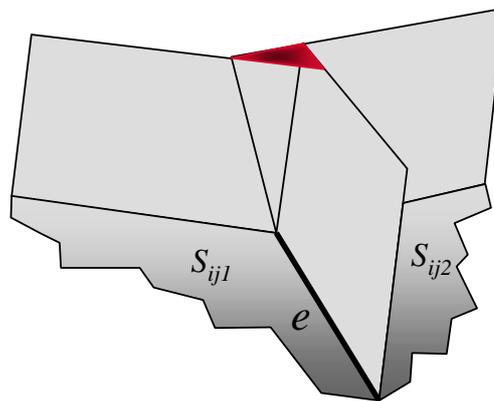


Abbildung 2.11: Disjunkte, aber nicht überdeckende Teilgebiete s_{ijk}

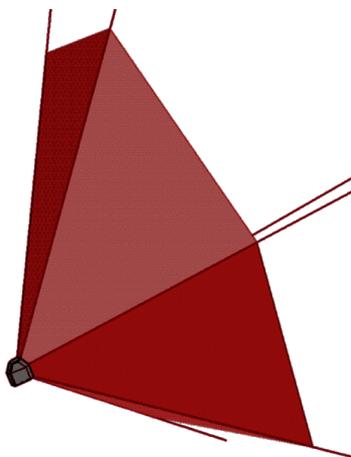


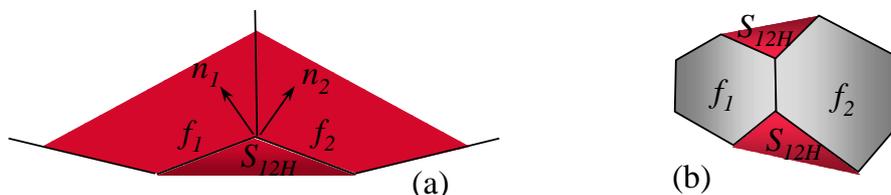
Abbildung 2.12: Extremalstrahlen und konvexe Hülle

jeder Kante e einer Facette wird jeweils die zur adjazenten Facette winkelhalbierende Ebene durch e gewählt. Dies liefert zwar disjunkte Teilgebiete, überdeckt aber nicht den gesamten Anordnungsraum, wie in Abb. 2.11 dargestellt ist. Wir betrachten hierzu eine Ecke mit drei inzidenten Kanten; durch diese Kanten ist jeweils eine Trennebene definiert. Diese Trennebenen besitzen jedoch nicht notwendigerweise eine gemeinsame Schnittgerade, so daß ein nicht überdeckter Bereich wie in der Abbildung rot dargestellt entsteht.

Die Teilgebiete werden daher wie folgt gebildet: Auf jeder Ecke v des Distanzpolyeders wird ein Extremalstrahl definiert. Als Richtungsvektor kann z. B. der Median der Normalenvektoren der inzidenten Flächen gewählt werden. Man beachte, daß die Kante e und die beiden Extremalstrahlen im allgemeinen windschief zueinander liegen. Diese Extremalstrahlen werden in einer hinreichend großen Entfernung von $D_{P,Q}$ gekappt (man betrachte $D_{P,Q}$ eingebettet in einen Ball mit hinreichend großem Radius). Das Teilgebiet ergibt sich aus der konvexen Hülle der Extremalstrahlen (siehe Abb. 2.12).

Wir betrachten nun nochmals Abb. 2.10, um die Zusammenfassung von Teilgebieten zu erläutern. Die Gebiete s_{ij1}, s_{ij2} des fünfeckigen Distanzpolyeders $D_{P,Q}$ werden zu einem Gebiet s_{ijH} verschmolzen. Das dunkelrot markierte Gebiet ergibt sich als zusätzlicher Flächenanteil durch die Hüllenbildung und liegt innerhalb des Distanzpolyeders, so daß sich bei Platzierung des Referenzpunktes von Q in diesem Bereich eine Überschneidung mit P ergibt. Dies entspricht der Relaxierung des Linearen Programms. Wir erhalten also eine untere Schranke, jedoch nicht notwendigerweise eine primal zulässige Lösung des ursprünglichen Problems.

Bei der Zusammenfassung von Teilgebieten zu Facetten f_1, f_2 des Distanzpolyeders sind zwei Zielsetzungen zu beachten:

Abbildung 2.13: Zusammenfassung von Teilgebieten s_{ijk} und konvexe Hülle

- der Schnitt des neuen Teilgebietes s_{ijH} mit dem Distanzpolyeder sollte so klein wie möglich sein, da Lösungen in diesem Bereich Überschneidungen enthalten und nicht zulässig sind (Abb. 2.13(a))
- die Differenz zwischen dem Volumen der konvexen Hülle und der Summe der Einzelvolumina der Teilgebiete¹ sollte so gering wie möglich sein, um die Überdeckung des Anordnungsraumes möglichst disjunkt zu halten und somit ineffiziente Mehrfachaufzählungen von Anordnungsvarianten zu vermeiden (Abb. 2.13(b))

Diese beiden Ziele werden in der vorliegenden Implementierung heuristisch verfolgt. Eine geringe Überschneidung der Hülle mit dem Distanzpolyeder ist dann zu erwarten, wenn der Winkel zwischen den Normalenvektoren der zugehörigen Flächen klein ist. Für die Abweichung des Gesamtvolumens von der Summe der Einzelvolumina wird die Projektion der beteiligten Facetten des Distanzpolyeders in eine Ebene betrachtet und die zweidimensionale Volumendifferenz zur konvexen Hülle betrachtet. Es ergibt sich eine Hierarchie vergrößerter Distanzpolyeder, die für ein dreidimensionales Beispiel in Abb. 2.14 dargestellt ist.

Die in diesem Abschnitt vorgestellte Vorgehensweise entspricht einer Vergrößerung des Distanzpolyeders $D_{P,Q}$ von außen. Es ist auch eine andere Sichtweise möglich, bei der die Polyederhierarchie von innen aufgebaut wird. Man beginnt z. B. mit einem einbeschriebenen Simplex, im dreidimensionalen Fall also einem Tetraeder, der von Ecken des Ursprungspolyeders aufgespannt wird und maximales Volumen hat. Dieser kann dann jeweils um eine Ecke des Distanzpolyeders erweitert werden, bis das gesamte Gebiet überdeckt ist. Für den zweidimensionalen Fall siehe hierzu [HL96].

¹Da die Teilgebiete unbegrenzt sind, ist das Volumen zunächst unendlich. Es wird daher der Schnitt mit einem hinreichend großen Ball betrachtet, um ein sinnvolles Maß zu erhalten. Siehe auch die Definition von Extremalstrahlen mit endlicher Länge.

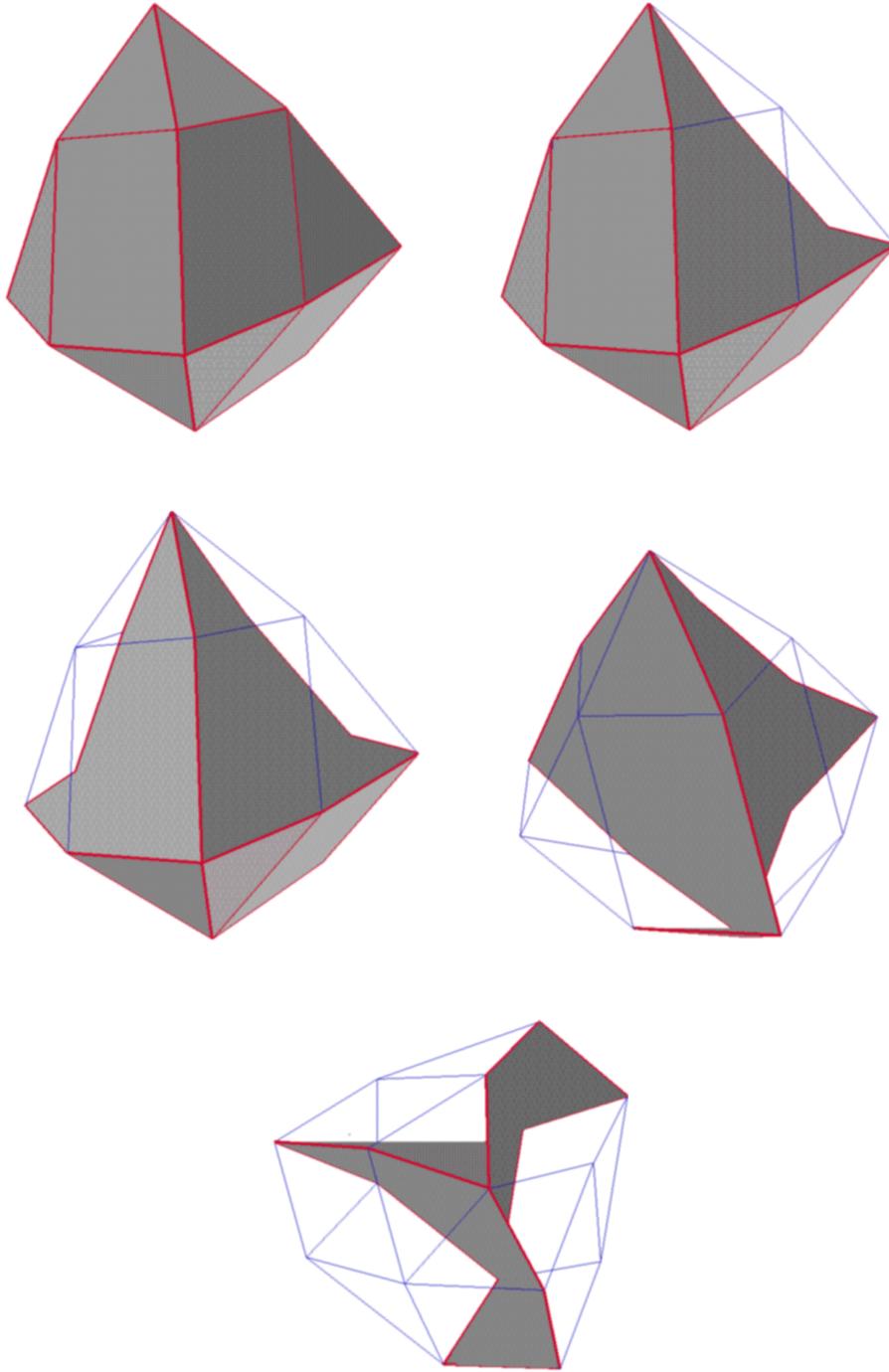


Abbildung 2.14: Hierarchie reduzierter Distanzpolyeder

Kapitel 3

Kompaktierung mittels lokaler Optimierung

In diesem Kapitel wird zunächst eine nichtlineare Formulierung für das Anordnungsproblem konvexer Polyeder unter Translation und Rotation vorgestellt. Da jedoch keine globalen Optimierungsverfahren für diese Problemklasse existieren, wird im folgenden das Kompaktierungsproblem betrachtet. Hierbei ist eine überschneidungsfreie Ausgangsanordnung der Polyeder vorgegeben, zu der dann eine lokal optimale Lösung bzw. dichte Anordnung der Polyeder erzeugt wird. Lokal optimal heißt hierbei, daß eine Verbesserung der Zielfunktion mittels kleiner Translationen und Rotationen der Objekte nicht mehr erzielt werden kann. Im Gegensatz zur globalen Optimierung bleibt die relative Lage der Objekte zueinander im allgemeinen entsprechend der Ausgangsanordnung erhalten.

In diesem Kapitel werden jeweils Mengen konvexer, polyedrischer Objekte betrachtet. Nicht polyedrische Objekte können jedoch zunächst durch geeignete Gitter polygonal approximiert werden und nicht konvexe Polyeder dann mittels geeigneter Zerlegungen bzw. Überdeckungen modelliert werden.

Es werden zwei lokale Optimierungsverfahren für das Kompaktierungsproblem vorgestellt. Das Grundprinzip beider Verfahren ist es, ausgehend von einer überschneidungsfreien Anordnung der Polyeder schrittweise verbesserte Drehwinkel und Positionen der Objekte festzulegen, bis ein lokales Optimum erreicht ist (siehe Abbildung 3.1). In jedem Optimierungsschritt werden Lineare Programme gelöst, die sich in unterschiedlicher Weise aus der nichtlinearen Formulierung des Optimierungsproblems ergeben. Beide Methoden stellen geometrisch-primale Optimierungsverfahren dar, da sie in jedem Zwischenschritt die Überschneidungsfreiheit der Lösungen erhalten.

Das erste Verfahren beruht auf Arbeiten von Victor Milenkovic [Mil98a] für zweidimensionale Problemstellungen, die auf den dreidimensionalen Fall erweitert werden. In diesem Verfahren wird die Restriktionsmenge geeignet linearisiert und in jedem Optimierungsschritt wird ein Lineares Programm gelöst, welches eine verbesserte Anordnung der Polyeder liefert. Bei Verwen-

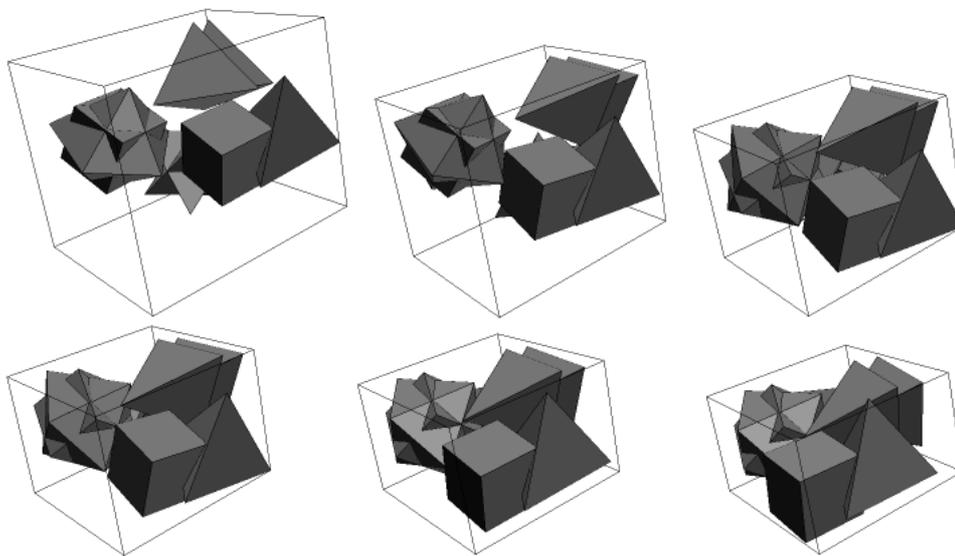


Abbildung 3.1: Sechs Schritte eines lokalen Optimierungsverfahrens

dung des Simplexalgorithmus zur Lösung der Linearen Programme steht jeweils eine zulässige Startbasis für den nächsten Optimierungsschritt zur Verfügung, was eine effiziente Implementierung des Verfahrens erlaubt. In dieser Arbeit wird zudem ein Konvergenzbeweis für dieses Optimierungsverfahren angegeben (Abschnitt 3.3), wobei die Schrittweitensteuerung aus dem zweiten Verfahren abgeleitet werden kann.

Das zweite Optimierungsverfahren ist ein Standardverfahren der nicht-linearen Optimierung, welches als Verfahren von Zoutendijk bekannt ist (siehe [BS79]). Hier werden in jedem Optimierungsschritt zwei Teilprobleme gelöst: Zunächst wird eine Optimierungsrichtung derart bestimmt, daß die Lösungen weiterhin zulässig sind und zudem die Zielfunktion verbessert wird. Dies wird durch die Betrachtung der Gradienten auf der Restriktionsmenge und des Gradienten der Zielfunktion zur aktuellen Lösung erreicht. Den zweiten Schritt bildet dann die Wahl der Schrittweite entlang der vom ersten Schritt vorgegebenen Richtung. Für die Richtungsbestimmung ergibt sich wiederum ein Lineares Programm, die Schrittweite kann dann mittels eines Line-Search Verfahrens bestimmt werden.

In beiden Fällen wird ein neuer Ansatz zur Linearisierung der Drehungsoperation benutzt. Es werden infinitesimale Rotationen betrachtet (siehe Kapitel 7), die auch in der Physik, hier insbesondere in der klassischen Mechanik, betrachtet werden. Für infinitesimale Rotationen wird in dieser Arbeit eine Hüllkörpereigenschaft hergeleitet (siehe Abschnitt 7.1, Satz 31), die überschneidungsfreie Anordnungen trotz der vorgenommenen Linearisierung sicherstellt. Für diesen Linearisierungsanteil ist daher die Zulässigkeit der Lösungen implizit gegeben und es muß insbesondere keine gesonderte

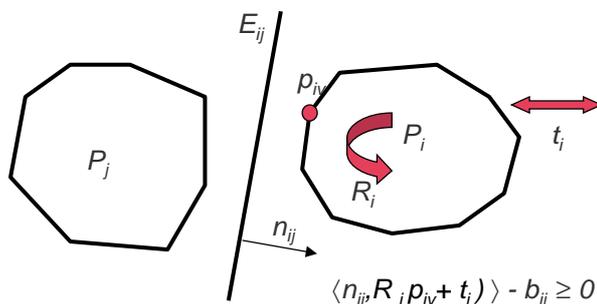
Schrittweitensteuerung erfolgen.

In der Praxis kann vollständig auf die Schrittweitensteuerung verzichtet werden. Zwar können in den Zwischenlösungen geringfügige Objektüberschneidungen auftreten, diese werden aber im folgenden Optimierungsschritt berücksichtigt und korrigiert. Durch die in der Praxis zu beobachtende Konvergenz des Verfahrens nehmen die Größe der nichtlinearen Terme und damit auch die Objektüberschneidungen im Laufe des Verfahrens ab. Zudem bewirkt die Hüllkörpereigenschaft bzw. die Vergrößerung der Objekte durch die linearisierte Rotationsoperation, daß bei größeren Drehungen, und insbesondere hier treten Überschneidungen auf, ein gewisses Spiel in den Ungleichungen entsteht. Der letzte Optimierungsschritt kann dann mit fixierten Variablen für die Separationsebenen erfolgen, wodurch die Überschneidungsfreiheit der Lösung sichergestellt ist.

Speziell zur Bestimmung von Separationsebenen innerhalb der in diesem Kapitel vorgestellten lokalen Optimierungsverfahren ist das Verfahren von Gilbert-Johnson-Keerthi geeignet (kurz GJK-Algorithmus), welches in Kapitel 5, Abschnitt 5.4 beschrieben wird. Es wird für die Erzeugung legaler Startlösungen und nach jedem Optimierungsschritt eingesetzt. Die Separationsebenen könnten zwar auch entsprechend der Variablenbelegung des Optimierungsschritts angepaßt werden; der Einsatz des GJK-Algorithmus ist jedoch aus zwei Gründen sinnvoll: Zum einen wird nach jedem Optimierungsschritt eine neue Objektlage innerhalb des linearisierten Hüllkörpers bestimmt. Zum anderen liefert der Simplexalgorithmus bindende Separationsebenen zwischen den Objekten, d. h. die Ebenen liegen an beiden Objekten an; eine symmetrische Lage mit „Spiel“ zwischen den Objekten ist jedoch für das Konvergenzverhalten des Gesamtverfahrens vorteilhafter. Der GJK-Algorithmus ist dabei selbst ein iteratives Verfahren, welches das Distanzpolyeder nur approximiert, wodurch die lokale Struktur der Lösungsfolge sehr effizient ausgenutzt wird.

Zu gradientenbasierten Ansätzen für Kompaktierungsprobleme siehe auch Stoyan [SGR⁺00, Sto83], wobei hier allerdings keine standardisierten Optimierungsmethoden zum Einsatz kommen. Ein wesentlicher Vorteil der beiden in diesem Abschnitt dargestellten Verfahren wird jedoch darin gesehen, daß ein Rückgriff auf Methoden der linearen Optimierung erfolgt, für die ausgereifte, effiziente und numerisch stabile Implementierungen verfügbar sind.

Dieses Kapitel ist wie folgt gegliedert: In Abschnitt 3.1 wird zunächst die nichtlineare Formulierung des Anordnungsproblems vorgestellt. Hiervon wird in Abschnitt 3.2 dann eine linearisierte Darstellung für das Kompaktierungsproblem abgeleitet, die als Grundlage des Verfahrens von Milenkovic dient und die in der praktischen Umsetzung des lokalen Optimierungsverfahrens verwendet wird. In Abschnitt 3.3 wird die Konvergenz dieses iterativen Lösungsverfahrens gezeigt, wobei eine maximale Schrittweite für die einzelnen Optimierungsschritte als gegeben vorausgesetzt wird. Spezi-

Abbildung 3.2: Lineare Separation zweier konvexer Polyeder P_i und P_j

elle Möglichkeiten zur Schrittweitenwahl unter dieser Problemformulierung werden in Abschnitt 3.4 vorgestellt. In Abschnitt 3.5 wird dann das alternative Verfahren von Zoutendijk betrachtet, wobei hier auch auf das allgemeine Problem der Schrittweitenwahl eingegangen wird. Für den GJK-Algorithmus zur Bestimmung von Separationsebenen wird auf Kapitel 5, Abschnitt 5.4 verwiesen.

3.1 Nichtlineare Formulierung

In diesem Abschnitt wird eine Problemformulierung für die volumenminimale Anordnung konvexer Polyeder mit kubischer Zielfunktion und nichtlinearen Nebenbedingungen angegeben, wobei kontinuierliche Objektdrehungen berücksichtigt werden. Die Grundlage für die Formulierung bildet der folgende Satz zur linearen Separierbarkeit von überschneidungsfrei angeordneten konvexen Mengen (siehe [SW70]):

Satz 5 (Lineare Separierbarkeit [SW70])

Zwei Punktmengen sind linear separierbar genau dann, wenn ihre konvexen Hüllen sich nicht schneiden.

Insbesondere sind also zwei konvexe Polyeder P_i, P_j genau dann überschneidungsfrei angeordnet, wenn es eine Ebene E_{ij} mit Ebenengleichung $\langle n_{ij}, x \rangle - b_{ij} = 0$ gibt, so daß P_i bezüglich des Normalenvektors n_{ij} vollständig im positiven Halbraum, P_j jedoch vollständig im negativen Halbraum liegt (siehe Abb. 3.2).

Für eine Menge konvexer Polyeder P_1, \dots, P_k mit festen Positionen kann dies durch ein Ungleichungssystem mit Vektorvariablen $n_{ij} \in \mathbb{R}^3$ und skalaren Variablen $b_{ij} \in \mathbb{R}$ ($1 \leq i \neq j \leq k$) wie folgt ausgedrückt werden:

$$\langle n_{ij}, p \rangle - b_{ij} \geq 0 \quad \forall p \in P_i \quad (3.1)$$

$$n_{ij} = -n_{ji}, \quad b_{ij} = -b_{ji} \quad (3.2)$$

$$\|n_{ij}\| = 1 \quad (3.3)$$

Da Ungleichung (3.1) linear in p ist und die konvexe Hülle eines Polyeders von seinen Ecken aufgespannt wird, ist es ausreichend statt aller Punkte $p \in P_i$ nur die Lage der Ecken p_{iv} von P_i zu betrachten. Die Gleichungen nach (3.2) sind technischer Natur und entsprechen der Lage des Polyeders im positiven bzw. negativen Halbraum. Gleichung (3.3) ist nur notwendig, um die triviale Lösung $n_{ij} = \mathbf{0}, b_{ij} = 0 \quad \forall i, j$ auszuschließen. P_1, \dots, P_k sind genau dann überschneidungsfrei angeordnet, wenn für das Ungleichungssystem (3.1)-(3.3) eine Lösung existiert.

In der bisherigen Formulierung ist die Position der Polyeder fest vorgegeben. Wir werden nun Ungleichung (3.1) um Terme erweitern, die Translationen und Rotationen der Polyeder entsprechen. Für den Translationsanteil wird eine Vektorvariable $t_i \in \mathbb{R}^3$ eingeführt und für die Rotation eine Matrixvariable $R_i \in SO(3) \subset \mathbb{R}^{3 \times 3}$, die einer Rotationsmatrix entspricht. Die Orthogonalitätsforderung $R_i \in SO(3)$ läßt sich auf quadratische Terme zurückführen (man betrachte hierzu die Quaternionen-Darstellung von Rotationen in Abschnitt 6.4). Da im folgenden jedoch infinitesimale Drehungen mit drei unabhängigen Parametern betrachtet werden sollen, wird auf eine explizite Ausformulierung der Orthogonalitätsbedingungen an dieser Stelle verzichtet. Damit ergibt sich die folgende Separationsungleichung mit t_i für die Translation und R_i für die Rotation:

$$\begin{aligned} \langle n_{ij}, R_i p_{iv} + t_i \rangle - b_{ij} &\geq 0 && \forall p_{iv} \text{ Ecke von } P_i \\ R_i &\in SO(3) \end{aligned} \quad (3.4)$$

Man beachte, daß sich in diesen Ungleichungen nun quadratische Terme in den Variablen n_{ij}, R_i und n_{ij}, t_i ergeben¹. Die Vektoren p_{iv} sind hingegen konstant und geben die beliebig, aber fest gewählte Ausgangslage der Eckpunkte an. Aufgrund der unbeschränkten Translationsvariablen existiert für das Ungleichungssystem nach (3.4) immer eine Lösung.

Zur Formulierung der Zielfunktion benötigen wir nun noch eine Variable für die Ausdehnung der Anordnung in x, y und z -Richtung. Für diese Längen führen wir eine Variable $l \in \mathbb{R}^3$ ein, wobei die Koordinaten aller Polygonecken jeweils kleiner als die entsprechende Komponente von l sein müssen. Zudem beschränken wir die Lage der Ecken und damit der gesamten Polyeder auf den positiven Oktanten:

$$0 \leq R_i p_{iv} + t_i \leq l \quad \forall p_{iv} \text{ Ecke von } P_i$$

Wir fassen die verwendeten Bezeichnungen nochmals zusammen:

¹Bem.: Quadratische Programme, kurz QP's, bezeichnen Formulierungen mit quadratischer Zielfunktion jedoch weiterhin linearen Nebenbedingungen; insbesondere ist die hier vorgestellte Formulierung also *kein* QP.

Objekte und Konstanten:

n	Anzahl konvexer Polyeder
$\mathcal{P} = \{P_1, \dots, P_n\}$	Menge konvexer Polyeder
m_i	Anzahl Ecken von Polyeder P_i
p_{iv}	Ecke/Vertex von P_i

Indizes:

$1 \leq i < j \leq n$	Indizes für Polyeder(-paare)
$d \in \{x, y, z\}$	Index für Koordinatenrichtung
$v \in \{1, \dots, m_i\}$	Index für Ecken der Polyeder

Variablen:

l_d	Länge der Anordnung in Richtung d
R_i	Rotationsmatrix aus $SO(3)$ zu Polyeder P_i
t_i	Translation von P_i
n_{ij}	Normalenvektor der Separationsebene zwischen P_i, P_j
b_{ij}	Absolute Glieder der Separationsebenen

Insgesamt ergibt sich hiermit die folgende nichtlineare Formulierung des Anordnungsproblems für konvexe Polyeder P_i ($1 \leq i \neq j \leq n$):

$$\min l_x l_y l_z \quad (3.5)$$

mit Nebenbedingungen:

$$0 \leq R_i p_{iv} + t_i \leq l \quad \forall p_{iv} \text{ Ecke von } P_i \quad (3.6)$$

$$\langle n_{ij}, R_i p_{iv} + t_i \rangle - b_{ij} \geq 0 \quad (3.7)$$

$$n_{ij} = -n_{ji}, \quad b_{ij} = -b_{ji} \quad (3.8)$$

$$\|n_{ij}\| = 1 \quad (3.9)$$

$$R_i \in SO(3) \quad (3.10)$$

$$R_i \in \mathbb{R}^{3 \times 3}, \quad n_{ij}, t_i, l \in \mathbb{R}^3, \quad b_{ij} \in \mathbb{R}$$

3.2 Linearisierung der Formulierung

Die Grundidee des Linearisierungsansatzes nach Milenkovic [Mil98a] ist folgende: Ausgehend von gegebenen Anfangswerten werden infinitesimale Änderungen der Variablen betrachtet. Für diese betragsmäßig sehr kleinen Werte sind die Terme höherer Ordnung vernachlässigbar und das nichtlineare Ungleichungssystem wird in einer hinreichend kleinen Umgebung der Ausgangslösung beliebig genau durch das linearisierte System approximiert.

Wir erweitern daher jede Variable x um einen differentiellen Term Δx , also z. B. $t_i \rightarrow t_i + \Delta t_i$. Man beachte, daß nunmehr nur die differentiellen Terme Δx als Variablen für den lokalen Optimierungsschritt betrachtet werden und die Werte der Variablen x aus dem vorangehenden Optimierungsschritt als konstant angesehen werden.

Die Änderung Δn des Normalenvektors n der Separationsebene muß zudem senkrecht zu n erfolgen, um eine Auslenkung im Gegensatz zu einer Streckung von n zu erreichen (Ziel: neigen der Separationsebene). Es ergeben sich hieraus zusätzliche Orthogonalitätsbedingungen der Form $\langle \Delta n_{ij}, n_{ij} \rangle = 0$. Es wird gezeigt, wie diese Bedingung formal aus der Linearisierung von Gleichung (3.9) folgt.

Für die absoluten Glieder b der Separationsebenen kann auf die Substitution $b_{ij} \rightarrow b_{ij} + \Delta b_{ij}$ verzichtet werden, da diese Variablen nicht in Termen höherer Ordnung vorkommen und unbeschränkt sind. Diese Eigenschaft kann bei einer Implementierung des Verfahrens ausgenutzt werden. Um infinitesimale Terme und Variablen jedoch einheitlich zu bezeichnen, wird die Substitution im folgenden vorgenommen.

Infinitesimale Drehungen, die eine Linearisierung orthogonaler Drehungsmatrizen darstellen, werden in Kapitel 7 formal eingeführt und ausführlich betrachtet (zur Linearisierung siehe insbesondere die Herleitung von Eigenschaft 18 infinitesimaler Rotationen über Eulersche Winkel; Seite 96). Wesentlich ist hier die Eigenschaft, daß infinitesimale Drehungen durch drei voneinander unabhängige Parameter α, β, γ gegeben sind. Damit entfallen die quadratischen Orthogonalitätsbedingungen zwischen den Matrixelementen von $R_i \in SO(3)$. In Kapitel 7, Eigenschaft 18 wird gezeigt, daß jede infinitesimale Drehung von der Form

$$\Delta R = \begin{pmatrix} 1 & \gamma & -\beta \\ -\gamma & 1 & \alpha \\ \beta & -\alpha & 1 \end{pmatrix}.$$

ist. Zudem wird gezeigt, daß die Änderung Δx eines Vektors x unter einer infinitesimalen Rotation ΔR durch das Kreuzprodukt $\Delta x = (\alpha, \beta, \gamma) \times x$ gegeben ist. Für den infinitesimalen bzw. differentiellen Drehungsanteil führen wir daher eine Variable $\Delta r_i = (\alpha, \beta, \gamma) \in \mathbb{R}^3$ ein.

Mit diesen differentiellen Anteilen und \doteq als Bezeichnung für Gleichheit unter Vernachlässigung infinitesimaler Terme höherer Ordnung, sowie $P_{iv} = R_i p_{iv} + t_i$ für den konstanten Term der Ausgangslage von Ecke p_{iv} erhalten wir aus der Separationsungleichung (3.4):

$$\begin{aligned} 0 &\leq \langle (n_{ij} + \Delta n_{ij}), (\Delta R R_i p_{iv}) + (t_i + \Delta t_i) \rangle - (b_{ij} + \Delta b_{ij}) \\ &= \langle (n_{ij} + \Delta n_{ij}), (P_{iv} + \Delta r_i \times R_i p_{iv} + \Delta t_i) \rangle - (b_{ij} + \Delta b_{ij}) \\ &= \langle n_{ij}, P_{iv} \rangle + \langle n_{ij}, \Delta r_i \times R_i p_{iv} \rangle + \langle n_{ij}, \Delta t_i \rangle - b_{ij} + \\ &\quad \langle \Delta n_{ij}, P_{iv} \rangle + \langle \Delta n_{ij}, \Delta r_i \times R_i p_{iv} \rangle + \langle \Delta n_{ij}, \Delta t_i \rangle - \Delta b_{ij} \\ &\doteq \langle n_{ij}, P_{iv} \rangle - b_{ij} + \langle \Delta n_{ij}, P_{iv} \rangle + \langle \Delta r_i, R_i p_{iv} \times n_{ij} \rangle + \langle \Delta t_i, n_{ij} \rangle - \Delta b_{ij} \end{aligned} \quad (3.11)$$

Die Orthogonalitätsbedingung $\langle \Delta n_{ij}, n_{ij} \rangle = 0$ ergibt sich aus der Linearisierung von (3.9) durch Subtraktion von 1 auf beiden Seiten der folgenden

Gleichung:

$$\begin{aligned}
1 &= \|n_{ij} + \Delta n_{ij}\| \\
&= \langle n_{ij} + \Delta n_{ij}, n_{ij} + \Delta n_{ij} \rangle \\
&= \langle n_{ij}, n_{ij} \rangle + 2\langle \Delta n_{ij}, n_{ij} \rangle + \langle \Delta n_{ij}, \Delta n_{ij} \rangle \\
&\doteq 1 + 2\langle \Delta n_{ij}, n_{ij} \rangle
\end{aligned} \tag{3.12}$$

Abschließend betrachten wir noch die Linearisierung der kubischen Zielfunktion. Als Linearisierung ergibt sich hier:

$$(l_x + \Delta l_x)(l_y + \Delta l_y)(l_z + \Delta l_z) \doteq l_x l_y l_z + l_x l_y \Delta l_z + l_x l_z \Delta l_y + l_y l_z \Delta l_x$$

Da $l_x l_y l_z$ ein konstanter Term ist, verbleibt als Zielfunktion: $\min l_x l_y \Delta l_z + l_x l_z \Delta l_y + l_y l_z \Delta l_x$. Das Lineare Programm für einen lokalen Optimierungsschritt ist damit wie folgt gegeben (Bezeichnungen der Variablen siehe Seite 36):

$$\min l_x l_y \Delta l_z + l_x l_z \Delta l_y + l_y l_z \Delta l_x \tag{3.13}$$

mit Nebenbedingungen:

$$0 \leq (\Delta r_i \times R_i p_{iv}) + (t_i + \Delta t_i) \leq (l + \Delta l) \quad \forall p_{iv} \text{ Ecke von } P_i \tag{3.14}$$

$$0 \leq \langle \Delta r_i, R_i p_{iv} \times n_{ij} \rangle + \langle \Delta t_i, n_{ij} \rangle + \langle \Delta n_{ij}, t_i \rangle - \Delta b_{ij} + \langle n_{ij}, t_i \rangle \tag{3.15}$$

$$\Delta n_{ij} = -\Delta n_{ji}, \quad \Delta b_{ij} = -\Delta b_{ji} \tag{3.16}$$

$$\langle \Delta n_{ij}, n_{ij} \rangle = 0 \tag{3.17}$$

$$\Delta n_{ij}, \Delta r_i, \Delta t_i, \Delta l \in \mathbb{R}^3, \quad \Delta b_{ij} \in \mathbb{R}$$

3.3 Konvergenzbeweis des lokalen Optimierungsverfahrens

In diesem Abschnitt wird gezeigt, daß die iterative Lösung der linearisierten Problemformulierung gegen ein lokales Optimum des Anordnungsproblems konvergiert. Der Beweis folgt dabei [BS79] Kapitel 7 Satz 7.2.3.

Hierzu definieren wir zunächst eine Iterationsabbildung \mathcal{I} , welche einer überschneidungsfreien Ausgangslösung v jeweils die Menge aller möglichen Lösungen des linearisierten Problems zuordnet, die in einem Iterations- bzw. einem lokalen Optimierungsschritt erreicht werden können. Polyeder werden hierbei als überschneidungsfrei² bezeichnet, wenn sie keine gemeinsamen inneren Punkte haben; gemeinsame Randpunkte sind wegen der nicht strikten Formulierung des Ungleichungssystems mit „ \leq “ zulässig.

Die Lösungsmenge $\mathcal{I}(v, \varepsilon_{max})$ ergibt sich aus der Variation der Schrittweite ε in einem Intervall $[0, \varepsilon_{max}]$, wobei $\varepsilon_{max} > 0$ für jede Iteration

²Bezogen auf Anordnungen werden die Begriffe *überschneidungsfrei* und *legal* synonym verwendet

so gewählt sei, daß keine Überschneidungen zwischen Objekten entstehen können. Dies ist wegen der Vernachlässigung von Termen höherer Ordnung in der linearisierten Formulierung notwendig. Auf die Bestimmung der Schrittweite wird in diesem Abschnitt nicht näher eingegangen; sie wird zunächst als gegeben vorausgesetzt. Für die Konvergenzaussage ist es jedoch insbesondere nicht notwendig ε_{max} größtmöglich zu wählen; ε_{max} muß lediglich die Legalität der im folgenden Iterationsschritt erzeugten Anordnung sicherstellen.

Sei $V_l \subset \mathbb{R}^N$ die Menge von Belegungen der Variablen für Rotationen R_i , Translationen t_i , Normalenvektoren n_{ij} und absolute Glieder b_{ij} der nichtlinearen Formulierung des Anordnungsproblems ($N = 3 \cdot (2n + n(n-1) + 1) + n(n+1)$), die legalen Lösungen entsprechen, d. h. die Ungleichungssystem (3.6)-(3.10) erfüllen. Sei zudem $\Omega_{lok} \subset V_l$ die Menge der lokalen Optima des Anordnungsproblems. Für die Variablen Δx des Linearen Programms (3.13)-(3.17) wird nun zudem die Schrittweitenbegrenzung $-\varepsilon \leq \Delta x \leq \varepsilon$ eingeführt.

Definition 5 (Iterationsabbildung \mathcal{I})

$\mathcal{I} : V_l \times \mathbb{R} \rightarrow V_l$

$\mathcal{I}(v, \varepsilon_{max}) := \{\tilde{v} \mid \tilde{v} \text{ ist zulässige Lösung des mit Ausgangslösung } v \text{ und Schrittweite } \varepsilon \in]0, \varepsilon_{max}] \text{ erzeugten Linearen Programms}\}$

Man beachte, daß \tilde{v} nur zulässige und nicht notwendigerweise Optimallösung des Linearen Programms sein muß. Sei v_0 die Variablenbelegung einer legalen Ausgangslösung. Eine Lösungsfolge $\{v_k\}$ wird dann iterativ wie folgt erzeugt:

$v_k \in \Omega_{lok}$: Stopp mit lokal optimaler Lösung.

$v_k \notin \Omega_{lok}$: Wähle beliebiges $v_{k+1} \in \mathcal{I}(v_k, \varepsilon_{max})$.

Die Zielfunktionswerte $f(v_k)$ (entsprechend $l_x l_y l_z$ aus (3.5)) sind streng monoton fallend, solange das iterative Verfahren fortgesetzt wird; d. h. es gilt $f(v_k) > f(v_{k+1})$ für alle $v_k \notin \Omega_{lok}$. Denn falls es eine Umgebung von v_k mit $f(v_{k+1}) \geq f(v_k) \quad \forall v_{k+1} \in \mathcal{I}(v_k, \varepsilon_{max})$ gäbe, so wäre v_k ein lokales Optimum im Widerspruch zu $v_k \notin \Omega_{lok}$.

Damit ist $\{v_k\}$ beschränkt, liegt also in einer kompakten Teilmenge von $V_l \subset \mathbb{R}^N$. Nach dem Satz von Bolzano-Weierstraß besitzt somit jede unendliche Folge mindestens einen Häufungspunkt.

Die Zielfunktion ist stetig über V_l , da sie ein Polynom in den Komponenten von l ist, wobei l linear von den wiederum linearen Transformationen Rotation R_i und Translation t_i der Eckpunkte der Polyeder abhängt.

Satz 6

Mit den obigen Bezeichnungen gilt:

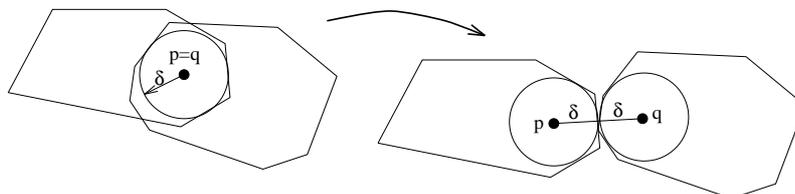


Abbildung 3.3: Minimale Separation zweier Polyeder

1. Der Grenzwert jeder konvergenten Teilfolge von $\{v_k\}$ bzw. jeder Häufungspunkt von v_k liegt in Ω_{lok} , ist also ein lokales Optimum.
2. $f(v_k) \rightarrow f(v)$ für ein $v \in \Omega_{lok}$

Beweis:

Der Beweis erfolgt in drei Schritten: Wir zeigen zunächst, daß die Menge V_l abgeschlossen ist. In Schritt 2 wird dann eine spezielle Stetigkeitseigenschaft der Iterationsabbildung \mathcal{I} hergeleitet. Die Konvergenz des Gesamtverfahrens wird dann abschließend in Schritt 3 gezeigt.

1. *Schritt:*

Wir zeigen, daß die Menge V_l abgeschlossen ist, d. h. der Grenzwert v jeder konvergenten Folge $\{v_k\}$ mit Folgengliedern aus V_l liegt ebenfalls in V_l .

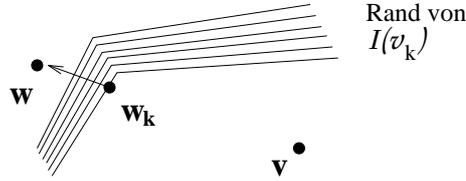
Annahme: Sei $v \notin V_l$, d. h. es gibt eine Überschneidung zwischen Polyedern; seien dies P und Q . Dann existieren Punkte p, q echt innerhalb von P bzw. Q , die in der Anordnung v übereinander liegen (vgl. Abb. 3.3). Da p und q echt innerhalb der Polyeder liegen, existiert ein $\delta > 0$, so daß sowohl der minimale Abstand von p zum Rand von P als auch von q zum Rand von Q größer gleich δ sind. Für jede überschneidungsfreie Anordnung v_k muß daher also $\|p - q\| > 2\delta$ gelten; andererseits gilt $\|p - q\| = 0$ in v . Da der Abstand $\|p - q\|$ stetig von $v - v_k$ abhängt, ist dies ein Widerspruch zu v Grenzwert von v_k .

2. *Schritt:*

Mit der Abgeschlossenheit von V_l zeigen wir nun eine spezielle Stetigkeitseigenschaft von \mathcal{I} . Sei $\{v_k\}$ eine konvergente Folge in V_l , deren Grenzwert v kein lokales Optimum sei, d. h. $v \notin \Omega_{lok}$. Für eine beliebige, jedoch als konvergent vorausgesetzte Wahl von Iterationsnachfolgern $w_k \in \mathcal{I}(v_k, \varepsilon_{max})$ mit Grenzwert w gilt dann: $w \in \mathcal{I}(v, \varepsilon_{max})$. Formal:

$$\left. \begin{array}{ll} v_k \in V_l & \text{mit } v_k \rightarrow v \\ w_k \in \mathcal{I}(v_k, \varepsilon_{max}) & \text{mit } w_k \rightarrow w \end{array} \right\} \Rightarrow w \in \mathcal{I}(v, \varepsilon_{max}) \quad (3.18)$$

Wegen der Abgeschlossenheit von V_l gilt sowohl $v \in V_l$ als auch $w \in V_l$. Wir betrachten nun die Abstandsfunktion $d_w(v_k, \varepsilon_{max})$, die jeweils den minima-

Abbildung 3.4: Konvergenz von $\mathcal{I}(v_k, \varepsilon_{max})$ gegen w

len Abstand von w zur Menge $\mathcal{I}(v_k, \varepsilon_{max})$ angibt (vgl. Abb. 3.4):

$$\begin{aligned} d_w(v_k, \varepsilon_{max}) &:= \min_{\tilde{w} \in \mathcal{I}(v_k, \varepsilon_{max})} \|w - \tilde{w}\| \\ &\leq \|w - w_k\|, \text{ da } w_k \in \mathcal{I}(v_k, \varepsilon_{max}) \end{aligned} \quad (3.19)$$

Der Rand von $\mathcal{I}(v_k, \varepsilon_{max})$ wird durch lineare Restriktionen in den Variablen Δx mit Parametern v_k und ε_{max} gebildet. Der Rand von $\mathcal{I}(v_k, \varepsilon_{max})$ ändert sich damit stetig mit v_k und ε_{max} . Insbesondere ist also auch die Abstandsfunktion $d_w(v_k, \varepsilon_{max})$ stetig.

Da $\|w - w_k\| \rightarrow 0$ (Voraussetzung w Grenzwert von $\{w_k\}$), muß für $v \rightarrow v_k$ wegen Ungleichung (3.19) auch $d_w(v_k, \varepsilon_{max}) \rightarrow 0$ gelten. Aus der Stetigkeit von d_w folgt damit $d_w(v, \varepsilon_{max}) = 0$. Da die Mengen $\mathcal{I}(v_k, \varepsilon_{max})$ abgeschlossen sind („ \leq “-Restriktionen), folgt daraus $w \in \mathcal{I}(v, \varepsilon_{max})$, was zu zeigen war.

3. Schritt:

Es wird nun die Konvergenz des Verfahrens gegen ein lokales Optimum gezeigt. Falls in einer Iteration $v_k \in \Omega_{lok}$ gilt, so wird das Verfahren beendet und die Aussagen von Satz 6 gelten.

Wir nehmen also an, daß das Verfahren eine unendliche Lösungsfolge $\{v_k\}$ generiere. Sei $\{v_k\}_{\mathcal{K}}$ eine beliebige konvergente Teilfolge mit Grenzwert $v \in V_l$ (diese existiert, da $\{v_k\}$ in einer kompakten Teilmenge des \mathbb{R}^N liegt). Wegen der Stetigkeit von f gilt: $\forall \varepsilon \exists K \in \mathcal{K} \forall k \geq K, k \in \mathcal{K} \quad f(v_k) - f(v) < \varepsilon$. Damit insbesondere

$$f(v_K) - f(v) < \varepsilon. \quad (3.20)$$

Sei nun $k > K$ beliebig, d. h. nicht notwendig aus \mathcal{K} . Da f monoton fallend über $\{v_k\}$ ist, gilt

$$f(v_k) < f(v_K) \quad \Leftrightarrow \quad f(v_k) - f(v_K) < 0 \quad \forall k > K. \quad (3.21)$$

Damit folgt:

$$f(v_k) - f(v) = \overbrace{f(v_k) - f(v_K)}^{(3.21)} + \overbrace{f(v_K) - f(v)}^{(3.20)} < \varepsilon \quad \forall k > K, \quad (3.22)$$

das heißt $\lim_{k \rightarrow \infty} f(v_k) = f(v)$. Bleibt zu zeigen, daß $v \in \Omega_{lok}$ gilt.

Annahme: $v \notin \Omega_{lok}$. Wir betrachten nun die Folge $\{v_{k+1}\}_{\mathcal{K}}$. Diese hat wiederum eine konvergente Teilfolge $\{v_{k+1}\}_{\tilde{\mathcal{K}} \subset \mathcal{K}}$ mit Grenzwert \tilde{v} in V_l . Wegen (3.22) muß $f(\tilde{v}) = f(v)$ gelten. Mit der Stetigkeitseigenschaft (3.18) der Iterationsabbildung \mathcal{I} in v folgt aus $k \in \mathcal{K}, v_k \rightarrow v$ und $v_{k+1} \in \mathcal{I}(v_k, \varepsilon_{max}), v_{k+1} \rightarrow \tilde{v}$, daß $\tilde{v} \in \mathcal{I}(v, \varepsilon_{max})$ gelten muß. Aus $v \notin \Omega_{lok}$ folgt damit aber $f(\tilde{v}) < f(v)$ im Widerspruch zu $\lim f(v_k) = f(v)$.

Daher ist also $v \in \Omega_{lok}$ und Punkt 1 von Satz 6 bewiesen. Punkt 2 folgt direkt aus Ungleichung (3.22). □

3.4 Schrittweitenregulierung

Im vorangehenden Abschnitt mußte die Schrittweite ε_{max} für den Konvergenzbeweis so gewählt werden, daß keine Überschneidungen entstehen bzw. stets legale Lösungen erzeugt werden. In der Praxis ist dies nicht zwingend erforderlich, da Überschneidungen im nächsten Optimierungsschritt automatisch beseitigt werden, wobei jedoch die Monotonie der Zielfunktion nicht notwendigerweise gegeben ist. In diesem Abschnitt sollen daher verschiedene Ansätze diskutiert werden, wie die Schrittweitensteuerung realisiert werden kann.

In Abschnitt 7.1 wird mit Satz 31 gezeigt, daß die Linearisierung der Drehungsoperation immer einen Hüllkörper des ursprünglichen Objektes liefert. Somit ist für die isoliert betrachteten Rotationsvariablen Δr keine Schrittweitenregulierung notwendig, da stets legale Objekttransformationen abgeleitet werden können. Die Schrittweitenregulierung erfolgt hier implizit über die Minimierung der Zielfunktion.

In Abschnitt 3.2 wurde die Linearisierung (3.11) der Separationsungleichung (3.4) hergeleitet. Es werden nun die beiden quadratischen Terme näher betrachtet, die in (3.11) vernachlässigt werden:

1. $\langle \Delta n_{ij}, \Delta t_i \rangle$
2. $\langle \Delta n_{ij}, \Delta r_i \times R_i v \rangle$

Hieraus folgt, daß bei Schrittweite 0 für Δn_{ij} (der Neigung der Separationsebene), stets legale Lösungen erzeugt werden, da die beiden quadratischen Terme verschwinden. Der Normalenvektor der Separationsebene n_{ij} bleibt in diesem Fall konstant; die Rotationen Δr_i und Translationen Δt_i der Polyeder sind jedoch weiterhin variabel. Ebenso kann für $\Delta r_i, \Delta t_i$ gemeinsam Schrittweite 0 gefordert werden, d. h. Rotation und Translation der Objekte werden konstant gehalten. In diesem Fall ist dann die Schrittweite für Δn_{ij} unbegrenzt.

Der quadratische Term $\langle \Delta n_{ij}, \Delta t_i \rangle$ verschwindet zudem, falls $\Delta t_i = \lambda n_{ij}$ gefordert wird. Dies folgt direkt aus der Orthogonalitätsbedingung (3.17) $\langle \Delta n_{ij}, n_{ij} \rangle = 0$.

Den zweiten Term betrachten wir nun in der äquivalenten Form $\langle R_i v, \Delta n_{ij} \times \Delta r_i \rangle$. Wird hier $\Delta n_{ij} = \lambda \Delta r_i$ gefordert, so verschwindet das Kreuzprodukt der beiden Variablen und damit der gesamte Term. Anschaulich bedeutet dies, daß die Drehachse Δr_i der infinitesimalen Rotation parallel zur Auslenkung des Normalenvektors der Separationsebene verläuft.

Falls Ungleichung (3.4) in der Ausgangslösung für den Optimierungsschritt keine bindende Restriktion ist, d. h. es gilt $\langle n_{ij}, R_i p_{iv} + t_i \rangle - b_{ij} > 0$, so können für Δn_{ij} , Δr_i und Δt_i echt positive Schrittweiten bestimmt werden, so daß der Betrag der beiden nichtlinearen Terme echt kleiner als der Schlupf der Ungleichung ist. Man beachte hierzu, daß die Schlupfvariablen am Ende des Simplexverfahrens ohne zusätzlichen Rechenaufwand zur Verfügung stehen.

Für die bindenden Restriktionen, d. h. (3.4) stellt eine Gleichung dar, müssen die linearen Restriktionen (3.11) zunächst noch weiter auf den Bereich der zulässigen Lösungen eingeschränkt werden. Die Wahl von zulässigen Richtungen und entsprechenden Schrittweiten werden im folgenden Abschnitt über das Verfahren von Zoutendijk behandelt.

3.5 Nichtlineare Optimierung: Verfahren von Zoutendijk

In diesem Abschnitt wird das Verfahren von Zoutendijk vorgestellt, welches ein lokales Optimierungsverfahren zur Lösung nichtlinearer Programme ist. Dies stellt zum einen eine eigenständige Alternative zu dem in Abschnitt 3.2 vorgestellten Linearisierungsansatz nach Milenkovic dar. Zum anderen wurde im Konvergenzbeweis zu diesem Linearisierungsansatz die Schrittweitenwahl als gegeben vorausgesetzt. Die Wahl der Schrittweite wird nun im Zusammenhang mit der Wahl von zulässigen Optimierungsrichtungen am Ende dieses Abschnittes näher behandelt.

Das Grundprinzip des Verfahrens ist wie folgt: In jedem Optimierungsschritt werden zwei Teilprobleme gelöst. Zunächst wird eine zulässige Optimierungsrichtung bestimmt, welche zudem die Zielfunktion verbessert (*verbessernde, zulässige Richtung*). Das zweite Teilproblem besteht dann in der Festlegung der Schrittweite entlang dieser Richtung. Da die Richtung bereits vorgegeben ist, ist die Bestimmung der Schrittweite ein eindimensionales Line-Search-Problem, welches mit Standardmethoden gelöst werden kann.

Für das Verfahren von Zoutendijk betrachten wir nichtlineare Programme der folgenden Form: Sei $f : \mathbb{R}^N \rightarrow \mathbb{R}$ die zu minimierende Zielfunktion. Ferner sei wie in Abschnitt 3.3 eingeführt $V_l \subset \mathbb{R}^N$ die Menge der

zulässigen Lösungen. Diese sei durch ein Ungleichungssystem $g_i(x) \leq 0$ mit m Ungleichungen gegeben und $I = \{1, \dots, m\}$ die Indexmenge, d. h. $V_l = \{v \mid g_i(v) \leq 0 \quad \forall i \in I\}$. Zielfunktion und Ungleichungen können nicht-linear sein, werden jedoch als differenzierbar über V_l vorausgesetzt³. Dies ist für die Formulierung des Anordnungsproblems nach (3.5)-(3.10) gegeben. Wir betrachten nun also das folgende Programm:

$$\min f(x) \tag{3.23}$$

mit Nebenbedingungen:

$$g_i(x) \leq 0 \quad \forall i \in I \tag{3.24}$$

Definition 6 (verbessernde, zulässige Richtung)

Ein Vektor $d_v \in \mathbb{R}^N \setminus \{0\}$ heißt zulässige Richtung in $v \in V_l$, wenn ein $\delta > 0$ existiert, so daß für alle $\lambda \in [0, \delta]$ folgt $v + \lambda d_v \in V_l$. d_v heißt verbessernde, zulässige Richtung, wenn zudem $f(v + \lambda d_v) < f(v)$ gilt.

Sei $\bar{I}_v \subset I$ die Menge der bindenden Restriktionen einer zulässigen Lösung $v \in V_l$, d. h. $\bar{I}_v = \{i \in I \mid g_i(v) = 0\}$.

Satz 7

Sei $v \in V_l$ eine zulässige Lösung des nichtlinearen Programms (3.23)-(3.24) und \bar{I}_v die Menge der bindenden Restriktionen. Falls $\nabla f(v) d_v < 0$ und $\nabla g_i(v) d_v < 0 \quad \forall i \in \bar{I}_v$, dann ist d_v eine verbessernde, zulässige Richtung.

Beweis:

Wir betrachten zunächst g_i mit $i \notin \bar{I}_v$. Aus der Differenzierbarkeit folgt Stetigkeit in v und damit folgt aus $g_i(v) < 0$, daß $g_i(v + \lambda d_v) \leq 0$ für hinreichend kleines $\lambda > 0$.

Für g_i mit $i \in \bar{I}_v$ folgt

$$g_i(v + \lambda d_v) = g_i(v) + \lambda \nabla g_i(v) d_v + \lambda \|d_v\| \alpha(v, \lambda d_v) \tag{3.25}$$

mit $\alpha(v, \lambda d_v) \rightarrow 0$ für $\lambda \rightarrow 0$. Aus $\nabla g_i(v) d_v < 0$ folgt damit $g_i(v + \lambda d_v) < g_i(v) \leq 0$ für hinreichend kleines $\lambda > 0$. Somit ist d_v eine zulässige Richtung.

Mit dem gleichen Argument folgt $f(v + \lambda d_v) < f(v)$ aus $\nabla f(v) d_v < 0$ und somit ist d_v eine verbessernde, zulässige Richtung. □

³Es genügt Differenzierbarkeit jeweils über der Menge \bar{I} der bindenden Restriktionen der tatsächlich erzeugten Zwischenlösungen zu fordern und Stetigkeit von $g_i(v)$ für $i \in I \setminus \bar{I}$.

Um eine verbessernde, zulässige Richtung zu bestimmen, wird das folgende Lineare Programm gelöst:

$$\min s \quad (3.26)$$

mit Nebenbedingungen:

$$\nabla f(v) d - s \leq 0 \quad (3.27)$$

$$\nabla g_i(v) d - s \leq 0 \quad \forall i \in \bar{I} \quad (3.28)$$

$$-1 \leq d_j \leq 1 \quad \forall j \in \{1, \dots, N\} \quad (3.29)$$

Man beachte, daß dies eine lineare Formulierung in der skalaren Variablen s und der Vektorvariablen d ist. Die Zielfunktion bzw. Schlupfvariable s ist genau dann gleich Null, wenn das strikte Ungleichungssystem $\nabla f(v)d < 0$, $\nabla g_i(v)d < 0$ keine Lösung besitzt. In diesem Fall gilt $s = 0$, $d = \mathbf{0}$ und v ist ein lokales Optimum. Falls das strikte Ungleichungssystem eine Lösung hat, so ist der Zielfunktionswert durch die Normalisierungsbedingung $-1 \leq d_j \leq 1$ beschränkt.

Nachdem eine verbessernde, zulässige Richtung festgelegt ist (d. h. $d_v \neq \mathbf{0}$), muß nun noch die zugehörige Schrittweite $\lambda > 0$ bestimmt werden. Dies stellt ein eindimensionales Line-Search-Problem der folgenden Art dar:

$$\min f(v + \lambda d_v) \quad (3.30)$$

mit Nebenbedingungen:

$$0 \leq \lambda \leq \varepsilon_{max} \quad (3.31)$$

Hierbei ist $\varepsilon_{max} = \sup\{\lambda \mid g_i(v + \lambda d_v) \leq 0 \quad \forall i \in I\}$, was im allgemeinen ebenfalls Line-Search Probleme darstellt. Für das hier betrachtete Anordnungsproblem mit linearisierter Drehungsoperation (man beachte hierzu nochmals, daß für die linearisierten Drehungsvariablen Δr keine Schrittweitensteuerung notwendig ist) sind die Restriktionen g_i jedoch höchstens quadratisch, womit eine analytische Lösung möglich ist.

Kapitel 4

Verbindung von globaler und lokaler Optimierung

In diesem Kapitel wird aufgezeigt, wie sich die Methoden der globalen und lokalen Optimierung zusammenführen lassen. Die Basis für die in Kapitel 2 eingeführten globalen Optimierungsverfahren bilden Distanzpolyeder, aus denen Separationsebenen abgeleitet werden. Als Alternative zur Bestimmung der Distanzpolyeder aus der Minkowski-Summe von Polyedern wird in Abschnitt 5.3 das Konzept der \mathcal{P} -Spuren eingeführt. Diese sind über den Schnitt der Winkelbereiche von Randobjekten definiert. Abb. 4.1 zeigt die \mathcal{P} -Spuren $\mathcal{P}_A, \mathcal{P}_B$ zweier Polygone. Hier entspricht der Winkelbereich einer Kante jeweils dem nach außen gerichteten Normalenvektor und der Winkelbereich einer Ecke dem zwischen den beiden inzidenten Kanten liegenden roten Bereich. Die formale Definition von Winkelbereichen dreidimensionaler Objekte erfolgt in Abschnitt 5.3.1, Definition 14¹.

Wesentlich ist die Feststellung, daß sich bei kontinuierlichen Drehungen der Objekte die Topologie der Distanzpolyeder bzw. \mathcal{P} -Spuren nur an endlich vielen Punkten ändert. Diese Änderungen ergeben sich genau dann, wenn ein Normalenvektor einer Kante (in \mathbb{R}^2) bzw. Fläche (in \mathbb{R}^3) den Winkelbereich einer Ecke verläßt. Die Distanzpolyeder mit identischer topologi-

¹Aus Def. 14 ergeben sich offene Mengen für Eckenwinkelbereiche, um insgesamt diskunkte Winkelbereiche zu erzeugen. Um die Anzahl der Äquivalenzklassen klein zu halten, werden hier jedoch halboffene Mengen für Eckenwinkelbereiche betrachtet, die einen der Normalenvektoren enthalten

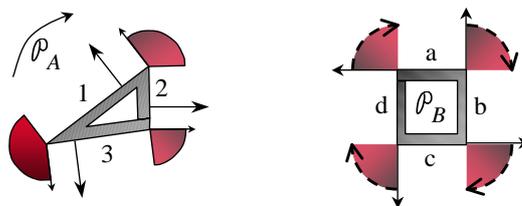


Abbildung 4.1: Winkelbereiche von \mathcal{P} -Spuren

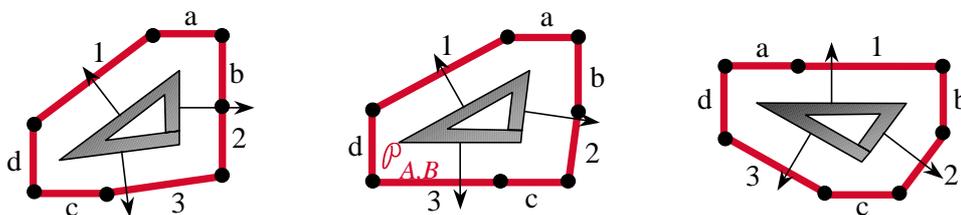


Abbildung 4.2: Äquivalenzklassen von Distanzpolyedern

scher Struktur werden zu Äquivalenzklassen zusammengefaßt und zu jeder Äquivalenzklasse wird jeweils ein Repräsentant ausgezeichnet. Das Optimierungsverfahren ist so strukturiert, daß eine global optimale Lösung auf der Menge der Repräsentanten gefunden wird, deren Drehwinkel durch die implizite lokale Optimierung angepaßt werden kann. Durch die implizite lokale Optimierung innerhalb der Äquivalenzklassen werden somit kontinuierliche Objektdrehungen möglich.

Die folgenden Abschnitte 4.1 und 4.2 führen die wesentlichen Elemente der global-lokalen Problemformulierung ein. In Abschnitt 4.3 werden daran anschließend alternative Optimierungsmethoden diskutiert.

4.1 Äquivalenzrelation auf Distanzpolyedern

Wir betrachten zunächst Abb. 4.1 und Abb. 4.2, um die Unterteilung in Äquivalenzklassen zu motivieren. Betrachtet werden ein Dreieck (mit Normalenvektoren der Kanten) und ein Quadrat (mit Winkelbereichen der Ecken), wobei die Randelemente mit Ziffern bzw. Buchstaben markiert sind und das Dreieck im Uhrzeigersinn gedreht wird. In der Ausgangskonfiguration steht Kante 2 senkrecht und der Normalenvektor liegt im Winkelbereich (b, c) des Quadrates. Beim ersten Drehungsschritt tritt der Normalenvektor von Kante 3 in den Winkelbereich (c, d) ein und schließlich verläßt der Normalenvektor 1 den Winkelbereich (d, a) . Hierzu ist jeweils das entsprechende Distanzpolyeder dargestellt, wobei dessen Topologie zwischen diesen Übergängen gleich bleibt. Die Äquivalenzklassen können auch mit den Kantenfolgen $(1ab23cd)$, $(1ab2c3d)$, $(a1b2c3d)$ bezeichnet werden. In Abbildung 4.3 sind alle Äquivalenzklassen mit jeweils einem Repräsentanten zusammengefaßt. Da im dreidimensionalen Raum keine kanonische Reihenfolge der Kanten gegeben ist, muß die Bezeichnung der Äquivalenzklassen in diesem Fall über die erzeugenden Objektpaare der Ursprungspolyeder $\mathcal{P}_A, \mathcal{P}_B$ erfolgen. In Abschnitt 5.3.1 auf Seite 75 in (5.13)-(5.15) sind die Paare definiert, die jeweils Randobjekte von $\mathcal{P}_{A,B}$ erzeugen. Für die Bildung der Äquivalenzklassen ist nur die Menge $\mathcal{F}_{A,B}$ der Flächen entscheidend, da die Mengen $\mathcal{E}_{A,B}, \mathcal{K}_{A,B}$ der Ecken und Kanten damit implizit festgelegt sind. Zwei Distanzpolyeder sind topologisch äquivalent genau dann, wenn die Mengen der erzeugenden

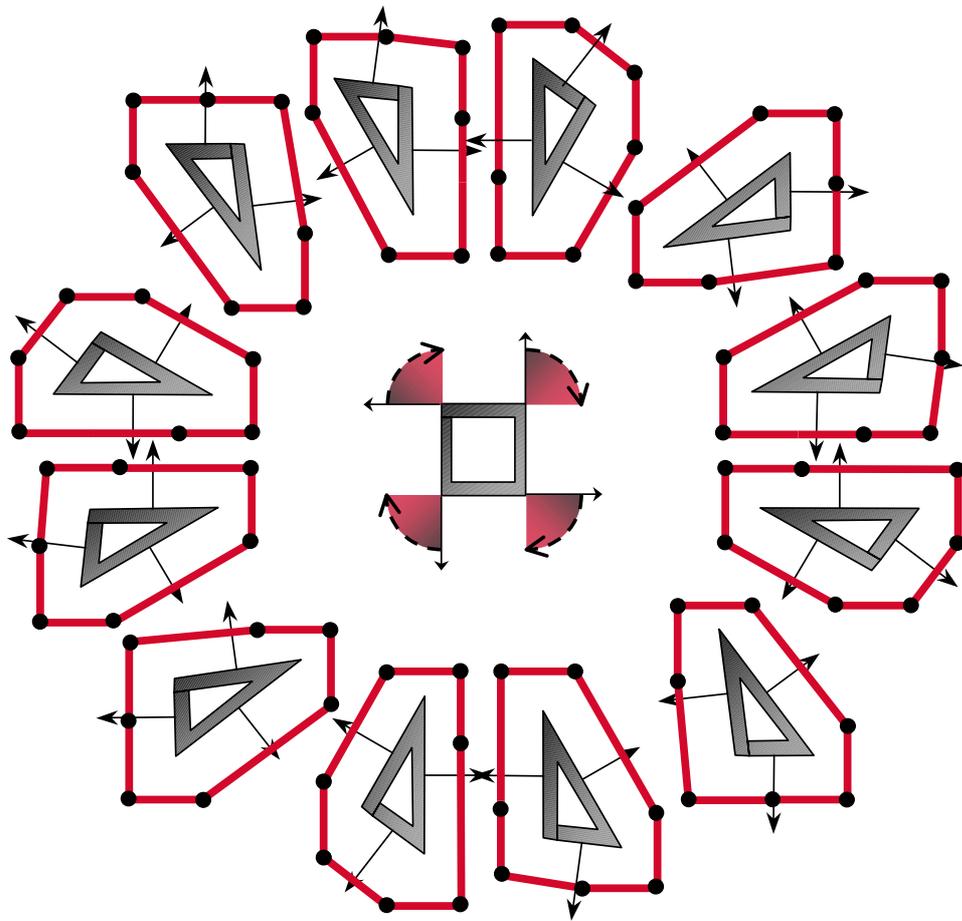


Abbildung 4.3: Repräsentanten der Äquivalenzklassen von Distanzpolyedern

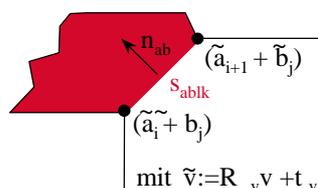


Abbildung 4.4: Facette des Distanzpolyeders mit erzeugenden Eckpunkten

Objektpaare $\{(f, v)\} \cup \{(e_a, e_b)\}$ aus $\mathcal{F}_{A,B}$ identisch sind.

4.2 Herleitung der Separationsungleichung

Im folgenden sei nun für ein Objektpaar P_A, P_B eine Äquivalenzklasse des Distanzpolyeders vorgegeben, d. h. die relative Orientierung o_{ab} der Objekte wird innerhalb der Grenzen dieser Klasse festgelegt. Dies kann wie in Kapitel 2 wegen der endlichen Anzahl von Klassen über Entscheidungsvariablen o_{abl} realisiert werden (l wird als neuer Index für die Äquivalenzklasse verwendet). Die Objektseparation erfolgt ebenfalls wie in Abschnitt 2.1 über die Lage des Referenzpunktes von P_B im positiven Halbraum einer Facette s_{ablk} des Distanzpolyeders. Das zugehörige Ungleichungssystem muß nun jedoch die variable Orientierung der Facette berücksichtigen, so daß sich wie bei der lokalen Optimierung in Kapitel 3 zunächst ein exaktes, jedoch nichtlineares Ungleichungssystem ergibt. Dieses wird nun für den zweidimensionalen Fall hergeleitet und kann auf den dreidimensionalen Fall übertragen werden.

Es seien zwei Polygone P_A, P_B gegeben. Die Ecken von P_A seien mit $\{a_1, \dots, a_n\}$ für eine beliebig, aber fest gewählte Orientierung gegeben. Die Ecken von P_B seien entsprechend $\{b_1, \dots, b_m\}$, wobei die Ausgangslage von B derart gewählt ist, daß sie in der Mitte des durch die Entscheidungsvariable o_{abl} festgelegten Winkelbereiches liegt. Sei α_{ab} die Gesamtgröße des Winkelbereiches, dann ergibt sich eine Drehung von maximal $\alpha_{ab}/2$ zu jeder Seite. Mit R_a, R_b für die Drehungen von P_A, P_B ergibt sich daraus die folgende Restriktion für einen beliebig aber fest gewählten Einheitsvektor e :

$$\cos \frac{\alpha_{ab}}{2} \leq \langle R_a e, R_b e \rangle \quad (4.1)$$

Wie in der Formulierung des lokalen Optimierungsverfahrens bezeichne t_a die Translation von Polyeder A . Für die transformierten Eckpunkte der Polygone führen wir zur Abkürzung die Notation \tilde{a}_i ein, sowie r_{ab} für die Lage des Referenzpunktes von B relativ zu A :

$$\tilde{a}_i := (R_a a_i + t_a) \quad (4.2)$$

$$r_{ab} := t_b - t_a \quad (4.3)$$

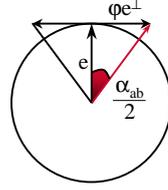


Abbildung 4.5: Linearisierung zweidimensionaler Rotationen

Sei nun die Entscheidungsvariable s_{ablk} für eine Facette des Distanzpolyeders festgelegt, wobei die Facette von einer Kante (a_i, a_{i+1}) von P_A und einer Ecke b_j von Polygon P_B erzeugt sei (siehe hierzu auch Abb. 4.4). Die Topologie ist dabei durch die Entscheidungsvariable o_{abl} vorgegeben und die Facette des Distanzpolyeders ist durch die Eckpunkte

$$(\tilde{a}_i + \tilde{b}_j) \text{ und } (\tilde{a}_{i+1} + \tilde{b}_j)$$

definiert. Da die Verschiebungsanteile t_a und \tilde{b}_j in beiden Punkten enthalten sind, werden sie zur Berechnung des Normalenvektors n_{ab} nicht benötigt. Der Normalenvektor der Facette ergibt sich somit wie folgt²:

$$n_{ab} := R_a(a_{i+1} - a_i)^\perp . \quad (4.4)$$

Da der Punkt $\tilde{a}_i + \tilde{b}_j$ auf dieser Kante liegt, erhalten wir als absolutes Glied der Geradengleichung

$$d_{ab} = \langle n_{ab}, \tilde{a}_i + \tilde{b}_j \rangle . \quad (4.5)$$

Die Separationsungleichung der Polygone A, B lautet damit:

$$\langle n_{ab}, r_{ab} \rangle - d_{ab} \geq 0 . \quad (4.6)$$

Da $n_{ab}, r_{ab}, \tilde{a}_i, \tilde{b}_j$ jeweils Variablen sind (sie enthalten Rotationsvariablen R_v oder Translationsvariablen t_v), ergeben die beiden Skalarprodukte quadratische Terme. Wir betrachten nun die Linearisierung des Ungleichungssystems. Der Matrixdarstellung einer zweidimensionalen Drehung eines Punktes p um den Winkel α

$$\begin{pmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} \quad (4.7)$$

entspricht mit $R = (\cos \alpha, \sin \alpha)$ die im folgenden verwendete Vektormultiplikation

$$R \cdot p := (R_1 p_1 - R_2 p_2, R_1 p_2 + R_2 p_1) . \quad (4.8)$$

² $v^\perp := (-v_2, v_1)$ ist der zu $v = (v_1, v_2)$ senkrechte Vektor

Die Linearisierung zweidimensionaler Rotationen eines Vektors e wird durch einen senkrechten Anteil e^\perp , wie in Abb. 4.5 dargestellt, realisiert. Das Bild eines normierten Vektors e unter einer infinitesimalen Drehung wird also durch $e + \varphi e^\perp$ approximiert.

Wie in Kapitel 3 werden die Variablen für Rotation und Translation der Objekte in einen konstanten Anteil R_a, t_a und einen linearisierten, variablen Anteil $\Delta R_a, \Delta t_a$ aufgeteilt. Für ΔR_a ergibt sich mit der oben eingeführten Linearisierung $\Delta R_a = \varphi_a R_a^\perp$. Für transformierte Eckpunkte erhalten wir damit insgesamt

$$\begin{aligned}\bar{a}_i &:= (R_a + \Delta R_a) \cdot a_i + (t_a + \Delta t_a) \\ &= R_a \cdot a_i + t_a + \Delta R_a \cdot a_i + \Delta t_a \\ &= \tilde{a}_i + \Delta R_a \cdot a_i + \Delta t_a\end{aligned}\tag{4.9}$$

Linearisierte Variablen und Terme x werden im folgenden immer mit \bar{x} gekennzeichnet. Für die Beschränkung des Winkelbereiches der relativen Objektlage α_{abl} auf $\pm\alpha_{ab}/2$ nach Ungleichung (4.2) ergibt sich in der linearisierten Form mit $\varphi_{ab} = \varphi_a - \varphi_b$

$$-\tan \frac{\alpha_{ab}}{2} \leq \varphi_{ab} \leq \tan \frac{\alpha_{ab}}{2} .\tag{4.10}$$

Für den Normalenvektor \bar{n}_{ab} der Facette des Distanzpolyeders erhalten wir in linearisierter Form:

$$\bar{n}_{ab} := (R_a + \Delta R_a) \cdot (a_{i+1} - a_i)^\perp\tag{4.11}$$

Damit ergibt sich die folgende linearisierte Form der Separationsungleichung 4.6 (die Vorzeichen $-\bar{r}_{ab}$ und $+\bar{d}_{ab}$ wurden zur Vereinfachung der Rechnung gewählt):

$$\begin{aligned}\langle \bar{n}_{ab}, -\bar{r}_{ab} \rangle + \bar{d}_{ab} &= \langle \bar{n}_{ab}, \bar{t}_a - \bar{t}_b \rangle + \langle \bar{n}_{ab}, \bar{a}_i + \bar{b}_j \rangle \\ &= \langle \bar{n}_{ab}, t_a + \Delta t_a - t_b - \Delta t_b \rangle + \\ &\quad \langle \bar{n}_{ab}, R_a \cdot a_i + t_a + \Delta R_a \cdot a_i + \Delta t_a \rangle + \\ &\quad \langle \bar{n}_{ab}, R_b \cdot b_j + t_b + \Delta R_b \cdot b_j + \Delta t_b \rangle \\ &= \langle \bar{n}_{ab}, R_a \cdot a_i + 2t_a + R_b \cdot b_j \rangle + \\ &\quad \langle \bar{n}_{ab}, \Delta R_a \cdot a_i + 2\Delta t_a + \Delta R_b \cdot b_j \rangle \\ &\stackrel{4.13\text{ff}}{=} \langle R_a \cdot (a_{i+1} - a_i)^\perp, 2t_a + R_b \cdot b_j \rangle + 2\langle a_i, a_{i+1}^\perp \rangle + \\ &\quad \langle R_a \cdot (a_{i+1} - a_i)^\perp, \Delta R_a \cdot a_i + 2\Delta t_a + \Delta R_b \cdot b_j \rangle + \\ &\quad \langle \Delta R_a \cdot (a_{i+1} - a_i)^\perp, R_a \cdot a_i + 2t_a + R_b \cdot b_j \rangle\end{aligned}\tag{4.12}$$

Es folgen Nebenrechnungen zu den Teiltermen von 4.12:

$$\begin{aligned}
& \langle \bar{n}_{ab}, R_a \cdot a_i + 2t_a + R_b \cdot b_j \rangle \\
&= \langle R_a \cdot (a_{i+1} - a_i)^\perp, R_a \cdot a_i + 2t_a + R_b \cdot b_j \rangle + \\
& \quad \langle \Delta R_a \cdot (a_{i+1} - a_i)^\perp, R_a \cdot a_i + 2t_a + R_b \cdot b_j \rangle \quad (4.13) \\
&= \langle R_a \cdot (a_{i+1} - a_i)^\perp, 2t_a + R_b \cdot b_j \rangle + \langle a_i, a_{i+1}^\perp \rangle + \\
& \quad \langle \Delta R_a \cdot (a_{i+1} - a_i)^\perp, R_a \cdot a_i + 2t_a + R_b \cdot b_j \rangle
\end{aligned}$$

Nebenrechnung zu $\langle \bar{n}_{ab}, \Delta R_a \cdot a_i \rangle$:

$$\begin{aligned}
\langle \bar{n}_{ab}, \Delta R_a \cdot a_i \rangle &= \langle R_a \cdot (a_{i+1} - a_i)^\perp, \Delta R_a \cdot a_i \rangle + \\
& \quad \langle \Delta R_a \cdot (a_{i+1} - a_i)^\perp, \Delta R_a \cdot a_i \rangle \quad (4.14) \\
&= \langle R_a \cdot (a_{i+1} - a_i)^\perp, \Delta R_a \cdot a_i \rangle + \langle a_i, a_{i+1}^\perp \rangle
\end{aligned}$$

Nebenrechnung zu $\langle \bar{n}_{ab}, \Delta t_a \rangle$:

$$\langle n_{ab}, 2\Delta t_a \rangle = \langle R_a \cdot (a_{i+1} - a_i)^\perp, 2\Delta t_a \rangle + 0 \quad (4.15)$$

Nebenrechnung zu $\langle \bar{n}_{ab}, \Delta R_b \cdot b_j \rangle$:

$$\langle \bar{n}_{ab}, \Delta R_b \cdot b_j \rangle = \langle R_a \cdot (a_{i+1} - a_i)^\perp, \Delta R_b \cdot b_j \rangle + 0 \quad (4.16)$$

Für den praktischen Einsatz ist jedoch die Komplexität dieses Verfahrens zu beachten. Es seien n Objekte gegeben und mit o_{ij} sei jeweils die Anzahl relativer Objektorientierungen bzw. Äquivalenzklassen von Distanzpolyedern mit gleicher Topologie bezeichnet. Zunächst erhalten wir für die Anzahl möglicher Objektorientierungen das Produkt der paarweisen Orientierungen, d. h. $O(n^2)$ Faktoren (wobei allerdings einige Kombinationen ausgeschlossen sind): $\prod_{1 \leq i < j \leq n} o_{ij}$. Zudem sind bei festgelegter Orientierung jeweils s_{ijk} Facetten des zugeordneten Distanzpolyeders zu betrachten, womit sich insgesamt ergibt:

$$\prod_{1 \leq i < j \leq n} \left(\sum_{1 \leq k \leq o_{ij}} s_{ijk} \right)$$

Es wird hier schon im zweidimensionalen Fall notwendig sein, ein hierarchisches Konzept und eine Zusammenfassung bzw. Vergrößerung der Distanzpolyeder zu betrachten.

4.3 Alternative Optimierungsmethoden

In dieser Arbeit werden Methoden der ganzzahligen Linearen Programmierung zur Formulierung und Optimierung der räumlichen Anordnungsprobleme verwendet. In den folgenden Abschnitten werden die Unterschiede,

sowie Vor- und Nachteile von zwei alternativen Verfahren diskutiert, die zur Lösung kombinatorischer Optimierungsprobleme eingesetzt werden können: Constraint Programming und Simulated Annealing.

4.3.1 Constraint Programming

Zunächst werden ganzzahlige Lineare Programmierung (ILP) und Finite Domain Constraint Programming (CP(FD)) auf einer gemeinsamen Sprachbasis eingeführt, anhand derer die Unterscheidung zwischen den Verfahren herausgestellt werden kann. Die Grundidee zu dieser Darstellung ist [BK98] entnommen. Als gemeinsame Sprachbasis wird formal die Prädikatenlogik erster Ordnung betrachtet. Eine Sprache ist hier durch eine Signatur $\Sigma = (F, P)$ definiert, wobei F eine Menge von Funktionssymbolen und P eine Menge von Prädikaten mit gegebener Stelligkeit ist. Funktionssymbole der Stelligkeit 0 entsprechen Konstanten. Zudem werden Variablensymbole $V = \{x_1, x_2, \dots\}$ und Terme t_i eingeführt. Ein Term t ist dabei rekursiv definiert, wobei eine einzelne Variable, Konstante oder Term jeweils ein gültiger Term sind und falls f ein n -stelliges Funktionssymbol ist, so ist $f(t_1, \dots, t_n)$ ebenfalls ein zulässiger Term. Der Wertebereich der Variablen ist für Finite Domain Constraint Programming auf diskrete, endliche Mengen eingeschränkt, so daß hier zur einheitlichen Darstellung die rationalen Zahlen \mathbb{Q} , bzw. geeignete Teilmengen hiervon, gewählt werden.

Die Menge der Funktionssymbole wird mit $F = \{0, 1, +, -, \cdot, \div, [], [e, l]\}$ für ganzzahlige Lineare Programmierung und Constraint Programming identisch gewählt ($[]$ entspricht der leeren Liste und $[e, l]$ der Konstruktion einer Liste aus Element e und einer bestehenden Liste l). Die Menge der Prädikate wird für ILP und CP(FD) unterschiedlich festgelegt:

$$P_{ILP} = \{\leq, \geq, =, \text{ganzzahlig}\}$$

$$P_{CP(FD)} = \{\leq, \geq, =, \neq, <, >, \text{ganzzahlig, verschieden, genau_eine}\}$$

Für die ganzzahlige Lineare Programmierung ergeben sich damit die beiden folgenden Mengen von arithmetischen und Ganzzahligkeitsbedingungen:

$$A_{ILP} = \left\{ \sum_{i \in I} a_i \cdot x_i \triangleleft b \mid a_i, b \in \mathbb{Q}, x_i \in V, \triangleleft \in \{\leq, \geq, =\} \right\}$$

$$G_{ILP} = \{\text{ganzzahlig}([x_1, \dots, x_n])\}$$

Für Constraint Programming wird neben arithmetischen und Ganzzahligkeitsbedingungen zusätzlich eine Menge symbolischer Restriktionen eingeführt. Dies spiegelt wider, daß Constraint Programming Systeme im Zusammenhang mit höheren Modellierungssprachen betrachtet werden und insbesondere für die Beschreibung kombinatorischer Problemstellungen mächtige

Sprachkonstrukte zur Verfügung stellen:

$$\begin{aligned}
 A_{CP(FD)} &= \left\{ \sum_{i \in I} a_i \cdot x_i \triangleleft b \mid a_i, b \in \mathbb{Q}, x_i \in V, \triangleleft \in \{\leq, \geq, =, \neq, <, >\} \right\} \cup \\
 &\quad \{x_i \triangleleft x_j \mid x_i, x_j \in V, \triangleleft \in \{\leq, \geq, =, \neq, <, >\}\} \\
 G_{CP(FD)} &= \{\text{ganzzahlig}([x_1, \dots, x_n])\} \\
 S_{CP(FD)} &= \{\text{verschieden}([x_1, \dots, x_n]), \text{genau_eine}([x_1, \dots, x_n])\}
 \end{aligned}$$

Die hier gewählten symbolischen Restriktionen haben dabei folgende Bedeutung: $\text{verschieden}([x_1, \dots, x_n])$ besagt, daß für je zwei Variablen der Liste $x_i \neq x_j$ gelten muß, entspricht also einer quadratischen Anzahl arithmetischer $x_i \neq x_j$ Restriktionen. $\text{genau_eine}([x_1, \dots, x_n])$ drückt aus, daß genau eine der Variablen x_i der Liste einen Wert ungleich 0 annehmen muß; dies entspricht z. B. der Auswahl genau einer 0/1-Entscheidungsvariable, wie bei den globalen Optimierungsverfahren in Kapitel 2 gefordert. Die verfügbaren Constraint Programming Systeme enthalten eine sehr viel größere Anzahl vordefinierter symbolischer Restriktionen. Wesentlich ist hierbei, daß für jede einzelne, isoliert betrachtete symbolische Restriktion immer effiziente Lösungsverfahren existieren müssen. Man beachte zudem, daß wegen der endlichen Wertemengen für die Variablen arithmetische Restriktionen der Form $x_i < x_j$ sinnvoll sind und zu gegebenem x_i einen eindeutigen kleinstmöglichen Wert für x_j definieren.

Aus rein deklarativer Sicht stellt CP(FD) damit zunächst die mächtigeren Werkzeuge zur Verfügung, da die Sprache eine echte Obermenge des Sprachumfangs von ILP ist. Wir betrachten nun die operationelle Semantik dieser Sprachkonstrukte und führen hierzu eine Unterscheidung zwischen primitiven und nicht primitiven Sprachelementen ein. Die Grundidee ist, daß Probleme mit ausschließlich primitiven Sprachkonstrukten einfach, d. h. in polynomieller Zeit, gelöst werden können, Probleme mit nicht primitiven Sprachkonstrukten jedoch schwierig sind und ggf. mehr als polynomielle Laufzeiten erfordern.

Für ILP gilt

$$\begin{aligned}
 \text{Prim}_{ILP} &= A_{ILP} \\
 \text{NPrim}_{ILP} &= G_{ILP}
 \end{aligned}$$

Die nicht primitiven Restriktionen von ILP sind also gerade die Ganzzahligkeitsbedingungen. Werden diese aus einer Problemdefinition entfernt, so ergibt sich die lineare Relaxierung. Die arithmetischen Restriktionen sind hingegen primitiv, können also global, und nicht nur isoliert bzw. lokal betrachtet, effizient gelöst werden.

Für CP(FD) kann z. B. gewählt werden

$$\begin{aligned} Prim_{CP(FD)} &= G_{CP(FD)} \cup \\ &\quad \{x_i \leq u, x_i \geq l, x_i \neq c, x_i = x_j \mid x_i \in V; u, l, c \in \mathbb{Z}\} \\ NPrim_{CP(FD)} &= \text{alle restlichen Sprachelemente} . \end{aligned}$$

Für CP(FD) sind also insbesondere die Ganzzahligkeitsbedingungen primitive Constraints, dafür fallen hingegen allgemeine arithmetische Bedingungen unter die nicht primitiven Sprachelemente. Für eine vertiefte Darstellung wird auf Bockmayer, Kasper „Branch-and-Infer: A Unifying Framework for Integer and Finite Domain Constraint Programming [BK98]“ und Barth, Bockmayer „Modelling Mixed-Integer Optimization Problems in Constraint Logic Programming [BB98]“ verwiesen.

Hier sollen nur einige Schlussfolgerungen für die in dieser Arbeit betrachteten Anordnungsprobleme abgeleitet werden. Das Kompaktierungsproblem bei festgelegten Entscheidungsvariablen für die relative Objektlage, siehe Abschnitt 2.3, ist eine reine LP-Formulierung, d. h. ohne Ganzzahligkeitsbedingung. Die enthaltenen Variablen der Objekttranslationen sind in natürlicherweise kontinuierlich und nicht auf eine endliche Menge, d. h. Finite Domain, eingeschränkt. Für diesen Problembereich sind also die LP-Methoden mit ihrer globalen Lösungsfähigkeit der Restriktionsmenge der geeignete Ansatz.

Für die Festlegung der Entscheidungsvariablen und den kombinatorischen Anteil bei der Festlegung von Anordnungsgraphen und -klassen, sind hingegen Constraint Programming Methoden ein durchaus geeignetes Hilfsmittel. Da für diese Arbeit jedoch keine integrierten Systeme für ILP und CP(FD) zur Verfügung standen, wurde der kombinatorische Anteil, d. h. die Branching-Strategie, separat implementiert und nur für den LP-Anteil entsprechende Löser eingesetzt. Insgesamt erscheinen die in dieser Arbeit betrachteten Anordnungsprobleme also als interessante Benchmark-Probleme für kombinierte ILP/CP(FD)-Systeme, wie sie z. B. mit der ILOG Concert Technology jetzt verfügbar werden.

Der reine CP(FD) Ansatz von Ribeiro, Carravilla und Oliveira in „Applying Constraint Logic Programming to the Resolution of Nesting Problems [RCO99]“ (polygonale 2D-Objekte ohne Rotation) über ein Raster zulässiger Anordnungspunkte in der Ebene, erscheint mir nicht weiterführend und insbesondere für den höherdimensionalen Fall der dreidimensionalen Anordnungsprobleme nicht sinnvoll einsetzbar. Dieses Vorgehen ist von der Modellierung des Anordnungsraumes her vergleichbar mit Beasley „An Exact Two-Dimensional Non-Guillotine Cutting Tree Search Procedure [Bea85]“. Hier werden nur quaderförmige Objekte betrachtet, wobei sich die Modellierung für dreidimensionale Bin-Packing Probleme aber schon als nicht mehr zielführend erwiesen hat (vgl. [FS97b]).

4.3.2 Simulated Annealing

Simulated Annealing ist ein stochastisches Optimierungsverfahren, welches auf eine sehr breite Klasse von Problemen angewandt werden kann, da es einzig die Auswertung der Zielfunktion zu einer Konfiguration benötigt und ein sogenanntes Move-Set, um die aktuelle Konfiguration lokal zu verändern. Ableitungen der Zielfunktion oder der Restriktionen werden nicht benötigt, wobei eine Integration dieser Informationen in bestimmte Move-Typen innerhalb des Move-Sets prinzipiell möglich wäre, jedoch im allgemeinen nicht betrachtet wird. Simulated Annealing kann wegen des vordefinierten Move-Sets auch den lokalen Suchverfahren zugeordnet werden. Hierbei ist jedoch zu beachten, daß Simulated Annealing wegen seiner stochastischen Natur in der Theorie ein lokales Optimum immer mit einer Wahrscheinlichkeit echt größer Null wieder verlassen kann. Insbesondere in der ingenieurwissenschaftlichen Literatur wird Simulated Annealing daher als globales Optimierungsverfahren propagiert. Dies trifft jedoch nur für den Grenzfall unendlicher Laufzeiten zu (siehe [LM86]), ist also in der Praxis nicht gegeben. Vielmehr muß hier oftmals mit einer deutlichen Varianz in der Lösungsgüte verschiedener, wegen der Randomisierung unterschiedlicher Läufe gerechnet werden.

Das Grundprinzip besteht darin, daß den einzelnen Move-Typen des Move-Sets Auswahlwahrscheinlichkeiten zugeordnet werden (diese können im Verlauf des Verfahrens angepaßt werden) und jeweils eine Konfigurationsveränderung bzgl. eines zufällig gewählten Move-Typs ausgeführt wird. Falls diese Konfiguration einen schlechteren Zielfunktionswert aufweist als die Ausgangskonfiguration, so wird sie nur mit einer bestimmten Wahrscheinlichkeit akzeptiert. Die Akzeptanzwahrscheinlichkeit für verschlechternde Lösungen nimmt dabei im Verlauf des Verfahrens über einen sogenannten Temperaturparameter gesteuert ab. Dies erfolgt in Analogie zu physikalischen Abkühlungsprozessen und der fallenden Wahrscheinlichkeit einer Verschlechterung des Energieniveaus des Systems.

Für Anordnungsprobleme sind hier verschiedene Umsetzungsmöglichkeiten vorhanden. In [SC93] werden z. B. nur überschneidungsfreie Konfigurationen betrachtet, wobei die Move-Typen Anordnungsregeln einer „Shape Grammar“ entsprechen. Problematisch ist hier jedoch, daß die Veränderungen der Zielfunktion nicht an die Verringerung der Akzeptanzwahrscheinlichkeiten angepaßt werden können und daher kein Konvergenzverhalten ermöglichen.

Bei der Anwendung von Simulated Annealing auf Anordnungsprobleme hat sich daher eine Modellierung durchgesetzt, bei der sich die Objekte zu Beginn des Verfahrens beliebig überschneiden dürfen (vgl. [SC94] und [HL95]). Die Zielfunktion wird im Verlauf des Verfahrens dann durch einen Term dominiert, der Objektüberschneidungen mit Strafkosten belegt, wodurch diese entfernt werden. Ein erheblicher Laufzeitanteil wird daher für

die Überschneidungsminimierung eingesetzt. Dies führte zu einer Reihe von Veröffentlichungen, zur Beschleunigung der Schnittberechnungen (siehe z. B. [CDY98]), zum anderen zu dem Versuch, vielversprechende Regionen des Konfigurationsraumes schneller zu erkennen. Eine gewisse Iterationszahl, um einen quasistationären Zustand zu erreichen, ist jedoch inhärenter Verfahrensbestandteil und kann nicht beliebig reduziert werden.

[HL96, Hec98] entwickelt daher randomisierte Verfahren, bei denen Überschneidungsfreiheit wieder a priori sichergestellt wird. Dies wird auch für den 3D-Fall als im Prinzip erfolgversprechende Methode angesehen, beruht jedoch auf effizienten und numerisch stabilen Algorithmen für Schnitt und Vereinigung von Distanzpolyedern, die im 3D-Fall nicht gegeben ist.

Der wesentliche Vorteil von Simulated Annealing Verfahren liegt in einer größtmöglichen Flexibilität bei der Wahl der Zielfunktion und der mit verhältnismäßig geringem Aufwand zu integrierenden Randbedingungen, da hierzu lediglich die Bewertung der Moves entsprechend angepaßt werden muß. Mit jedem Lauf wird jedoch jeweils nur genau eine Lösung erzeugt, wobei diese legale, überschneidungsfreie Lösung erst am Ende der Laufzeit zur Verfügung stehen.

Die in dieser Arbeit vorgestellten Methoden zur globalen Optimierung können demgegenüber substantiell unterschiedliche Lösungen anbieten, da prinzipiell alle möglichen Anordnungen betrachtet werden. Dies ermöglicht dem Konstrukteur die abschließende Beurteilung und Auswahl zwischen Lösungen mit ähnlichem Zielfunktionswert. Zudem können bei geeigneter Implementierung mittels Tiefensuche in geeigneten Teilbäumen schon frühzeitig erste legale Lösungen geliefert werden, die mit fortschreitender Zeit verbessert werden. Das globale Optimum wird im allgemeinen sogar weit vor Ende der Gesamtlaufzeit gefunden, so daß sich ein vorzeitiger Abbruch des Verfahrens bei Erreichen von zufriedenstellenden Lösungen im Gegensatz zu Simulated Annealing Verfahren anbietet.

Kapitel 5

Polyederseparation

In diesem Kapitel werden zunächst in Abschnitt 5.1 Polygone und Polyeder definiert. Dies sind jeweils abgeschlossene, linear beschränkte Teilmengen der Ebene bzw. des Raumes. Daran anschließend werden verschiedene geometrische Strukturen und Verfahren zur Separation polyedrischer Objekte vorgestellt.

Insbesondere wird der Schnittest zwischen zwei Polyedern darauf zurückgeführt, ob ein Referenzpunkt innerhalb eines aus den gegebenen Polyedern A, B konstruierten *Distanzpolyeders* $D_{A,B}$ liegt. Die wesentliche Operation zur Konstruktion des Distanzpolyeders ist die Bildung der *Minkowski-Summe* zweier Polyeder, was in Abschnitt 5.2 näher betrachtet wird (zu Minkowski-Operationen siehe auch Ghosh [Gho90]). Da die paarweisen Distanzpolyeder vorberechnet werden können und die Problemstellung „Punkt in Polyeder“ einfacher zu lösen ist als eine vollständige Neuberechnung von Polyederschnitten, ergibt sich hieraus ein effizientes Verfahren für den Schnittest. Zudem kann diese Charakterisierung überschneidungsfreier Objektlagen mit Methoden der Linearen Programmierung formuliert werden, so daß für die Berechnung von Objektanordnungen robuste und effiziente Optimierungsverfahren zur Verfügung stehen. In der Literatur werden insbesondere bei der Anordnung von Polygonen in der Ebene auch die Begriffe Hodograph und No-Fit-Polygon anstelle von Distanzpolygon verwendet (siehe [ST93, TLS87]).

Eine Alternative zu Distanzpolyedern stellen *Randspuren polyedrischer Objekte* (Polyhedral Tracings, kurz \mathcal{P} -Spuren) dar, die in Abschnitt 5.3 eingeführt werden. Über eine spezielle Faltungsoperation (Convolution) auf diesen Randspuren erhält man ebenfalls eine Charakterisierung überschneidungsfreier relativer Anordnungen für polyedrische Objekte. Insbesondere ist der Rand der Minkowski-Summe im Faltungsprodukt zweier \mathcal{P} -Spuren enthalten. \mathcal{P} -Spuren können im Gegensatz zu einfachen Polyedern Selbstüberschneidungen enthalten, woraus sich zwei wesentliche Vorteile gegenüber der Minkowski-Summe ableiten lassen: Zunächst ist die Worst-Case Beschreibungskomplexität von \mathcal{P} -Spuren für nicht konvexe Objekte deutlich niedriger als die der Minkowski-Summe, da die Aufspaltung von Flächen in

Teilgebiete durch Überschneidungen nicht explizit dargestellt werden muß (im Worst-Case $O(nm)$ gegenüber $O(n^3m^3)$ für Ausgangspolyeder mit $O(n)$ und $O(m)$ Randobjekten). Der zweite Vorteil ist algorithmischer Natur: Da die in der Faltung enthaltenen Randobjekte, insbesondere Flächen, bis auf Verschiebung unverändert von den Ausgangsobjekten übernommen werden, müssen keine Objektschnitte berechnet werden. Numerische Probleme werden daher vermieden und die topologisch korrekte Erzeugung der Faltung ist auf kombinatorische Entscheidungen bzw. die Topologie der Ausgangspolyeder zurückgeführt.

Distanzpolyeder und \mathcal{P} -Spuren können jeweils für Objekte mit fester Orientierung im Raum berechnet werden. Im lokalen Optimierungsverfahren in Kapitel 3 werden jedoch kontinuierliche Objektdrehungen betrachtet, so daß eine explizite Vorberechnung dieser Strukturen ausscheidet. Wie in Kapitel 3 gezeigt, existiert zwar nur eine endliche Anzahl unterschiedlicher Topologien, die über eine Äquivalenzrelation klassifiziert werden können; deren vollständige Berechnung ist für das lokale Optimierungsverfahren jedoch nicht sinnvoll, da im Gegensatz zum globalen Verfahren nicht alle relativen Objektlagen betrachtet werden und keine Information zu Topologieübergängen benötigt wird.

Speziell für die Anpassung von Separationsebenen im lokalen Optimierungsverfahren wird in Abschnitt 5.4 der Algorithmus von Gilbert-Johnson-Keerthi [GJK88] eingeführt. In diesem Verfahren wird das Distanzpolyeder nicht explizit berechnet, sondern jeweils durch einen Simplex approximiert, der am Ende des iterativen Verfahrens zum minimalen Objektabstand bzw. einer Separationsebene korrespondiert. Bei entsprechender Implementierung (siehe hierzu insbesondere van den Bergen [vdB99]) können die Berechnungen der vorangehenden Iterationsschritte jeweils effizient weiterverwendet werden, was insbesondere auch nach einem lokalen Optimierungsschritt möglich ist.

5.1 Definition polyedrischer Objekte

Definition 7 (Ebenes Polygon)

Ein Polygon $P \subset \mathbb{E}^2$ ist eine abgeschlossene Teilmenge der euklidischen Ebene, die linear begrenzt ist, wobei gilt: P besteht aus einer endlichen Punktmenge $\{e_1, \dots, e_k\}$, den Ecken, sowie einer gleich mächtigen Menge von Strecken $\overline{e_i, e_{i+1}}$ sowie $\overline{e_k, e_1}$, den Kanten, die jeweils zwei aufeinanderfolgende Eckpunkte verbinden, sowie dem von diesem Streckenzug begrenzten, abgeschlossenen Gebiet der Ebene. Jede Ecke ist dabei Schnittpunkt von genau zwei Kanten.

Ein Polygon heißt *einfach*, wenn sich Kanten nur in Eckpunkten schneiden.

Definition 8 (Polyeder)

Ein Polyeder $P \subset \mathbb{E}^3$ ist eine abgeschlossene Teilmenge des euklidischen Raumes, die durch ebene Flächen, d. h. linear begrenzt ist, wobei gilt: P besteht aus einer endlichen Menge von Ecken und Kanten, die jeweils einfache, ebene Polygone bilden, die Flächen, sowie dem von diesen Flächen begrenzten, abgeschlossenen Gebiet der Ebene. Jede Kante muß dabei in genau zwei Flächen enthalten sein; diese Flächen werden als adjazent bezeichnet. Die Menge der Flächen sei ferner zusammenhängend, d. h. eine minimale Menge mit den vorgenannten Eigenschaften.

Ein Polyeder heißt *einfach*, wenn nur adjazente Flächen gemeinsame Punkte enthalten. Polygone bzw. Polyeder heißen konvex, falls die zugrunde liegenden Teilmengen des \mathbb{R}^n konvex sind (siehe Def. 26).

Die Basis für die Objektseparation bilden der im folgenden nochmals aufgeführte Satz 5 (Seite 34) zur linearen Separierbarkeit konvexer Mengen und der Satz von Minkowski, der für konvexe Polyeder die Beschränkung der Separationsbedingungen auf die Eckenmenge rechtfertigt.

Satz (Lineare Separierbarkeit [SW70])

Zwei Punktmengen sind linear separierbar genau dann, wenn ihre konvexen Hüllen sich nicht schneiden.

Satz 8 (Satz von Minkowski [Fis78])

Jedes konvexe Polyeder $P \subset \mathbb{R}^n$ ist die konvexe Hülle seiner Extrempunkte.

Hieraus folgt, daß zwei konvexe Polyeder genau dann überschneidungsfrei angeordnet sind, wenn sie linear separierbar sind. Um Separationsebenen für Polyeder zu bestimmen, wird im folgenden der Begriff der Minkowski-Summe eingeführt. Ziel ist es, eine translationsinvariante Beschreibung des überschneidungsfreien Gebietes zwischen zwei Polyedern zu definieren. Dies wird über die Definition des Distanzpolyeders erreicht. Hiermit wird der Schnitttest von Polyedern darauf zurückgeführt, ob ein beliebig aber fest gewählter Referenzpunkt innerhalb oder außerhalb des Distanzpolyeders liegt. Wir präzisieren zunächst den Distanzbegriff zwischen Polyedern als minimalen Punktabstand:

Definition 9 (Distanz zwischen Polyedern)

Die Distanz zwischen zwei Polyedern A, B wird definiert als

$$d(A, B) := \min\{\|a - b\| \mid a \in A, b \in B\}$$

Man beachte, daß nach dieser Definition immer $d(A, B) \geq 0$ gilt. Im Falle einer Objektüberschneidung liefert sie alleine noch kein Maß für die Tiefe der Überschneidung (Intersection Depth) und keine Separationsrichtung. Für alternative Distanzmaße siehe z. B. [GO94] und [Mil98b].

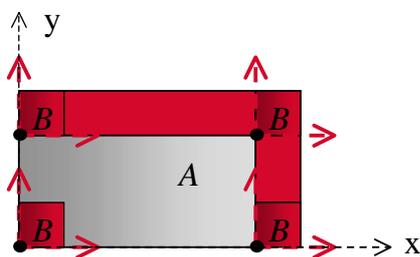
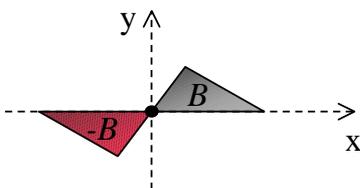


Abbildung 5.1: Minkowski-Summe eines Rechtecks und eines Quadrates

Abbildung 5.2: Punktgespiegelte Menge $-B$

5.2 Distanzpolyeder und Minkowski-Summe (Hodograph)

Die Minkowski-Summe zweier Mengen $A, B \subset \mathbb{R}^n$ ist die punktweise Addition aller Punkte aus A mit allen Punkten aus B . Anschaulich kann sie auch als Vergrößerung von Objekt A um Objekt B angesehen werden. Hierbei wird der lokale Koordinatenursprung von Objekt B , dieser wird auch als *Referenzpunkt* bezeichnet, an jeden Punkt von A verschoben bzw. angelegt. Siehe hierzu Abb. 5.1, in der auch jeweils das lokale Koordinatensystem von B dargestellt ist.

Definition 10 (Minkowski-Summe)

Seien $A, B \subset \mathbb{R}^n$. Die Minkowski-Summe ist wie folgt definiert:

$$A \oplus B := \{a + b \mid a \in A, b \in B\} \quad (5.1)$$

Zudem benötigen wir für die Definition des Distanzpolyeders die am Koordinatenursprung punktgespiegelte Menge $(-B)$ zu einer gegebenen Ausgangsmenge B (siehe Abb. 5.2); im folgenden auch als *symmetrische Menge* bezeichnet.

Definition 11 (Symmetrische Menge)

$$(-B) := \{-b \mid b \in B\} \quad (5.2)$$

5.2.1 Distanzpolyeder: algebraische Definition

Der Zusammenhang zwischen der Minkowski-Summe und der überschneidungsfreien Anordnung von Teilmengen des \mathbb{R}^n , insbesondere also von Polyedern, ergibt sich aus der Definition des Distanzpolyeders $D_{A,B}$. Es folgen zunächst die formale Definition und die zentrale Aussage zur Überschneidungsfreiheit von Polyedern. Diese werden anschließend in Abschnitt 5.2.2 geometrisch motiviert.

Definition 12 (Distanzpolyeder / Hodograph)

Seien A, B zwei Polyeder. Das Distanzpolyeder $D_{A,B}$ ist definiert als

$$D_{A,B} := A \oplus (-B) \quad (5.3)$$

Satz 9 (Distanzpolyeder und Überschneidungsfreiheit)

1. Zwei Polyeder A, B überschneiden sich genau dann, wenn der Koordinatenursprung $\mathbf{0}$ innerhalb des Distanzpolyeders $D_{A,B}$ liegt.
2. Die Distanz, d. h. der minimale Abstand zwischen zwei Polyedern A, B entspricht dem minimalen Abstand des Koordinatenursprungs $\mathbf{0}$ zu $D_{A,B}$.

Beweis:

zu 1: Mit den obigen Definitionen gilt

$$\begin{aligned} A, B \text{ überschneiden sich} \\ &\Leftrightarrow \exists a \in A, b \in B \text{ mit } a = b \\ &\Leftrightarrow \mathbf{0} = a - b = a + (-b) \\ &\Leftrightarrow \mathbf{0} \in A \oplus (-B) \\ &\Leftrightarrow \text{Koordinatenursprung } \mathbf{0} \in D_{A,B} \end{aligned}$$

zu 2: Um $d(A, B) = d(D_{A,B}, \{\mathbf{0}\})$ zu zeigen, werden die beiden folgenden Annahmen zum Widerspruch geführt:

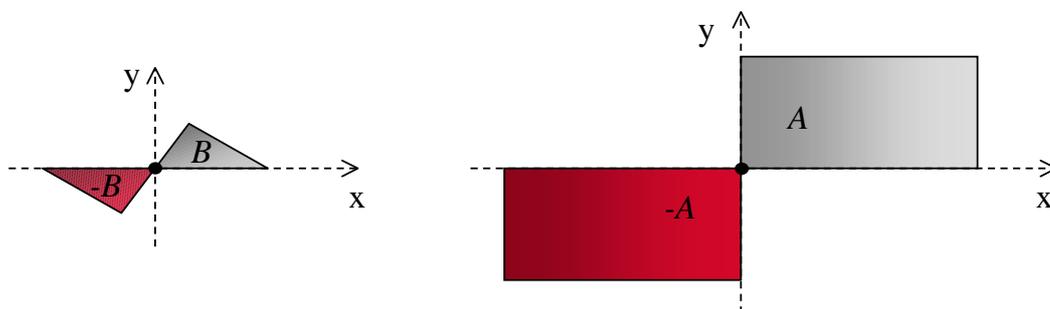
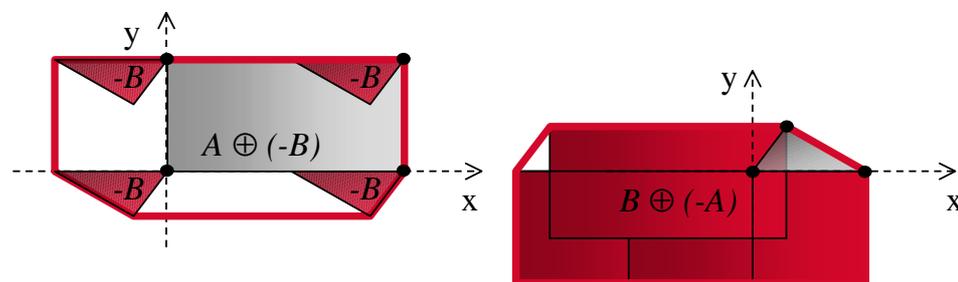
$$\begin{aligned} a) \quad &d(A, B) < d(D_{A,B}, \{\mathbf{0}\}) \\ b) \quad &d(A, B) > d(D_{A,B}, \{\mathbf{0}\}). \end{aligned}$$

zu 2a:

$$\begin{aligned} d(A, B) &\stackrel{a)}{<} d(D_{A,B}, \{\mathbf{0}\}) \\ &\Leftrightarrow \exists a \in A, b \in B \text{ mit } \|a - b\| < d(D_{A,B}, \{\mathbf{0}\}) \end{aligned} \quad (5.4)$$

$$\begin{aligned} \forall a \in A, b \in B : \quad &(a + (-b)) \in D_{A,B} \\ &\Rightarrow d(D_{A,B}, \{\mathbf{0}\}) \leq \|(a + (-b)) - \mathbf{0}\| = \|a - b\| \end{aligned} \quad (5.5)$$

$$\text{Insgesamt: } \|a - b\| \stackrel{5.4}{<} d(D_{A,B}, \{\mathbf{0}\}) \stackrel{5.5}{\leq} \|a - b\| \quad \text{Widerspruch!}$$

Abbildung 5.3: Ausgangspolyeder A, B und symmetrische MengenAbbildung 5.4: Distanzpolyeder $D_{A,B} = A \oplus (-B)$ und $D_{B,A} = B \oplus (-A)$

zu 2b:

$$\exists p \in D_{A,B} \text{ mit } d(D_{A,B}, \{\mathbf{0}\}) = \|p\| \quad (5.6)$$

$$p \in D_{A,B} \Rightarrow \exists a \in A, b \in B \text{ mit } p = (a + (-b)) \quad (5.7)$$

$$\begin{aligned} \text{Insgesamt: } d(D_{A,B}, \{\mathbf{0}\}) &\stackrel{5.6}{=} \|p\| \stackrel{5.7}{=} \|a + (-b)\| = \|a - b\| \\ &\geq d(A, B) \stackrel{b)}{>} d(D_{A,B}, \{\mathbf{0}\}) \quad \text{Widerspruch!} \end{aligned}$$

□

In den Abb. 5.3 und 5.4 sind zwei Polygone A, B jeweils mit ihrer symmetrischen Menge und den beiden Distanzpolygonen $A \oplus (-B)$ und $B \oplus (-A)$ dargestellt. Man beachte, daß die beiden Distanzpolyeder zwar symmetrische Mengen sind, aber nicht identisch. Die Reihenfolge der Summanden ist nur solange unerheblich, wie die Lage des Koordinatenursprungs bzgl. des Distanzpolyeders betrachtet wird, was jedoch nur für Polyeder mit fester Position im Raum bzw. in Ursprungslage gilt.

Im folgenden werden Translationen t_a, t_b der Polyeder A, B betrachtet, wobei der Schnittest auf das Distanzpolyeder in Ursprungslage $D_{A,B} = A \oplus (-B)$ zurückgeführt werden soll. In diesem Fall muß entschieden werden, ob der relative Ursprung $t_b - t_a$ innerhalb oder außerhalb des Distanzpolyeders $D_{A,B}$ liegt, wobei die Differenz $t_b - t_a$ konsistent mit der Wahl der Summanden bei der Definition des Distanzpolyeders sein muß.

Eigenschaft 10 (Überschneidungsfreiheit unter Translation)

Seien $\tilde{A} := A \oplus \{t_a\}$ und $\tilde{B} := B \oplus \{t_b\}$ die um t_a bzw. t_b verschobenen Bilder der Polyeder A, B . Dann gilt:

$$\tilde{A}, \tilde{B} \text{ überschneiden sich} \quad \Leftrightarrow \quad (t_b - t_a) \in A \oplus (-B)$$

Beweis:

$$\begin{aligned} \tilde{A}, \tilde{B} \text{ überschneiden sich} & \\ \Leftrightarrow \mathbf{0} \in \tilde{A} \oplus (-\tilde{B}) & \\ \Leftrightarrow \exists \tilde{a} \in \tilde{A}, \tilde{b} \in \tilde{B} \text{ mit } \mathbf{0} = \tilde{a} - \tilde{b} & \\ \Leftrightarrow \exists a \in A, b \in B \text{ mit } \tilde{a} = a + t_a, \tilde{b} = b + t_b & \\ \text{und } \mathbf{0} = (a + t_a) - (b + t_b) & \\ \Leftrightarrow \mathbf{0} = (a - b) + (t_a - t_b) & \\ \Leftrightarrow (t_b - t_a) = (a - b) & \\ \Leftrightarrow (t_b - t_a) \in A \oplus (-B) & \end{aligned}$$

□

In dieser Arbeit wird der Schwerpunkt auf die überschneidungsfreie Anordnung polyedrischer Objekte gelegt. Man kann jedoch auch fordern, daß ein Objekt innerhalb einer bestimmten Region plaziert werden muß. In diesem Fall ist eine vollständig überschneidende Anordnung von Objekt und zulässiger Region gesucht. Dieses Problem kann mit Hilfe der Minkowski-Differenz gelöst werden (siehe hierzu auch [Gho90]).

Um die Definition der Minkowski-Differenz einzuführen, betrachten wir eine zu Definition 10 äquivalente Beschreibung der Minkowski-Summe. Sei A_p die um den Vektor p verschobene Menge A ; mit obiger Notation $A_p = A \oplus \{p\}$. Dann gilt:

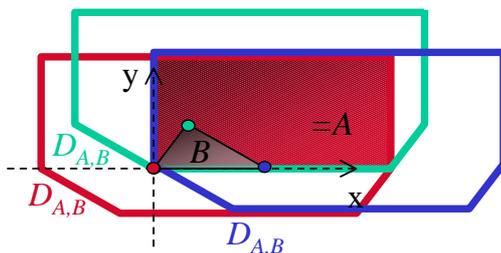
$$A \oplus B = \bigcup_{a \in A} B_a = B \oplus A = \bigcup_{b \in B} A_b \quad (5.8)$$

Bei der Minkowski-Differenz wird nun statt der Vereinigung der Schnitt aller Mengen A_b gebildet:

Definition 13 (Minkowski-Differenz)

$$A \ominus B := \bigcap_{b \in (-B)} A_b \quad (5.9)$$

In Abbildung 5.5 ist die Minkowski-Differenz $D_{A,B} \ominus (-B)$ zu dem in Abb. 5.4 gezeigten Distanzpolyeder $D_{A,B}$ dargestellt. Sie ergibt sich als Schnitt der drei eingezeichneten Umrisse, deren Referenzpunkt jeweils an einem Eckpunkt von B anliegt. Man beachte, daß die Differenz gerade dem Polyeder A entspricht. Allgemein gilt:

Abbildung 5.5: Minkowski Differenz $D_{A,B} \ominus (-B) = A$ **Eigenschaft 11 (Symmetrie Minkowski-Summe/Differenz)**

Seien $A, B \subset \mathbb{R}^n$ und $S = A \oplus B$. Dann gilt:

$$S = A \oplus B \Rightarrow S \ominus B = A \quad (5.10)$$

Man beachte, daß die Gleichung $(S \ominus B) \oplus B = S$ nicht für beliebige Mengen S, B gilt, sondern daß die Voraussetzung $\exists A$ mit $S = A \oplus B$ notwendig ist. Dies kann durch die Definition eines negativen Flächen- bzw. Volumenbegriffs vermieden werden, der hier jedoch nicht eingeführt werden soll (siehe hierzu [Sch91a]).

5.2.2 Distanzpolyeder: geometrische Motivation

Nach der algebraischen Definition des Distanzpolyeders soll nun die geometrische Vorgehensweise motiviert werden, da diese die Grundlage der algorithmischen Konstruktion ist. In Abb. 5.6 sind die beiden wesentlichen Sichtweisen zur Konstruktion des Distanzpolyeders $D_{A,B}$ skizziert: Die in (a) dargestellte Verschiebung von B um A ist mit dem in Abschnitt 5.3 behandelten algorithmischen Konzept von Randspuren polyedrischer Objekte (Polyhedral Tracings) verwandt; die in (b) dargestellte Addition von $-B$ an allen Eckpunkten von A mit der Bildung der konvexen Hülle. Die folgenden Abschnitte gehen näher auf die beiden Konstruktionsprinzipien und die jeweils damit verbundenen Eigenschaften des Distanzpolyeders ein.

In Abb. 5.6(a) ist die intuitive Konstruktionsmethode dargestellt. Hierbei wird Polyeder B berührend, aber nicht überschneidend am Rand von Polyeder A entlang verschoben. Dabei wird die Spur des Referenzpunktes von B aufgezeichnet. Aus der Konstruktion ist unmittelbar ersichtlich, daß die Polyeder A, B genau dann überschneidungsfrei sind, wenn der Referenzpunkt von B außerhalb von $D_{A,B}$ liegt¹.

Man beachte, daß die Kontur von $D_{A,B}$ aus den Kanten der Polyeder A und B gebildet wird. Für konvexe Polygone in der euklidischen Ebene \mathbb{E}^2

¹Mit den Bezeichnungen aus Eigenschaft 10 entspricht die Lage des Referenzpunktes von B der Verschiebung t_b ; die Lage von Polyeder A ist fix und wird mit $t_a = \mathbf{0}$ gewählt.

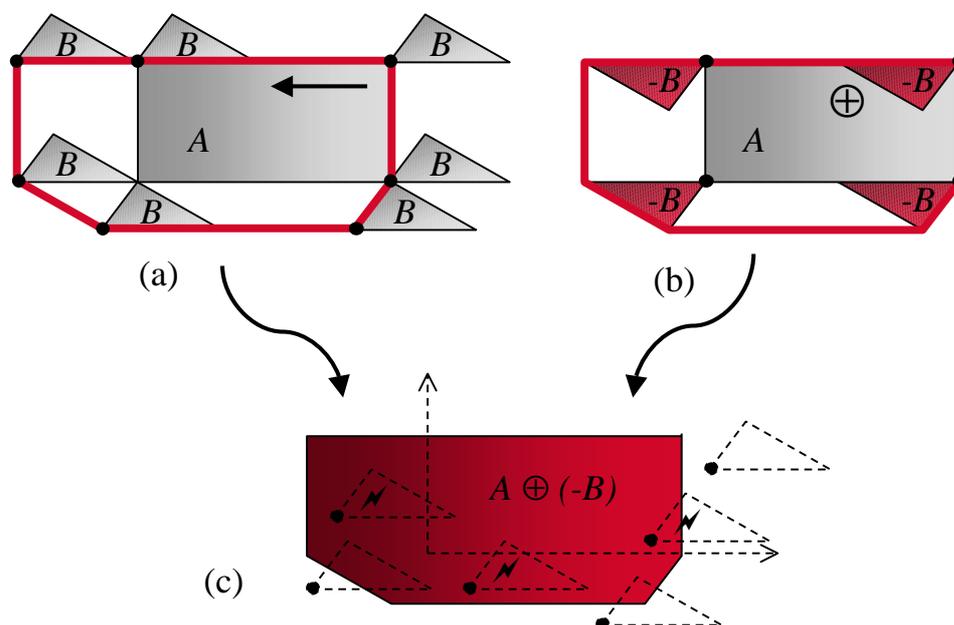


Abbildung 5.6: Konstruktion von $D_{A,B} = A \oplus (-B)$ durch Verschiebung (a), Hüllenbildung (b), und zulässige/überschneidende Lagen von B (c)

ergibt sich die Abfolge der Kanten aus der Sortierung nach Steigungswinkeln, womit ein einfaches und robustes Konstruktionsverfahren existiert. Bei räumlichen Polyedern wird die Kontur ebenfalls von Randobjekten der Ausgangspolyeder gebildet, wobei neue Flächen in Form von Parallelogrammen entstehen können, die von je einer Kante aus den beiden Ursprungspolyedern aufgespannt werden. Die Konstruktion des dreidimensionalen Distanzpolyeders ist mit einem lokalen Suchalgorithmus möglich, der auf dem Vergleich von Winkelbereichen beruht, die in Abschnitt 5.3 im Zusammenhang mit dem Konzept der Randspur polyedrischer Objekte eingeführt werden.

In Abb. 5.6 (b) ist die Konstruktion entsprechend der algebraischen Definition dargestellt. In diesem Fall wird der Referenzpunkt des punktgespiegelten Polyeders $(-B)$ formal an jedem Punkt von A angelegt. Algorithmisch kann man sich hierbei jedoch auf Ecken von A beschränken, da innere Punkte wegen der Linearität der Grenzflächen implizit überdeckt werden. Im Gegensatz zur vorangehenden Konstruktionsmethode wird $D_{A,B}$ nun nicht durch die Spur des Referenzpunktes von $-B$ bestimmt (diese entspräche dem Rand von A), sondern durch die Gesamtmenge bzw. Vereinigung aller von $-B$ überdeckten Punkte (vgl. Definition 13 von Minkowski-Summe und Differenz über Mengenvereinigung/-schnitt).

Für konvexe Polyeder entspricht das Distanzpolyeder $D_{A,B}$ daher gerade der konvexen Hülle der paarweisen Summen aller Eckpunkte von A mit Eckpunkten von $-B$. Es steht somit ein effizientes Verfahren zur Berech-

nung des Distanzpolyeders konvexer Objekte zur Verfügung. Nicht konvexe Objekte müssen jedoch zunächst durch geeignete Überdeckungen bzw. Zerlegungen dargestellt werden.

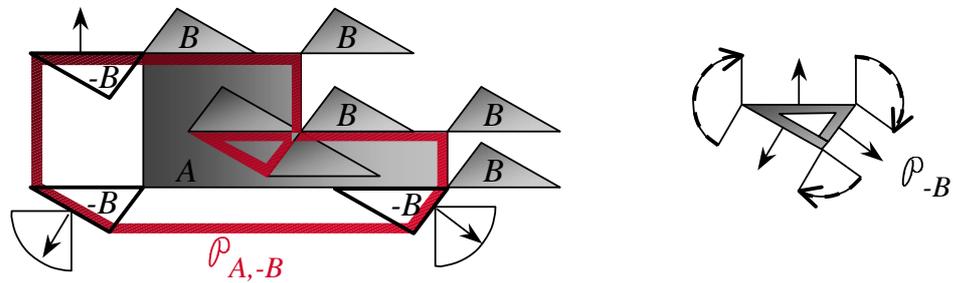
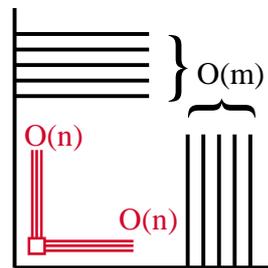
Bei der direkten Berechnung der Minkowski-Summe nicht konvexer Objekte können beliebige Schnitte der Ausgangsflächen erzeugt werden. Es sind somit numerische Implementierungsgesichtspunkte und Spezialfälle wie bei Polyederschnitt und -vereinigung zu berücksichtigen. Diese Problematik kann mit der im folgenden Abschnitt vorgestellten Randspur polyedrischer Objekte vermieden werden.

5.3 Randspur polyedrischer Objekte (\mathcal{P} -Spur, Polyhedral Tracing)

In diesem Abschnitt wird eine Alternative zu dem im vorigen Abschnitt vorgestellten Konzept der Minkowski-Summe eingeführt. Polyhedral Tracings², hier im folgenden als Randspur polyedrischer Objekte bzw. kurz \mathcal{P} -Spur bezeichnet, wurden von Guibas, Ramshaw und Stolfi in „A Kinetic Framework for Computational Geometry [GRS83]“ zunächst für zweidimensionale, polygonale Objekte eingeführt (siehe hierzu auch [Ram96]) und in [BGRR96] auf den dreidimensionalen Fall erweitert. Jedem Polyeder A ist eine \mathcal{P} -Spur \mathcal{P}_A zugeordnet, die nur die Randobjekte des Polyeders enthält, diese jedoch um eine lokale Orientierung bzw. Winkelbereiche erweitert. Entsprechend dem über die Minkowski-Summe \oplus definierten Distanzpolyeder $D_{A,B}$ wird hier über eine spezielle Faltungsoperation \otimes (Convolution) eine \mathcal{P} -Spur $\mathcal{P}_{A,B} = \mathcal{P}_A \otimes \mathcal{P}_B$ erzeugt; der Schnitttest wird dann wie im Falle des Distanzpolyeders $D_{A,B}$ darauf zurückgeführt, ob ein Punkt innerhalb oder außerhalb von $\mathcal{P}_{A,B}$ liegt.

Im Gegensatz zur Minkowski-Summe, die für polyedrische Ausgangsobjekte wieder ein einfaches Polyeder definiert, kann die \mathcal{P} -Spur Selbstüberschneidungen aufweisen (siehe Abb. 5.7). Die Beschreibungskomplexität der \mathcal{P} -Spur kann hierdurch wesentlich geringer sein, da die Selbstüberschneidungen nicht explizit in einzelne Randelemente aufgebrochen werden müssen. Die Minkowski-Summe kann für zwei Ausgangspolyeder mit $O(m)$ und $O(n)$ Randelementen im ungünstigsten Fall $O(m^3n^3)$ Randobjekte enthalten (siehe Abb. 5.8 und [KOS91]), die Größe der \mathcal{P} -Spur ist jedoch auf $O(mn)$ begrenzt. Eine \mathcal{P} -Spur mit k Randelementen kann dabei in einer von der Ergebnisgröße abhängigen Laufzeit von $O((m+n+k) \log^3(m+n) \alpha(m+n))$ berechnet werden [BGRR96, BGR96]. Man beachte hierzu, daß die Randobjekte der Minkowski-Summe $D_{A,B}$ und der \mathcal{P} -Spur $\mathcal{P}_{A,B}$ im Falle konvexer Polyeder identisch sind und die Beschreibungskomplexität in diesem Fall durch $O(m+n)$ begrenzt ist. Mit der \mathcal{P} -Spur $\mathcal{P}_{A,B}$ sind zudem zusätzliche

²Wir beschränken uns in dieser Arbeit auf dreidimensionale, polyedrische Objekte und deren Tracings. Der Begriff des Tracings kann jedoch allgemeiner definiert werden.

Abbildung 5.7: \mathcal{P} -Spur zweier Polygone mit SelbstüberschneidungAbbildung 5.8: Worst Case Komplexität $O(m^2 n^2)$ der Minkowski-Summe [KOS91]

Informationen über die Topologie der Überschneidungsregion gegeben, da sie die Euler-Poincaré Charakteristik und die Windig Number zu jedem Punkt liefert (siehe Abschnitt 5.3.3; insbesondere Satz 12).

Für die algorithmische Umsetzung ist jedoch entscheidend, daß alle Randobjekte der \mathcal{P} -Spur $\mathcal{P}_{A,B}$ schon mit den Ausgangsobjekten gegeben sind und keine numerisch instabilen und kombinatorisch aufwendigen Schnittbildungen vorgenommen werden müssen. Zusätzlich zu den gegebenen Randobjekten entstehen lediglich Parallelogrammflächen, wobei je eine Randkante aus Polyeder A und eine aus B entstammt. Grundsätzlich sind daher sehr einfache und robuste Implementierungen zur Berechnung der \mathcal{P} -Spur möglich³.

In dieser Arbeit wird die Faltungsoperation bzw. Convolution nur für den speziellen Fall abgeschlossener, räumlicher, polyedrischer Objekte und deren Randspuren betrachtet. Dies stellt eine Vereinfachung bzgl. der allgemeineren Theorie dar, die mit unterschiedlichen Bezeichnungen und unterschiedlichem Fokus in der Literatur behandelt wird. Basch, Guibas, Ramkumar, Ramshaw [BGRR96] beschränkt sich auf den \mathbb{R}^3 und setzt unter der Bezeich-

³Der in [BGRR96] skizzierte Algorithmus enthält demgegenüber eine numerisch kritische Koordinatentransformation (zur Transformation und Datenstruktur siehe auch [BGR96]), wodurch die oben genannten Vorteile teilweise wieder aufgehoben werden. Es ist hier eine Abwägung zwischen numerischer Robustheit und garantierter Worst-Case Laufzeit einzugehen.

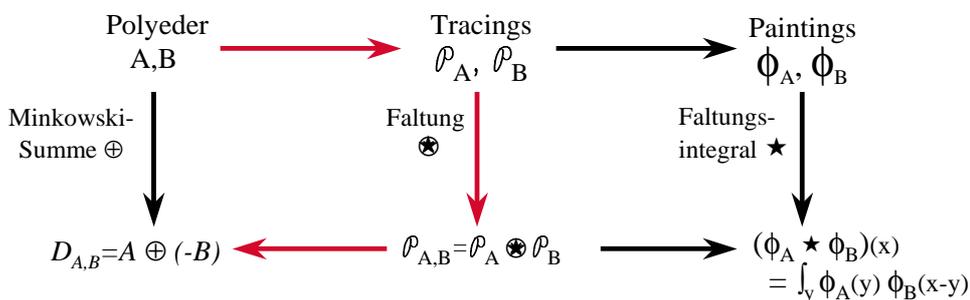


Abbildung 5.9: Minkowski-Summe, Faltungsoperation auf \mathcal{P} -Spuren und Faltungsintegral über Färbungen

nung *Tracing* den Schwerpunkt auf geometrische Anwendungen. Tracings werden dort jedoch allgemeiner definiert, insbesondere erfolgt zunächst keine Einschränkung auf polyedrische Objekte, und erlauben zudem die Charakterisierung offener und abgeschlossener Mengen. Schapira [Sch91b] diskutiert unter der Bezeichnung *Constructible Functions* den Fall für konvexe, kompakte Teilmengen des \mathbb{R}^n aus algebraischer Sicht, stellt aber den Bezug zur geometrischen Interpretation her. Chen stellt in [Che99] unabhängig von den beiden vorgenannten Arbeiten eine rein algebraische Behandlung vor. Dort wird eine *Minkowski Algebra on Indicator Functions* offener und abgeschlossener konvexer Teilmengen des \mathbb{R}^n eingeführt und mit algebraischen Mitteln untersucht.

In dieser Arbeit gehen wir wie folgt vor: Im folgenden Abschnitt 5.3.1 werden \mathcal{P} -Spuren und die darauf definierte Faltungsoperation \otimes zunächst algorithmisch eingeführt. Die formale Definition und der theoretische Hintergrund mit Rückgriff auf Färbungen (Paintings) und die darauf über ein Faltungsintegral definierte Operation \star (man beachte den Unterschied zu \otimes) wird dann in Abschnitt 5.3.3 behandelt. Abb. 5.9 zeigt den Zusammenhang zwischen der Minkowski-Summe, der Faltungsoperation auf \mathcal{P} -Spuren und dem Faltungsintegral $\int \phi$ über Färbungen. Aus Polyedern können Tracings abgeleitet werden, die wiederum speziellen Paintings entsprechen. Aus den Verknüpfungsergebnissen können dann wiederum $\mathcal{P}_{A,B}$ und das Distanzpolyeder $D_{A,B}$ gewonnen werden. Für die algorithmische Behandlung ist der rot dargestellte Weg von Bedeutung. Die inhaltliche Darstellung in Abschnitt 5.3.1 orientiert sich an [BGRR96] und beruht teilweise auf gemeinsamen Arbeiten mit C. Duvenbeck (siehe auch [Duv01]).

5.3.1 Faltungsoperation auf \mathcal{P} -Spuren polyedrischer Objekte

In diesem Abschnitt wird zunächst die anschauliche Bedeutung der \mathcal{P} -Spur eines einzelnen Polyeders vorgestellt und dann gezeigt, wie mit Hilfe einer speziellen Faltungsoperation \otimes zu zwei gegebenen polyedrischen Objekten

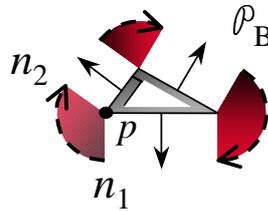


Abbildung 5.10: Normalenvektoren und Winkelbereiche eines Polyeders

A, B eine \mathcal{P} -Spur $\mathcal{P}_{A,B}$ bestimmt werden kann, so daß der Überschneidungstest für die Polyeder A, B wie im vorigen Abschnitt auf die Lage eines Referenzpunktes innerhalb oder außerhalb von $\mathcal{P}_{A,-B}$ zurückgeführt werden kann.

Grundidee der \mathcal{P} -Spur ist es, den Randpunkten p eines Polyeders zusätzliche Richtungsindikatoren n_p zuzuordnen, so daß sich jeweils Zustandstupel (p, n_p) ergeben. Wir betrachten zunächst Abb. 5.10 mit einem zweidimensionalen Beispiel. Für einen Punkt p auf einer Kante (Fläche bei Polyedern) ist der Richtungsindikator n_p eindeutig bestimmt und entspricht dem nach außen weisenden Normalenvektor⁴. Für einen Eckpunkt p überstreicht der Richtungsindikator hingegen den gesamten *Winkelbereich* zwischen den Normalenvektoren n_1, n_2 der beiden angrenzenden Kanten. Alle zwischen (p, n_1) und (p, n_2) liegenden Zustände sind in der \mathcal{P} -Spur enthalten, womit ein stetiger Zustandsübergang realisiert wird. Die Winkelbereiche von Ecken stellen offene Mengen dar (siehe Def. 14 auf Seite 72).

Formal können zur Definition der Zustandstupel (p, n_p) zwei Abbildungen für den Orts- (Komponente 1, p) und den Richtungsindikator (Komponente 2, n_p) über einer geeigneten Grundmenge⁵, z. B. der Einheitssphäre \mathcal{E}^3 wie in Abb. 5.11, betrachtet werden. Der Ortsindikator $o : \mathcal{E}^3 \rightarrow \mathbb{R}^3$ liefert jeweils Punkte auf der Polyederoberfläche; der Richtungsindikator $r : \mathcal{E}^3 \rightarrow \mathbb{E}$ jeweils Normalenvektoren. Hierbei wird gefordert, daß der Richtungsindikator r immer senkrecht zum Ortsindikator o variiert und daß sich stetige Zustandsübergänge ergeben. Konkret ergibt sich für ein polyedrisches Objekt P eine Unterteilung der Grundmenge in die verschiedenen Klassen von Randobjekten (vgl. Abb. 5.11):

Fläche f : Der Richtungsindikator r ist der nach außen gerichtete Normalenvektor und ist für alle roten Punkte der Einheitssphäre gleich bzw. konstant, während der Ortsindikator einer Bijektion auf die quadratische Polyederfläche entspricht.

Kante k : Der Richtungsindikator r variiert auf dem orthogonal zum Ort-

⁴Die Unterscheidung zwischen abgeschlossenen/offenen Mengen kann über die Orientierung des Normalenvektors nach außen/innen erfolgen; in dieser Arbeit sollen aber nur abgeschlossene Mengen betrachtet werden.

⁵allgemein: orientierte, kompakte Mannigfaltigkeit ohne Rand

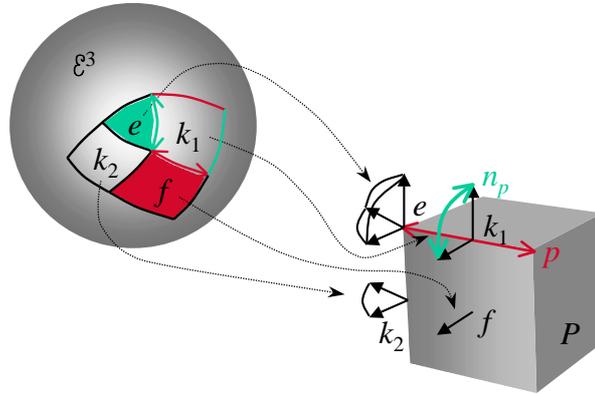


Abbildung 5.11: Skizze zu Orts- (rot) und Richtungsindikator (grün) der \mathcal{P} -Spur eines polyedrischen Objektes [BGR96]

sindikator der Kante verlaufenden Großkreisbogen zwischen den Normalenvektoren der beiden adjazenten Flächen; es wird immer der kürzere Bogen bzw. der kleinere Winkelbereich zwischen den Normalenvektoren gewählt. Der Ortsindikator o variiert jeweils über die Punkte der Polyederkante.

Ecke e : In diesem Fall ist der Ortsindikator o konstant über der grünen Fläche der Einheitskugel und liefert jeweils den Eckpunkt des Polyeders; der Richtungsindikator r variiert innerhalb des von den Normalenvektoren der adjazenten Flächen aufgespannten Kegels.

Algorithmisch werden jedoch nur Winkelbereiche der Randobjekte Fläche, Kante und Ecke betrachtet, da der Richtungsindikator für alle Punkte einer Fläche bzw. für alle Punkte einer Kante dieselbe Menge von Normalenvektoren liefert (die Ortsauflösung ist durch das betrachtete Randobjekt selbst gegeben).

Definition 14 (Winkelbereich $\omega(o)$)

Sei $\eta(f)$ der Normalenvektor einer Randfläche f . Der Winkelbereich $\omega(o)$ für ein Randobjekt o ist definiert als ($\overrightarrow{\quad}$: normierter Vektor):

$$\omega(o) := \begin{cases} \text{Fläche (2D Kante):} & \{\eta(o)\} \\ \text{Kante (2D Ecke)} & : \{\overrightarrow{\lambda \eta(f_1) + (1 - \lambda) \eta(f_2)} \mid \\ & \lambda \in]0, 1[, f_1, f_2 \text{ adjazent an } o\} \\ \text{Ecke} & : \{\overrightarrow{\sum_i \lambda_i \eta(f_i)} \mid \\ & \lambda_i > 0, f_i \text{ inzident zu } o\} \end{cases}$$

Mit diesem Grundverständnis der Randspur eines polyedrischen Objektes kann die Faltungsoperation $\mathcal{P}_A \otimes \mathcal{P}_B$ für zwei Polyeder eingeführt

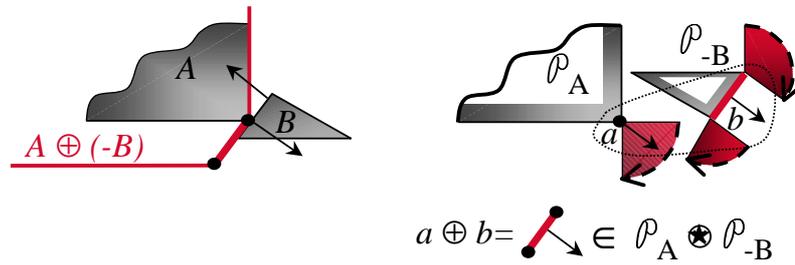


Abbildung 5.12: Überschneidung von Winkelbereichen und neue Randobjekte

werden. Wir betrachten zunächst wieder ein zweidimensionales Beispiel. Abb. 5.12 zeigt eine Gegenüberstellung mit dem Distanzpolyeder $D_{A,B} = A \oplus (-B)$. Eine Kante von B kann genau dann berührend an einer Ecke von A entlang verschoben werden, wenn die entsprechenden Richtungsvektoren von A und B entgegengesetzt orientiert sind, die Randobjekte also in einem naiven Sinne einander „zugewandt“ sind. Der Normalenvektor der Kante b von \mathcal{P}_{-B} muß also innerhalb des Winkelbereiches der Ecke a von \mathcal{P}_A liegen, d. h. $\omega(a) \cap \omega(b) \neq \emptyset$. Die Randobjekte werden hierbei punktweise addiert, d. h. das neue Randobjekt $o_{a,b}$ der Faltung $\mathcal{P}_{A,-B}$ entspricht $a \oplus b$. Wir halten fest:

$$\left. \begin{array}{l} a \in \mathcal{P}_A, b \in \mathcal{P}_{-B} \text{ mit} \\ \omega(a) \cap \omega(b) \neq \emptyset \end{array} \right\} \Leftrightarrow o_{a,b} = a \oplus b \in \mathcal{P}_{A,-B} \quad (5.11)$$

Der Richtungsindikator bzw. Winkelbereich von $o_{a,b}$ entspricht $\pm(\omega(a) \cap \omega(b))$, wobei die Vorzeichenwahl von den Konvexitätseigenschaften der Ausgangsobjekte abhängt. Wir betrachten hierzu Abb. 5.13. Sowohl die konvexe als auch die nicht konvexe Ecke von \mathcal{P}_A erzeugen mit der rechten, oberen Kante von \mathcal{P}_{-B} ein Randobjekt von $\mathcal{P}_{A,-B}$. Im Falle der konvexen Ecke stimmt der Richtungsindikator mit dem Schnitt der Winkelbereiche überein und ist nach außen orientiert. Im Falle der nicht konvexen Ecke muß der Richtungsindikator jedoch umgekehrt werden, um eine konsistente Orientierung des Faltungsproduktes zu erhalten. In der Abbildung ist eine Orientierung der Randspur durch die Pfeilspitzen symbolisiert; das Außengebiet von $\mathcal{P}_{A,-B}$ muß immer in Pfeilrichtung gesehen rechts liegen.

Um dies formal zu fassen, definieren wir für jedes Randobjekt o neben dem Winkelbereich $\omega(o)$ ein zusätzliches Vorzeichen $\sigma(o)$, wobei in Abb. 5.14 die Vorzeichen für dreidimensionale Polyederecken dargestellt sind. Ecken, an denen ausschließlich konvexe bzw. nicht konvexe Kanten angrenzen, erhalten ein positives Vorzeichen; Ecken mit sowohl konvexen, als auch nicht konvexen Kanten ein negatives Vorzeichen. Man beachte, daß hier schon eine Normalisierung der Polyederdarstellung auf genau drei Kanten je Ecke vorausgesetzt wird, die im folgenden noch näher beschrieben wird.

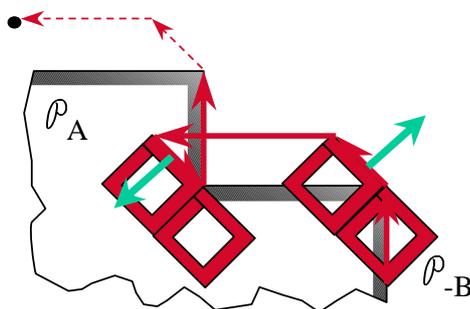
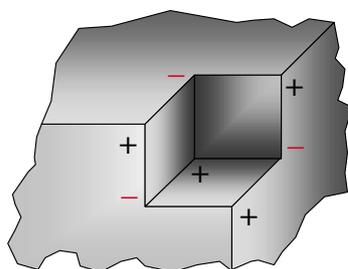


Abbildung 5.13: Richtungsindikator einer konvexen und nicht konvexen Ecke

Abbildung 5.14: Vorzeichen $\sigma(o)$ für (nicht-)konvexe Ecken**Definition 15 (Vorzeichen für Randobjekte $\sigma(o)$)**

Sei o Randobjekt einer \mathcal{P} -Spur. Dann wird das Vorzeichen $\sigma(o)$ wie folgt definiert:

$$\sigma(o) := \begin{cases} \text{Fläche (2D Kante):} & +1 \\ \text{Kante (2D Ecke)} & : +1, \text{ falls } o \text{ konvex} \\ & \quad -1, \text{ sonst} \\ \text{Ecke} & : +1, \text{ falls alle inzidenten Kanten konvex} \\ & \quad \text{oder alle inzidenten Kanten konkav} \\ & \quad -1, \text{ sonst (siehe Abb. 5.14)} \end{cases}$$

Damit wird nun die Faltungsoperation auf Randspuren polyedrischer Objekte eingeführt. Diese wird zum einen auf der Menge der Zustandstapel und zum anderen auf der Menge der Randobjekte betrachtet.

Definition 16 (Faltung)

Seien $\mathcal{P}_A, \mathcal{P}_B$ Randspuren polyedrischer Objekte. Für die Zustandstapel der Faltung $\mathcal{P}_{A,B} := \mathcal{P}_A \otimes \mathcal{P}_B$ gilt:

$$\left. \begin{array}{l} (a, n_a) \in \mathcal{P}_A, (b, n_b) \in \mathcal{P}_B \\ \text{mit } (n_a = n_b) =: n \end{array} \right\} \Leftrightarrow (a \oplus b, \sigma(a)\sigma(b)n) \in \mathcal{P}_{A,B} \quad (5.12)$$

Seien ferner $\mathcal{E}_A, \mathcal{K}_A, \mathcal{F}_A$ jeweils die Menge der Eckpunkte, Kanten und Flächen einer Randspur \mathcal{P}_A . Dann gilt für die Mengen der Randobjekte von

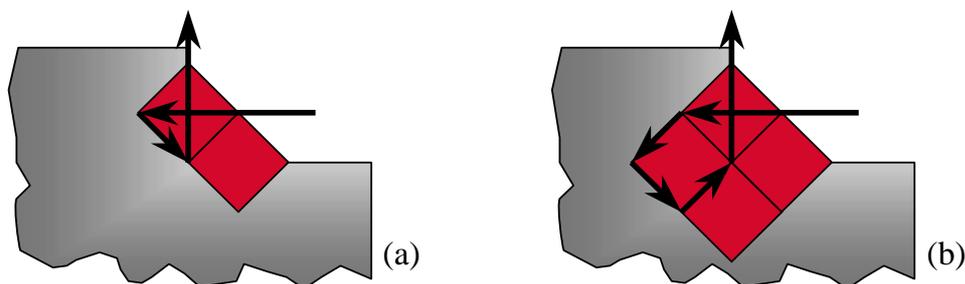


Abbildung 5.15: Winkelbereiche nicht konvexer Ecken: zwei Definitionsmöglichkeiten

$\mathcal{P}_{A,B}$:

$$\mathcal{E}_{A,B} := \{(e_a + e_b) \mid e_a \in \mathcal{E}_A, e_b \in \mathcal{E}_B \text{ und } \omega(e_a) \cap \omega(e_b) \neq \emptyset\} \quad (5.13)$$

$$\mathcal{K}_{A,B} := \{(k \oplus e) \mid (k, e) \in \mathcal{K}_A \times \mathcal{E}_B \cup \mathcal{K}_B \times \mathcal{E}_A \text{ und } \omega(k) \cap \omega(v) \neq \emptyset\} \quad (5.14)$$

$$\mathcal{F}_{A,B} := \{(f \oplus e) \mid (f, e) \in \mathcal{F}_A \times \mathcal{E}_B \cup \mathcal{F}_B \times \mathcal{E}_A \text{ und } \omega(f) \in \omega(e)\} \cup \\ \{(k_a \oplus k_b) \mid k_a \in \mathcal{K}_A, k_b \in \mathcal{K}_B \text{ und } \omega(k_a) \cap \omega(k_b) \neq \emptyset\} \quad (5.15)$$

Zwei Ecken erzeugen also in der Faltung wieder eine Ecke, aus den Paaren Ecke, Kante bzw. Ecke, Fläche entsteht ebenfalls wieder eine verschobene Kante bzw. Fläche. Einzig aus der Kombination Kante, Kante entsteht mit einer Parallelogrammfläche ein neues geometrisches Objekt.

Es werden jetzt noch zwei Punkte der Definition von Winkelbereichen betrachtet, die bisher schon vorausgesetzt, aber nicht diskutiert wurden. Zunächst bestehen zwei Möglichkeiten für die Definition der Winkelbereiche nicht konvexer Ecken. Wir betrachten hierzu Abb. 5.15. In Abb. (a) ist die Faltung dargestellt, die aus der oben gegebenen Definition des Winkelbereiches resultiert, bei der die Normalenvektoren n_1, n_2 immer über den kürzeren Bogen ineinander überführt werden, die Winkelbereiche in 2D also z. B. immer kleiner π sind. In Abb. (b) ist hingegen die Faltung dargestellt, wenn die Normalenvektoren immer mit einem festen Drehsinn, in 2D also z. B. entgegen dem Uhrzeigersinn, ineinander überführt werden. Für nicht konvexe Ecken würde dann der längere Bogen gewählt und damit gerade der zur bisherigen Definition komplementäre Winkelbereich und die komplementären Randobjekte. Dies ergibt bei entsprechender Wahl der Vorzeichen ebenfalls eine topologisch korrekte Struktur (ohne dies hier im einzelnen zu definieren). Bei der Wahl des kürzeren Bogens für den Winkelbereich ergeben sich jedoch Vorteile beweistechnischer und algorithmischer Natur, da die Menge der zugehörigen Normalenvektoren auf eine Halbkugel der Einheitssphäre beschränkt ist und somit eine Lokalitätseigenschaft aufweist.

Bei der Definition der Vorzeichen für Ecken nicht konvexer Polyeder wurde schon darauf hingewiesen, daß eine Normalisierung der Polyeder auf

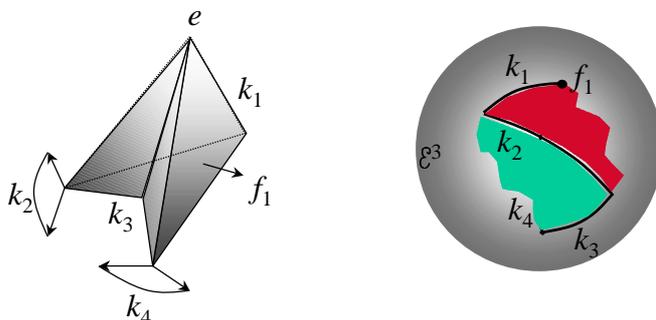
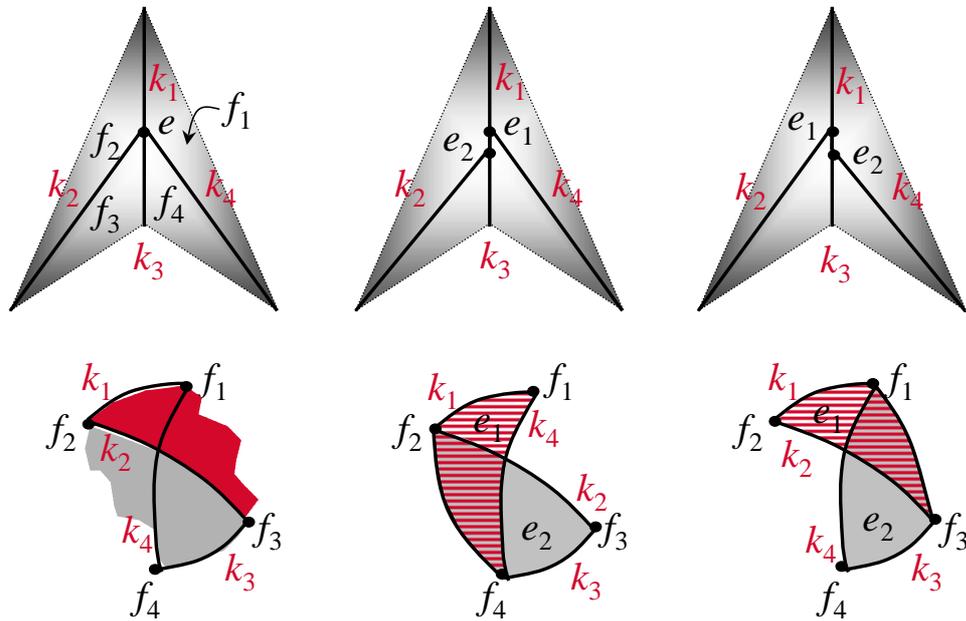


Abbildung 5.16: Selbstüberschneidung der Winkelbereiche von k_2 und k_4

genau drei Kanten je Ecke vorgenommen wird. Wir betrachten zunächst Abb. 5.16 um die Problematik aufzuzeigen. Dargestellt ist die Ecke e einer nicht konvexen Pyramide mit inzidenten Kanten k_1, \dots, k_4 und deren Winkelbereiche als Spur der Normalenvektoren auf der Einheitssphäre⁶. Der Winkelbereich einer Fläche entspricht jeweils einem Punkt (dem eindeutig festgelegten Normalenvektor), für Kanten jeweils dem Ausschnitt des Großkreisbogens zwischen den Normalenvektoren der adjazenten Flächen und für Ecken jeweils einer abgeschlossenen, durch die Winkelbereiche der inzidenten Kanten begrenzten Fläche.

Für die Ecke e weist der begrenzende Kantenzug jedoch eine Selbstüberschneidung der Kanten k_2 und k_4 auf. Die Definition eines von diesen Kanten begrenzten Winkelbereiches ist nicht ohne weiteres möglich, da die Gebiete sowohl rechts als auch links dieser Kanten unbegrenzt sind. Wir betrachten hierzu Abb. 5.17, in welcher die Ecke e in zwei Ecken e_1, e_2 aufgeteilt ist, die durch eine virtuelle Kante mit Länge 0 verbunden sind. Dargestellt sind jeweils eine Aufsicht von oben, sowie die zugehörigen Winkelbereiche der Ecken. Jede nicht konvexe Ecke wird derart normalisiert, daß zu jeder bestehenden bzw. neu eingefügten virtuellen Ecke genau drei Kanten inzident sind. Hiermit sind Selbstüberschneidungen in den Kantenzügen der zugehörigen Winkelbereiche ausgeschlossen; man kann dies auch als konvexe Zerlegung bzw. Triangulierung der Ecke e interpretieren. Diese Zerlegung ist nicht eindeutig; sie ist jedoch topologisch korrekt in dem Sinne, daß je genau zwei Flächen inzident zu einer Kante sind und der Winkelbereich der Kante einem stetigen Übergang zwischen den Normalenvektoren dieser Flächen entspricht.

⁶In Abb. 5.16 wird die Einheitssphäre zur Darstellung von Winkelbereichen bzw. Normalenvektoren verwendet. Die Einheitssphäre in Abb. 5.11 hat als Grundmenge des Richtungs- sowie des Ortsindikators eine andere Bedeutung.

Abbildung 5.17: Normalisierung der Winkelbereiche zur Ecke e

5.3.2 \mathcal{P} -Spuren und Schnitttest

Eine Anwendung der Randspur polyedrischer Objekte ist die Definition von Separationsebenen zwischen Polyedern für das in Kapitel 2 Abschnitt 2.4 vorgestellte globale Optimierungsverfahren. Diese Ebenen sind mit den im vorigen Abschnitt definierten Randobjekten gegeben.

Eine weitere Anwendung der Randspur $\mathcal{P}_{A,B}$ ist der effiziente Schnitttest von Polyedern unter Translationen (wie bei der Minkowski-Summe ist die relative Objektorientierung für \mathcal{P} -Spuren fixiert). Dieser wird auf die Bestimmung der *Winding Number* zum Referenzpunkt von Objekt B zurückgeführt. Die Winding Number zu einer geschlossenen Kurve im Raum ist allgemein wie folgt definiert:

Definition 17 (Winding Number)

Sei K eine geschlossene, orientierte Raumkurve; z. B. der Rand eines Polyeders oder eine \mathcal{P} -Spur. Für einen Punkt $p \in \mathbb{R}^3$ ist die Winding Number bzgl. K wie folgt definiert:

Man betrachte einen stetigen Pfad von p nach ∞ (fiktiver Punkt in beliebiger Richtung mit Abstand ∞ zu K). Schneidet dieser Pfad die Raumkurve K von innen nach außen, so wird $+1$ addiert, bei einem Schnitt von außen nach innen -1 . Die Gesamtsumme entlang des Pfades ist die Winding Number von p bzgl. K .

In Abb. 5.18 ist die Winding Number verschiedener Punkte bzgl. einer \mathcal{P} -Spur $\mathcal{P}_{A,B}$ dargestellt. Die Winding Number ist unabhängig von dem

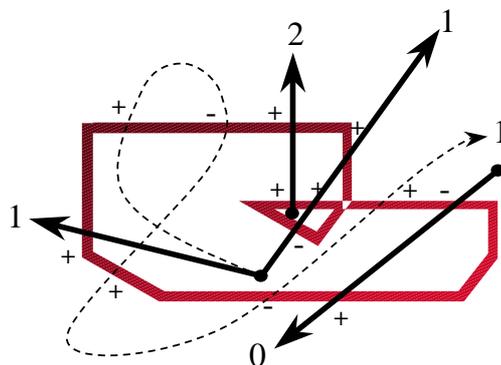


Abbildung 5.18: Winding Number von Referenzpunkt r_b bzgl. $\mathcal{P}_{A,-B}$ gleich Null $\Leftrightarrow A, B$ überschneidungsfrei

gewählten Pfad gegen ∞ (nicht notwendig eine Gerade); für die Implementierung eines Schnitttests wird ein Strahl gewählt, da in diesem Fall Ray-Shooting Verfahren eingesetzt werden können.

Für Punkte außerhalb von $\mathcal{P}_{A,B}$ ergibt sich immer der Wert 0; für Punkte innerhalb mindestens ein Wert von 1, da der Rand mindestens einmal von innen nach außen passiert werden muß. Wegen möglicher Selbstüberschneidungen der \mathcal{P} -Spur kann sich für Punkte innerhalb von $\mathcal{P}_{A,B}$ eine Winding Number echt größer 1 ergeben. Für die Punkte auf der \mathcal{P} -Spur ist eine Zusatzvereinbarung notwendig, die im wesentlichen festlegt, ob diese Punkte innerhalb, oder außerhalb von $\mathcal{P}_{A,B}$ liegen.

Zwei Polyeder A, B überschneiden sich genau dann, wenn die Winding Number des Referenzpunktes r_b von B bzgl. $\mathcal{P}_{A,-B}$ echt größer Null ist.

5.3.3 Faltungsintegral über Färbungen; Algebraische Definition der Faltungsoperation

In diesem Abschnitt sollen kurz die wesentlichen Begriffe zur rein algebraischen Behandlung der Faltungsoperation skizziert und weiterführende Referenzen angegeben werden. Der Zusammenhang zwischen Faltung und Polyederschnitt wird mit Satz 12 hergestellt, wobei sich der Begriff der Faltung aus dem dort verwendeten Faltungsintegral ableitet. Zur Vertiefung und insbesondere zur formalen Definition des Integrals über polyedrischen Färbungen und den Beweisen siehe Schapira [Sch91b] und Chen [Che99].

Eine Färbung ϕ des Raumes ist eine Abbildung $\phi : \mathbb{R}^3 \rightarrow \mathbb{N}$, die Punkten des Raumes einen ganzzahligen Wert zuordnet. Wir beschränken uns hier auf den Fall polyedrischer Färbungen, die polyedrischen Teilmengen des \mathbb{R}^3 konstante Werte auf den Elementen Ecke, Kante, Facette und inneren Punkten zuordnen.

Definition 18 (Polyedrische Färbung / Painting)

Seien $\mathcal{E}_{\mathcal{P}}, \mathcal{K}_{\mathcal{P}}, \mathcal{F}_{\mathcal{P}}, \mathcal{I}_{\mathcal{P}}$ die Menge der Ecken, Kanten, Flächen und der inneren Punkte eines Polyeders P , $\mathcal{O}_{\mathcal{P}}$ Platzhalter für eine dieser Mengen und $\phi : \mathbb{R}^3 \rightarrow \mathbb{N}$ eine Abbildung. ϕ heißt polyedrische Färbung (Painting), falls der Wert von ϕ auf den Mengen $\mathcal{E}_{\mathcal{P}}, \mathcal{K}_{\mathcal{P}}, \mathcal{F}_{\mathcal{P}}, \mathcal{I}_{\mathcal{P}}$ konstant ist, d. h.

$$(o_1, o_2) \in (\mathcal{O}_{\mathcal{P}} \times \mathcal{O}_{\mathcal{P}}) \Rightarrow \phi(o_1) = \phi(o_2) . \tag{5.16}$$

Ein einfaches Beispiel für eine polyedrische Färbung ist die *charakteristische Färbung* eines Polyeders $P \subset \mathbb{R}^3$:

$$\phi_P(p) := \begin{cases} 1, & \text{für alle } p \in P \\ 0, & \text{sonst} \end{cases} \tag{5.17}$$

Ein weiteres Beispiel ist die von einer Randspur \mathcal{P} induzierte Färbung, wobei die Winding Number (siehe Def. 17 im vorigen Abschnitt) von p bzgl. \mathcal{P} den Wert der Färbung in p definiert:

$$\Phi_{\mathcal{P}}(p) := \text{Winding Number von } p \text{ bzgl. } \mathcal{P} \tag{5.18}$$

Eine \mathcal{P} -Spur kann also auch als Randstruktur (Boundary Representation) einer Färbung aufgefaßt werden.

Wir benötigen nun noch den Begriff der Euler-Poincaré Charakteristik:

Definition 19 (Euler-Poincaré-Charakteristik)

Sei P eine polyedrische Teilmenge des \mathbb{R}^n . Die Euler-Poincaré Charakteristik von P ist definiert als die Anzahl der Komponenten abzüglich der Anzahl von Tunnel plus der Anzahl interner Hohlräume von P .

Für polyedrische Färbungen ϕ kann ein Integral $\int_p \phi(p) dp$ betrachtet werden. Dieses wird mit Hilfe einer Triangulierung des zugrunde liegenden Polyeders definiert (siehe *Triangulation Theorem (2.1)* in [Sch91b] bzw. *Euler Integral/Characteristic (1)* in [Che99]), wobei die Integralbildung unabhängig von der Wahl der Triangulierung und somit wohldefiniert ist. Für die charakteristische Färbung ϕ eines Polyeders P entspricht der Wert des Integrals $\int \phi$ der Euler-Poincaré Charakteristik des Polyeders. Den Zusammenhang zwischen der Faltung $\phi \star \psi$ von Färbungen und der Separation bzw. dem Schnitt von Polyedern stellt der folgende Satz her:

Satz 12 (Faltung und Euler-Poincaré Charakteristik)

Seien ϕ_A, ϕ_B charakteristische Färbungen zweier Polyeder A und B . Dann entspricht der Wert der Faltung⁷ $(\phi_A \star \phi_B)(t_b) = \int_{p \in \mathbb{R}^3} \phi_A(p) \phi_B(t_b - p) dp$ (Faltungsintegral) für einen Vektor $t_b \in \mathbb{R}^3$ der Euler-Poincaré Charakteristik des Schnittes von A mit dem am Ursprung punktgespiegelten und um t_b verschobenen Polyeder B .

⁷Man vergleiche hierzu die Definitionen der Faltungsoperation für die Laplacetransformation und ihre diskrete Entsprechung, die \mathcal{Z} -Transformation. Die \mathcal{Z} -Transformation wird auf die Triangulierung der polygonalen Grundmenge angewandt. Der Faltungssatz für die Fouriertransformation ist zum Vergleich ebenfalls aufgeführt. Zu den Faltungsoper-

Die in Abschnitt 5.3.1 auf \mathcal{P} -Spuren definierte Faltung \otimes hat insbesondere die folgende Eigenschaft: Seien $\mathcal{P}_A, \mathcal{P}_B$ die \mathcal{P} -Spuren zweier Polyeder, $\Phi_{\mathcal{P}_A}, \Phi_{\mathcal{P}_B}$ die hiervon induzierten Färbungen und $\mathcal{P}_{A,B}$ die zugehörige Faltung. Dann gilt für die von $\mathcal{P}_{A,B}$ induzierte Färbung $\Phi_{\mathcal{P}_{A,B}}$:

$$\Phi_{\mathcal{P}_{A,B}} = \Phi_{\mathcal{P}_A} \star \Phi_{\mathcal{P}_B} . \quad (5.19)$$

5.4 Gilbert-Johnson-Keerthi-Algorithmus: Separationsebenen für lokale Optimierung

Die Konstruktion des Distanzpolyeders bzw. der Randspur polyedrischer Objekte ist für die Anordnungsoptimierung bei fester Orientierung der Polyeder im Raum günstig, da sie in diesem Falle vorberechnet werden können. Für die lokale Optimierung wird jedoch ein Verfahren benötigt, welches für die potentiell kontinuierlichen, jedoch schrittweise ausgeführten Objektrotationen jeweils eine effiziente Anpassung der Separationsebenen ermöglicht.

Der Gilbert-Johnson-Keerthi-Algorithmus [GJK88, GF90] ist ein iteratives Verfahren, welches das Distanzpolyeder D_{AB} zweier Polyeder durch einen einbeschriebenen Simplex approximiert. Der Algorithmus terminiert, wenn der Simplex die Facette des Distanzpolyeders enthält, die zum minimalen Punktabstand der Polyeder A, B korrespondiert. Für den effizienten Einsatz des Verfahrens im Zusammenspiel mit den lokalen Optimierungsverfahren ist zu beachten, daß die Simplexinformation aus dem vorangehenden Optimierungsschritt ausgenutzt und mit der Basisiteration des GJK-Algorithmus effizient und lokal an die veränderte Objektorientierung angepaßt werden kann; siehe Abschnitt 5.4.1 (man beachte hierzu insbesondere die Implementierungen von S. Cameron [Cam97] und G. van den Bergen [vdB99]).

Das Distanzpolyeder $D_{P,Q}$ zweier konvexer Polyeder entspricht der konvexen Hülle der Summe der Polyederecken von P, Q ; siehe Abschnitt 5.2 zur Konstruktion des Distanzpolyeders. Im GJK-Algorithmus wird $D_{P,Q}$ nicht explizit konstruiert; die erzeugte Folge von Simplexes ergibt sich jedoch ebenfalls aus Summen von (Extremal-)Ecken der Ursprungspolyeder. Der Algorithmus wird in diesem Abschnitt für die Ausgabe eines Punktepaars mit minimalem Abstand formuliert. Aus diesem Punktepaar läßt sich jedoch leicht eine Separationsebene ableiten, die orthogonal zur Verbindungsstrecke der Punkte liegt und durch deren Mittelpunkt verläuft. Es

rationen von Transformierten siehe auch [All78, BS81, Bol89].

Faltung Z-Transformierte:

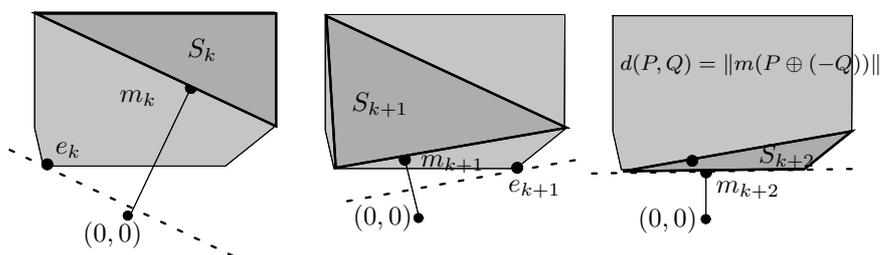
$$\mathcal{Z}(f_n \star g_n) = \mathcal{Z}\left(\sum_{k=0}^n f_k g_{n-k}\right) = \mathcal{Z}(f_n)\mathcal{Z}(g_n)$$

Faltung Laplacetransformation:

$$\mathcal{L}((f \star g)(t)) = \mathcal{L}\left(\int_0^t f(\tau)g(t-\tau)d\tau\right) = \mathcal{L}(f(t))\mathcal{L}(g(t))$$

Faltung Fouriertransformation:

$$\mathcal{F}((f \star g)(x)) = \mathcal{F}\left(\int_{-\infty}^{\infty} f(y)g(x-y)dy\right) = \sqrt{2\pi}\mathcal{F}(f(x))\mathcal{F}(g(x))$$

Abbildung 5.19: Iterationen des GJK-Algorithmus: Wahl von m_k und e_k

wird zunächst der Distanzbegriff für Polyeder P, Q wiederholt, der dem minimalen Punktabstand entspricht (siehe Definition 9 auf Seite 61):

$$d(P, Q) := \min\{\|p - q\| \mid p \in P, q \in Q\}$$

Wir betrachten den Zusammenhang zwischen $d(P, Q)$ und dem für konvexe Polyeder ebenfalls konvexen Distanzpolyeder $D_{P, Q} := P \oplus (-Q)$ und definieren hierzu den Minimalpunkt eines Polyeders:

Definition 20 (Minimalpunkt)

Sei P ein konvexes Polyeder. Der Minimalpunkt $m(P)$ von P ist der Punkt mit dem geringsten Abstand zum Koordinatenursprung. Dieser Punkt ist für konvexe Polyeder eindeutig bestimmt. Formal:

$$m(P) := \{p \in P \mid \|p\| = \min_{q \in P} \{\|q\|\}\}$$

Es gilt:

$$d(P, Q) = \|m(P \oplus (-Q))\| = \|m(D_{P, Q})\|$$

Es wird nun die Basisiteration des GJK-Algorithmus beschrieben und an einem zweidimensionalen Beispiel verdeutlicht. Gesucht ist der Minimalpunkt des Distanzpolygons, also $m(D_{P, Q})$. Der Algorithmus erzeugt eine Folge S_k von Simplizes (in \mathbb{R}^3 : Tetraeder, Dreiecksflächen, Kanten oder Punkte), deren Ecken auf dem Rand des Distanzpolyeders liegen und damit innerhalb von $D_{P, Q}$. Die Folge der Simplizes wird so gewählt, daß deren Minimalpunkte m_k gegen den Minimalpunkt von $D_{P, Q}$ und damit gegen die gesuchte Lösung konvergieren; siehe hierzu auch Abb. 5.19. Für polyedrische Objekte terminiert der Algorithmus nach einer durch die Eckenzahl von $D_{P, Q}$ begrenzten, endlichen Anzahl von Schritten (für den Beweis siehe [GJK88]).

Um die Folge der Simplizes S_k näher zu beschreiben, benötigen wir den Begriff der Extrempunkte eines Polyeders P in Richtung r . Extrempunkte sind die Schnittpunkte von P mit einer maximal in Richtung r parallelverschobenen Stützebene H_r mit Normalenvektor r (gestrichelte Linien in Abb. 5.19).

Definition 21 (Extremalpunkte)

Sei P ein konvexes Polyeder und $n \in \mathbb{R}^3$ eine vorgegebene Richtung bzw. der Normalenvektor einer Ebene. Die Abbildung e_P , welche die Menge der Extremalpunkte von P bzgl. n liefert, ist gegeben durch:

$$e_P(n) := \{e \in P \mid \langle n, e \rangle = \max\{\langle n, p \rangle \mid p \in P\}\}$$

Im GJK-Algorithmus wird jeweils zu m_k , dem Minimalpunkt des aktuellen Simplex S_k , ein Extremalpunkt e_k des Distanzpolyeders $D_{P,Q}$ in Richtung $-m_k$ bestimmt; $e_k := e_{D_{P,Q}}(-m_k)$. Da

$$e_{D_{P,Q}}(-m_k) = e_{P \oplus (-Q)}(-m_k) = e_P(-m_k) \oplus -e_Q(m_k)$$

gilt, müssen zur Bestimmung von $e_k \in D_{P,Q}$ nur die einzelnen „Summanden“ P, Q bekannt sein. e_k wird nun in den nächsten Simplex S_{k+1} integriert, wobei dafür im allgemeinen eine Ecke des bisherigen Simplex S_k entfernt wird (Ausnahme: S_k ist niederdimensional, d. h. Dreieck oder Kante; siehe hierzu Abschnitt 5.4.1). Bei diesem Eckenaustausch wird sichergestellt, daß weiterhin $m_k \in S_{k+1}$ liegt. Typischerweise wird also eine Tetraederecke von S_k gegen e_k ausgetauscht, um S_{k+1} zu erzeugen.

Wir betrachten nun das Terminierungskriterium für den Algorithmus. Der Abstand der durch m_k und den Extremalpunkt e_k bestimmten Stützebene zum Koordinatenursprung $a_k := \langle m_k, e_k \rangle / \|m_k\| = \langle \vec{m}_k, e_k \rangle$ stellt immer eine untere Schranke für die Distanz $\|m(D_{P,Q})\|$ dar; $\|m_k\|$ ist jeweils eine obere Schranke. Da die Folge der a_k im Gegensatz zu $\|m_k\|$ nicht notwendig monoton fallend ist, betrachten wir $\bar{a}_k := \max\{0, a_0, \dots, a_k\}$. Der Algorithmus terminiert, sobald der Abstand der Stützebene \bar{a}_k gleich dem Abstand des gefundenen Minimalpunktes m_k vom Koordinatenursprung ist, d. h. $\bar{a}_k = \|m_k\|$ gilt.

Es ergibt sich der folgende Pseudo-Code für den GJK-Algorithmus:

m_0 beliebiger Punkt in $D_{P,Q}$;

$e_0 := e_P(-m_0) - e_Q(m_0)$;

$S_0 := \{m_0\}$;

$k := 0$;

$\bar{a} := \max\{0, \langle \vec{m}_0, e_0 \rangle\}$;

WHILE $\bar{a} < \|m_k\|$ **DO**

$S_{k+1} := \min_{|X|} \{X \subset S_k \mid m_k \in \text{Conv}(X)\} \cup \{e_k\}$;

S_k ist jeweils die Eckenmenge des aktuellen Simplex.

$k := k + 1$;

$m_k := m(\text{Conv}(S_k))$;

m_k ist Punkt des Simplex S_k mit minimalem Abstand zum Koordinatenursprung. m_k liegt im Normalfall echt innerhalb von $D_{P,Q}$, während die Ecken e_i des Simplex S_k auf dem Rand von $D_{P,Q}$ liegen. Ferner gilt $\|m_k\| \geq m(D_{P,Q})$ und monoton fallend.

$e_k := e_P(-m_k) - e_Q(m_k);$
 e_k ist Extrempunkt in Richtung $-m_k$ an $D_{P,Q}$.

$\bar{a} := \max\{\bar{a}, \langle \vec{m}_k, e_k \rangle\};$
 $\langle \vec{m}_k, e_k \rangle$ ist der Abstand der Stützebene in Richtung m_k vom Koordinatenursprung.

END WHILE

return Punktepaar $(e_P(-m_k), e_Q(m_k));$

Auf die Bestimmung von m_k wird im folgenden Abschnitt 5.4.1 näher eingegangen. Bei der Bestimmung der Extrempunkte $e_P(m_k)$ ist zu beachten, daß man sich hier auf die Eckenmenge des Polyeders P beschränken kann und somit nur eine endliche Punktmenge betrachten muß. Insbesondere steht im GJK-Algorithmus mit Ausnahmen der ersten Iteration jeweils der Extrempunkt der vorigen Iteration zur Verfügung, so daß eine lokale Suche entlang des Adjazenzgraphen der Ecken in der Praxis in konstanter Zeit den neuen Extrempunkt liefert (vgl. [Cam97, OG97]). Dies gilt insbesondere auch beim Einsatz des GJK-Algorithmus im lokalen Optimierungsverfahren von Abschnitt 3.2.

5.4.1 Algebraische Bestimmung des Simplex S_k

Zu zeigen bleibt noch, wie der Minimalpunkt m_k und die minimale Menge $X \subset S_k$ mit $m_k \in \text{Conv}(X)$ zu gegebenem S_k algebraisch bestimmt werden können. Es werden hierzu alle Teilmengen X von S_k mit ein bis drei Elementen gebildet (höchstens 14 verschiedene Teilmengen) und jeweils getestet, ob m_k als Konvexkombination der enthaltenen Vektoren dargestellt werden kann. Die Begründung für die Einschränkung von S_k auf Mengen mit maximal vier Elementen liefert der Satz von Carathéodory:

Satz 13 (Satz von Carathéodory [GW93])

Sei $P \subset \mathbb{R}^n$ und p ein Punkt in $\text{Conv}(P)$. Dann gibt es eine Teilmenge $S \subseteq P$ mit $n + 1$ oder weniger Punkten und $p \in \text{Conv}(S)$.

Sei im folgenden $X = \{s_{i_1}, \dots, s_{i_n}\} \subset S_k$ ($1 \leq n \leq 3$) mit Indexmenge $I = \{i_1, \dots, i_n\}$. Aus der Bedingung $m_k \in \text{Conv}(X)$ erhalten wir die folgenden Gleichungen:

$$m_k = \sum_{i \in I} \lambda_i s_i \quad (5.20)$$

$$\text{mit } \sum_{i \in I} \lambda_i = 1 \quad (5.21)$$

$$\text{und } \lambda_i > 0 \quad (5.22)$$

Zudem gilt für den Minimalpunkt m_k , daß er orthogonal zu $\text{Aff}(X)$ sein muß. Dies heißt insbesondere, daß m_k orthogonal zu allen Vektoren $(s_i - s_j)$ mit

$s_i, s_j \in X$ sein muß. Für einen beliebig aber fest gewählten Index $i^* \in I$ erhalten wir hieraus Gleichungen der Form

$$\langle m_k, (s_{i^*} - s_j) \rangle = 0 \quad \forall j \in I \setminus \{i^*\} . \quad (5.23)$$

Wir setzen nun Gleichung (5.20) in Gleichung (5.23) ein und erhalten daraus:

$$\sum_{i \in I} \lambda_i \langle s_i, (s_{i^*} - s_j) \rangle = 0 \quad \forall j \in I \setminus \{i^*\} \quad (5.24)$$

Die Gleichungen (5.21) und (5.24) bilden zusammengenommen ein lineares Gleichungssystem in den Variablen λ_i . Falls $\lambda_i > 0 \quad \forall i \in I$ gilt und damit auch Bedingung (5.22) erfüllt ist, so ist X die gesuchte Teilmenge von S_k .

Es wird nun noch gezeigt, wie die Menge der Gleichungssysteme zu den verschiedenen Teilmengen X von S_k effizient gelöst werden kann. Die Lösungsidee basiert auf der Cramerschen Regel und hat hier die folgende spezielle Form von Rekursionsgleichungen:

$$c_i(\{s_i\}) := 1 \quad (5.25)$$

$$c_i(Y \cup \{s_i\}) := \sum_{j \in I_Y} c_j(Y) (\langle s_j, s_{i^*} \rangle - \langle s_j, s_i \rangle) \quad (5.26)$$

$$c_X = \sum_{i \in I_X} d_i(X) \quad (5.27)$$

$$\lambda_i = c_i(X) / c_X \quad (5.28)$$

Es wurde gezeigt, daß S_k aus maximal vier Elementen besteht. Für die effiziente Berechnung der obigen Rekursionsgleichungen werden die maximal 16 paarweisen Skalarprodukte $\langle s_i, s_j \rangle$ gespeichert und in den weiteren Berechnungen wiederverwendet. In jeder Iteration müssen nur diejenigen Skalarprodukte neu berechnet werden, in denen der hinzugefügte Vektor e_k vorkommt. Zudem werden auch die Werte $c_i(Y)$ gespeichert, wobei auch hier immer nur diejenigen Teilmengen neu zu berechnen sind, in denen e_k enthalten ist.

Kapitel 6

Darstellung räumlicher Drehungen

In diesem Kapitel werden verschiedene Darstellungsmöglichkeiten für räumliche Drehungsoperationen vorgestellt und mit ihren Vor- und Nachteilen diskutiert. Im wesentlichen sind vier Darstellungsformen verbreitet:

- Drehachse und Winkel
- Eulersche Winkel
- Matrixdarstellung
- Quaternionen

Über die Angabe einer Drehachse und des entsprechenden Drehwinkels ist eine intuitive Charakterisierung von Drehungen möglich. Sie wird daher häufig für die Eingabe bzw. Definition von Drehungen in interaktiven Systemen verwendet.

Eulersche Winkel beziehen sich hingegen auf ein System von drei Drehachsen, zu denen jeweils die Drehwinkel in einer festgelegten Reihenfolge angegeben werden. Meist werden hier die drei Koordinatenachsen als Drehachsen verwendet und jeweils kleine Drehwinkel angegeben. Die intuitive Annahme, die Gesamtdrehung sei unabhängig von der Reihenfolge der Teildrehungen, ist zwar nicht korrekt, die Abweichung fällt bei kleinen Drehwinkeln aber nur unwesentlich ins Gewicht. Auch diese Darstellung ist daher zur interaktiven Eingabe von Drehungen verbreitet. Bei geeigneter Wahl der Drehachsen sind die Eulerschen Winkel eindeutig bestimmt und liefern eine Darstellung mit drei unabhängigen Parametern.

Die Matrixdarstellung von Drehungen entspricht der Standardrepräsentation linearer Abbildungen in der linearen Algebra. Sie wird zudem von vielen standardisierten Graphikumgebungen, wie z. B. OpenGL oder SGI Open Inventor, verwendet bzw. gefordert. Zur Visualisierung von Objekten wird diese Darstellung daher häufig direkt verwendet oder es sind effiziente Konvertierungen in diese Darstellung notwendig. Zur interaktiven Eingabe eignet sich die Matrixform jedoch nicht.

Quaternionen¹ bilden eine weitere Möglichkeit, räumliche Drehungen darzustellen. Sie eignen sich insbesondere dann, wenn häufig Verkettungen von Rotationen berechnet werden müssen, da dies mit wesentlich weniger Rechenoperationen als z. B. bei einer Matrixmultiplikation erreicht werden kann. Sie werden daher im Bereich von Computeranimationen, insbesondere auch Computerspielen, häufig eingesetzt. Eine Konvertierung dieser Darstellung in Matrixform wird auf Seite 94 angegeben. Zur interaktiven Definition von Drehungen sind sie jedoch ebenfalls nicht geeignet.

6.1 Algebraische Definition und Matrixdarstellung

Räumliche Drehungen sind lineare Abbildungen von \mathbb{R}^3 nach \mathbb{R}^3 und können daher in Form reeller 3×3 -Matrizen dargestellt werden (siehe hierzu Standardwerke zur Linearen Algebra, z. B. [KK72, Fis78, Fis86, Bri83]). Für Drehungsmatrizen gelten dabei spezielle Eigenschaften, die im folgenden kurz zusammengefaßt werden. Wie bei allen linearen Abbildungen in Matrixform entsprechen die Spalten der Transformationsmatrix jeweils den Bildern der drei kanonischen Basisvektoren e_i . Es sei eine Drehungsmatrix wie folgt gegeben:

¹Zur geschichtlichen Einführung der Quaternionen folgendes Zitat aus [Bri83] Seite 333:

„Zur Entstehung und Durchsetzung des Vektorraumbegriffs:

Die von Gauß (1799), Wessel (1799), Argand (1806) und anderen entwickelte geometrische Darstellung der komplexen Zahlen in der Ebene gestattet eine algebraische Behandlung der ebenen Geometrie, die insbesondere die Addition gerichteter Strecken, darüberhinaus aber auch deren Multiplikation ermöglicht. Da das für die mathematische Beherrschung physikalischer Zusammenhänge eine große Bedeutung hat, versuchte Hamilton ein umfassenderes „System imaginärer Größen“ zu finden, die eine entsprechende Rechnungsweise für den dreidimensionalen Raum gestatten. Er fand dabei heraus, daß dies für drei Dimensionen nicht unmittelbar zu realisieren ist, stieß aber 1843 auf das vierdimensionale System der Quaternionen. Um mit diesen dennoch eine für die Physik nützliche Rechnungsweise zu ermöglichen, spaltete er eine Quaternion ($a + ix + jy + kz$) in einen Vektor ($ix + jy + kz$) und einen Skalar a auf. Die Bezeichnung „Vektor“ und „Skalar“ führte Hamilton deswegen ein, weil er mit dem Vektoranteil einer Quaternion geometrische und physikalische gerichtete Größen wie gerichtete Strecken, geradlinige Bewegungen, Geschwindigkeit, Kräfte, Beschleunigungen usw. beschrieb (vehere (lat) führen, tragen, ziehen) und mit dem skalaren Anteil der Quaternion ungerichtete, skalierbare Größen, wie etwa die Zeit. Hamilton entwickelte Methoden der Algebra und der Analysis zum Rechnen in Quaternionen und wendet sie für die verschiedensten Untersuchungen der mathematischen Physik an.“

Zur zeitlichen Einordnung beachte man, daß die Peano-Axiome für unendlich dimensionale Vektorräume über \mathbb{R} aus dem Jahr 1888 stammen, die Erweiterung auf Vektorräume über beliebigen Körpern auf Krull 1925 und Emmy Noether 1929 zurückgehen und der heute übliche allgemeine Vektorraumbegriff über Schiefkörpern 1947 von Bourbaki eingeführt wurde, Quaternionen also schon früher bekannt waren.

$$R = \begin{pmatrix} b_{11} & b_{21} & b_{31} \\ b_{12} & b_{22} & b_{32} \\ b_{13} & b_{23} & b_{33} \end{pmatrix} \quad \text{Drehungsmatrix}$$

$$\underbrace{\qquad}_{\mathbf{b}_1} \quad \underbrace{\qquad}_{\mathbf{b}_2} \quad \underbrace{\qquad}_{\mathbf{b}_3} \quad \text{Bilder } e_i$$

Im dreidimensionalen Raum entsprechen Drehungen den Matrizen $R \in SO(3)$, der Gruppe der speziellen orthogonalen 3×3 -Matrizen. Orthogonale Matrizen sind durch die Gleichung $R^{-1} = R^t$ definiert, wobei R^t die transponierte Matrix mit $R_{ij} = R_{ji}^t$ darstellt. Für die Bilder der Basisvektoren gilt $\langle b_i, b_j \rangle = 0$ ($i \neq j$).

Für die speziellen orthogonalen Matrizen gilt zusätzlich $\det R = +1$ (vgl. [KK72, Fis78, Fis86]), woraus $\|b_i\| = 1$ folgt. Man beachte, daß räumliche Drehungen im Gegensatz zum zweidimensionalen Fall nicht kommutativ sind. Für Drehungsmatrizen A, B gilt also im allgemeinen $AB \neq BA$.

Für eine reelle 3×3 -Matrix stehen zunächst neun Parameter bzw. Koeffizienten zur Verfügung. Durch die definierenden Gleichungen $\langle b_i, b_j \rangle = 0$ und $\|b_i\| = 1$ werden diese jedoch auf drei Freiheitsgrade eingeschränkt. Die Parameterabhängigkeiten sind somit durch quadratische Ungleichungen gegeben.

6.2 Drehachse und -winkel

Bei dieser Darstellung einer Rotation wird die Drehachse d und ein Drehwinkel α angegeben. Man beachte, daß der Drehwinkel auch durch die Länge der Drehachse definiert werden kann, d.h. $\alpha := \|d\|$. Hieraus folgt unmittelbar, daß räumliche Drehungen maximal drei Freiheitsgrade zulassen. Die wesentliche Eigenschaft der Drehachse ist, daß sie bei der Transformation auf sich selbst abgebildet wird. Dies bildet die Grundlage für die Konvertierung in Matrixform:

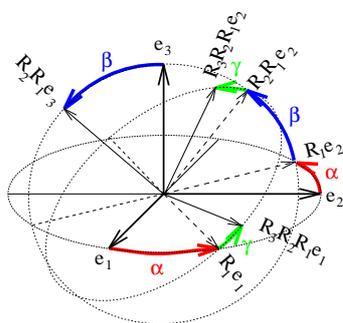
$$Rd = d \Leftrightarrow 0 = Rd - d = Rd - \mathbf{1}d = (R - \mathbf{1})d .$$

Es ist also ein homogenes lineares Gleichungssystem mit Unbekannten $d = (d_1, d_2, d_3)$ zu lösen.

Zudem liefert diese Darstellung eine Normalform für die Matrixdarstellung einer Rotation bzgl. eines geeignet gewählten Orthonormalsystems mit $e_1 = d$. Die Normalform ist dann von der Gestalt

$$R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{pmatrix} .$$

Betrachtet man die Summe der Diagonalelemente, die *Spur* der Matrixdarstellung, so erhält man für die obige Normalform $\text{Spur } R = 1 + 2 \cos \alpha$.

Abbildung 6.1: Drehung mit Eulerschen Winkeln α , β , γ

Wenn auch uneigentliche Drehungen, d. h. Spiegelungen mit $\det R = -1$, betrachtet werden, so gilt:

$$\text{Spur } R = \det R + 2 \cos \alpha$$

Für eine gegebene lineare Abbildung ist die Spur insbesondere unabhängig von der gewählten Basis (Folgerung aus der Invarianz des charakteristischen Polynoms bzgl. der Basis; zum Begriff Spur siehe [Fis86]).

6.3 Eulersche Winkel

Eulersche Winkel beziehen sich immer auf ein vorgegebenes System von drei, nicht notwendig unterschiedlichen, Drehachsen. Sie geben jeweils den Drehwinkel um die entsprechenden Achsen an, wobei die Reihenfolge der Drehungen wegen der nicht kommutativen Verknüpfung ebenfalls festgelegt sein muß. In der Literatur wird der Begriff in zwei Varianten benutzt.

In der englischsprachigen Literatur wird meist eine beliebige Wahl der Drehachsen zugelassen, insbesondere werden oft die x, y und z -Achse gewählt. In diesem Fall sind die zu einer gegebenen Drehung korrespondierenden Winkel jedoch nicht eindeutig bestimmt.

Die deutschsprachige Literatur verwendet den Begriff hingegen im allgemeinen für Systeme von Drehachsen, bei denen die Wahl der Drehwinkel zu einer gegebenen Rotation eindeutig festgelegt ist. Dies ist z. B. für die Wahl z, x, z -Achse der Fall. Im folgenden wird gezeigt, wie diese Winkel aus einer gegebenen Matrixdarstellung gewonnen werden können.

Satz 14

Sei $R \in SO(3)$. Dann gibt es eindeutig bestimmte Winkel $\alpha, \beta, \gamma \in [0, 2\pi[$, so daß

$$R = \begin{pmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \beta & -\sin \beta \\ 0 & \sin \beta & \cos \beta \end{pmatrix} \begin{pmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Beweis:

Gezeigt wird, daß Winkel $\varphi, \psi, \beta \in [0, 2\pi[$ existieren, so daß

$$\underbrace{\begin{pmatrix} \cos \varphi & -\sin \varphi & 0 \\ \sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{pmatrix} R \begin{pmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{pmatrix}}_X = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \beta & -\sin \beta \\ 0 & \sin \beta & \cos \beta \end{pmatrix}.$$

Die in der Formulierung des Satzes gewählten Matrizen ergeben sich dann mit $\gamma = -\varphi$ und $\alpha = -\psi$. Sei $X = (x_{ij})$ die links vom Gleichheitszeichen stehende Produktmatrix.

1. *Schritt:* Es ist $x_{33} = r_{33}$. Da R orthogonal ist, folgt aus $RR^t = RR^{-1} = \mathbf{1}$, daß $r_{31}^2 + r_{32}^2 + r_{33}^2 = 1$ und damit $\|r_{33}\| \leq 1$. Wähle $\beta \in [0, 2\pi[$ mit $\cos \beta = r_{33}$.

2. *Schritt:* Es ist ferner $x_{13} = r_{13} \cos \varphi - r_{23} \sin \varphi$; wähle $\varphi \in [0, 2\pi[$ so, daß $x_{13} = 0$ gilt.

Dies ist möglich, da der Wertebereich der Tangensfunktion das Intervall $[-\infty, +\infty]$ überdeckt, wobei $\tan \varphi = \frac{\sin \varphi}{\cos \varphi} = \frac{r_{13}}{r_{23}}$ (mit Sonderfall $\cos \varphi = 0$, falls $r_{23} = 0$).

3. *Schritt:* Wähle $\psi \in [0, 2\pi[$ mit $\bar{r} = (\cos \varphi, -\sin \varphi) \begin{pmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{pmatrix} = (\cos \psi, \sin \psi)$. Dies ist möglich, da \bar{r} nach Wahl von φ die Norm 1 hat (siehe unten). Damit gilt $x_{11} = \cos^2 \psi + \sin^2 \psi = 1$.

Nebenrechnung zu $\|\bar{r}\| = 1$. Aus R orthogonal folgt

$$\begin{aligned} r_{11}^2 + r_{12}^2 &= 1 - r_{13}^2, & r_{21}^2 + r_{22}^2 &= 1 - r_{23}^2 & \text{und} \\ r_{11}r_{21} + r_{12}r_{22} &= 0 - r_{13}r_{23} = -r_{13}r_{23}. \end{aligned} \quad (6.1)$$

Damit:

$$\begin{aligned} \|\bar{r}\| &= r_{11}^2 \cos^2 \varphi - 2r_{11}r_{21} \cos \varphi \sin \varphi + r_{21}^2 \sin^2 \varphi + \\ &\quad r_{12}^2 \cos^2 \varphi - 2r_{12}r_{22} \cos \varphi \sin \varphi + r_{22}^2 \sin^2 \varphi \\ &\stackrel{(6.1)}{=} (1 - r_{13}^2) \cos^2 \varphi + 2r_{13}r_{23} \cos \varphi \sin \varphi + (1 - r_{23}^2) \sin^2 \varphi \\ &= \cos^2 \varphi + \sin^2 \varphi + \underbrace{(r_{13} \cos \varphi - r_{23} \sin \varphi)^2}_{= 0 \text{ (Schritt 2)}} = 1 \end{aligned}$$

4. *Schritt:* Da X orthogonal ist (Produkt orthogonaler Matrizen), folgt aus $x_{11} = 1$ mit $XX^t = \mathbf{1}$ wie in Schritt 1, daß $x_{12} = x_{21} = x_{31} = 0$. Da $\det X = +1$ folgt hiermit dann $\begin{pmatrix} x_{22} & x_{23} \\ x_{32} & x_{33} \end{pmatrix} = \begin{pmatrix} \cos \beta & -\sin \beta \\ \sin \beta & \cos \beta \end{pmatrix}$

Die Winkel α , β und γ sind eindeutig bestimmt, sofern R eine von $(0, 0, 1)$ verschiedene Drehachse hat. Entspricht R jedoch einer Drehung um die z -Achse, so sind $r_{13} = r_{23} = 0$ und in Schritt 2 ist φ nicht eindeutig bestimmt. Setze φ in diesem Fall gleich Null. Mit dieser zusätzlichen Vereinbarung sind die Eulerschen Winkel zu einer gegebenen Rotation R immer eindeutig. \square

Die Matrixdarstellung ergibt sich aus den Eulerschen Winkeln durch Multiplikation der zu den Teildrehungen korrespondierenden Matrizen.

6.4 Quaternionen (Cayley-Kleinsche Parameter, \mathbb{H})

Die Darstellung von Drehungen als sogenannte Quaternionen beruht auf einer speziellen Erweiterung der komplexen Zahlen, wobei die multiplikative Verknüpfung entsprechend der Verkettung von Drehungen im Raum nicht kommutativ ist. Dies entspricht der Struktur eines Schiefkörpers bzw. eines Divisionsrings (hierüber können Divisionsalgebren gebildet werden) und im folgenden wird die Algebra der Quaternionen über den reellen Zahlen mit \mathbb{H} bezeichnet (die Bezeichnung ist in der Literatur nicht einheitlich; Klingenberg/Klein [KK72] verwendet z. B. \mathbb{L}). Es gilt:

Satz 15 (Satz von Frobenius)

Jede endlich dimensionale Divisionsalgebra A über \mathbb{R} ist isomorph zu \mathbb{R} , \mathbb{C} oder \mathbb{H} .

Insofern bilden die Quaternionen die einzige und kleinstmögliche Einbettung von Rotationen in eine \mathbb{R} -Algebrastruktur, denn die Multiplikation in \mathbb{R} und \mathbb{C} ist kommutativ.

In der Literatur sind zwei Darstellungsformen für Quaternionen gebräuchlich. In der Physik, speziell der klassischen Mechanik, wird eine Darstellung von Quaternionen als 2×2 -Matrizen über \mathbb{C} bevorzugt und häufig als Cayley-Kleinsche Parameter bezeichnet. Zur Isomorphie zwischen den unitären komplexen 2×2 -Matrizen (genauer: Paaren von Matrizen) und den reellen orthogonalen 3×3 -Matrizen siehe [Gol89].

Quaternionen können alternativ als Erweiterung der komplexen Zahlen eingeführt werden (siehe z. B. [MM97]), woraus eine einfach zu implementierende Darstellung gewonnen werden kann. Zur Implementierung siehe auch direkt Abschnitt 6.4.3.

6.4.1 Cayley-Kleinsche Parameter; 2×2 -Matrixdarstellung über \mathbb{C}

Wir gehen zunächst auf die Darstellung mittels komplexer 2×2 -Matrizen ein und betrachten hierzu den folgenden Satz von Matrizen (Paulische Spin

Matrizen):

$$\mathbf{1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}; \rho_x := \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}; \rho_y := \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}; \rho_z := \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Diese sind linear unabhängig und die Darstellung eines Punktes $p = (p_1, p_2, p_3) \in \mathbb{R}^3$ erfolgt durch die komplexe Matrix der Form

$$\hat{P} = p_1\rho_x + p_2\rho_y + p_3\rho_z = \begin{pmatrix} p_3 & p_1 - ip_2 \\ p_1 + ip_2 & -p_3 \end{pmatrix}.$$

Für eine Drehung mit Drehachse $d \in \mathbb{R}^3$ mit $\|d\| = 1$ und Drehwinkel θ definieren wir zudem die folgende Matrix:

$$Q = \cos \frac{\theta}{2} \mathbf{1} + i \sin \frac{\theta}{2} (d_1\rho_x + d_2\rho_y + d_3\rho_z)$$

Die Drehung eines Vektors bzw. Punktes p mit Matrixdarstellung \hat{P} ist durch das Produkt

$$\tilde{P} = Q\hat{P}Q^a$$

gegeben. Q ist eine unitäre komplexe Matrix mit Determinante $\det Q = 1$. Unitäre Matrizen sind durch die Gleichung $Q^a Q = \mathbf{1} = Q Q^a$ definiert, wobei $Q^a = \bar{Q}^t$ die adjungierte Matrix zu Q bezeichnet mit \bar{Q} für die konjugiert komplexe Matrix. Es gilt also insbesondere $Q^a = Q^{-1}$ (vgl. Definition orthogonaler Matrizen durch $R^t = R^{-1}$). Man kann zeigen, daß die Matrix Q mit $\alpha, \beta \in \mathbb{C}$ immer von der Gestalt

$$Q = \begin{pmatrix} \alpha & \beta \\ -\bar{\beta} & \bar{\alpha} \end{pmatrix}$$

ist, wobei zusätzlich

$$\alpha\bar{\alpha} + \beta\bar{\beta} = 1$$

gelten muß. Durch diese Bedingung werden die vier Freiheitsgrade bei der Wahl der Komponenten von α und β auf die drei Freiheitsgrade bei räumlichen Drehungen eingeschränkt.

6.4.2 \mathbb{R}^4 -Vektordarstellung von \mathbb{H}

Im folgenden wird nun die alternative Betrachtungsweise von Quaternionen als Erweiterung der komplexen Zahlen vorgestellt. Wie üblich ist $i^2 = -1$. Zudem führen wir zwei neue Basiselemente j und k ein, wobei gilt:

$$i^2 = j^2 = k^2 = ijk = -1 \quad (6.2)$$

$$ij = -ji = k, \quad jk = -kj = i, \quad ki = -ik = j. \quad (6.3)$$

Quaternionen besitzen nun eine Darstellung als $\hat{q} = q_0 + q_1i + q_2j + q_3k$ mit $q_i \in \mathbb{R}$. Zur Vereinfachung der Schreibweise führen wir den Vektor $\mathbf{I} = (i, j, k)$ ein, der den imaginären Basisanteil bzw. Unterraum repräsentiert. Mit $q := (q_1, q_2, q_3) \in \mathbb{R}^3$ ergibt sich damit $\hat{q} = q_0 + \langle q, \mathbf{I} \rangle$. Aus (6.3) folgt direkt, daß die Quaternionenmultiplikation, entsprechend der Verkettung von Drehungen, nicht kommutativ ist. Das Produkt zweier Quaternionen a, b ergibt sich insgesamt wie folgt:

$$\begin{aligned}
\hat{a}\hat{b} &= (a_0 + a_1i + a_2j + a_3k)(b_0 + b_1i + b_2j + b_3k) \\
&= (a_0b_0 - a_1b_1 - a_2b_2 - a_3b_3) + \\
&\quad (a_0b_1 + a_1b_0 + a_2b_3 - a_3b_2)i + \\
&\quad (a_0b_2 - a_1b_3 + a_2b_0 + a_3b_1)j + \\
&\quad (a_0b_3 + a_1b_2 - a_2b_1 + a_3b_0)k \\
&= (a_0b_0 - \langle a, b \rangle) + \\
&\quad (a_0b_1 + b_0a_1 + a_2b_3 - a_3b_2)i + \\
&\quad (a_0b_2 + b_0a_2 + a_3b_1 - a_1b_3)j + \\
&\quad (a_0b_3 + b_0a_3 + a_1b_2 - a_2b_1)k \\
&= (a_0b_0 - \langle a, b \rangle) + \langle a_0b + b_0a + a \times b, \mathbf{I} \rangle
\end{aligned} \tag{6.4}$$

Die Darstellung eines Punktes bzw. Vektors p im dreidimensionalen Raum (d.h. $p \in \mathbb{R}^3$) ergibt sich aus

$$\hat{p} = p_1i + p_2j + p_3k = \langle p, \mathbf{I} \rangle \tag{6.5}$$

und die Darstellung einer Rotation mit Drehachse $d \in \mathbb{R}^3$, $\|d\| = 1$ und Drehwinkel θ aus

$$\hat{q} = \cos \frac{\theta}{2} + \sin \frac{\theta}{2} \hat{d} . \tag{6.6}$$

Das Produkt $\hat{p} := \hat{q}\hat{p}\hat{q}$ ist die Quaternionendarstellung für den bzgl. der Drehachse d um den Winkel θ gedrehten Punkt p . Das konjugierte Quaternion ist definiert durch $\bar{q} := q_0 - q_1i - q_2j - q_3k = q_0 - \langle q, \mathbf{I} \rangle$. Man beachte, daß die zu \hat{q} äquivalente Drehung um die Achse $-d$ und den Winkel $2\pi - \theta$ eine andere Quaternionendarstellung besitzt. Eine gegebene Drehung kann also im allgemeinen durch genau zwei verschiedene Quaternionen erzeugt werden. Für die einer Drehung entsprechenden Quaternionen $\hat{q}\bar{q}$ gilt wegen $\cos^2 \alpha + \sin^2 \alpha = 1$ und $\|d\| = 1$ (vgl. auch Matrixdarstellung von Quaternionen $QQ^a = \mathbf{1}$):

$$\hat{q}\bar{q} = q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1 = \bar{q}\hat{q} \tag{6.7}$$

Für die Drehungsoperation $\hat{p} = \hat{q}\hat{p}\hat{q}$ wird nun eine explizite Darstellung abgeleitet (siehe auch die nachfolgende Nebenrechnung). Man beachte, daß

wieder ein rein imaginärer Anteil entsteht, welcher der Darstellung eines Punktes entspricht.

$$\begin{aligned}
q\hat{p}\bar{q} &\stackrel{(6.4)}{=} \overbrace{(-\langle q, p \rangle + \langle q_0 p + q \times p, \mathbf{I} \rangle)}^{q\hat{p} \text{ mit } p_0=0} \overbrace{(q_0 - \langle q, \mathbf{I} \rangle)}^{\bar{q}} \\
&\stackrel{(6.4)}{=} -q_0 \langle q, p \rangle + \langle q_0 p + q \times p, q \rangle + \\
&\quad \langle \langle q, p \rangle q + q_0(q_0 p + q \times p) + (q_0 p + q \times p) \times (-q), \mathbf{I} \rangle \\
&= -q_0 \langle q, p \rangle + q_0 \langle p, q \rangle + \langle q \times p, q \rangle + \\
&\quad \langle \langle q, p \rangle q + q_0^2 p + q_0(q \times p) + q_0(p \times (-q)), \mathbf{I} \rangle + \\
&\quad \langle (q \times p) \times (-q), \mathbf{I} \rangle \\
&= \langle \langle q, p \rangle q + q_0^2 p + 2q_0(q \times p), \mathbf{I} \rangle + \langle q \times (q \times p), \mathbf{I} \rangle \\
&\stackrel{(6.9)}{=} \langle \langle q, p \rangle q + q_0^2 p + 2q_0(q \times p), \mathbf{I} \rangle + \langle (q_0^2 - 1)p + \langle q, p \rangle q, \mathbf{I} \rangle \\
&= \langle 2\langle q, p \rangle q + (2q_0^2 - 1)p + 2q_0(q \times p), \mathbf{I} \rangle \\
&= 2\langle \langle q, p \rangle q + (q_0^2 - 0, 5)p + q_0(q \times p), \mathbf{I} \rangle
\end{aligned} \tag{6.8}$$

Nebenrechnung zu (6.8):

$$\begin{aligned}
q \times (q \times p) &= q \times (q_2 p_3 - q_3 p_2, q_3 p_1 - q_1 p_3, q_1 p_2 - q_2 p_1) \\
&= (q_2(q_1 p_2 - q_2 p_1) - q_3(q_3 p_1 - q_1 p_3), \\
&\quad q_3(q_2 p_3 - q_3 p_2) - q_1(q_1 p_2 - q_2 p_1), \\
&\quad q_1(q_3 p_1 - q_1 p_3) - q_2(q_2 p_3 - q_3 p_2)) \\
&= ((-q_2^2 - q_3^2)p_1 + (q_2 p_2 + q_3 p_3)q_1, \\
&\quad (-q_3^2 - q_1^2)p_2 + (q_3 p_3 + q_1 p_1)q_2, \\
&\quad (-q_1^2 - q_2^2)p_3 + (q_1 p_1 + q_2 p_2)q_3) \\
&\stackrel{(6.7)}{=} ((q_0^2 - 1 + q_1^2)p_1 + (q_2 p_2 + q_3 p_3)q_1, \\
&\quad (q_0^2 - 1 + q_2^2)p_2 + (q_1 p_1 + q_3 p_3)q_2, \\
&\quad (q_0^2 - 1 + q_3^2)p_3 + (q_1 p_1 + q_2 p_2)q_3) \\
&= (q_0^2 - 1)p + \langle q, p \rangle q
\end{aligned} \tag{6.9}$$

Es wird nun hergeleitet, wie die Matrixdarstellung einer Rotation aus der Quaternionendarstellung abgeleitet werden kann. Hierzu betrachten wir zunächst nochmals das Produkt zweier Quaternionen:

$$\begin{aligned}
\hat{a}\hat{b} &= (a_0 b_0 - a_1 b_1 - a_2 b_2 - a_3 b_3) + \\
&\quad (a_0 b_1 + a_1 b_0 + a_2 b_3 - a_3 b_2)i + \\
&\quad (a_0 b_2 - a_1 b_3 + a_2 b_0 + a_3 b_1)j + \\
&\quad (a_0 b_3 + a_1 b_2 - a_2 b_1 + a_3 b_0)k
\end{aligned}$$

Bezüglich der Basis $(1, i, j, k)$ des Quaternionenraumes kann diese Multiplikation durch eine 4×4 -Matrix ausgedrückt werden, wobei die Quaternionenmultiplikation mit a von links bzw. mit b von rechts jeweils durch eine Matrix $L(a)$ bzw. $R(b)$ beschrieben ist:

$$\hat{a}\hat{b} = L(a)b = \begin{pmatrix} a_0 & -a_1 & -a_2 & -a_3 \\ a_1 & a_0 & -a_3 & a_2 \\ a_2 & a_3 & a_0 & -a_1 \\ a_3 & -a_2 & a_1 & a_0 \end{pmatrix} b = R(b)a = \begin{pmatrix} b_0 & -b_1 & -b_2 & -b_3 \\ b_1 & b_0 & b_3 & -b_2 \\ b_2 & -b_3 & b_0 & b_1 \\ b_3 & b_2 & -b_1 & b_0 \end{pmatrix} a$$

Damit ergibt sich für die Drehungsoperation $\tilde{p} = \hat{q}\hat{p}\hat{q}$ die folgende 4×4 -Matrixdarstellung:

$$\hat{q}\hat{p}\hat{q} = L(\hat{q})\hat{p}\hat{q} = L(\hat{q})R(\hat{q})\hat{p} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 0 & 2(q_1q_2 + q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 0 & 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{pmatrix} \begin{pmatrix} 0 \\ p_1 \\ p_2 \\ p_3 \end{pmatrix} \quad (6.10)$$

Der Matrixeintrag $(1,1)$ ergibt sich aus $q_0^2 + q_1^2 + q_2^2 + q_3^2 \stackrel{(6.7)}{=} 1$. Die 3×3 -Untermatrix des imaginären Basisanteils i, j, k entspricht der Rotationsmatrix im \mathbb{R}^3 .

6.4.3 Implementierung der Quaternionenalgebra \mathbb{H}

Die Implementierung kann mittels eines Skalars $w \in \mathbb{R}$ für den reellen Anteil und eines Vektors $v \in \mathbb{R}^3$ für die Koeffizienten der imaginären Anteile i, j, k erfolgen. Dies hat den Vorteil, daß die Basisoperationen für Quaternionen aufbauend auf dem üblichen Skalar- und Vektorprodukt des \mathbb{R}^3 implementiert werden können. Für eine Drehung mit Drehachse d , $\|d\| = 1$ und Winkel θ ergibt sich nach (6.6):

$$w := \cos \frac{\theta}{2}; \quad v := \sin \frac{\theta}{2} d$$

Die Transformation eines Punktes $p \in \mathbb{R}^3$ kann dann mit (6.8) wie folgt berechnet werden:

$$\tilde{p} := 2(\langle v, p \rangle v + (w^2 - 0.5)p + w(v \times p))$$

Eine Verkettung zweier Rotationen, siehe (6.4), ist gegeben durch

$$\begin{aligned} w_{\text{gesamt}} &:= w_1 w_2 - \langle v_1, v_2 \rangle \\ v_{\text{gesamt}} &:= w_1 v_2 + w_2 v_1 + v_1 \times v_2 \end{aligned}$$

Die Konvertierung von der Quaternionendarstellung in eine reelle 3×3 -Rotationsmatrix ergibt sich aus (6.10).

Kapitel 7

Infinitesimale Drehungen

In Kapitel 6 wurden Eigenschaften räumlicher Drehungen zusammengefaßt und verschiedene Darstellungsmöglichkeiten diskutiert. In diesem Abschnitt wird nun untersucht, welche zusätzlichen Eigenschaften für sehr kleine Drehungen gelten. Eine *infinitesimale Drehung* ist eine Drehungsoperation, durch welche die Vektorkomponenten nur geringfügig verändert werden, d. h. es soll gelten:

$$\tilde{x}_i = x_i + \varepsilon_{i1}x_1 + \varepsilon_{i2}x_2 + \varepsilon_{i3}x_3 . \quad (7.1)$$

\tilde{x} bezeichnet im folgenden immer das Bild von x unter einer infinitesimalen Drehung. Die ε_{ij} werden als infinitesimale Größen angesehen, so daß in allen folgenden Rechnungen nur die erste, nichtverschwindende Ordnung der ε_{ij} berücksichtigt wird. Gleichheit unter Vernachlässigung von Termen höherer Ordnung wird durch \doteq gekennzeichnet. Die komponentenweisen Beziehungen $\tilde{x}_i = x_i + \sum_j \varepsilon_{ij}x_j$ lauten in Matrixschreibweise $\tilde{x} = (\mathbf{1} + \varepsilon)x$.

Eigenschaft 16

Infinitesimale Drehungen verhalten sich kommutativ.

Beweis:

Seien $(\mathbf{1} + \varepsilon_1)$ und $(\mathbf{1} + \varepsilon_2)$ infinitesimale Drehungen. Damit folgt:
 $(\mathbf{1} + \varepsilon_1)(\mathbf{1} + \varepsilon_2) = \mathbf{1}^2 + \varepsilon_1\mathbf{1} + \mathbf{1}\varepsilon_2 + \varepsilon_1\varepsilon_2 \doteq \mathbf{1} + \varepsilon_1 + \varepsilon_2$

□

Eigenschaft 17

Das inverse Element zu $(\mathbf{1} + \varepsilon)$ ist $(\mathbf{1} - \varepsilon)$.

Beweis:

Dies folgt aus $(\mathbf{1} + \varepsilon)(\mathbf{1} - \varepsilon) = \mathbf{1}^2 + \varepsilon\mathbf{1} - \mathbf{1}\varepsilon + \varepsilon^2 \doteq \mathbf{1}$

□

Eigenschaft 18

Jede infinitesimale Drehung $R = (\mathbf{1} + \varepsilon)$ ist von der Form

$$R = \mathbf{1} + \begin{pmatrix} 0 & \gamma & -\beta \\ -\gamma & 0 & \alpha \\ \beta & -\alpha & 0 \end{pmatrix} = \begin{pmatrix} 1 & \gamma & -\beta \\ -\gamma & 1 & \alpha \\ \beta & -\alpha & 1 \end{pmatrix} .$$

Beweis:

Orthogonale Matrizen sind durch $R^{-1} = R^t$ definiert, so daß für infinitesimale Drehungen gilt: $\mathbf{1} - \varepsilon \doteq R^{-1} = R^t = \mathbf{1} + \varepsilon^t$ und damit folgt $-\varepsilon \doteq \varepsilon^t$, d. h. ε ist antisymmetrisch. Für jede antisymmetrische Matrix ε gilt nun: $\varepsilon_{ii} = 0$ und $\varepsilon_{ij} = -\varepsilon_{ji}$ für $i \neq j$.

Die Vorzeichenverteilung kann beliebig gewählt werden. Im Hinblick auf eine einfache Berechnung der Koordinatentransformation als Vektorprodukt (siehe Eigenschaft 19), wird sie jedoch wie oben angegeben vereinbart. \square

Eigenschaft 18 kann auch angelehnt an die Darstellung durch Eulersche Winkel abgeleitet werden. Für eine infinitesimale Drehung sind die eindeutig bestimmten Eulerschen Winkel nach Satz 14 bzgl. des Drehachsensystems z, x, z jedoch nicht notwendigerweise klein. In diesem Falle ist die Betrachtung des Drehachsensystems x, y, z günstiger, da die Bilder der Koordinatenachsen nur geringfügig von ihren Urbildern abweichen. Zudem sind infinitesimale Drehungen nach Eigenschaft 16 kommutativ und die Reihenfolge der Operationen damit unerheblich.

Für infinitesimale Winkel φ ergeben sich für $\cos \varphi$ und $\sin \varphi$ die folgenden Linearisierungen aus der Reihenentwicklung der Funktionen:

$$\begin{aligned}\sin \varphi &= \varphi^1 - \frac{\varphi^3}{3!} + \frac{\varphi^5}{5!} - \dots \doteq \varphi \\ \cos \varphi &= \varphi^0 - \frac{\varphi^2}{2!} + \frac{\varphi^4}{4!} - \dots \doteq 1\end{aligned}$$

Aus den Matrixdarstellungen für die einzelnen Teildrehungen folgt damit:

$$\begin{aligned}& \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \varphi & -\sin \varphi \\ 0 & \sin \varphi & \cos \varphi \end{pmatrix} \begin{pmatrix} \cos \beta & 0 & -\sin \beta \\ 0 & 1 & 0 \\ \sin \beta & 0 & \cos \beta \end{pmatrix} \begin{pmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \doteq \\ & \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & -\varphi \\ 0 & \varphi & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & -\beta \\ 0 & 1 & 0 \\ \beta & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & -\psi & 0 \\ \psi & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \doteq \begin{pmatrix} 1 & -\psi & -\beta \\ \psi & 1 & -\varphi \\ \beta & \varphi & 1 \end{pmatrix} \quad (7.2)\end{aligned}$$

Mit $\psi = -\gamma$ und $\varphi = -\alpha$ entspricht dies der Form nach Eigenschaft 18:

$$R = \mathbf{1} + \begin{pmatrix} 0 & \gamma & -\beta \\ -\gamma & 0 & \alpha \\ \beta & -\alpha & 0 \end{pmatrix} = \mathbf{1} + \varepsilon .$$

Eigenschaft 19

Für die Änderung Δx eines Vektors x unter einer infinitesimalen Drehung $(\mathbf{1} + \varepsilon)$ in der obigen Darstellung gilt $\Delta x = (\alpha, \beta, \gamma) \times x$.

Beweis:

Die Änderung Δx eines Vektors x hat nach Gleichung 7.1 die Form $\Delta x = \tilde{x} - x = \varepsilon x$. Mit Eigenschaft 18 ist damit

$$\Delta x = (x_3\beta - x_2\gamma, x_1\gamma - x_3\alpha, x_2\alpha - x_1\beta) = (\alpha, \beta, \gamma) \times x$$

□

Eigenschaft 20

Eine infinitesimale Rotation eines konvexen Objektes erzeugt einen Hüllkörper für dieses Objekt unter einer orthogonalen Transformation.

Der Beweis dieser für das lokale Optimierungsverfahren zentralen Eigenschaft wird im folgenden Abschnitt gegeben. Eigenschaft 20 entspricht der Aussage von Satz 31.

7.1 Hüllkörpereigenschaft infinitesimaler Rotationen

In diesem Abschnitt wird bewiesen, daß jede infinitesimale Rotation eines konvexen Objektes einen Hüllkörper für dieses Objekt unter einer orthogonormalen Rotation liefert. Für konvexe Metriken (siehe Def. 28, sowie [CD85, GO97]), wie insbesondere den gewöhnlichen euklidischen Abstand, muß hierzu lediglich gezeigt werden, daß eine Einbettung des Einheitswürfels in sein Bild unter der infinitesimalen Rotation existiert.

Das Bild des Einheitswürfels unter einer infinitesimalen Rotation \mathbf{R} ist ein Spat (siehe auch Abb. 7.9 auf Seite 114), der von den Bildern der drei kanonischen Basisvektoren $\mathbf{R}e_i = \mathbf{b}_i$ aufgespannt wird. Im folgenden werden die Bilder der Basisvektoren unter einer infinitesimalen Rotation immer mit \mathbf{b}_i und als Spatvektoren bezeichnet.

Die Einbettung des Einheitswürfels in den Spat wird durch eine orthonormale Rotationsabbildung angegeben. Die entsprechenden Bilder der e_i werden für diese Abbildung jeweils mit o_i bezeichnet und bilden ein rechtshändiges Orthonormalsystem. Der Beweis für die Existenz der Einbettung erfolgt konstruktiv; es werden insbesondere die entsprechenden Bilder o_i angegeben. Satz 31 liefert diese Aussage unter Verwendung der hierfür notwendigen vorangehenden Lemmata. Der Aufbau der Hilfssätze gliedert sich wie folgt:

In Lemma 21 wird gezeigt, daß für eine infinitesimale Rotation die paarweisen Winkel zwischen den Spatvektoren \mathbf{b}_i immer größer gleich 90° angenommen werden können, was im folgenden dann vorausgesetzt wird. Insbesondere gilt für jedes Paar von Spatvektoren mit $i \neq j$ damit $\langle \mathbf{b}_i, \mathbf{b}_j \rangle \leq 0$. Diese Eigenschaft bleibt auch erhalten, wenn zwei Spatvektoren $\mathbf{b}_i, \mathbf{b}_j$ auf die Ebene E_k senkrecht zum dritten Basisvektor \mathbf{b}_k projiziert werden (siehe Lemma 23).

In Lemma 24 bis 26 werden dann Aussagen über Längen und Abstände innerhalb des Spates hergeleitet, die für den Beweis benötigt werden.

In Lemma 27 wird dann hergeleitet, wann der von einem Orthonormalsystem o_1, o_2, o_3 aufgespannte Würfel innerhalb des Spates liegt. Es zeigt sich, daß über die Einbettung der Würfeldiagonalen $d = o_1 + o_2 + o_3$ die Einbettung aller anderen Spatecken sichergestellt werden kann. d ist gleichzeitig die Drehachse der gesuchten orthonormalen Transformation.

Die folgenden Lemmata 28 und 29 zeigen, wie die Berechnung des vollständigen Orthonormalsystems erfolgt, wenn die Drehachse gegeben ist. Die Verwendung trigonometrischer Funktionen ist hierzu nicht erforderlich, so daß sich effiziente Implementierungen ergeben.

Zusammenfassend hier die Bezeichnungen, die in diesem Abschnitt verwendet werden:

$$\mathbf{R} = \begin{pmatrix} 1 & \gamma & -\beta \\ -\gamma & 1 & \alpha \\ \beta & -\alpha & 1 \end{pmatrix} \quad \text{infinitesimale Rotation}$$

$$\underbrace{\quad}_{\mathbf{b}_1} \quad \underbrace{\quad}_{\mathbf{b}_2} \quad \underbrace{\quad}_{\mathbf{b}_3} \quad \text{Spatvektoren}$$

i, j, k	Indizes jeweils aus $\{1, 2, 3\}$
o_1, o_2, o_3	Orthonormalsystem/Rotationsabbildung
$d := o_1 + o_2 + o_3$	Diagonale des Einheitswürfels/Drehachse
$\theta := \arccos(1/\sqrt{3})$	Winkel zwischen d und o_i
$\eta := \arccos(\sqrt{2}/\sqrt{3})$	Winkel zwischen d und Flächen o_i, o_j
$\frac{\vec{v}}{v}$	Normierter Vektor in Richtung v der Länge 1

Lemma 21

Für mindestens eine Ecke des von $\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3$ aufgespannten Spates gilt, daß alle paarweisen Winkel zwischen den Spatschenkeln größer gleich 90° sind. Für diese Ecke gilt dann insbesondere $\langle \mathbf{b}_i, \mathbf{b}_j \rangle \leq 0$ für alle Paare mit $i \neq j$.

Beweis:

Zu zeigen ist, daß für eine Ecke des Spates die paarweisen Skalarprodukte $\langle \mathbf{b}_i, \mathbf{b}_j \rangle$ ($i \neq j$) zwischen den Spatschenkeln jeweils kleiner gleich Null sind. Zu betrachten sind also die Vorzeichen der folgenden Terme:

$$\begin{aligned} \langle \mathbf{b}_1, \mathbf{b}_2 \rangle &= \langle (1, -\gamma, \beta), (\gamma, 1, -\alpha) \rangle = \gamma - \gamma - \beta\alpha = -\alpha\beta \\ \langle \mathbf{b}_2, \mathbf{b}_3 \rangle &= \langle (\gamma, 1, -\alpha), (-\beta, \alpha, 1) \rangle = -\gamma\beta + \alpha - \alpha = -\beta\gamma \\ \langle \mathbf{b}_1, \mathbf{b}_3 \rangle &= \langle (1, -\gamma, \beta), (-\beta, \alpha, 1) \rangle = -\beta - \gamma\alpha + \beta = -\alpha\gamma \end{aligned} \quad (7.3)$$

Die nachfolgende Tabelle zeigt die Vorzeichenverteilung der Skalarprodukte für die acht möglichen Vorzeichenkombinationen von α, β, γ . Der Eintrag „-“ in der Tabelle bedeutet hier „ ≤ 0 “. Man beachte, daß die positiven Vorzeichen jeweils paarweise auftreten. Damit diese Skalarprodukte ebenfalls kleiner gleich Null werden, muß das Vorzeichen genau eines Spatvektors

geändert werden. Man kann leicht verifizieren, daß dies durch die in der letzten Spalte angegebenen Basistransformationen erreicht wird. Hierbei wird zusätzlich die Position von zwei Spatvektoren getauscht, um trotz des Vorzeichenwechsels ein rechtshändiges System zu erhalten.

α	β	γ	$\langle \mathbf{b}_1, \mathbf{b}_2 \rangle$ $-\alpha\beta$	$\langle \mathbf{b}_1, \mathbf{b}_3 \rangle$ $-\beta\gamma$	$\langle \mathbf{b}_2, \mathbf{b}_3 \rangle$ $-\alpha\gamma$	Basistransformation
+	+	+	-	-	-	$(\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3)$
-	-	-	-	-	-	$(\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3)$
+	-	+	+	+	-	$(-\mathbf{b}_1, \mathbf{b}_3, \mathbf{b}_2)$
-	+	-	+	+	-	$(-\mathbf{b}_1, \mathbf{b}_3, \mathbf{b}_2)$
-	+	+	+	-	+	$(\mathbf{b}_3, -\mathbf{b}_2, \mathbf{b}_1)$
+	-	-	+	-	+	$(\mathbf{b}_3, -\mathbf{b}_2, \mathbf{b}_1)$
+	+	-	-	+	+	$(\mathbf{b}_2, \mathbf{b}_1, -\mathbf{b}_3)$
-	-	+	-	+	+	$(\mathbf{b}_2, \mathbf{b}_1, -\mathbf{b}_3)$

□

Im folgenden seien $\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3$ immer entsprechend Lemma 21 gewählt, d. h. es gelte jeweils $\langle \mathbf{b}_i, \mathbf{b}_j \rangle \leq 0$ für $i \neq j$.

Korollar 22

Die Transformation $\mathbf{R} = (\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3)$ ist orthogonal genau dann, wenn mindestens zwei der Werte α, β, γ gleich Null sind.

Beweis:

Man betrachte die drei Skalarprodukte in (7.3). Offenbar sind diese genau dann gleichzeitig Null, wenn mindestens zwei der Werte α, β, γ gleich Null sind.

□

Lemma 23

Der Winkel zwischen zwei Spatschenkeln $\mathbf{b}_i, \mathbf{b}_j$ projiziert auf die Ebene orthogonal zum dritten Spatvektor \mathbf{b}_k ist jeweils größer gleich 90° .

Beweis:

Wir betrachten den Fall für $i = 1, j = 2$ und $k = 3$. Ferner seien p_1, p_2 die Projektionen von $\mathbf{b}_1, \mathbf{b}_2$ auf die Ebene senkrecht zu \mathbf{b}_3 . Zu zeigen ist also $\angle(p_1, p_2) \geq 90^\circ$. Dies ist äquivalent zu $\langle \vec{p}_1, \vec{p}_2 \rangle \leq 0$ und wegen positiver Längen zu $\langle p_1, p_2 \rangle \leq 0$.

Es gilt (für $i \in \{1, 2\}$):

$$p_i = \mathbf{b}_i^\perp \mathbf{b}_3 = \mathbf{b}_i - \frac{\langle \mathbf{b}_i, \mathbf{b}_3 \rangle}{\langle \mathbf{b}_3, \mathbf{b}_3 \rangle} \mathbf{b}_3$$

Damit

$$\begin{aligned}
\langle p_1, p_2 \rangle &= \left\langle \mathbf{b}_1 - \frac{\langle \mathbf{b}_1, \mathbf{b}_3 \rangle}{\langle \mathbf{b}_3, \mathbf{b}_3 \rangle} \mathbf{b}_3, \mathbf{b}_2 - \frac{\langle \mathbf{b}_2, \mathbf{b}_3 \rangle}{\langle \mathbf{b}_3, \mathbf{b}_3 \rangle} \mathbf{b}_3 \right\rangle \\
&= \langle \mathbf{b}_1, \mathbf{b}_2 \rangle - 2 \frac{\langle \mathbf{b}_1, \mathbf{b}_3 \rangle \langle \mathbf{b}_2, \mathbf{b}_3 \rangle}{\langle \mathbf{b}_3, \mathbf{b}_3 \rangle} + \frac{\langle \mathbf{b}_1, \mathbf{b}_3 \rangle \langle \mathbf{b}_2, \mathbf{b}_3 \rangle}{\langle \mathbf{b}_3, \mathbf{b}_3 \rangle^2} \langle \mathbf{b}_3, \mathbf{b}_3 \rangle \\
&= \langle \mathbf{b}_1, \mathbf{b}_2 \rangle - \frac{\langle \mathbf{b}_1, \mathbf{b}_3 \rangle \langle \mathbf{b}_2, \mathbf{b}_3 \rangle}{\langle \mathbf{b}_3, \mathbf{b}_3 \rangle} \tag{7.4}
\end{aligned}$$

Nach Voraussetzung bzw. Lemma 21 gilt $\langle \mathbf{b}_i, \mathbf{b}_j \rangle \leq 0$ für alle gemischten Terme mit $i \neq j$ und es gilt zudem $\langle \mathbf{b}_3, \mathbf{b}_3 \rangle > 0$. Somit sind die beiden Summanden in (7.4) negativ und damit $\langle p_1, p_2 \rangle \leq 0$, was zu zeigen war. \square

In den folgenden drei Hilfssätzen werden Aussagen über Seitenlängen und Abstände innerhalb des Spates hergeleitet.

Lemma 24

Für die Längen der Spatseiten bzw. der Spatdiagonalen gilt:

1. Für alle Seitenlängen $\|\mathbf{b}_i\|$ des durch $\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3$ aufgespannten Spates gilt $\|\mathbf{b}_i\| \geq 1$.
2. Für die beiden Seitendiagonalen $(\mathbf{b}_i \pm \mathbf{b}_j)$ ($i \neq j$) innerhalb der von $\mathbf{b}_i, \mathbf{b}_j$ aufgespannten Spatflächen gilt jeweils $\|\mathbf{b}_i \pm \mathbf{b}_j\| \geq \sqrt{2}$.
3. Für die Spatdiagonalen $\mathbf{b}_1 \pm \mathbf{b}_2 \pm \mathbf{b}_3$ gilt entsprechend $\|\mathbf{b}_1 \pm \mathbf{b}_2 \pm \mathbf{b}_3\| \geq \sqrt{3}$.

Beweis:

$\|\mathbf{b}_i\| \geq 1$ trivial, da eine Koordinate gleich Eins und die beiden restlichen quadratischen Terme größer gleich Null sind.

Sei $i = 1, j = 2$. Für die beiden Diagonalen $\mathbf{b}_1 + \mathbf{b}_2$ und $\mathbf{b}_1 - \mathbf{b}_2$ auf der von $\mathbf{b}_1, \mathbf{b}_2$ aufgespannten Spatfläche gilt damit

$$\begin{aligned}
\|\mathbf{b}_1 + \mathbf{b}_2\|^2 &= (1 + \gamma)^2 + (-\gamma + 1)^2 + (\beta - \alpha)^2 \\
&= 2 + 2\gamma^2 + (\beta - \alpha)^2 \geq 2 \tag{7.5}
\end{aligned}$$

sowie

$$\begin{aligned}
\|\mathbf{b}_1 - \mathbf{b}_2\|^2 &= (1 - \gamma)^2 + (-\gamma - 1)^2 + (\beta + \alpha)^2 \\
&= 2 + 2\gamma^2 + (\beta + \alpha)^2 \geq 2 \tag{7.6}
\end{aligned}$$

Der Beweis für die beiden anderen Seitendiagonalen erfolgt analog hierzu. Für die Spatdiagonalen betrachten wir die Summe mit positiven Spatvektoren. Die restlichen Fälle folgen wie oben analog hierzu:

$$\begin{aligned}
\|\mathbf{b}_1 + \mathbf{b}_2 + \mathbf{b}_3\|^2 &= (1 + \gamma - \beta)^2 + (-\gamma + 1 + \alpha)^2 + (\beta - \alpha + 1)^2 \\
&= 3 + 2\alpha^2 + 2\beta^2 + 2\gamma^2 - 2\beta\gamma - 2\alpha\gamma - 2\alpha\beta \tag{7.7} \\
&= 3 + (\alpha - \beta)^2 + (\alpha - \gamma)^2 + (\beta - \gamma)^2 \geq 3 \ . \quad \square
\end{aligned}$$

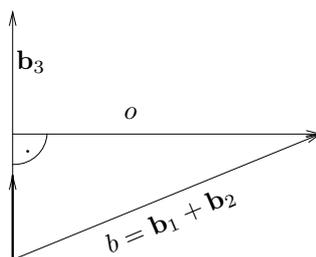


Abbildung 7.1: Bezeichnungen der Vektoren zu Lemma 25

Lemma 25

Die Länge einer Seitendiagonalen $\mathbf{b}_i \pm \mathbf{b}_j$ projiziert auf die Ebene orthogonal zum dritten Spatvektor \mathbf{b}_k ist größer $\sqrt{2}$. Dies entspricht dem Abstand zwischen zwei sich diagonal gegenüberliegenden Spatkanten.

Beweis:

Wir betrachten den Fall $i = 1, j = 2, k = 3$; die anderen Fälle sind analog hierzu. Die Projektion von $\mathbf{b}_1 + \mathbf{b}_2$ auf die Ebene orthogonal zu \mathbf{b}_3 sei mit o bezeichnet; abkürzend sei $b = \mathbf{b}_1 + \mathbf{b}_2$ (siehe Abb. 7.1). Zu zeigen ist $\|o\| \geq \sqrt{2}$ bzw. $\|o\|^2 \geq 2$.

Mit

$$o = (\mathbf{b}_1 + \mathbf{b}_2)^\perp \mathbf{b}_3 = b - \frac{\langle b, \mathbf{b}_3 \rangle}{\|\mathbf{b}_3\|^2} \mathbf{b}_3 \quad (7.8)$$

ist

$$\begin{aligned} \|o\|^2 &\stackrel{(7.8)}{=} \left\langle b - \frac{\langle b, \mathbf{b}_3 \rangle}{\|\mathbf{b}_3\|^2} \mathbf{b}_3, b - \frac{\langle b, \mathbf{b}_3 \rangle}{\|\mathbf{b}_3\|^2} \mathbf{b}_3 \right\rangle \\ &= \langle b, b \rangle - 2 \left\langle b, \frac{\langle b, \mathbf{b}_3 \rangle}{\|\mathbf{b}_3\|^2} \mathbf{b}_3 \right\rangle + \frac{\langle b, \mathbf{b}_3 \rangle^2}{\|\mathbf{b}_3\|^4} \langle \mathbf{b}_3, \mathbf{b}_3 \rangle \\ &= \|b\|^2 - \frac{\langle b, \mathbf{b}_3 \rangle^2}{\|\mathbf{b}_3\|^2} \\ &\stackrel{(7.5)}{=} 2 + 2\gamma^2 + (\beta - \alpha)^2 - \frac{\langle b, \mathbf{b}_3 \rangle^2}{\|\mathbf{b}_3\|^2} . \end{aligned} \quad (7.9)$$

Nach Subtraktion von 2 und Multiplikation mit $\|\mathbf{b}_3\|^2$, ist $2 \leq \|o\|^2$ damit äquivalent zu

$$\begin{aligned} 0 &\stackrel{(7.9)}{\leq} (2\gamma^2 + (\beta - \alpha)^2) \|\mathbf{b}_3\|^2 - \langle b, \mathbf{b}_3 \rangle^2 \\ &= (\beta - \alpha)^2 \|\mathbf{b}_3\|^2 + 2\gamma^2 \|\mathbf{b}_3\|^2 - \langle b, \mathbf{b}_3 \rangle^2 \end{aligned} \quad (7.10)$$

was im folgenden zu zeigen ist.

Nebenrechnungen:

$$\|\mathbf{b}_3\|^2 = 1 + \alpha^2 + \beta^2 \quad (7.11)$$

$$2\gamma^2\|\mathbf{b}_3\|^2 = 2\gamma^2 + 2\gamma^2\alpha^2 + 2\gamma^2\beta^2$$

$$\begin{aligned} -\langle \mathbf{b}, \mathbf{b}_3 \rangle^2 &= -\langle (1 + \gamma, 1 - \gamma, \beta - \alpha), (-\beta, \alpha, 1) \rangle^2 \\ &= -(-\gamma\beta - \gamma\alpha)^2 \\ &= -\gamma^2\alpha^2 - \gamma^2\beta^2 - 2\gamma^2\beta\alpha \end{aligned}$$

$$\begin{aligned} 2\gamma^2\|\mathbf{b}_3\|^2 - \langle \mathbf{b}, \mathbf{b}_3 \rangle^2 &= 2\gamma^2 + \gamma^2\alpha^2 + \gamma^2\beta^2 - 2\gamma^2\beta\alpha \\ &= 2\gamma^2 + \gamma^2(\beta - \alpha)^2 \end{aligned} \quad (7.12)$$

Damit ergibt sich für 7.10 insgesamt:

$$\begin{aligned} 0 &\leq (\beta - \alpha)^2\|\mathbf{b}_3\|^2 + \overbrace{2\gamma^2 + \gamma^2(\beta - \alpha)^2}^{(7.12)} \\ &\stackrel{(7.11)}{=} 2\gamma^2 + (1 + \alpha^2 + \beta^2 + \gamma^2)(\beta - \alpha)^2 \end{aligned} \quad (7.13)$$

□

Lemma 26

Der Abstand der parallelen Flächen in dem durch $\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3$ aufgespannten Spat ist größer gleich Eins.

Beweis:

Der Abstand der parallelen Spatflächen entspricht der Höhe h des Spates bzgl. der jeweiligen Grundfläche. Es gilt: Spatvolumen = Grundfläche \times Höhe und es ist zu zeigen $h \geq 1$. Wir betrachten die von $\mathbf{b}_1, \mathbf{b}_2$ aufgespannte Fläche. $\|\mathbf{b}_1 \times \mathbf{b}_2\|$ ist der Flächeninhalt des durch $\mathbf{b}_1, \mathbf{b}_2$ definierten Parallelogramms und $|\langle \mathbf{b}_3, \mathbf{b}_1 \times \mathbf{b}_2 \rangle|$ ist das Volumen des aufgespannten Spates. Damit folgt:

$$\begin{aligned} |\langle \mathbf{b}_3, \mathbf{b}_1 \times \mathbf{b}_2 \rangle| &= \|\mathbf{b}_1 \times \mathbf{b}_2\| h \\ \Leftrightarrow h &= \frac{|\langle \mathbf{b}_3, \mathbf{b}_1 \times \mathbf{b}_2 \rangle|}{\|\mathbf{b}_1 \times \mathbf{b}_2\|} \\ \Rightarrow h^2 &= \frac{\langle \mathbf{b}_3, \mathbf{b}_1 \times \mathbf{b}_2 \rangle^2}{\langle \mathbf{b}_1 \times \mathbf{b}_2, \mathbf{b}_1 \times \mathbf{b}_2 \rangle} \end{aligned} \quad (7.14)$$

Nebenrechnung:

$$\begin{aligned}
 \mathbf{b}_1 \times \mathbf{b}_2 &= (1, -\gamma, \beta) \times (\gamma, 1, -\alpha) \\
 &= (\gamma\alpha - \beta, \beta\gamma + \alpha, 1 + \gamma^2) \\
 &= (-\beta, \alpha, 1) + \gamma(\alpha, \beta, \gamma) \\
 &= \mathbf{b}_3 + \gamma(\alpha, \beta, \gamma)
 \end{aligned} \tag{7.15}$$

$$\langle \mathbf{b}_3, (\alpha, \beta, \gamma) \rangle = \langle (-\beta, \alpha, 1), (\alpha, \beta, \gamma) \rangle = \gamma \tag{7.16}$$

$$\begin{aligned}
 \langle \mathbf{b}_3, \mathbf{b}_1 \times \mathbf{b}_2 \rangle &\stackrel{(7.15)}{=} \langle \mathbf{b}_3, \mathbf{b}_3 + \gamma(\alpha, \beta, \gamma) \rangle \\
 &\stackrel{(7.16)}{=} \|\mathbf{b}_3\|^2 + \gamma^2
 \end{aligned} \tag{7.17}$$

$$\begin{aligned}
 \langle \mathbf{b}_3, \mathbf{b}_1 \times \mathbf{b}_2 \rangle^2 &\stackrel{(7.17)}{=} \|\mathbf{b}_3\|^4 + 2\|\mathbf{b}_3\|^2\gamma^2 + \gamma^4 \\
 &\stackrel{(7.11)}{=} \|\mathbf{b}_3\|^4 + \gamma^2(2 + 2\alpha^2 + 2\beta^2 + \gamma^2)
 \end{aligned} \tag{7.18}$$

$$\begin{aligned}
 \langle \mathbf{b}_1 \times \mathbf{b}_2, \mathbf{b}_1 \times \mathbf{b}_2 \rangle &\stackrel{(7.15)}{=} \langle \mathbf{b}_3 + \gamma(\alpha, \beta, \gamma), \mathbf{b}_3 + \gamma(\alpha, \beta, \gamma) \rangle \\
 &= \langle \mathbf{b}_3, \mathbf{b}_3 \rangle + 2\gamma\langle \mathbf{b}_3, (\alpha, \beta, \gamma) \rangle + \gamma^2(\alpha^2 + \beta^2 + \gamma^2) \\
 &\stackrel{(7.16)}{=} \|\mathbf{b}_3\|^2 + 2\gamma^2 + \gamma^2(\alpha^2 + \beta^2 + \gamma^2) \\
 &= \|\mathbf{b}_3\|^2 + \gamma^2(2 + \alpha^2 + \beta^2 + \gamma^2)
 \end{aligned} \tag{7.19}$$

Mit (7.18),(7.19) damit:

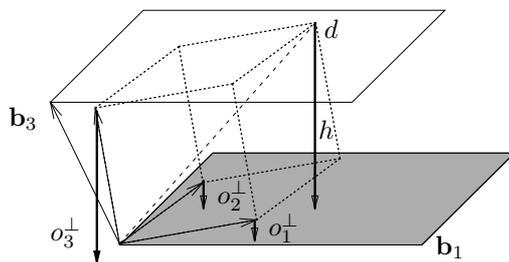
$$h^2 = \frac{\|\mathbf{b}_3\|^4 + \gamma^2(2 + 2\alpha^2 + 2\beta^2 + \gamma^2)}{\|\mathbf{b}_3\|^2 + \gamma^2(2 + \alpha^2 + \beta^2 + \gamma^2)} \tag{7.20}$$

Differenz von Zähler und Nenner in (7.20):

$$\begin{aligned}
 \|\mathbf{b}_3\|^4 - \|\mathbf{b}_3\|^2 + \gamma^2(\alpha^2 + \beta^2) &= (\|\mathbf{b}_3\|^2 - 1)\|\mathbf{b}_3\|^2 + \gamma^2(\alpha^2 + \beta^2) \\
 &\stackrel{(7.11)}{=} (\alpha^2 + \beta^2)\|\mathbf{b}_3\|^2 + \gamma^2(\alpha^2 + \beta^2) \\
 &= (\alpha^2 + \beta^2)(\|\mathbf{b}_3\|^2 + \gamma^2) \\
 &\stackrel{(7.11)}{=} (\alpha^2 + \beta^2)(1 + \alpha^2 + \beta^2 + \gamma^2)
 \end{aligned} \tag{7.21}$$

Vereinfachung Nenner:

$$\begin{aligned}
 \|\mathbf{b}_3\|^2 + \gamma^2(2 + \alpha^2 + \beta^2 + \gamma^2) &= (\|\mathbf{b}_3\|^2 + \gamma^2) + \gamma^2(1 + \alpha^2 + \beta^2 + \gamma^2) \\
 &\stackrel{(7.11)}{=} (1 + \gamma^2)(1 + \alpha^2 + \beta^2 + \gamma^2)
 \end{aligned} \tag{7.22}$$

Abbildung 7.2: Richtungskomponenten der o_i senkrecht zu einer Spatfläche

Gekürzte Form für h^2 nach (7.21),(7.22):

$$h^2 = 1 + \frac{\alpha^2 + \beta^2}{1 + \gamma^2} \quad (7.23)$$

Offensichtlich gilt nun $h^2 \geq 1$ und damit $h \geq 1$. □

Das folgende Lemma gibt zwei notwendige und zusammen hinreichende Bedingungen an, wann der von o_1, o_2, o_3 gebildete Einheitswürfel innerhalb des durch eine infinitesimale Rotation gegebenen Spates liegt.

Lemma 27

Sei $\mathbf{R} = (\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3)$ eine infinitesimale Rotation und o_1, o_2, o_3 ein Orthornormalsystem. Die folgenden beiden Eigenschaften bilden zusammen eine hinreichende Bedingung für die Einbettung des durch o_1, o_2, o_3 gegebenen Einheitswürfels in den durch $\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3$ aufgespannten Spat:

1. o_1, o_2, o_3 liegen innerhalb des Kegels $\{\lambda \mathbf{b}_1 + \mu \mathbf{b}_2 + \nu \mathbf{b}_3 \mid \lambda, \mu, \nu \geq 0\}$
2. $d = o_1 + o_2 + o_3$ liegt vollständig innerhalb des Spates

Beweis:

Eigenschaft 1 stellt sicher, daß keine der im Koordinatenursprung liegenden Würfel­flächen die entsprechenden Spat­flächen schneidet.

Zu zeigen ist noch, daß aus Eigenschaft 2 folgt, daß alle Ecken und damit der gesamte Einheitswürfel innerhalb des Spates liegen. Wir betrachten hierzu jeweils zwei parallele Spat­flächen und zeigen, daß die Ecken des Einheitswürfels zwischen diesen Flächen liegen, falls d zwischen den Flächen liegt.

In Abb. 7.2 sind für die Spat­fläche $\mathbf{b}_1, \mathbf{b}_2$ die Richtungsanteile von o_1, o_2, o_3 dargestellt, die senkrecht auf dieser Fläche stehen. Diese Anteile sind mit o_i^\perp bezeichnet und man beachte, daß wegen Bedingung 1 alle o_i im positiven Halbraum von $\mathbf{b}_1 \times \mathbf{b}_2$ liegen. Zudem ist der entsprechende Anteil von d dargestellt, welcher mit h bezeichnet wird. Es gilt: $h = \sum_{i=1}^3 o_i^\perp$. Bedingung 2 stellt sicher, daß der Abstand zwischen parallelen Spatebenen mindestens

$\|h\|$ ist. Die Abstände aller anderen Ecken von der Spatfläche müssen aber kleiner gleich $\|h\|$ sein, da sie sich aus Teilsummen der o_i^\perp zusammensetzen.

□

Die Existenz von Orthonormalsystemen, die Bedingung 1 von Lemma 27 erfüllen, wird im wesentlichen durch Lemma 21 sichergestellt. Die Fallunterscheidungen im Beweis zu Satz 31 sind eine Folge von Bedingung 2.

Eigenschaft 1 stellt keine notwendige Bedingung dar, wenn Translationen des Einheitswürfels aus dem Koordinatenursprung hinaus betrachtet werden. In diesem Falle ist für große Werte von α, β, γ wegen der Streckung des Spates eine nahezu beliebige Rotation erlaubt.

Man beachte zudem, daß Bedingung 2 nicht direkt aus Lemma 24 oder 26 folgt, in denen gezeigt wird, daß der Abstand zwischen den Spatflächen größer gleich 1 ist. Die Ausdehnung des Einheitswürfels senkrecht zu einer Spatfläche ist echt größer Eins, wenn die Würfelfläche nicht parallel zu dieser Fläche ausgerichtet ist.

Im weiteren wird mit d immer die Diagonale des Einheitswürfels nach der Transformation bzw. die Drehachse der orthonormalen Transformation bezeichnet. Es wird gezeigt, wie die vollständige orthonormale Transformation berechnet werden kann, wenn die Drehachse d und das Bild o_i eines Basisvektors e_i gegeben sind.

Sei $\theta := \arccos(1/\sqrt{3})(\approx 54,7^\circ)$ der Winkel zwischen d und o_i und $\eta := \arccos(\sqrt{2}/\sqrt{3})(\approx 35,3^\circ)$ der Winkel zwischen d und den Flächen des Einheitswürfels.

Lemma 28

Sei d mit $\|d\| = \sqrt{3}$ die Drehachse einer Rotationsabbildung und sei zudem das Bild o_1 des Basisvektors e_1 gegeben. Die zugehörigen Basisvektoren o_2, o_3 und die orthonormale Transformationsmatrix R sind dann wie folgt gegeben:

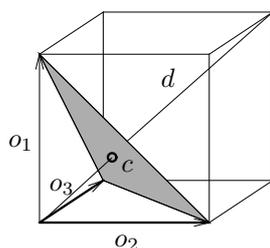
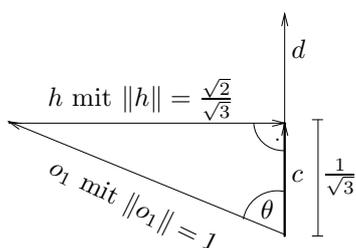
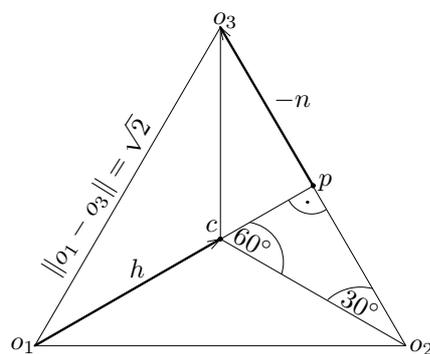
$$o_2 = 0.5(d - o_1 + (d \times o_1)) \quad (7.24)$$

$$o_3 = 0.5(d - o_1 - (d \times o_1)) \quad (7.25)$$

$$R = \begin{pmatrix} o_{11} & o_{21} & o_{31} \\ o_{12} & o_{22} & o_{32} \\ o_{13} & o_{23} & o_{33} \end{pmatrix} \quad (7.26)$$

Beweis:

Wir betrachten die durch o_1, o_2, o_3 gebildete Raumecke des Koordinatensystems, die in Abb. 7.3 dargestellt ist. Die Abbildungen 7.4 und 7.5 zeigen jeweils zweidimensionale Ansichten dieser Raumecke. Der im Zentrum liegende Punkt c ist dabei der Schnittpunkt der Würfel diagonalen mit der durch die Endpunkte von o_1, o_2, o_3 definierten Ebene E_Δ , die in Abb. 7.3 grau eingezeichnet ist.

Abbildung 7.3: Raumecke gebildet von o_1, o_2, o_3 Abbildung 7.4: Zweidimensionaler Schnitt; (o_1, d) -EbeneAbbildung 7.5: Durch o_1, o_2, o_3 definierte Ebene E_Δ der Raumecke

In Abbildung 7.4 sind die Würfeldiagonale d und das gegebene Bild $o_1 = Re_1$ des Einheitsvektors e_1 dargestellt. Mit \vec{d} wird die normierte Drehachse der Länge Eins bezeichnet. Es gilt für die in Abb. 7.4 dargestellten Vektoren c und h :

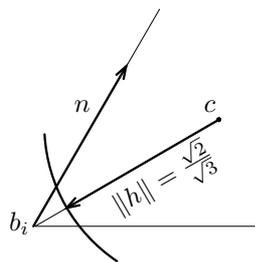
$$c = (\cos \theta) \vec{d} = \frac{1}{\sqrt{3}} \vec{d} = \frac{1}{3} d \quad (7.27)$$

$$h = c - o_1 \quad \text{mit } \|h\| = \sin \theta = \frac{\sqrt{2}}{\sqrt{3}} \quad (7.28)$$

Abbildung 7.5 zeigt den Schnitt durch den Einheitswürfel, der durch o_1, o_2, o_3 definiert wird (Ebene E_Δ). Im folgenden werden Darstellungen für die beiden Vektoren p und n abgeleitet. Es gilt:

$$p = c + 0.5 h \stackrel{(7.28)}{=} c + 0.5 (c - o_1) = 0.5 (3c - o_1) \stackrel{(7.27)}{=} 0.5 (d - o_1) \quad (7.29)$$

Für den Vektor $\tilde{n} := d \times o_1$ orthogonal zur (o_1, d) -Ebene sowie zum Vektor h gilt, daß $\|\tilde{n}\|$ dem Flächeninhalt des aus d und o_1 gebildeten Parallelogramms entspricht. Die Höhe dieses Parallelogramms ist gerade h (siehe nochmals Abb. 7.4), so daß gilt:

Abbildung 7.6: Extremalposition von o_i bzgl. n

$$\|\tilde{n}\| = \|d \times o_1\| = \|d\| \|h\| \stackrel{(7.28)}{=} \sqrt{3} \frac{\sqrt{2}}{\sqrt{3}} = \sqrt{2} \quad (7.30)$$

Die Richtung von \tilde{n} und n ist identisch und mit

$$\|n\| = 0.5 \|o_2 - o_3\| = 0.5 \sqrt{2} \stackrel{(7.30)}{=} 0.5 \|\tilde{n}\| \quad (7.31)$$

ergibt sich

$$n \stackrel{(7.31)}{=} 0.5 \tilde{n} = 0.5 (d \times o_1). \quad (7.32)$$

Für o_2 und o_3 ergeben sich damit insgesamt die folgenden Darstellungen:

$$\begin{aligned} o_2 &= p + n \\ &\stackrel{7.29, 7.32}{=} 0.5 (d - o_1) + 0.5 (d \times o_1) \\ &= 0.5 (d - o_1 + (d \times o_1)) \\ o_3 &= p - n = 0.5 (d - o_1 - (d \times o_1)) \end{aligned}$$

□

Im folgenden Lemma wird angegeben, wie die Extremalposition für das Bild eines Basisvektors o_i auf dem Rand des Spates bestimmt werden kann, wenn \mathbf{b}_i und die Drehachse der Transformation gegeben sind.

Lemma 29

Sei d die Drehachse einer Rotationsabbildung und E_Δ die Ebene mit Normalenvektor d und Abstand $1/\sqrt{3}$ vom Koordinatenursprung. Ferner sei \mathbf{b}_i ein Spatvektor mit $\angle(d, \mathbf{b}_i) \geq \theta$, der E_Δ schneidet. Zudem bezeichne n ($\|n\| = 1$) den Richtungsvektor der Schnittgeraden zwischen E_Δ und einer an \mathbf{b}_i angrenzenden Spatfläche. Eine Extremalposition für o_i ist dann gegeben durch $o_i = \mathbf{b}_i + \lambda n$, mit λ siehe Herleitung.

Beweis:

Es seien c, h wie in Lemma 28 definiert, d. h. c ist der Schnittpunkt von d mit E_Δ und h bezeichnet den Vektor von c nach o_i . Es gilt (siehe hierzu auch Abb. 7.6):

$$\|c - o_i\| = \|h\| \stackrel{(7.28)}{=} \sqrt{2}/\sqrt{3} \quad (7.33)$$

$$\Rightarrow \|c - (\mathbf{b}_i + \lambda n)\| - \sqrt{2}/\sqrt{3} = 0 \quad (7.34)$$

Sei $l = c - \mathbf{b}_i$. Nach quadrieren ergibt sich damit

$$0 = \langle l - \lambda n, l - \lambda n \rangle - 2/3 \quad (7.35)$$

$$\Leftrightarrow 0 = \lambda^2 - 2\langle l, n \rangle \lambda + \langle l, l \rangle - 2/3 \quad (7.36)$$

$$\Rightarrow \lambda_{1/2} = \langle l, n \rangle \pm \sqrt{\langle l, n \rangle^2 - \langle l, l \rangle + 2/3} \quad (7.37)$$

□

Es folgt nun ein Lemma, welches die Steigung der Diagonalen d projiziert auf eine Seitenfläche des Spates auf unter 45° einschränkt, wenn bestimmte Voraussetzungen an die Winkel zu den Spatschenkeln gegeben sind.

Lemma 30

Es seien $\angle(\mathbf{b}_i, d), \angle(\mathbf{b}_j, d) \leq \theta$, sowie $\angle(\mathbf{b}_i, \mathbf{b}_j) \geq 90^\circ$. $1/\sqrt{3}d = \vec{d} =: (d_1, d_2, d_3)$ bezeichne die normierte Diagonale. Zudem sei \hat{d} die Projektion von \vec{d} auf die $(\mathbf{b}_i, \mathbf{b}_i \times \mathbf{b}_j)$ -Ebene. Dann ist $\angle(\mathbf{b}_i, \hat{d}) \leq 45^\circ$.

Beweis:

Da nur Winkel der Vektoren relativ zueinander betrachtet werden, kann eine Skalierung und Drehung in ein geeignetes Koordinatensystem erfolgen. Es sei daher $\mathbf{b}_i = (1, 0, 0)$ und $\mathbf{b}_j = (-\lambda, 1, 0)$, wobei aus $\angle(\mathbf{b}_i, \mathbf{b}_j) \geq 90^\circ$ folgt, daß $\lambda \geq 0$ sein muß.

$$\angle(\mathbf{b}_i, d) \leq \theta \Rightarrow 1/\sqrt{3} \leq \langle \mathbf{b}_i, d_n \rangle = d_1 \quad (7.38)$$

$$\angle(\mathbf{b}_j, d) \leq \theta \Rightarrow 1/\sqrt{3} \leq \frac{\langle \mathbf{b}_j, d_n \rangle}{\|\mathbf{b}_j\|} = \frac{d_2 - \lambda d_1}{1 + \lambda^2} \quad (7.39)$$

Es gilt ferner

$$\hat{d} = (d_1, 0, d_3) \quad (7.40)$$

$$\|d_n\| = d_1^2 + d_2^2 + d_3^2 = 1 \quad (7.41)$$

und damit

$$\max \angle(\mathbf{b}_i, \hat{d}) \Leftrightarrow \min \frac{\langle \mathbf{b}_i, \hat{d} \rangle}{\|\mathbf{b}_i\| \|\hat{d}\|} \stackrel{(7.40)}{=} \min \frac{d_1}{d_1^2 + d_3^2} \stackrel{(7.41)}{=} \min \frac{d_1}{1 - d_2^2} \quad (7.42)$$

Um diesen Ausdruck zu minimieren, müssen d_1 und d_2 minimal gewählt werden. Mit (7.38) folgt $d_1 = 1/\sqrt{3}$ und damit aus (7.39) mit $\lambda = 0$ auch $d_2 = 1/\sqrt{3}$. Mit (7.41) muß dann ebenfalls $d_3 = 1/\sqrt{3}$ gelten. Insgesamt also $\angle(\mathbf{b}_i, \hat{\mathbf{d}}) \leq \angle((1, 0, 0), (1/\sqrt{3}, 0, 1/\sqrt{3})) = 45^\circ$.

□

Satz 31

Zu jeder infinitesimalen Rotation \mathbf{R} gibt es eine orthonormale Rotation R und Translation T , so daß das Bild eines jeden konvexen Objektes O unter $(T \circ R)$ vollständig im Bild von O unter \mathbf{R} enthalten ist. Die Transformation $(T \circ R)$ kann effektiv konstruiert werden.

Beweis:

Zu zeigen ist, daß der Einheitswürfel e_1, e_2, e_3 transformiert durch $(T \circ R)$ und aufgespannt von Vektoren o_1, o_2, o_3 , vollständig in dem durch die infinitesimale Rotation \mathbf{R} gegebenen und von $\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3$ aufgespannten Spat enthalten ist. Da die euklidische Norm eine konvexe Abstandsfunktion auf \mathbb{R}^3 definiert, überträgt sich die Hülleneigenschaft auf alle konvexen Objekte.

Nach Lemma 21 können wir durch geeignete Transformation der Spatbasis erreichen, daß die paarweisen Winkel zwischen den Spatschenkeln \mathbf{b}_i jeweils größer 90° sind. Dies wird im folgenden für $\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3$ angenommen. Die Translation T ist nur im Zusammenhang mit dieser Basistransformation erforderlich und wird zunächst nicht betrachtet.

Lemma 27 nennt zwei hinreichende Bedingungen für ein Orthonormalsystem (o_1, o_2, o_3) , so daß der Einheitswürfel vollständig im Spat enthalten ist. Im folgenden wird gezeigt, wie ein solches Orthonormalsystem konstruiert werden kann. Wir betrachten die Spatdiagonale $\tilde{\mathbf{d}} = \mathbf{b}_1 + \mathbf{b}_2 + \mathbf{b}_3$ als potentielle Drehachse der orthonormalen Rotation R und unterscheiden die folgenden Fälle:

1. $\angle(\tilde{\mathbf{d}}, \mathbf{b}_i) \geq \theta$ für alle Spatschenkel \mathbf{b}_i . Weitere Fallunterscheidung:
 - (a) $\tilde{\mathbf{d}}$ hat einen Winkel größer gleich η zu allen Spatflächen: In diesem Fall ist $\tilde{\mathbf{d}}$ die gesuchte Drehachse der Rotation.
 - (b) $\tilde{\mathbf{d}}$ hat einen Winkel kleiner η zu einer der Spatflächen $(\mathbf{b}_i, \mathbf{b}_j)$: Die Einbettung erfolgt wie unter Fall 3, da $\tilde{\mathbf{d}}$ bzgl. der Ebene $(\mathbf{b}_i, \mathbf{b}_j)$ „zu flach“ verläuft.
2. $\angle(\tilde{\mathbf{d}}, \mathbf{b}_i) \leq \theta$ für genau einen Spatschenkel \mathbf{b}_i . Anschaulich verläuft die Spatdiagonale $\tilde{\mathbf{d}}$ „zu steil“. Die Kante e_i des Einheitswürfels wird in diesem Fall auf den Spatschenkel \mathbf{b}_i abgebildet.
3. $\angle(\tilde{\mathbf{d}}, \mathbf{b}_i) \leq \theta$ und $\angle(\tilde{\mathbf{d}}, \mathbf{b}_j) \leq \theta$ für genau zwei Spatschenkel $\mathbf{b}_i, \mathbf{b}_j$. Anschaulich verläuft die Spatdiagonale $\tilde{\mathbf{d}}$ in diesem Fall gegenüber der Ebene $(\mathbf{b}_i, \mathbf{b}_j)$ „zu flach“. Eine Seitenfläche des Einheitswürfels wird daher in die $(\mathbf{b}_i, \mathbf{b}_j)$ -Ebene eingebettet.

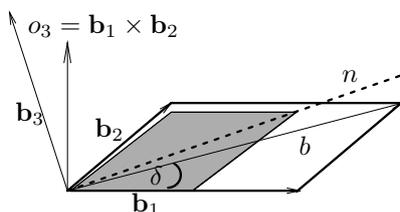


Abbildung 7.7: Bezeichnungen zum Beweis von Theorem 31, Fall 2

Der Fall, daß die Spatdiagonale \tilde{d} zu allen drei Spatschenkeln einen Winkel kleiner θ aufweist, kann nicht auftreten, da die paarweisen Winkel zwischen den Spatschenkeln nach Voraussetzung größer 90° sind. Im folgenden werden nun die Einbettungen angegeben und es wird gezeigt, daß sie die notwendigen und hinreichenden Bedingungen von Lemma 27 erfüllen.

zu Fall 1a:

Aus den Voraussetzungen über die Winkel von \tilde{d} zu den Spatschenkeln und Flächen folgt, daß ein Orthonormalsystem o_1, o_2, o_3 mit $d := \sqrt{3} \vec{\tilde{d}} = o_1 + o_2 + o_3$ existiert, welches Bedingung 1 von Lemma 27 erfüllt. Die Wahl der o_i erfolgt entsprechend Lemma 29 aus der Menge der Extremalpositionen für diejenigen Spatschenkel \mathbf{b}_i , welche die Ebene E_Δ schneiden.

Die Ebene E_Δ wird zumindest von demjenigen Spatschenkel geschnitten, der den kleinsten Winkel zu d einschließt. Wird sie von mehreren Spatschenkeln geschnitten, so ergibt sich ein maximaler gemeinsamer Drehwinkelbereich, der von zwei Extremalpositionen begrenzt wird. Von diesen wird eine als Bild o_i des kanonischen Basisvektors e_i gewählt und die Basis nach Lemma 28 vervollständigt.

Zu zeigen bleibt noch Bedingung 2 von Lemma 27. Die Drehachse d liegt aber in trivialer Weise innerhalb des Spates, da sie der Spatdiagonalen entspricht, die nach Lemma 24 mindestens die Länge $\sqrt{3}$ besitzt.

zu Fall 2 und 3 [Bedingung 1 von Lemma 27]:

Es wird zunächst gezeigt, wie ein Orthonormalsystem o_1, o_2, o_3 für Fall 3 konstruiert werden kann, welches innerhalb des Spates liegt. Fall 2 wird dann darauf zurückgeführt.

Es seien $\mathbf{b}_1, \mathbf{b}_2$ die Spatschenkel mit $\angle(\tilde{d}, \mathbf{b}_1), \angle(\tilde{d}, \mathbf{b}_2) \leq \theta$ und zudem gelte $\|\mathbf{b}_1\| \geq \|\mathbf{b}_2\|$. Wir betrachten die $(\mathbf{b}_1, \mathbf{b}_2)$ -Ebene des Spates, wobei das Bild von e_3 durch den Normalenvektor der Ebene $o_3 := \overrightarrow{\mathbf{b}_1 \times \mathbf{b}_2}$ gegeben ist (siehe Abb. 7.7). Wegen der Voraussetzungen $\angle(\mathbf{b}_1, \mathbf{b}_3), \angle(\mathbf{b}_2, \mathbf{b}_3) \geq 90^\circ$ liegt o_3 innerhalb des von den \mathbf{b}_i gebildeten Kegels.

Sei $b = \mathbf{b}_1 + \mathbf{b}_2$ die Diagonale des von $\mathbf{b}_1, \mathbf{b}_2$ aufgespannten Parallelogramms und $\delta := \angle(\mathbf{b}_1, b)$ der zwischen \mathbf{b}_1 und b eingeschlossene Winkel. Aus der Längenannahme $\|\mathbf{b}_1\| \geq \|\mathbf{b}_2\|$ folgt $\angle(\mathbf{b}_2, b) \geq \angle(\mathbf{b}_1, b)$ und wegen

$\angle(\mathbf{b}_1, \mathbf{b}_2) \geq 90^\circ$ gilt damit $\angle(\mathbf{b}_2, b) \geq 45^\circ$. Je nach Größe von $\angle(\mathbf{b}_1, b)$ werden zwei Fälle unterschieden (siehe auch Abb. 7.8):

A. $\angle(\mathbf{b}_1, b) \geq 45^\circ$: o_1, o_2 werden in der $\mathbf{b}_1, \mathbf{b}_2$ -Ebene symmetrisch im Winkel von 45° zu b gewählt.

B. $\angle(\mathbf{b}_1, b) < 45^\circ$: Es wird $o_1 := \vec{\mathbf{b}}_1$ und orthogonal hierzu $o_2 := o_3 \times o_1$ gewählt.

Die Bilder o_1, o_2 liegen nach Konstruktion jeweils innerhalb des von $\mathbf{b}_1, \mathbf{b}_2$ aufgespannten Parallelogramms und damit insbesondere innerhalb des Spates.

Fall 2 wird darauf wie folgt zurückgeführt: Es wird $o_3 := \vec{\mathbf{b}}_3$ festgelegt, womit o_3 trivialerweise innerhalb des \mathbf{b}_i -Kegels liegt. Die Spatvektoren $\mathbf{b}_1, \mathbf{b}_2$ werden auf die Ebene senkrecht zu o_3 projiziert; seien dies p_1, p_2 . Nach Lemma 23 gilt für die Projektionen ebenfalls $\angle(p_1, p_2) \geq 90^\circ$ und sie liegen innerhalb des Spates, da die Projektion in die positive Richtung von \mathbf{b}_3 erfolgt. Damit sind alle Voraussetzungen für die obige Konstruktion erfüllt, wobei p_1, p_2 an Stelle von $\mathbf{b}_1, \mathbf{b}_2$ einzusetzen sind.

zu Fall 2 und 3 [Bedingung 2 von Lemma 27]:

Es bleibt zu zeigen, daß die Drehachse d des jeweiligen Orthonormalsystems innerhalb des Spates liegt. Hierzu wird gezeigt, daß die Projektion von d auf eine Grundfläche innerhalb des Spates liegt und die entsprechende Höhe ausreichend ist. Die Länge der Projektion und die Höhe werden über Steigungsargumente von \tilde{d} gekoppelt.

zu Fall 2A:

d liegt vollständig innerhalb des Spates, da die Projektion von b auf die Ebene orthogonal zu \mathbf{b}_3 nach Lemma 25 mindestens Länge $\sqrt{2}$ hat. Zudem folgt aus $\angle(\tilde{d}, \mathbf{b}_3) \leq \theta = \angle(\mathbf{b}_3, d)$, daß der Spat höher als der Endpunkt von d ist (\tilde{d} steiler und mindestens so lang wie $d \Rightarrow$ höher als d).

zu Fall 2B:

Man betrachte hierzu Abb. 7.8. Nach Lemma 26 ist sichergestellt, daß der Abstand zwischen parallelen Spatseiten größer gleich 1 ist, womit auch der Abstand des in Abb. 7.8 dargestellten Punktes p von \mathbf{b}_1 größer gleich 1 ist. Um diesen Abstand bei einem Winkel $\delta < 45^\circ$ zu erreichen, muß $\lambda > 1$ gelten ($\lambda =$ Länge der senkrechten Projektion von p auf \mathbf{b}_1). Damit liegt die Projektion der Würfelfläche innerhalb der Spatfläche. Für die Höhe des Spates gilt dasselbe Argument wie in Fall 2 A.

zu Fall 3A:

Nach Lemma 26 gilt in Richtung o_3 für die Höhe des Spates $h \geq 1$, so daß sie

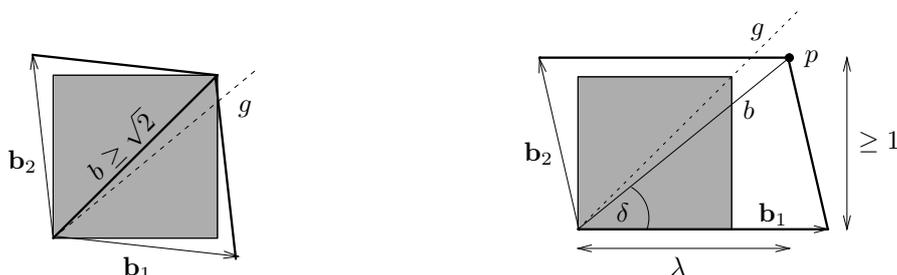


Abbildung 7.8: Schematische Darstellung zu Fall 2A und 2B

wegen der Ausrichtung des Würfels auf der $(\mathbf{b}_1, \mathbf{b}_2)$ -Ebene des Spates ausreichend ist. Für die Diagonale \tilde{d} folgt aus den Voraussetzungen $\angle(\tilde{d}, \mathbf{b}_1) \leq \theta$ und $\angle(\tilde{d}, \mathbf{b}_2) \leq \theta$ mit $\angle(\mathbf{b}_1, \mathbf{b}_2) \geq 90^\circ$, daß der Winkel von \tilde{d} zur Spatfläche $\mathbf{b}_1, \mathbf{b}_2$ kleiner η ist. Damit muß die Länge von \tilde{d} in Richtung b aber größer $\sqrt{2}$ und damit größer als die Länge von d sein (\tilde{d} flacher \Rightarrow länger um Höhe 1 zu erreichen).

In Fall 1b gilt ebenfalls Winkel \tilde{d} zu Spatfläche $\mathbf{b}_1, \mathbf{b}_2$ kleiner η , so daß das gleiche Längenargument greift. Man beachte, daß wegen $\angle(\tilde{d}, \mathbf{b}_i) \geq \theta$ in 1b immer eine Einbettung entlang b erfolgt.

zu Fall 3B:

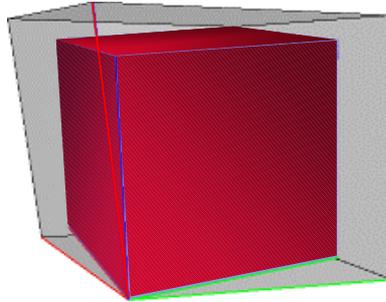
Für die Höhe des Spates gilt $h \geq 1$ wie in 3 A. Die Länge der senkrechten Projektion der Raumdiagonale \tilde{d} auf o_1 bzw. o_2 ist ebenfalls größer 1. Dies folgt mit den gegebenen Voraussetzungen $\angle(\tilde{d}, \mathbf{b}_1), \angle(\tilde{d}, \mathbf{b}_2) \leq \theta$ aus Lemma 30, da bei einem Winkel von unter 45° zum Erreichen von $h \geq 1$ eine Strecke von mehr als einer Längeneinheit erforderlich ist. Somit ist auch in diesem Fall die Würfeläche in der Spatfläche eingebettet.

Zu zeigen bleibt noch, wie der Translationsanteil T aus der Basistransformation nach Lemma 21 folgt. Nach der bisherigen Konstruktion liegen sowohl der durch die infinitesimale Rotation \mathbf{R} definierte Spat, als auch der durch R transformierte Einheitswürfel jeweils im Ursprung des Koordinatensystems. Insbesondere wurde angenommen, daß eine Spatecke mit der Eigenschaft $\angle(\mathbf{b}_i, \mathbf{b}_j) \geq 90^\circ$ für alle paarweisen Winkel im Koordinatenursprung liegt. In Lemma 21 wurde gezeigt, daß dies durch die Vorzeichenumkehr genau eines Spatvektors \mathbf{b}_i erreicht werden kann, was wegen der Symmetrie des Spates einer Verschiebung um genau jenen Vektor \mathbf{b}_i entspricht. Die Translation T kann also entsprechend \mathbf{b}_i gewählt werden. Der Einheitswürfel liegt in diesem Falle weiterhin in einer Ecke des Spates.

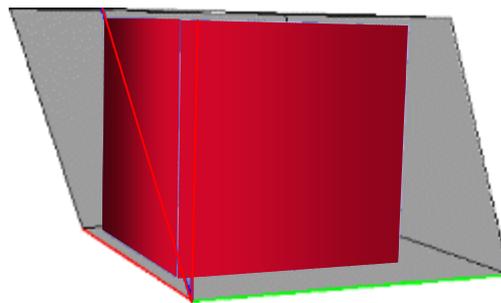
Intuitiv sollte das Bild des Einheitswürfels jedoch zentriert in der Mitte des Spates liegen. Sei $m_e := 0.5(o_1 + o_2 + o_3)$ der Mittelpunkt des Einheitswürfels und $m_s := 0.5(\mathbf{b}_1 + \mathbf{b}_2 + \mathbf{b}_3)$ das Zentrum des untransformierten Spates (d. h. vor Anwendung von Lemma 21). Durch die Translation

$T = (m_s - m_e)$ wird der Mittelpunkt des Einheitswürfels auf das Zentrum des Spates abgebildet. Zu beachten ist, daß der Einheitswürfel weiterhin innerhalb des Spates liegt, da m_s das Symmetriezentrum des Spates ist. \square

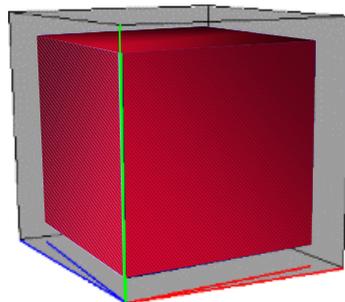
Abbildung 7.9 zeigt Beispiele der drei verschiedenen Einbettungen des Einheitswürfels entsprechend den Fällen 1-3 im Beweis von Theorem 31.



Fall 1: Einbettung entlang der Diagonalen ($\alpha = \beta = \gamma = 0.5$)



Fall 2: Einbettung an einer Kante ($\alpha = \gamma = 0.5; \beta = 0$)



Fall 3: Einbettung in einer Ebene ($\alpha = 1.0; \beta = 0.5; \gamma = 0$)

Abbildung 7.9: Beispiele für die Einbettung des Einheitswürfels unter verschiedenen infinitesimalen Rotationen

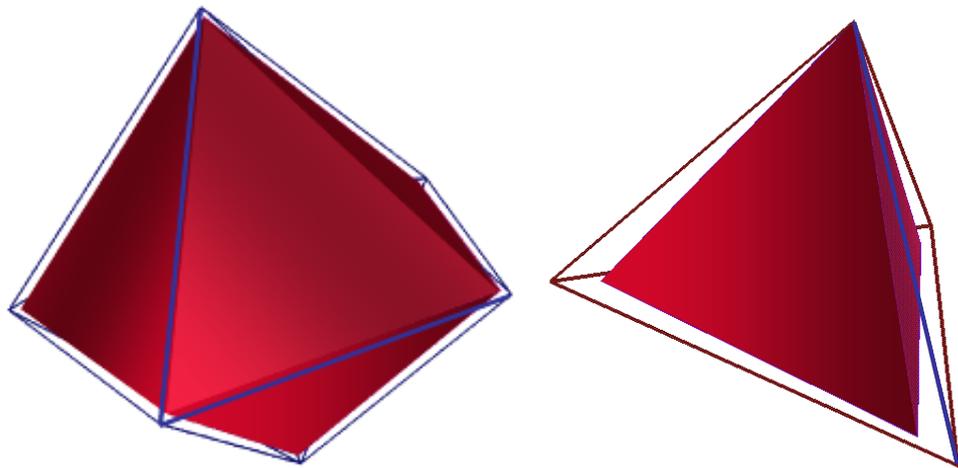


Abbildung 7.10: Beispiele für Hüllkörper und Einbettungen polyedrischer Objekte

Kapitel 8

Anwendung auf industrielle Problemstellungen

Der Entwurf kompakter Anordnungen ist in vielen industriellen Anwendungsgebieten von großer Bedeutung. Die automatisierte Generierung solcher Anordnungen hat insbesondere mit der Entwicklung von VLSI-Schaltkreisen einen entscheidenden Durchbruch erzielt, da die Bauteilkomplexitäten in diesem Bereich nicht mehr manuell beherrschbar sind. In anderen industriellen Anwendungsgebieten, wie etwa der Schnittbildgenerierung für die leder- oder textilverarbeitende Industrie, kommen bislang vollautomatische Systeme kaum zum Einsatz, da hier die von Experten erzielten Ergebnisse oftmals nahezu oder sogar beweisbar optimal sind. Diese Güte muß zunächst von automatisierten Systemen erreicht werden, um die nötige Akzeptanz zu schaffen und die Vorteile einer durchgängigen Automatisierung zum Tragen zu bringen. Für den Bereich der Textilindustrie sind nun aber unabhängig voneinander entwickelte Systeme industriell verfügbar (vgl. [HL96, LM95, DLM92]). Für weitere Anwendungsgebiete siehe auch die Einleitung Abschnitt 1.1, wo neben den methodischen Konzepten auch die jeweils betrachteten Problemstellungen aufgeführt sind.

Im Bereich der dreidimensionalen mechanischen Konstruktion sind bisher noch keine automatisierten Verfahren in Entwurfswerkzeuge, wie z. B. parametrische CAD-Systeme, integriert. Dies liegt zum Teil an der Vielfalt von Randbedingungen, die in diesem Umfeld anzutreffen sind und eine Automatisierung erschweren. Es werden jedoch selbst für generische Teilprobleme, wie z. B. die Kompaktierung vorgegebener Anordnungen, keine Lösungen angeboten. Dies liegt zum Teil darin begründet, daß in industrie-relevanten Systemen bisher ausschließlich mit exakten (Freiform-)Geometrien gearbeitet wurde. Diese lassen selbst einfach erscheinende Operationen wie die interaktive Prüfung auf Überschneidungsfreiheit nur für stark eingeschränkte Modelle zu. In jüngster Zeit sind jedoch hybride Systeme entwickelt und auf den Markt gebracht worden (z. B. DMU-Navigator zu IBM Catia), in denen neben der Freiformgeometrie eine tesselierte Objektdarstellung mitgeführt wird. Diese Systeme liefern nun gerade die benötigten

Eingabedaten für die in Kapitel 2 und 3 vorgestellten globalen und lokalen Optimierungsverfahren bzw. Anordnungs- und Kompaktierungsverfahren.

Ein schnell, einfach und durchgängig zu bedienendes Konzept für die Integration von Randbedingungen stellt für die globalen Optimierungsverfahren zur Zeit noch ein ungelöstes Problem dar. Zudem sind diskrete Drehwinkel nicht in jedem Falle ausreichend. Für die lokalen Optimierungsmethoden entfällt die Beschränkung auf diskrete Drehwinkel, da kontinuierliche Rotationen betrachtet werden können. Auch eine Vielzahl von Randbedingungen wird durch die vorgegebene, z. B. manuell erstellte Ausgangsanordnung bereits berücksichtigt und wird durch die Kompaktierung nicht verletzt. Für die Berücksichtigung von Mindestabständen können zudem einzelne Objektfacetten durch Einfärbungen entsprechend einer Abstandstabelle markiert werden, was sich in der Praxis als unproblematische Benutzerschnittstelle erwiesen hat. Für die lokalen Optimierungsmethoden sind daher die wesentlichen Voraussetzungen für die Integration in industrielle Prozeßketten gegeben. In Abschnitt 8.1 wird nochmals näher auf die Modellierungsgesichtspunkte eingegangen.

In den folgenden drei Abschnitten werden Beispielanwendungen vorgestellt, die in Zusammenarbeit mit unserem Kooperationspartner aus der Automobilindustrie realisiert wurden. In Abschnitt 8.2 wird ein Verfahren zur Anordnung elektronischer Module mit Hilfe eines globalen Optimierungsverfahrens basierend auf den Grundideen von Kapitel 2 betrachtet. Die wesentliche Erweiterung besteht hier in der Modellierung und Einbeziehung von Kabelraumabschätzungen in die Problemformulierung. Abschnitt 8.3 zeigt dann eine Anwendung, in der zunächst die globale Anordnungsstruktur einer Modulgruppe mittels der Verfahren aus Kapitel 2 festgelegt wurde und in einem zweiten Schritt dann erfolgversprechende Varianten mittels des lokalen Optimierungsverfahrens aus Kapitel 3 der nicht konvexen Karosseriestruktur angepaßt wurden. In Abschnitt 8.4 werden dann zwei weitere Einsatzmöglichkeiten der in dieser Arbeit vorgestellten Verfahren aufgezeigt. Zum einen ist eine Einpassung von Bauteilen in bestehende Umgebungen möglich, zum anderen wurde eine erste Phase zur Ausliterung von PKW-Kofferräumen entsprechend einer europäischen und US-amerikanischen Norm betrachtet.

8.1 Modellierung

Um die in dieser Arbeit vorgestellten Methoden für die in industriellen CAD-Systemen vorliegenden Objektbeschreibungen einsetzbar zu machen, sind zwei Verfahrensschritte erforderlich: Zunächst muß für Freiformflächen und -objekte eine polyedrische Approximation erzeugt werden. Dies erfolgt mit Hilfe von Gittergeneratoren, wobei tesselierte, insbesondere mit einem Dreiecksgitter versehene Objektdarstellungen oft schon zur Visualisierung vor-

liegen. In einem weiteren Schritt wird dann eine konvexe Zerlegung oder Überdeckung dieser polyedrischen Objekte erzeugt. Die Tesselierung der Datensätze wurde von unserem Kooperationspartner innerhalb des eingesetzten CAD-Systems vorgenommen, was problemlos möglich war, jedoch von Hand für die einzelnen Objekte durchgeführt werden mußte¹. Es erfolgte zudem eine nachträgliche Gitterbearbeitung, um überschneidende Netzsegmente zu entfernen, Lücken zu schließen und eine weitere Vergrößerung durchzuführen.

Die Berechnung konvexer Zerlegungen führte jedoch teilweise zu Problemen, so daß hier noch nicht von einem vollständig automatisierten Prozeß zu sprechen ist, wobei diese Problemstellung nicht im Fokus der Arbeit stand. Zur Vermeidung einer konvexen Zerlegung ist prinzipiell eine direkte Berechnung des Distanzpolyeders nicht konvexer Objekte möglich, um hieraus Separationsebenen zwischen je zwei Objekten abzuleiten. Hierbei sind jedoch beliebige Schnittbildungen von Flächen der Ausgangsobjekte möglich, so daß die fehlerfreie und robuste Implementierung nicht trivial ist. Eine Alternative hierzu bilden die in Abschnitt 5.3 eingeführten \mathcal{P} -Spuren, bei deren Berechnung keine Schnittbildungen notwendig sind und nur Elemente der Ausgangsobjekte in geeigneter Form zusammengefügt werden.

Ein weiterer Modellierungsaspekt betrifft die Vergrößerung des Konfigurationsraumes im globalen Branch & Bound Optimierungsansatz. Um das Branching, bzw. die Laufzeit des Verfahrens so gering wie möglich zu halten und einen größtmöglichen Nutzen aus den berechneten unteren Schranken zu ziehen, werden Teile des Lösungsraumes hierarchisch zusammengefaßt. Unzulässige Lösungen in diesen Bereichen werden dann in nachfolgenden Schritten weiter bearbeitet. Beim Aufbau dieser Separationshierarchie sind jeweils konvexe Hüllen der entsprechenden Teilbereiche zu bilden. Im Gegensatz zur konvexen Zerlegung existieren hierfür numerisch ausgereifte Verfahren und die durchgeführten Berechnungen mit dem Programm Qhull (siehe Anhang C) ergaben ein robustes und problemloses Verhalten.

8.2 E-Modulboxen mit Kabelraumabschätzung

In diesem Abschnitt wird ein integrierter Ansatz zur Anordnungsoptimierung dreidimensionaler elektronischer Bauteile mit Kabelraumabschätzung vorgestellt. Dieser stellt eine Erweiterung des in Kapitel 2 betrachteten globalen Optimierungsverfahrens dar, wobei für jede Anordnungsvariante eine Abschätzung des benötigten Kabelraumes erzeugt wird. Für ausgewählte Anordnungen wird dann in einer zweiten Phase der detaillierte Verlauf der Verdrahtung innerhalb des reservierten Raumes festgelegt. Die Aufteilung in Raumabschätzung und detaillierte Verdrahtung erfolgt dabei aus Lauf-

¹Hierfür möchte ich mich bei Frau Jutta Müller und Herrn Christian Erbe von der DaimlerChrysler AG herzlich bedanken.

zeitgründen. Die Abschätzung kann in die LP-Formulierung des Kompaktierungsproblems integriert werden und somit effizient innerhalb des Branch & Bound Ansatzes mitberechnet werden.

Für die Kabelraumabschätzung wurde eine explizite Formulierung gewählt. Bei einer expliziten Formulierung wird für jede Kabelverbindung eine Anzahl unterschiedlicher Verdrahtungswege vorberechnet, zwischen denen dann eine Auswahl mittels LP-Relaxierung getroffen wird. Es können dabei sowohl Punkt-zu-Punkt als auch Verbindungen zwischen mehr als zwei Endpunkten, d. h. Kabelbäume, betrachtet werden. Die Raumabschätzung kann dabei sowohl für orthogonale Verdrahtungen (wie meist im VLSI-Design gefordert) oder für Kabelverläufe mit freier Richtungswahl durchgeführt werden. Zielkriterium ist zunächst die Volumenminimierung der Gesamtanordnung, wobei jedoch als sekundäres Kriterium z. B. die Länge der Verdrahtungswege betrachtet werden kann.

Die verwendeten Methoden sind zum Teil im Rahmen der Designautomatisierung von VLSI-Schaltkreisen entwickelt worden. Insbesondere die Grundidee der Berechnung globaler Verdrahtungen schon während der Platzierungsphase von Bauteilen ist dort wegen der hohen Konnektivität der Schaltkreise ein unverzichtbarer Lösungsbestandteil. Einen sehr guten Gesamtüberblick hierzu gibt Lengauer [Len90]. Zur Integration von Verdrahtung in ILP-Formulierungen siehe auch Lengauer, Lügering [LL93].

Dort werden insbesondere die beiden Varianten der *impliziten* und *expliziten Formulierung* dargestellt. Bei einer impliziten Formulierung werden potentiell alle möglichen Verdrahtungsverläufe berücksichtigt, wobei die Restriktionen sicherstellen, daß ein Verdrahtungsweg für alle geforderten Verbindungen existiert. Falls nicht alle Wege zulässig sind, müssen entsprechende Ausschlußbedingungen gefunden und in die ILP-Formulierung integriert werden. Für den in dieser Arbeit gewählten expliziten Ansatz wird demgegenüber eine Menge von Verdrahtungswegen für jede Verbindung vorberechnet. Diese Berechnung erfolgt für die einzelnen Verbindungen unabhängig voneinander und spezielle Einschränkungen für deren Verlauf können hierbei berücksichtigt werden, was eine hohe Flexibilität für die Einarbeitung neuer Restriktionen bedeutet. Für diese Varianten werden 0/1-Entscheidungsvariablen eingeführt, die jeweils genau einen Verdrahtungsweg zu einer gegebenen Verbindungsanforderung festlegen. Die lineare Relaxierung dieser Formulierung liefert dann Werte im Intervall $[0, 1]$, die als Gewichtungsfaktoren für die Auswahl der Varianten interpretiert werden und auf Grundlage derer Verdrahtungsraum reserviert wird. Im Gegensatz zu sequentiellen Verfahren, bei denen die Verdrahtungswege nacheinander bestimmt werden und somit früher festgelegte Wege Hindernisse für spätere Verdrahtungen bilden, werden in der hier vorgestellten Problemformulierung alle Verbindungen gleichzeitig betrachtet.

Die für den Schaltkreisentwurf oftmals verwendeten Min-Cut-Placement Ansätze sind für die in dieser Arbeit betrachteten Problemstellungen mit

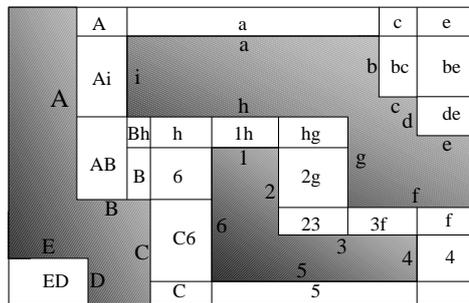


Abbildung 8.1: Definition der Verdrahtungszellen (Markierung mit Namen der Facetten)

einer relativ geringen Anzahl von Objekten und Verbindungen weniger geeignet, da sie eine hohe Verbindungsdichte voraussetzen; siehe z. B. Lengauer, Müller [LM93] für einen solchen hierarchischen Ansatz. Für die detaillierte Verdrahtung ausgewählter Anordnungsvarianten innerhalb des durch die Kabelraumabschätzung reservierten Raumes können verschiedene Verfahren eingesetzt werden. Für nicht orthogonale Verdrahtungen sind zum Teil Ideen aus dem Bereich der Leiterplattenfertigung übertragbar; siehe hierzu Lengauer [Len90] Kapitel 9.9 „Detailed Routing in Gridless Routing Models“. Ein Simulated Annealing Ansatz hierzu ist in Szykman, Cagan [SC96b] zu finden; Verfahren basierend auf Expertensystemen werden in Zhu, Latombe [ZL91] und Mitsuta et al [MKW⁺86] vorgestellt. Für die im folgenden betrachtete Problemstellung wurde ein orthogonales, sequentielles Rip-Up-and-Reroute Verfahren für die detaillierte Verdrahtung implementiert, in welchem die einzelnen Verbindungen nacheinander realisiert werden, vorher festgelegte Verbindungen jedoch stückweise aufgebrochen werden können, falls sie Blockaden für nachfolgende Verdrahtungswege bilden. Dieses Verfahren wird jedoch nicht im einzelnen diskutiert und auf die entsprechende VLSI-Design Literatur verwiesen.

8.2.1 Definition des Verdrahtungsgraph

Die unterschiedlichen Verdrahtungsvarianten für jede zu realisierende Verbindung werden in einem Verdrahtungsgraph generiert, der aus einer vorgegebenen Anordnungsvariante abgeleitet wird. Hierzu wird zunächst der für Verkabelungen zulässige Freiraum in quaderförmige Verdrahtungszellen aufgeteilt. Diese Zellen werden dann durch die Knoten des Verdrahtungsgraph repräsentiert, wobei Zellen mit sich berührenden Seitenflächen durch Kanten verbunden werden. In diesem Verdrahtungsgraph werden dann kürzeste Wege bzw. Steiner-Bäume für die einzelnen Verbindungen berechnet.

Die Definition des Verdrahtungsgraph ist an die im zweidimensionalen VLSI Design benutzten *Track Graphs* (siehe [Len90]) angelehnt und kann als

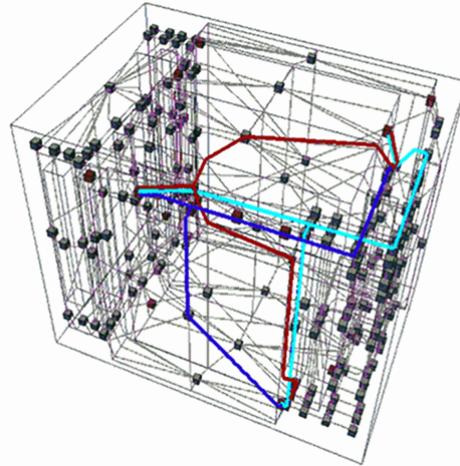


Abbildung 8.2: Beispiel für Verdrahtungsgraph mit Verdrahtungsvarianten

Erweiterung auf den dreidimensionalen Fall angesehen werden. Die Grundidee bei der Konstruktion des Verdrahtungsgraph ist es, die Facetten der bereits angeordneten Objekte nach vorn auszudehnen, bis sie ein anderes Objekt oder den Rand des Anordnungsraumes treffen (siehe Abb. 8.1), wodurch eine Partitionierung des Freiraumes entsteht. Formal wird eine Gitterstruktur aufgebaut, die Gitterebenen für jede Objektfacette enthält. Falls sich zwei Objekte in der Anordnung berühren, so wird zwischen ihnen eine zusätzliche Ebene eingefügt, die einem Freiraum der Breite 0 entspricht, so daß Verdrahtungswege dort möglich sind. Alle Zellen vor einer Facette werden mit einer entsprechenden Markierung versehen, bis ein Teil eines anderen Objektes erreicht wird. Gitterzellen mit den gleichen Markierungen werden dann zu einer Verdrahtungszelle zusammengefaßt, wobei evtl. vorhandene L-förmige Elemente entlang einer beliebig gewählten Trennrichtung aufgebrochen werden. Anschlußpunkte für die Verkabelung liegen an fest vorgegebenen Punkten der Objekte oder am Rand des begrenzenden Bauraumes und werden den entsprechenden Verdrahtungszellen zugeordnet. Zudem kann eine Abgangsrichtung für jeden Anschlußpunkt vorgegeben werden, welche die Richtung des ersten Verdrahtungssegmentes festlegt. Der Verdrahtungsgraph spiegelt somit die Topologie des Verdrahtungsraumes wider; siehe Abb. 8.2 für ein typisches Beispiel. Man beachte hierzu, daß die Knotendichte in Bereichen mit inhomogener geometrischer Struktur, d. h. vielen kleinen Facetten, höher ist als in Bereichen mit homogener Struktur, d. h. wenigen und großen Facetten.

Die Kantengewichte des Verdrahtungsgraph können auf verschiedene Weisen definiert werden. Falls die Verdrahtungslänge minimiert wird, so kann die Euklidische oder Manhattan Distanz zwischen den Knoten verwendet werden. Da jedoch insbesondere der Verdrahtungsraum minimiert

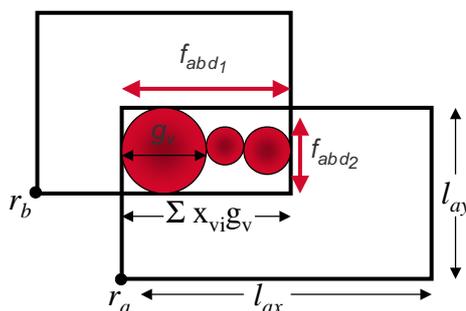


Abbildung 8.3: Variablen für die Verdrahtungsraumabschätzung

werden soll, werden zudem Strafkosten addiert, falls der Kabeldurchmesser größer als die Breite der gemeinsamen Facette zwischen den Verdrahtungszellen ist, da in diesem Fall zusätzlicher Raum bereitgestellt werden muß. Zur Berechnung der Steiner-Bäume wird eine Heuristik basierend auf kürzesten Wegen und minimal spannenden Bäumen verwendet. Um unterschiedliche Verdrahtungswege für die Varianten zu generieren, werden Strafkosten auf die jeweils schon benutzten Kanten des Verdrahtungsgraph addiert. Abb. 8.2 zeigt drei unterschiedliche Varianten für eine Verbindung, die mit diesem Verfahren erzeugt wurden.

8.2.2 Definition des Linearen Programms

Wir betrachten nun die Formulierung der Verdrahtungsraumabschätzung als ILP und die zugehörige lineare Relaxierung. Für jede Verbindung v wird, wie in Abschnitt 8.2.1 dargestellt, eine Menge von Verdrahtungsvarianten berechnet. Für diese Varianten T_{vi}^2 ($1 \leq i \leq m_v$) werden Entscheidungsvariablen $x_{vi} \in \{0, 1\}$ eingeführt, die den Wert 1 annehmen, wenn Variante i für Verbindung v ausgewählt wird. Um genau eine Verdrahtungsvariante für jede Verbindung zu selektieren, wird $\sum_{i=1}^{m_v} x_{vi} = 1$ gefordert (Gleichung (8.8) der ILP-Formulierung auf Seite 125).

Zur Einführung der Bezeichnungen betrachten wir Abb. 8.3. Die Position jeder Verdrahtungszelle Z_a im Raum wird wie bei dem globalen Optimierungsverfahren in Kapitel 2 durch eine Vektorvariable r_a für den Referenzpunkt festgelegt (Ecke vorne links unten). Für die Ausdehnung der Verdrahtungszellen führen wir Vektorvariablen $l_a \in \mathbb{R}^3$ ein, deren Komponenten $(l_a)_d$ ($d \in \{x, y, z\}$) die Länge, Breite und Tiefe der Zelle sind. Die Werte von $(l_a)_d$ werden durch das Lineare Programm entsprechend der gewählten Verdrahtungsvarianten angepaßt, so daß alle über die Facetten verlaufenden Kabelquerschnitte aufgenommen werden können. Die vorgegebenen Objekte stellen hierbei spezielle Zellen dar, deren Ausdehnung l_a unveränderlich ist.

²T für *tree*

Wir betrachten nun zwei in der Ausgangsanordnung benachbarte Verdrahtungszellen Z_a und Z_b und deren gemeinsame Facette. Durch die Translation und Größenanpassung der Verdrahtungszellen werden diese gegeneinander verschoben, wobei jedoch eine hinreichend große Fläche für die Verkabelung erhalten bleiben muß. Für die Mindestlänge der Überlappung in Richtung d_1 bzw. d_2 führen wir Variablen f_{ab} ein. Restriktionen der Form $(r_a)_d + (f_{ab})_d \leq (r_b)_d + (l_b)_d$ ($d \in \{d_1, d_2\}$) stellen eine entsprechende Überlappung sicher. Für die verbleibende dritte Dimension $d_3 \neq d_{1/2}$ wird die räumliche Trennung der Verdrahtungszellen durch Ungleichungen $(r_a)_{d_3} + (l_a)_{d_3} \leq (r_b)_{d_3}$ sichergestellt (siehe Ungleichungen (8.3),(8.4) des ILP).

Wir betrachten nun die Abschätzung der zur Verkabelung benötigten Überlappungsfläche. Sei hierzu q_v der Querschnitt bzw. Durchmesser von Verbindung v . Wir fordern, daß die kürzere Seite $\min_{ab} \leq (f_{ab})_{d_{1/2}}$ mindestens die Größe des maximalen Querschnitts der über diese Facette verlaufenden Verbindungen hat. Da mehrere Verdrahtungsvarianten einer Verbindung über dieselbe Facette verlaufen können, ergibt sich mit der Gewichtung x_{vi} der Varianten im Linearen Programm die Restriktion $\min_{ab} \geq \sum_i x_{vi} q_v$ (Ungleichungen (8.5),(8.6)).

Falls eine der Seitenlängen f_{ab} gerade dem maximalen Kabeldurchmesser entspricht, so fordern wir, daß die andere, längere Seite mindestens die Länge der Summe der Kabeldurchmesser annimmt. Anders formuliert bedeutet dies, daß die Summe $(f_{ab})_{d_1} + (f_{ab})_{d_2}$ größer als $\min_{ab} + \sum_{v,i} x_{vi} q_v$ sein muß (siehe Ungleichung (8.7)). Falls die Variablen f_{ab} nicht die Extremalwerte \min_{ab} und $\sum_{v,i} x_{vi} q_v$ annehmen, so wird die zur Verfügung stehende Überlappungsfläche $(f_{ab})_{d_1} * (f_{ab})_{d_2}$ dadurch nochmals vergrößert, d. h. es handelt sich bei dieser Linearisierung um eine konservative Abschätzung.

Es folgt nun eine Kurzübersicht der verwendeten Bezeichnungen und die ILP-Formulierung der Verdrahtungsraumabschätzung:

Objekte und Konstanten:

$\mathcal{Z} = \{Z_1, \dots, Z_k\}$	Menge der Verdrahtungszellen
k	Anzahl Verdrahtungszellen inkl. Überdeckung der Objekte
q_v	Querschnitt/Durchmesser von Verbindung v

Indizes:

$1 \leq a \neq b \leq k$	Indizes für benachbarte Verdrahtungszellen
$d \in \{x, y, z\}$	Index für Koordinatenrichtung
v	Index für Verbindungen mit Varianten T_{vi}
i	Index für Verdrahtungsvarianten

Variablen:

l	Länge der Gesamtanordnung
r_a	Referenzpunkt von Z_a
$(l_a)_d$	Länge, Höhe, Tiefe von Verdrahtungszelle Z_a

$(f_{ab})_d$	Überlappungslänge Z_a, Z_b in Richtung $d_{1/2}$
min_{ab}	Minimallänge für die Überlappung Z_a, Z_b
$x_{vi} \in \{0, 1\}$	Entscheidungsvariable für Variante T_{vi}

ILP-Verdrahtungsraumabschätzung:

$$\min f(l, r_i) \quad (8.1)$$

mit Nebenbedingungen:

$$r_a + l_a \leq l \quad (8.2)$$

$$(r_a)_d + (f_{ab})_d \leq (r_b)_d + (l_b)_d \quad d = d_{1/2} \quad (8.3)$$

$$(r_a)_d + (l_a)_d \leq (r_b)_d \quad d \neq d_{1/2} \quad (8.4)$$

$$min_{ab} \leq (f_{ab})_{d_{1/2}} \quad (8.5)$$

$$\sum_i x_{vi} q_v \leq min_{ab} \quad (8.6)$$

$$min_{ab} + \sum_{v,i} x_{vi} q_v \leq (f_{ab})_{d_1} + (f_{ab})_{d_2} \quad (8.7)$$

$$\sum_i x_{vi} = 1 \quad (8.8)$$

$$r_i \geq 0, x_{vi} \in \{0, 1\} \quad (8.9)$$

(8.3)-(8.7): $\forall Z_a, Z_b$ adjazente Verdrahtungszellen mit gemeinsamer Facette in Koordinatenrichtung d_1, d_2

(8.6)-(8.7): \forall Verbindungsvarianten T_{vi} , die Facette Z_a, Z_b schneiden

8.2.3 Ergebnisse für E-Modul-Box SG129

In diesem Abschnitt werden die Ergebnisse für das von unserem Kooperationspartner zur Verfügung gestellte Testbeispiel SG129 zusammengefaßt. Es besteht aus fünf elektronischen Modulen (ABS, Motorelektronik, adaptives Dämpfungssystem, etc.), die jeweils aus einem Rumpfbauteil und Befestigungsglaschen bestehen und durch mehrere Quader überdeckt wurden. Zudem sind acht Kabelstränge mit unterschiedlichem Durchmesser vorhanden, die nach außen geführt werden. Die Module müssen in eine wasserdichte Box eingepaßt werden, deren maximale Ausmaße $31 \times 22 \times 19 \text{ cm}$ vorgegeben sind. Zudem sind die Positionen für die Kabelverbindungen nach außen durch eine Referenzanordnung vorgegeben, wobei die Verbindungen jeweils Punkt-zu-Punkt sind. Die Rotation der Module ist nicht eingeschränkt, wobei jedoch die meisten Kabel auf der Oberseite der Module angeschlossen sind. Diese Orientierung wurde auch von den Algorithmen aus Platzgründen bevorzugt, da die Kabeldurchführungen nach außen ebenfalls am oberen Rand

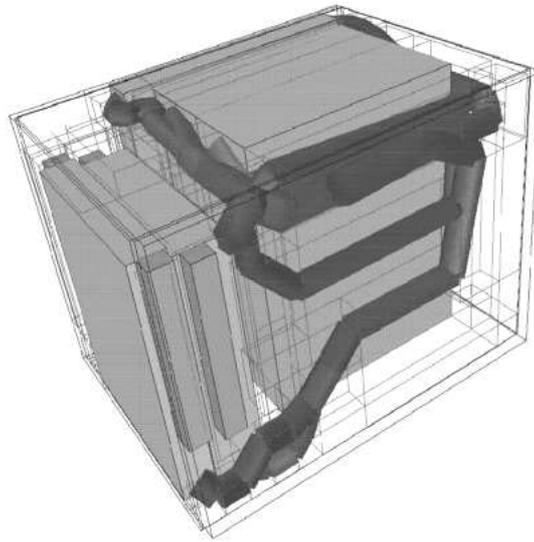


Abbildung 8.4: Referenzanordnung des Industriepartners für SG129

des Bauraumes liegen. Für die Berechnungen wurde eine einfache, zweistufige Objekthierarchie benutzt. Die erste Stufe bestand jeweils nur aus dem Bauteilrumpf ohne Befestigungslaschen, d. h. jeweils genau ein Quader je Modul. In der zweiten Stufe wurden dann 2 bis 4 zusätzliche Quader für die Befestigungslaschen hinzugenommen.

In den Abbildungen 8.4 und 8.5 sind die industrielle Referenzanordnung und die volumenminimale Anordnung mit Kabelraumabschätzung für die E-Modul-Box SG129 dargestellt. Das Volumen der ursprünglichen Anordnung beträgt 12.958cm^3 (angegeben ist immer das Volumen eines minimal umschließenden Quaders). Die berechnete Anordnung ist ca. 10cm kürzer und weist inkl. Verdrahtungsraumabschätzung ein Volumen von 7.874cm^3 auf. Das Volumen ohne Kabelraumabschätzung beträgt für dieses Beispiel 7.417cm^3 . Die berechnete Anordnung konnte an Hand eines Prototyps physikalisch realisiert werden. Der wesentliche Unterschied liegt in der Anordnung eines Moduls direkt unterhalb der Kabeldurchbrüche in der Außenwand, sowie der Drehung der beiden vorderen Module derart, daß die Anschlußpunkte der Verkabelung an der Oberseite liegen. Neben der gezeigten Optimallösung wurden zudem eine Reihe weiterer, ebenfalls sehr kompakter Anordnungen im Verlauf des Branch & Bound Verfahrens berechnet. Diese können als Alternativvorschläge ebenfalls angeboten und von Konstrukteuren abschließend beurteilt werden. Dies stellt in der Praxis einen wesentlichen Vorteil gegenüber Verfahren dar, die nur eine Lösung liefern.

In Tabelle 8.1 sind Laufzeitindikatoren der Berechnungen zusammengefaßt. Betrachtet wird das vorgestellte Beispiel SG129, wobei zum einen die Problemstellung ohne Kabelraumabschätzungen und zum anderen oh-

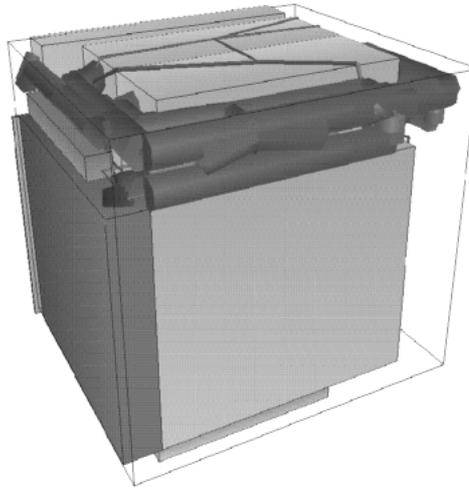


Abbildung 8.5: Volumenminimale Anordnung mit Kabelraumabschätzung

	Knoten im Suchbaum	Knoten „pruned“	Knoten max.aktiv	Laufzeit bis Opt.	Gesamtlaufzeit
ohne Kabel	209.050	152.086	7.852	11s	88s
ohne Kabel (Bauraum frei)	778.610	574.778	22.458	31s	358s
mit Kabel	2.256.116	1.758.225	50.907	775s	11.136s

Tabelle 8.1: Laufzeitindikatoren für E-Modulbox SG129

ne die Begrenzung des Bauraumes („Bauraum frei“) betrachtet wurde. Die maximale Anzahl gleichzeitig aktiver Knoten im Branching Tree ist das wesentliche Maß für den Speicherplatzbedarf. Dieser kann über verschiedene Parameter zur Tiefen- bzw. Breitensuche beeinflusst werden. Die Gesamtzahl der Knoten ist ein maschinenunabhängiges Maß für die Laufzeitentwicklung des Verfahrens, wobei diese entscheidend von der Effektivität der berechneten unteren Schranken abhängt. Die dritte Spalte der Tabelle zeigt daher die Anzahl der Knoten, die nach Berechnung der unteren Schranke nicht weiter verfolgt wurden („Pruned Nodes“).

Tabelle 8.2 zeigt gemessene Laufzeiten auf einem SUN Sparc Server 1000 bei Verwendung von CPLEX 4.0 zur Lösung der Linearen Programme. Neben der Gesamtlaufzeit ist insbesondere die Laufzeit bis zum Erreichen der Optimallösung von Interesse. Es kann zwar nicht a priori bestimmt werden, wann das Optimum erreicht ist, die in der Tabelle aufgeführten Laufzeiten zeigen jedoch, daß schon nach einem geringen Laufzeitanteil mit guten bzw. sogar schon optimalen Lösungen gerechnet werden kann und die restliche Laufzeit zum Beweis der Optimalität verwendet wird.

Anzahl Quader	Knoten im Suchbaum	Knoten „pruned“	Knoten max. aktiv	Laufzeit
5	21,710	18,085	813	2s
6	77,332	64,435	3,714	10s
7	135,247	112,695	5,765	19s
8	419,588	349,645	13,451	66s
9	1,473,879	1,228,225	34,323	260s
10	7,610,955	6,342,450	119,598	1443s

Tabelle 8.2: Laufzeitindikatoren für Branch&Bound-Verfahren

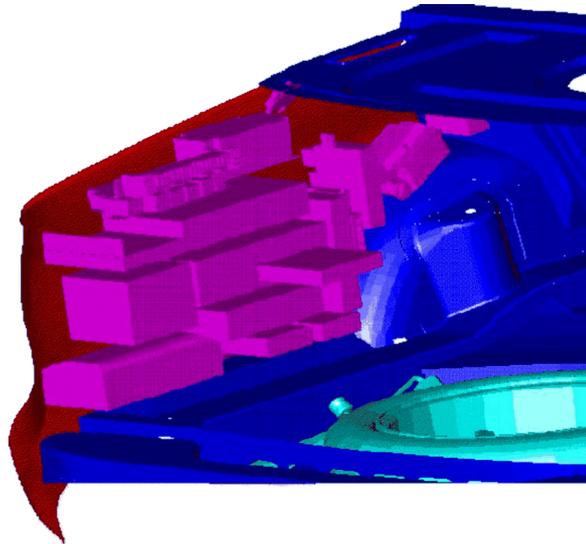


Abbildung 8.6: Referenzanordnung des Industriepartners

8.3 Optimierung von Bauteilanordnungen im Kofferraum eines PKW

Für diese Problemstellung sind 19 verschiedene Module gegeben, die an der hinteren, linken Kofferraumwand eines PKW platziert werden müssen. Unter anderem sind ein CD-Player und ein Navigationssystem mit CD-Schublade enthalten, so daß für diese Geräte die Zugreifbarkeit von außen gewährleistet sein muß. Die Referenzanordnung unseres Kooperationspartners und der Anordnungsraum sind in Abb. 8.6 dargestellt.

Für die globale Anordnungsoptimierung erfolgte die Modellierung des Anordnungsraumes zunächst durch die in Abb. 8.7 gezeigte Approximation mit einem vorplazierten Element zur Modellierung des Radkastens. Für die globale Optimierungsphase wurden 5 der Objekte durch Überdeckungen von Quadern modelliert, für die restlichen 14 Objekte wurde die konvexe Hülle betrachtet. Die Zugreifbarkeit der CD-Schubladen wurde durch

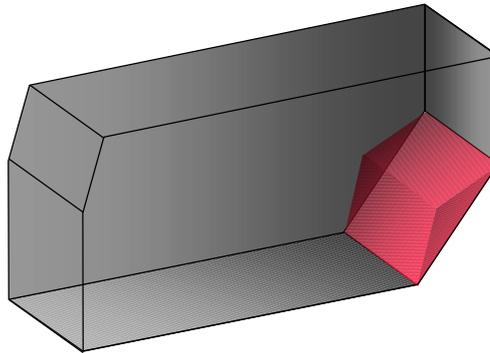


Abbildung 8.7: Modellierung des Anordnungsraumes

Laufzeit glob. Optimierung	Ø Anzahl Iterationen	Ø Anzahl Überschneidungen	Ø Laufzeit lok. Optimierung
7.349s	8,3	1.783	2.385s

Tabelle 8.3: Zusammenfassung der wesentlichen Kenngrößen zur globalen und lokalen Optimierung

Plazierungsrestriktionen an der Vorderseite bzw. Oberseite der Anordnung berücksichtigt, wobei für diese Module keine Drehungen zugelassen wurden.

Das Gesamtverfahren bestand aus einer ersten Phase, in der verschiedene globale Anordnungen der Module in dem approximierten Anordnungsraum bestimmt wurden. Hier wurde das globale Optimierungsverfahren für polyedrische Objekte aus Kapitel 2 eingesetzt, wovon 7 Anordnungen zur weiteren Bearbeitung betrachtet wurden. Für den globalen Optimierungsschritt wurde eine Rechenzeit von ca. zwei Stunden benötigt (siehe Tabelle 8.3).

In einer zweiten Phase wurde dann eine Variante des lokalen Optimierungsverfahrens eingesetzt, um Überschneidungen mit der Außenhaut des Kofferraums zu beseitigen und eine Kompaktierung der Elemente durchzuführen. Um die Überschneidungen mit der Außenhaut zu bestimmen, wurde ein in [Hel96] beschriebenes Verfahren eingesetzt. Anschließend wurden jeweils Separationsbedingungen zu Dreieckselementen der Außenhaut eingefügt, die eine Überschneidung mit einem Objekt aufwiesen. In Tabelle 8.3 ist die durchschnittliche Anzahl von Überschneidungen mit der Außenhaut im ersten Iterationsschritt angegeben. Die durchschnittliche Iterationszahl des lokalen Optimierungsverfahrens zur Kompaktierung und Überschneidungsbeseitigung lag bei 8,3 Iterationen.

In Abb. 8.8 ist die beste gefundene Anordnung dargestellt, wobei die ursprüngliche Anordnung als roter Umriß dargestellt ist. Das rechts außerhalb der Hauptanordnung liegende Bauteil entspricht einer Steckdose und mußte aufgrund einer Vorgabe unseres Kooperationspartners an dieser Stelle ver-

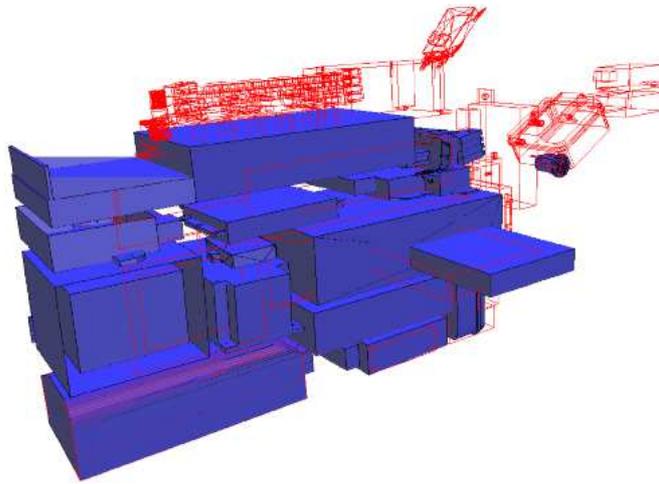


Abbildung 8.8: Alternative Anordnung mit geringerem Volumen

bleiben. Die Höhe der Anordnung wurde um mehr als 10cm reduziert, die Länge um ca. 60cm, wobei jeweils Mindestabstände zwischen den Bauelementen freigehalten wurden.

8.4 Weitere Anwendungen

Die in dieser Arbeit vorgestellten Methoden wurden auf zwei weitere Problemstellungen angewandt.

Zum einen wurde modellhaft aufgezeigt, wie ein Bauteil in eine bestehende Aggregatanordnung integriert werden kann. Hierzu wurde ein Szenario aus sechs Polyedern zusammengestellt, in das ein neues Objekt eingefügt werden soll. Zudem wurde die Kofferraumbegrenzung der Problemstellung aus Abschnitt 8.3 verwendet, um eine komplexer geformte seitliche Begrenzung zu erhalten. Abb. 8.9 zeigt sechs von insgesamt 11 Schritten eines hierfür angepaßten lokalen Optimierungsverfahrens. Hierbei ist zu beachten, daß die Überschneidungsfreiheit jeweils nur für die eingenommenen Positionen des Bauteils garantiert ist. Bei einer zu groß gewählten Schrittweite könnte daher ein Engpaß übersprungen werden, der eigentlich nicht überschneidungsfrei passiert werden kann. Die Laufzeit für einen lokalen Optimierungsschritt des dargestellten Beispiels liegt bei etwa einer Minute, so daß sich für die 11 Optimierungsschritte eine Gesamtlaufzeit von ca. 15 Minuten ergab.

Eine weitere konkrete Aufgabenstellung ergibt sich aus zwei unterschiedlichen Normen zur Volumenbestimmung von PKW-Kofferräumen (dies wird auch als Ausliterung bezeichnet). Hierbei unterscheiden sich die Normen für den europäischen und US-amerikanischen Markt und erfordern unterschiedliche Optimierungsansätze.

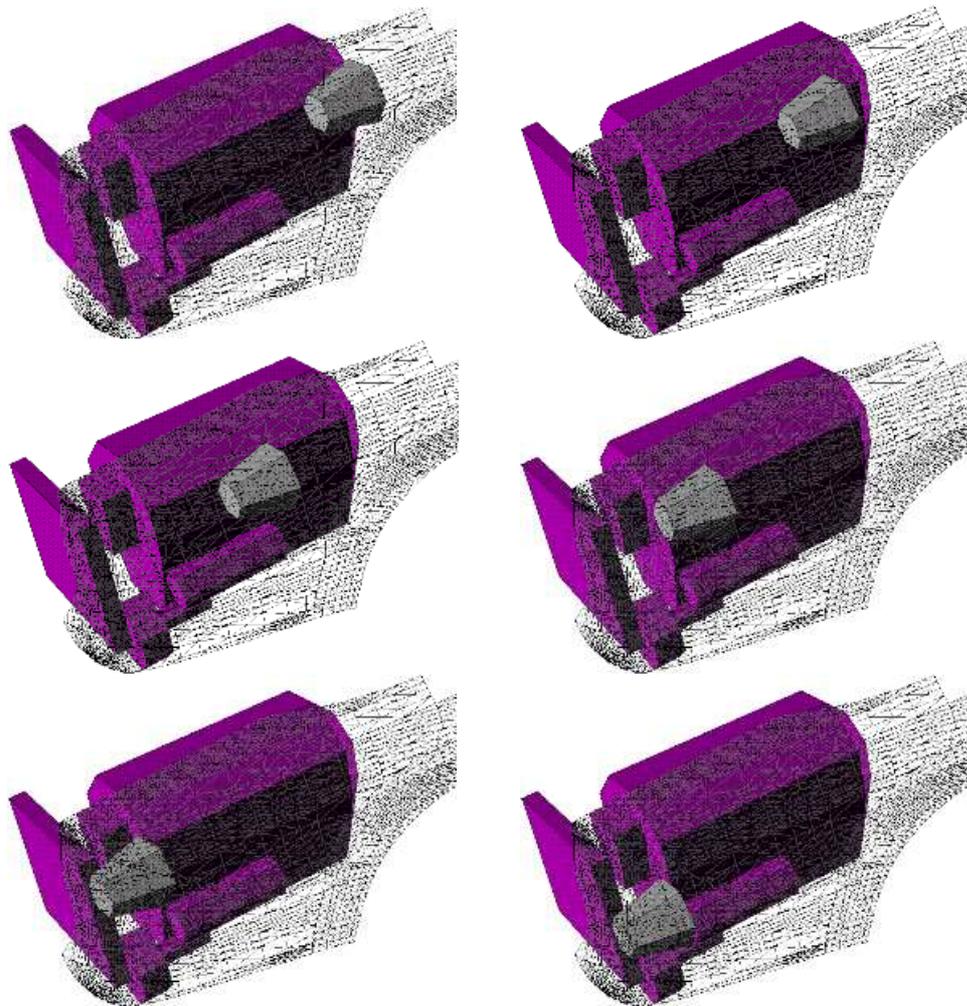


Abbildung 8.9: Einpassen eines Bauteils in eine bestehende Aggregatanordnung

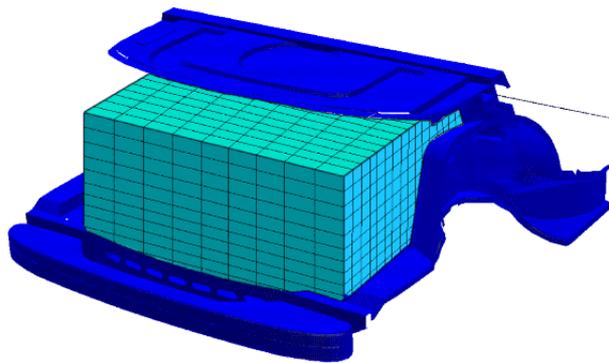


Abbildung 8.10: Kofferraumauslastung mit optimal gefülltem Quader

Für den europäischen Markt wird das Kofferraumvolumen mittels DIN-Normquadern von 5x10x20cm Kantenlänge bestimmt. Ziel ist es, eine maximale Anzahl solcher Quader in die vorgegebene Kofferraumbegrenzung zu plazieren. Neben den in dieser Arbeit diskutierten Methoden müssen hier auch Verfahren für die regelmäßige Anordnung von vielen gleichartigen Objekten eingesetzt werden (siehe die Literaturhinweise in der Einleitung). Prototypisch implementiert wurde ein Verfahren, mit dem zunächst heuristisch ein Quader maximaler Größe innerhalb der Kofferraumgeometrie bestimmt werden kann. Das erzeugte Quader ist dabei abhängig von einem frei gewählten Ausgangspunkt. Mit einem auf dynamischer Programmierung beruhenden Verfahren wird dieses Quader dann optimal mit Normquadern ausgefüllt, wobei die Anordnungen durch Guillotine Schnitte zerlegbar sind. Abb. 8.10 zeigt eine solche Anordnung.

Für den amerikanischen Markt wird eine SAE-Norm verwendet. In dieser ist ein Standardkofferset definiert, welches zunächst im Kofferraum verstaut werden muß. Können alle Koffer untergebracht werden, so dürfen im folgenden beliebig viele sogenannte H-Boxes (in etwa von der Größe eines Schuhkartons) zum Auffüllen verwendet werden. Zur Grobplazierung des Koffersets können die in dieser Arbeit vorgestellten Verfahren zur globalen Optimierung verwendet werden, wobei die Feinplazierung dann mit Hilfe der lokalen Optimierungsmethoden erfolgen kann.

Kapitel 9

Zusammenfassung und Ausblick

Hauptergebnis der vorliegenden Arbeit ist ein lokales Optimierungsverfahren zur Kompaktierung von Anordnungen konvexer, polyedrischer Objekte, sowie die Verbindung dieses Verfahrens mit einem globalen Optimierungsansatz. Die für dieses Resultat verwendete mathematische Methodik enthält sowohl Elemente der diskreten, wie auch der kontinuierlichen, nichtlinearen Optimierung, sowie eine geometrische Komponente.

Die zunächst nichtlineare Problemformulierung für das globale Anordnungsproblem konvexer, polyedrischer Objekte wurde in eine linearisierte Formulierung für das lokale Kompaktierungsproblem überführt. Hierzu wurde in Kapitel 7 eine linearisierte Beschreibung räumlicher Drehungen hergeleitet und eine Hüllkörpereigenschaft für diese bewiesen, was das Hauptresultat aus theoretischer Sicht darstellt (Satz 31, Seite 31). Das Optimierungsmodell kann mit ausgereiften und effizienten Methoden der Linearen Programmierung gelöst werden. Für das Verfahren wurde in Abschnitt 3.3 ein Konvergenzbeweis geführt.

Für die effiziente Implementierung des Verfahrens sind dabei die folgenden Punkte von Bedeutung:

- Die Verwendung von linearer Programmierung als Optimierungskern; hierfür stehen ausgereifte und numerisch robuste Implementierungen zur Verfügung.
- Bei Verwendung des Simplex-Verfahrens zur Lösung der Linearen Programme steht jeweils eine zulässige Startbasis aus dem vorangehenden Optimierungsschritt zur Verfügung.
- Die Anpassung der Separationsebenen durch den GJK-Algorithmus ist ebenfalls sehr effizient möglich, da die Approximation des Distanzpolyeders nur lokal angepaßt werden muß.
- Die Schrittweitensteuerung kann entfallen, da in der Praxis nur geringe Überschneidungen auftreten, die im Verlauf des Verfahrens automatisch korrigiert werden.

In Kapitel 4 wurde die Integration des globalen und lokalen Optimierungsansatzes betrachtet. Die Basis bildet das algebraische Konzept der „Convolution Theory of Closed Convex Sets“ [Che99], welches aus geometrischer Sicht und für dreidimensionale Objekte in [BGR96] eingeführt wird. Aus dieser Charakterisierung überschneidungsfreier Anordnungen zu zwei polyedrischen Objekten ergibt sich eine endliche Menge topologischer Strukturen für \mathcal{P} -Spuren und Distanzpolyeder (siehe hierzu auch [Duv01]). Der kontinuierliche Konfigurationsraum kann somit in eine endliche Anzahl von Äquivalenzklassen unterteilt werden, welche die Basis für ein Branch & Bound Verfahren darstellen. Innerhalb dieser Äquivalenzklassen greifen dann die Methoden der lokalen Optimierung. Die Formulierung stellt in diesem Fall ein ganzzahliges Lineares Programm dar. Wegen der Linearisierung und lokalen Optimierung innerhalb der Äquivalenzklassen ergibt sich jedoch nicht notwendig ein globales Optimum; hierzu wäre eine weitere Untersuchung und Charakterisierung möglicher lokaler Optima bzgl. des gewählten Optimierungsverfahrens innerhalb der einzelnen Äquivalenzklassen notwendig.

Neben diesen Ergebnissen aus mathematischer Sicht werden im folgenden die anwendungsorientierten Ergebnisse und Modellierungsgesichtspunkte zusammengefaßt. Im Rahmen des Forschungsvorhabens wurden konkrete Problemstellungen aus dem Bereich des Fahrzeugbaus untersucht und es wurde gezeigt, daß mit den vorgestellten Methoden eine teilweise Automatisierung und Optimierung von Bauteilanordnungen erreicht werden kann. Eine Darstellung der Testdatensätze und der generierten Beispielanordnungen ist in Kapitel 8 enthalten. Die Anordnungen wurden von Konstrukteuren als ernsthafte Alternativen mit Praxisrelevanz gewertet. Der globale Optimierungsansatz ist dabei in Konkurrenz mit wissensbasierten Systemen zu sehen. Hier wird die Entscheidung zugunsten desjenigen Systems ausfallen, in welches die problemspezifischen Randbedingungen besser integriert werden können. Man vergleiche hierzu auch die Anmerkungen zur formalen Spezifikation funktionaler Zusammenhänge in Abschnitt 1.1.2.

Schon eine teilweise Integration der hier vorgestellten Methoden in CAD-Systeme, wie zum Beispiel eine automatische Kompaktierungsfunktion, würde die Arbeit der Konstrukteure jedoch erheblich erleichtern. Hierbei ist zu beachten, daß selbst moderne CAD-Systeme im Normalfall keine interaktive Kollisionserkennung durchführen und Funktionalitäten wie „Verschiebe Bauteil B maximal in Richtung r bis es auf ein anderes Bauteil stößt“ nicht standardmäßig realisieren. Dies liegt zum Teil an der sehr genauen Objektdarstellung mittels Freiformflächen, die einen effizienten Schnittest nicht zulassen. Am Markt etablieren sich nun erste hybride, d. h. zusätzlich auf tesselierten, polyedrischen Daten arbeitende Systeme. In derartige Systeme sind die hier vorgestellten Methoden ohne aufwendige Datenkonvertierungen

integrierbar. Der Verarbeitung tessellierter Daten kommt zudem durch den im Internet verbreiteten VRML-Standard (VRML: Virtual Reality Modeling Language) eine verstärkte Aufmerksamkeit zu. Hier werden orientierte, planare Flächenstücke definiert, die direkt als Eingabedaten für die in dieser Arbeit vorgestellten Methoden benutzt werden können.

Für den lokalen Optimierungsansatz ergibt sich zudem ein zusätzlicher Anwendungsbereich bei der Planung der Wege von Objekten durch Hindernisse. Bisher wird dieses Problem hauptsächlich für starr orientierte Objekte und Hindernisse betrachtet (Configuration-Space Ansatz in der Robotik). Mit Hilfe des lokalen Optimierungsansatzes kann nun eine kontinuierliche Drehung bestimmt werden, um eine Engstelle zu passieren. Dies wurde für eine einfache Einbausimulation bei unserem Kooperationspartner vorgestellt und stieß auf großes Interesse.

Für den lokalen Optimierungsansatz kommen zudem zahlreiche Problemstellungen in Frage, bei denen geometrische Paßform eine Rolle spielt. Insgesamt ergibt sich somit ein breit gefächertes Anwendungsspektrum für die in dieser Arbeit entwickelten Methoden, wobei die Fortschritte im Bereich der Linearen Programmierung die Behandlung auch sehr großer, d. h. praxisrelevanter Problemstellungen erlauben.

Anhang A

Definitionen und Bezeichnungen

Es sei $\lambda \in \mathbb{R}$, $a, b, c \in \mathbb{R}^n$ und $A, B \subset \mathbb{R}^n$.

Definition 22 (Skalarprodukt)

$$\langle a, b \rangle := a_1 b_1 + a_2 b_2 + \dots + a_n b_n$$

Eigenschaften des Skalarproduktes:

$$\langle a, b \rangle = \langle b, a \rangle \tag{A.1}$$

$$\langle a + b, c \rangle = \langle a, c \rangle + \langle b, c \rangle \tag{A.2}$$

$$\langle \lambda a, b \rangle = \lambda \langle a, b \rangle \tag{A.3}$$

$$\langle a, a \rangle \geq 0 \tag{A.4}$$

$$\text{und } \langle a, a \rangle = 0 \Leftrightarrow a = 0$$

$$\langle a, b \rangle = \|a\| \|b\| \cos \angle(a, b) \tag{Winkel} \tag{A.5}$$

$$\text{und } \langle a, b \rangle = 0 \Leftrightarrow a \perp b \tag{Orthogonalitat}$$

$$\|a\| := \sqrt{\langle a, a \rangle} \tag{Norm/Lange} \tag{A.6}$$

$$\|a + b\| \leq \|a\| + \|b\| \tag{Dreiecksungleichung} \tag{A.7}$$

$$\|a + b\|^2 = a^2 + b^2 + 2\langle a, b \rangle \tag{Satz von Pythagoras} \tag{A.8}$$

$$|\langle a, b \rangle| \leq \|a\| \|b\| \tag{Cauchy-Schwarz. Ungl.} \tag{A.9}$$

$$\text{und } |\langle a, b \rangle| = \|a\| \|b\| \Leftrightarrow a = \lambda b$$

Fur die folgende Definition des Vektorprodukts seien $a, b, c \in \mathbb{R}^3$.

Definition 23 (Vektorprodukt)

$$a \times b := (a_2 b_3 - a_3 b_2, a_3 b_1 - a_1 b_3, a_1 b_2 - a_2 b_1)$$

Eigenschaften des Vektorproduktes:

$$a \times b = -(b \times a) \quad (\text{rechts-/linkshänd. Orien.}) \quad (\text{A.10})$$

$$(a + b) \times c = a \times c + b \times c \quad (\text{A.11})$$

$$\lambda a \times b = \lambda(a \times b) \quad (\text{A.12})$$

$$a \times a = 0 \quad (\text{A.13})$$

$$\text{und } a \times b = 0 \Leftrightarrow a = \lambda b$$

$$\langle a \times b, a \rangle = \langle a \times b, b \rangle = 0 \quad (\text{A.14})$$

$$\|a \times b\| = \|a\| \|b\| \sin \angle(a, b) \quad (\text{Parallelogrammfläche}) \quad (\text{A.15})$$

$$\|a \times b\|^2 = \|a\|^2 \|b\|^2 - \langle a, b \rangle^2 \quad (\text{A.16})$$

$$|\langle a, b \times c \rangle| = \|a\| \|b \times c\| \cos \angle(a, b \times c) \quad (\text{Spatprodukt/-volumen}) \quad (\text{A.17})$$

Definition 24 (Normierter Vektor)

$$\vec{a} := \frac{1}{\|a\|} a$$

Definition 25 (Parallele/senkrechte Komponenten)

Sei $e \in \mathbb{R}^3$ mit $\|e\| = 1$.

$$a^{\parallel e} = \langle a, e \rangle e \quad (\text{Anteil von } a \text{ parall. zu } e)$$

$$a^{\perp e} = a - a^{\parallel e} = a - \langle a, e \rangle e \quad (\text{Anteil von } a \text{ orthog. zu } e)$$

Falls b kein Einheitsvektor:

$$a^{\parallel b} = \left\langle a, \frac{b}{\|b\|} \right\rangle \frac{b}{\|b\|} = \frac{1}{\|b\|^2} \langle a, b \rangle b = \frac{\langle a, b \rangle}{\langle b, b \rangle} b$$

Definition 26 (Konvexe Menge, konvexe/affine Hülle)

Eine Teilmenge $A \subset \mathbb{R}^n$ heißt *konvex*, wenn mit je zwei Punkten $a, b \in A$ auch die gesamte Strecke $\overline{(a, b)}$ vollständig in A liegt:

$$a, b \in A \Rightarrow \lambda a + (1 - \lambda)b \in A \text{ mit } \lambda \in [0, 1] .$$

Die konvexe und die affine Hülle einer Teilmenge $A \subset \mathbb{R}^n$ werden gebildet durch

$$\text{Conv}(A) := \{c = \lambda a + (1 - \lambda)b \mid \lambda \in [0, 1], a, b \in A\}$$

$$\text{Aff}(A) := \{c = \lambda a + (1 - \lambda)b \mid \lambda \geq 0, a, b \in A\}$$

Definition 27 (Minkowski-Summe)

Seien $A, B \subset \mathbb{R}^n$. Die Minkowski-Summe ist wie folgt definiert:

$$A \oplus B := \{a + b \mid a \in A, b \in B\}$$

Definition 28 (Konvexe Distanzfunktion)

Sei $B \subset \mathbb{R}^n$ eine konvexe Menge (Ball), die den Koordinatenursprung enthält. Dann induziert B eine konvexe Distanzfunktion wie folgt:

$$d_B(p, q) := \min\{\lambda \mid q \in p \oplus \lambda B\}$$

Ist B zudem symmetrisch zum Koordinatenursprung, so ist d_B eine Metrik.

Definition 29 (L_p -Metrik / Minkowski-Metrik)

Seien $a, b \in \mathbb{R}^n$. Für $1 \leq p \leq \infty$ ist die L_p -Metrik wie folgt definiert:

$$d_p(a, b) := \left(\sum_{i=1}^n |a_i - b_i|^p \right)^{1/p} \quad (\text{A.18})$$

L_p -Metriken sind konvexe Distanzfunktionen. Ferner entsprechen sich die folgenden üblichen Bezeichnungen:

L_2 -Metrik: Euklidische Distanz

L_1 -Metrik: Manhattan-Distanz (Summe achsenparalleler Abstände)

L_∞ -Metrik: Maximum-Norm (maximaler achsenparalleler Abstand)

Definition 30 (Komplementäre/symmetrische Menge)

Das Komplement einer Menge $A \subset \mathbb{R}^n$ wird mit \bar{A} bezeichnet.

Die symmetrische bzw. punktspiegelte Menge zu A ist $-A := \{-a \mid a \in A\}$

Anhang B

Abschnittsübersicht nach Problemklassen

Globale Optimierung/rekagonale Objekte

Weitgehend abgeschlossene Behandlung in 2.1- 2.3 mit Einführung von Anordnungsgraphen und -klassen zur Berechnung unterer Schranken. Erweiterung auf Kabelraumabschätzung und Anwendung in 8.2.

Globale Optimierung/polyedrische Objekte/diskrete Orientierungen

Abschnitte 2.1, 2.2 und 2.4 zum Optimierungsverfahren; zur Diskretisierung des Anordnungsraumes mit Hilfe von Distanzpolyedern siehe 5.2. Im weiteren siehe 5.3 zu \mathcal{P} -Spuren als Alternative zu Distanzpolyedern.

Lokale Optimierung/polyedrische Objekte/kontinuierliche Rotationen

Abschnitte 3.1, 3.2 und 3.4 zum Optimierungsverfahren; Behandlung von Drehungen in Kapitel 6, Eigenschaften infinitesimaler Drehungen in Kapitel 7 und Anpassung von Separationsebenen in Abschnitt 5.4.

Globale Optimierung/polyedrische Objekte/lokale Rotationsanpassung

Abschnitte 2.1, 2.2 zur Modellierung mit Entscheidungsvariablen; Abschnitte 3.1, 3.2 zur Linearisierungsidee; Abschnitt 5.3 zu Winkelbereichen topologisch äquivalenter Rotationen (Abschnitt 5.2 hilfreich zum Verständnis von 5.3).

Anhang C

Softwarekomponenten

Für die Durchführung des Forschungsvorhabens wurden die nachfolgend aufgeführten Softwarekomponenten von dritter Seite eingesetzt, die für wissenschaftliche Zwecke frei verfügbar sind und aus verschiedenen öffentlichen Quellen gefördert wurden. Der Quellcode in C/C++ ist jeweils frei verfügbar, jedoch mit unterschiedlichem Urheberrechtsschutz. Den fördernden Institutionen und insbesondere den Autoren Dank und Anerkennung für die geleisteten Programmimplementierungen, ohne die die in dieser Arbeit erzielten Ergebnisse nicht möglich gewesen wären.

In Abb. C.1 ist die Integration aufgezeigt, wobei die meisten Komponenten direkt in die Geometriebibliothek eingebunden wurden. Das Modul zu infinitesimalen Rotationen stellt noch eine externe Erweiterung dar.

PlaGeo/SpaGeo 2D/3D Geometriepaket, Universität Utrecht, Autor: Geert-Jan Giezeman (jetzt Projekt „CGAL - Computational Geometry Algorithms Library“)

SoPlex Simplex-Implementierungen zur Lösung linearer Programme des Konrad-Zuse-Zentrum für Informationstechnik, Berlin. Autor: Roland Wunderling

Qhull Berechnung konvexer Hüllen, The Geometry Center, University of Minnesota. Autoren: Barber, Dobkin und Huhdanpaa

GeomView Visualisierung, The Geometry Center, University of Minnesota

GJK-engine Berechnung von Separationsebenen, Technische Universität Eindhoven. Autor: Gino van den Bergen.

Zudem wurden die folgenden kommerziell erhältlichen Programmpakete eingesetzt:

CasCade CAD-Entwicklungsumgebung von EADS Matra Datavision (Quellcode als OpenSource verfügbar)

CPLX Programmpaket zur Lösung linearer Programme (Simplex- und Interior Point Verfahren), ILOG

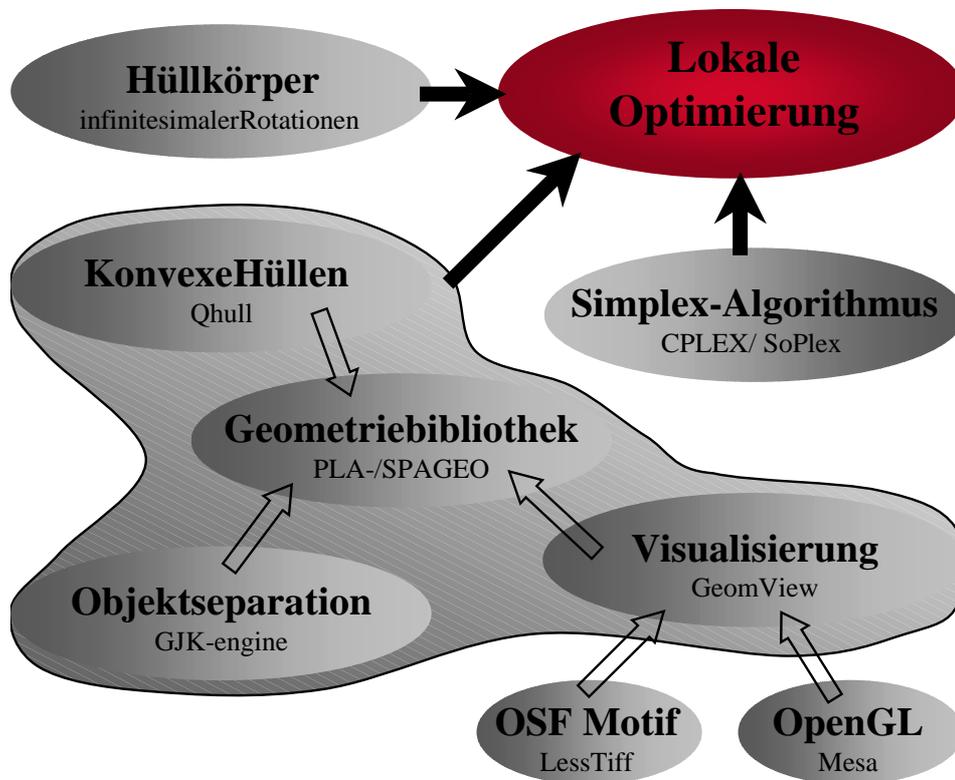


Abbildung C.1: Softwarekomponenten für lokales Optimierungsverfahren

Abbildungsverzeichnis

1.1	Querbezüge zwischen den Kapiteln	4
2.1	Überschneidungsfreie Anordnung Rechteck / Quadrat	15
2.2	Anordnungsgraph $G_x = (V, A_x)$	19
2.3	Minimale Koordinate $(r_i)_x$ und längste Pfade in $G_x = (V, A_x)$	19
2.4	Dreiecksungleichung $r_i + c_{ij} \leq r_j$	20
2.5	Anordnungsclassen G_d^C und Anordnungsgraphen G_d	22
2.6	Alternative Orientierung von G_x mit Anordnung	22
2.7	Alternative Orientierung von G_y mit Anordnung	22
2.8	Gewinn- und Verlustzonen für u^ε	24
2.9	Gewinn- und Verlustzonen von u^k (für u^1, \dots, u^4)	25
2.10	Zusammenfassung von Teilgebieten s_{ijk} und konvexe Hülle	27
2.11	Disjunkte, aber nicht überdeckende Teilgebiete s_{ijk}	27
2.12	Extremalstrahlen und konvexe Hülle	28
2.13	Zusammenfassung von Teilgebieten s_{ijk} und konvexe Hülle	29
2.14	Hierarchie reduzierter Distanzpolyeder	30
3.1	Sechs Schritte eines lokalen Optimierungsverfahrens	32
3.2	Lineare Separation zweier konvexer Polyeder P_i und P_j	34
3.3	Minimale Separation zweier Polyeder	40
3.4	Konvergenz von $\mathcal{I}(v_k, \varepsilon_{max})$ gegen w	41
4.1	Winkelbereiche von \mathcal{P} -Spuren	47
4.2	Äquivalenzklassen von Distanzpolyedern	48
4.3	Repräsentanten der Äquivalenzklassen von Distanzpolyedern	49
4.4	Facette des Distanzpolyeders mit erzeugenden Eckpunkten	50
4.5	Linearisierung zweidimensionaler Rotationen	51
5.1	Minkowski-Summe eines Rechtecks und eines Quadrates	62
5.2	Punktgespiegelte Menge $-B$	62
5.3	Ausgangspolyeder A, B und symmetrische Mengen	64
5.4	Distanzpolyeder $D_{A,B} = A \oplus (-B)$ und $D_{B,A} = B \oplus (-A)$	64
5.5	Minkowski Differenz $D_{A,B} \ominus (-B) = A$	66
5.6	Konstruktion von $D_{A,B} = A \oplus (-B)$	67
5.7	\mathcal{P} -Spur zweier Polygone mit Selbstüberschneidung	69
5.8	Worst Case Komplexität $O(m^2n^2)$ der Minkowski-Summe	69
5.9	Minkowski-Summe, \mathcal{P} -Spuren und Faltungsintegral	70

5.10	Normalenvektoren und Winkelbereiche eines Polyeders	71
5.11	Orts- und Richtungsindikator einer \mathcal{P} -Spur	72
5.12	Überschneidung von Winkelbereichen und neue Randobjekte	73
5.13	Richtungsindikator einer konvexen und nicht konvexen Ecke	74
5.14	Vorzeichen $\sigma(o)$ für (nicht-)konvexe Ecken	74
5.15	Winkelbereiche nicht konvexer Ecken: zwei Def.-möglichkeiten	75
5.16	Selbstüberschneidung der Winkelbereiche von k_2 und k_4	76
5.17	Normalisierung der Winkelbereiche zur Ecke e	77
5.18	Winding Number bzgl. $\mathcal{P}_{A,-B}$	78
5.19	Iterationen des GJK-Algorithmus: Wahl von m_k und e_k	81
6.1	Drehung mit Eulerschen Winkeln α, β, γ	88
7.1	Bezeichnungen der Vektoren zu Lemma 25	101
7.2	Richtungskomponenten der o_i senkrecht zu einer Spatfläche	104
7.3	Raumecke gebildet von o_1, o_2, o_3	106
7.4	Zweidimensionaler Schnitt; (o_1, d) -Ebene	106
7.5	Durch o_1, o_2, o_3 definierte Ebene E_Δ der Raumecke	106
7.6	Extremalposition von o_i bzgl. n	107
7.7	Bezeichnungen zum Beweis von Theorem 31, Fall 2	110
7.8	Schematische Darstellung zu Fall 2A und 2B	112
7.9	Beispiele für die Einbettung des Einheitswürfels	114
7.10	Hüllkörper und Einbettungen polyedrischer Objekte	115
8.1	Definition der Verdrahtungszellen	121
8.2	Beispiel für Verdrahtungsgraph mit Verdrahtungsvarianten	122
8.3	Variablen für die Verdrahtungsraumabschätzung	123
8.4	Referenzanordnung des Industriepartners für SG129	126
8.5	Volumenminimale Anordnung mit Kabelraumabschätzung	127
8.6	Referenzanordnung des Industriepartners	128
8.7	Modellierung des Anordnungsraumes	129
8.8	Alternative Anordnung mit geringerem Volumen	130
8.9	Einpassen eines Bauteils in bestehende Aggregatanordnung	131
8.10	Kofferraumausliterung mit optimal gefülltem Quader	132
C.1	Softwarekomponenten für lokales Optimierungsverfahren	142

Literaturverzeichnis

- [All78] ALLEN, ARNOLD O.: *Probability, Statistics, and Queueing Theory*. Computer Science and Applied Mathematics. Academic Press, London, UK, 1978.
- [BB98] BARTH, P. und A. BOCKMAYR: *Modelling discrete optimisation problems in constraint logic programming*. Annals of Operations Research, 81:467–496, 1998.
- [Bea85] BEASLEY, J. E.: *An Exact Two-Dimensional Non-Guillotine Cutting Tree Search Procedure*. Operations Research, 33(1):49–64, 1985.
- [BGR96] BASCH, JULIEN, LEONIDAS J. GUIBAS und G. D. RAMKUMAR: *Reporting Red-Blue Intersections Between Connected Sets of Line Segments*. In: *Proc. 4th European Symposium on Algorithms (ESA '96)*, Band 1136 der Reihe *Lecture Notes in Computer Science*, Seiten 302–319, Berlin, 1996. Springer-Verlag.
- [BGR96] BASCH, JULIEN, LEONIDAS J. GUIBAS, G. D. RAMKUMAR und LYLE RAMSHAW: *Polyhedral Tracings and their Convolution*. In: *Algorithms for Robotic Motion and Manipulation: Workshop on the Algorithmic Foundations of Robotics*, Seiten 171–184. A K Peters, Ltd., 1996.
- [BK98] BOCKMAYR, A. und T. KASPER: *Branch-and-Infer: A Unifying Framework for Integer and Finite Domain Constraint Programming*. INFORMS Journal on Computing, 10(3):287–300, 1998.
- [Bol89] BOLCH, GUNTER: *Leistungsbewertung von Rechensystemen mittels analytischer Warteschlangenmodelle*. Teubner, Stuttgart, 1989.
- [Bri83] BRIESKORN, EGBERT: *Lineare Algebra und analytische Geometrie*. Vieweg, Braunschweig, 1983.
- [BS79] BAZARAA, MOKHTAR S. und C. M. SHETTY: *Nonlinear Programming*. John Wiley & Sons, New York, NY, 1979.

- [BS81] BRONŠTEIN, IL'JA NIKOLAEVIČ und KONSTANTIN ADOL'FOVIČ SEMENDJAEV: *Taschenbuch der Mathematik*. Thun, Frankfurt/Main, 1981.
- [BW95] BISCHOFF, E. E. und G. WÄSCHER: *Cutting and Packing*. European Journal of Operational Research, 84:503–505, 1995.
- [CAC97] CAMPBELL, M. I., C. H. AMON und J. CAGAN: *Optimal Three-Dimensional Placement of Heat Generating Electronic Components*. ASME Journal of Electronic Packaging, 119(2):106–113, 1997.
- [Cag94] CAGAN, JONATHAN: *Shape Annealing Solution to the Constrained Geometric Knapsack Problem*. Computer Aided Design, 26(10):763–769, 1994.
- [Cam97] CAMERON, STEPHEN: *Enhancing GJK: Computing Minimum and Penetration Distances between Convex Polyhedra*. In: *Proc. IEEE Int. Conference on Robotics and Automation*, Seiten 3112–3117, Albuquerque, 22.-24. April 1997.
- [CCD⁺96] CAGAN, JONATHAN, RICHARD CLARK, PRATIP DASTIDAR, SIMON SZYKMAN und PAUL WEISSER: *HVAC CAD Layout Tools: a Case Study of University/Industry Collaboration*. In: *Proc. 22nd ASME Design Theory and Methodology Conference [DETC96/DTM-1505]*, Irvine, 18-22 August 1996.
- [CD85] CHEW, L. P. und R. L. DRYSDALE: *Voronoi Diagrams Based on Convex Distance Functions*. In: *Proc. 1st ACM Symposium on Computational Geometry*, Seiten 234–244, 1985.
- [CDY98] CAGAN, JONATHAN, DREW DEGENTESH und SU YIN: *A Simulated Annealing-Based Algorithm Using Hierarchical Models for General Three-Dimensional Component Layout*. Computer Aided Design, 30(10):781–790, 1998.
- [CGJ84] COFFMAN, JR., E. G., M. R. GAREY und D. S. JOHNSON: *Approximation Algorithms for Bin-Packing — An Updated Survey*. In: AUSIELLO, G., M. LUCERTINI und P. SERAFINI (Herausgeber): *Algorithm Design for Computer System Design*, Seiten 49–106. Springer-Verlag, Berlin, 1984.
- [Che99] CHEN, BEIFANG: *Minkowski Algebra I: A Convolution Theory of Closed Convex Sets and Relatively Open Convex Sets*. Asian Journal of Mathematics, 3(3):609–634, 1999.

- [DD92] DOWSLAND, KATHRYN A. und WILLIAM B. DOWSLAND: *Packing Problems*. European Journal of Operational Research, 56:2–14, 1992.
- [DD95] DOWSLAND, KATHRYN A. und WILLIAM B. DOWSLAND: *Solution Approaches to Irregular Nesting Problems*. European Journal of Operational Research, 84:506–521, 1995.
- [DF92] DYCKHOFF, HARALD und UTE FINKE: *Cutting and Packing in Production and Distribution: Typology and Bibliography*. Contributions to Management Science. Physica-Verlag, Heidelberg, 1992.
- [DLM92] DANIELS, KAREN, ZHENYU LI und VICTOR MILENKOVIC: *Placement and Compaction of Non-Convex Polygons for Clothing Manufacture*. In: WANG, C. A. (Herausgeber): *Proc. 4th Canadian Conference on Computational Geometry*, Seiten 236–243, August 1992.
- [Duv01] DUVENBECK, CLEMENS: *Charakterisierung überschneidungsfreier Anordnungen polyedrischer Objekte unter Berücksichtigung von Translation und Rotation*. Diplomarbeit, Universität Bonn, Institut für Informatik, 2001.
- [Fis78] FISCHER, GERD: *Analytische Geometrie*. Vieweg-Studium. Vieweg, Braunschweig, 1978.
- [Fis86] FISCHER, GERD: *Lineare Algebra*. Vieweg-Studium. Vieweg, Braunschweig, 1986.
- [FS97a] FEKETE, SÁNDOR und JÖRG SCHEPERS: *New Classes of Lower Bounds for Bin Packing Problems*. ZPR Report 97.265, Universität zu Köln, Center for Parallel Computing, Cologne, Germany, 1997.
- [FS97b] FEKETE, SÁNDOR und JÖRG SCHEPERS: *A New Exact Algorithm for General Orthogonal d -Dimensional Knapsack Problems*. In: *Proc. 5th European Symposium on Algorithms ESA'97*, Band 1284 der Reihe *Lecture Notes in Computer Science*, Seiten 144–156. Springer-Verlag, 1997.
- [FW95] FLEMMING, ULRICH und ROBERT F. WOODBURY: *Software Environment to Support Early Phases in Building Design (SEED): Overview*. ASCE Journal of Architectural Engineering, 1(4):147–152, Dezember 1995.

- [GF90] GILBERT, ELMER G. und C.-P. FOO: *Computing the Distance Between General Convex Objects in Three-Dimensional Space*. Transactions on Robotics and Automation, 6(1):53–61, Februar 1990.
- [Gho90] GHOSH, PIJUSH K.: *A Solution of Polygon Containment, Spatial Planning, and Other Related Problems Using Minkowski Operations*. Computer Vision, Graphics, and Image Processing, 49:1–35, 1990.
- [GJ79] GAREY, M. R. und D. S. JOHNSON: *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, New York, NY, 1979.
- [GJK88] GILBERT, ELMER G., DANIEL W. JOHNSON und S. SATHIYA KEERTHI: *A Fast Procedure for Computing the Distance Between Complex Objects*. IEEE Journal on Robotics and Automation, 4(2):193–203, April 1988.
- [GO94] GILBERT, ELMER G. und CHONG-JIN ONG: *New Distances for the Separation and Penetration of Objects*. In: *Proc. Int. Conference on Robotics and Automation*, Seiten 579–586, San Diego, Mai 1994. IEEE.
- [GO97] GOODMAN, JACOB E. und JOSEPH O’ROURKE: *Handbook of Discrete and Computational Geometry*. CRC Press, Boca Raton, NY, 1997.
- [Gol89] GOLDSTEIN, HERBERT: *Klassische Mechanik*. Aula-Verlag, Wiesbaden, 1989. 10. Auflage.
- [GRS83] GUIBAS, LEONIDAS J., LYLE RAMSHAW und JORGE STOLFI: *A Kinetic Framework for Computational Geometry*. In: *Proc. 24th IEEE Symposium on Foundations of Computer Science*, Seiten 100–111, 1983.
- [GS87] GUIBAS, LEONIDAS J. und RAIMUND SEIDEL: *Computing Convolutions by Reciprocal Search*. Discrete Computational Geometry, 2:175–193, 1987.
- [GW93] GRUBER, PETER M. und JÖRG. M. WILLS: *Handbook of Convex Geometry*. North-Holland, Amsterdam, NL, 1993.
- [Hec98] HECKMANN, RALF: *Nesting - Berechnung oberer und unterer Schranken*. Dissertation, Rheinische Friedrich-Wilhelms-Universität Bonn, Mathematisch-Naturwissenschaftliche Fakultät, 1998.

- [Hel96] HELD, MARTIN: *Reliable C Code for Computing Triangle-Triangle Intersections and Triangle-Segment Intersections in 2D and 3D*. Technical Note, AMS Dept., SUNY Stony Brook, Stony Brook, USA, 8 Mai 1996.
- [HL95] HECKMANN, RALF und THOMAS LENGAUER: *A Simulated Annealing Approach to the Nesting Problem in the Textile Manufacturing Industry*. *Annals of Operations Research*, 57:103–133, 1995.
- [HL96] HECKMANN, RALF und THOMAS LENGAUER: *Computing Upper and Lower Bounds on Textile Nesting Problems*. In: DIAZ, JOSEP und MARIA SERNA (Herausgeber): *Proc. 4th European Symposium on Algorithms ESA'96*, Band 1136 der Reihe *Lecture Notes in Computer Science*, Seiten 392–405. Springer-Verlag, 1996.
- [KCR96] KOLLI, ASHISH, JONATHAN CAGAN und ROB RUTENBAR: *Packing of Generic, Three-Dimensional Components Based on Multi-Resolution Modeling*. In: *Proc. 22nd ASME Design Automation Conference [DETC96/DAC-1479]*, Irvine, 18-22 August 1996.
- [KK71] KLINGENBERG, WILHELM und PETER KLEIN: *Lineare Algebra und analytische Geometrie*. B.I.-Hochschultaschenbücher. Bibliographisches Institut, Mannheim, 1971.
- [KK72] KLINGENBERG, WILHELM und PETER KLEIN: *Lineare Algebra und analytische Geometrie, Band 2*. B.I.-Hochschultaschenbücher. Bibliographisches Institut, Mannheim, 1972.
- [KOS91] KAUL, A., M. A. O'CONNOR und V. SRINIVASAN: *Computing Minkowski Sums of Regular Polygons*. In: *Proc. 3rd Canad. Conference on Computational Geometry*, Seiten 74–77, 1991.
- [KS98] KIM, GERARD JOUGHYUN und SIMON SZYKMAN: *Combining Interactive Exploration and Optimization for Assembly Design*. *ASME Journal of Mechanical Design*, 120(1):24–31, 1998.
- [KW84] KEDEM, GERSHON und HIROYUKI WATANABE: *Graph-Optimization Techniques for IC Layout and Compaction*. *IEEE Transactions on Computer-Aided Design*, 3(1):12–20, 1984.
- [Len90] LENGAUER, THOMAS: *Combinatorial Algorithms for Integrated Circuit Layout*. *Applicable Theory in Computer Science*. Wiley-Teubner, Chichester-Stuttgart, 1990.

- [LL93] LENGAUER, THOMAS und MARTIN LÜGERING: *Integer Program Formulations of Global Routing and Placement Problems*. In: SARRAFZADEH, M. und D. T. LEE (Herausgeber): *Algorithmic Aspects of VLSI Layout*, Band 2 der Reihe *Lecture Notes Series on Computing*, Seiten 167–197. World Scientific, Singapore, 1993.
- [LM86] LUNDY, M. und A. MEES: *Convergence of an Annealing Algorithm*. *Mathematical Programming*, 34:111–124, 1986.
- [LM93] LENGAUER, THOMAS und ROLF MÜLLER: *Robust and Accurate Hierarchical Floorplanning with Integrated Global Wiring*. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 12(6):802–809, Juni 1993.
- [LM95] LI, ZHENYU und VICTOR MILENKOVIC: *Compaction and Separation Algorithms for Non-Convex Polygons and their Applications*. *European Journal of Operational Research*, 84:539–561, 1995.
- [LMR86] LANGRANA, N. A., T. M. MITCHELL und N. RAMACHANDRAN: *Progress toward a Knowledge-Based Aid for Mechanical Design*. Technical Report WP-TR-83, Rutgers University, Department of Computer Science, New Brunswick, NJ, 1986.
- [Mil98a] MILENKOVIC, VICTOR J.: *Rotational Polygon Containment and Minimum Enclosure*. In: *Proc. 14th ACM Symposium on Computational Geometry*, Juni 1998.
- [Mil98b] MILENKOVIC, VICTOR J.: *Rotational Polygon Overlap Minimization and Compaction*. *Computational Geometry*, 10(4):305–318, 1998.
- [MKW⁺86] MITSUTA, TORU, YASUHIRO KOBAYASHI, YUTAKA WADA, TAKASHI KIGUCHI und TOSHIAKI YOSHINAGA: *A Knowledge-Based Approach to Routing Problems in Industrial Plant Design*. In: *Proc. 6th Int. Workshop on Expert Systems & Their Applications*, Seiten 237–256, Avignon, April 1986.
- [MM97] MILENKOVIC, VICTOR J. und V. MILENKOVIC: *Rational Orthogonal Approximations to Orthogonal Matrices*. *Computational Geometry: Theory and Applications*, 7:25–35, 1997.
- [OG97] ONG, CHONG-JIN und ELMER G. GILBERT: *The Gilbert-Johnson-Keerthi Distance Algorithm: A Fast Version for Incremental Motions*. In: *Proc. IEEE Int. Conference on Robotics and Automation*, Seiten 1183–1189, 1997.

- [PS85] PREPARATA, F. P. und M. I. SHAMOS: *Computational Geometry: An Introduction*. Springer-Verlag, Berlin, 1985.
- [Ram96] RAMKUMAR, G. D.: *An Algorithm to Compute the Minkowski Sum Outer-Face of Two Simple Polygons*. In: *Proc. 12th ACM Symposium on Computational Geometry*, Seiten 234–241, 1996.
- [Rau84] RAULEFS, PETER: *Foundation of Expert Systems for Conceptual Design in Mechanical Engineering*. Interner Bericht 115/84, Universität Kaiserslautern, Fachbereich Informatik, 1984.
- [RCO99] RIBEIRO, CRISTINA, MARIA ANTANIA CARRAVILLA und JOS F. OLIVEIRA: *Applying Constraint Logic Programming to the Resolution of Nesting Problems*. *Pesquisa Operacional*, 19:239–247, Dezember 1999.
- [SC93] SZYKMAN, SIMON und JONATHAN CAGAN: *Automated Generation of Optimally Directed Three Dimensional Component Layouts*. In: *Proc. 19th ASME Design Automation Conference*, Band 1, Seiten 527–537, Albuquerque, 19–22 September 1993.
- [SC94] SZYKMAN, SIMON und JONATHAN CAGAN: *A Simulated Annealing-Based Approach to Three Dimensional Component Packing*. In: *Proc. 20th ASME Design Automation Conference*, Minneapolis, 11–14 September 1994.
- [SC96a] SCHMIDT, LINDA C. und JONATHAN CAGAN: *Configuration Design: An integrated Approach Using Grammars*. In: *Proc. 22nd ASME Design Theory and Methodology Conference [DETC96/DTM-1511]*, Irvine, 18–22 August 1996.
- [SC96b] SZYKMAN, SIMON und JONATHAN CAGAN: *Synthesis of Optimal Non-Orthogonal Routes*. *ASME Journal of Mechanical Design*, 118(3):419–424, 1996.
- [SC97] SZYKMAN, SIMON und JONATHAN CAGAN: *Constrained Three-Dimensional Component Layout using Simulated Annealing*. *ASME Journal of Mechanical Design*, 119(1):28–35, 1997.
- [Sch91a] SCHANUEL, S. H.: *Negative Sets have Euler Characteristic and Dimension*. In: *Proc. Int. Conference on Category Theory*, Band 1488 der Reihe *Lecture Notes in Mathematics*, Seiten 380–385, Como, IT, 22–28 Juli 1991. Springer-Verlag.
- [Sch91b] SCHAPIRA, PIERRE: *Operations on Constructible Functions*. *Journal of Pure and Applied Algebra*, 72(1):83–93, 1991.

- [SCW98] SZYKMAN, SIMON, JONATHAN CAGAN und PAUL WEISSER: *An Integrated Approach to Optimal Three Dimensional Layout and Routing*. ASME Journal of Mechanical Design, 120(3):510–512, 1998.
- [SD88] SANDGREN, E. und T. DWORAK: *Part Layout Optimization using a Quadtree Representation*. In: *Proc. 14th ASME Design Automation Conference*, Seiten 211–219, Kissimmee, 25-28 September 1988.
- [SGR⁺00] STOYAN, Y., M. GIL, T. ROMANOVA, J. TERNO und G. SCHEITHAUER: *Construction of a ϕ -Function for Two Convex Polytopes*. Pre-Print, Seiten 1–17, September 2000.
- [SL01] SCHÄFER, MIKE und THOMAS LENGAUER: *Automated Layout Generation and Wiring Area Estimation for 3D Electronic Modules*. Journal of Mechanical Design, 123(3):330–336, 2001.
- [SLar] SCHÄFER, MIKE und THOMAS LENGAUER: *Combinatorial Optimization Techniques for three-dimensional Arrangement Problems*. to appear.
- [SLW83] SCHLAG, M., Y.-Z. LIAO und C. K. WONG: *An Algorithm for Optimal Two-Dimensional Compaction of VLSI Layouts*. Integration, 1:179–209, 1983.
- [SP92] SWEENEY, P. E. und E. R. PATERNOSTER: *Cutting and Packing Problems: A Categorized Application-Oriented Research Bibliography*. Journal of the Operational Research Society, 43:691–706, 1992.
- [SSBR99] SZYKMAN, SIMON, RAM D. SRIRAM, C. BOCHENEK und JANUSZ W. RACZ: *The NIST Design Repository Projekt*. In: ROY, R., T. FURUHASHI und P. K. CHAUDHRY (Herausgeber): *Advances in Software and Computing - Engineering Design and Manufacturing*, Seiten 5–19. Springer-Verlag, Berlin, 1999.
- [SSS97] SZYKMAN, SIMON, LINDA C. SCHMIDT und HARSHAWARDHAN SHETTY: *Improving the Efficiency of Simulated Annealing Optimization through Detection of Productive Search*. In: *Proc. 23th ASME Design Automation Conference [DETC97/DAC-398]*, Sacramento, 14-17 September 1997.
- [ST93] SCHEITHAUER, GUNTRAM und JOHANNES TERNO: *Modelling of Packing Problems*. Optimization, 28:63–84, 1993.

- [Sto83] STOYAN, Y. G.: *Mathematical Methods for Geometry Design*. In: ELLIS, T. M. R. und O. I. SEMENKOV (Herausgeber): *Advances in CAD/CAM*, Seiten 67–86. North-Holland Publishing Company, 1983.
- [SW70] STOER, J. und C. WITZGALL: *Convexity and Optimization in Finite Dimensions I*. Springer-Verlag, Berlin, 1970.
- [TLS87] TERNO, JOHANNES, R. LINDEMANN und GUNTRAM SCHEITHAUER: *Zuschnittprobleme und ihre Praktische Lösung*. Verlag Harry Deutsch, Thun, 1987.
- [UBBL88] UDY, JERRY L., RICHARD J. BALLING, STEVEN E. BENZLEY und MARK D. LANDON: *Computation of Interferences between Three-Dimensional Objects and the Optimal Packing Problem*. *Advances in Engineering Software*, 10(1):8–14, 1988.
- [vdB98] BERGEN, GINO VAN DEN: *Efficient Collision Detection of Complex Deformable Models using AABB Trees*. *Journal of Graphics Tools*, 4(2):1–14, November 1998.
- [vdB99] BERGEN, GINO VAN DEN: *A Fast and Robust GJK Implementation for Collision Detection of Convex Objects*. *Journal of Graphics Tools*, 4(2):7–25, 1999.
- [Wat84] WATANABE, HIROYUKI: *IC Layout Generation and Optimization Using Mathematical Optimization*. PhD Dissertation, University of Rochester, Department of Computer Science, 1984.
- [WC90] WOLTER, J. und P. CHANDRASEKARAN: *Knowledge Representation for Functional Design of Mechanical Assemblies*. Technical Report 90-026, Texas A&M University, Department of Computer Science, 1990.
- [WKC89] WIMER, SHMUEL, ISRAEL KOREN und ISRAEL CEDERBAUM: *Optimal Aspect Ratios of Building Blocks in VLSI*. *IEEE Transactions on Computer-Aided Design*, 8(2):139–145, Februar 1989.
- [YC00] YIN, SU und JONATHAN CAGAN: *An Extended Pattern Search Algorithm for Three-Dimensional Component Layout*. *ASME Journal of Mechanical Design*, 122(1):102–108, 2000.
- [ZL91] ZHU, DAVID und JEAN-CLAUDE LATOMBE: *Mechanization of Spatial Reasoning for Automatic Pipe Layout Design*. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 5(1):1–20, 1991.