



DGK Deutsche Geodätische Kommission
bei der Bayerischen Akademie der Wissenschaften

Reihe C

Dissertationen

Heft Nr. 647

Christian Clemen

**Ein geometrisch-topologisches Informationsmodell
für die Erfassung und Validierung von
flächenparametrisierten 3d-Gebäudemodellen**

München 2010

Verlag der Bayerischen Akademie der Wissenschaften
in Kommission beim Verlag C. H. Beck

ISSN 0065-5325

ISBN 978-3-7696-5059-4



Ein geometrisch-topologisches Informationsmodell
für die Erfassung und Validierung von
flächenparametrisierten 3d-Gebäudemodellen

Von der Fakultät VI – Planen Bauen Umwelt
der Technischen Universität Berlin
zur Erlangung des Grades
Doktor der Ingenieurwissenschaften (Dr.-Ing.)
genehmigte Dissertation

von

Dipl.-Ing. Christian Clemen

D 83

München 2010

Verlag der Bayerischen Akademie der Wissenschaften
in Kommission beim Verlag C. H. Beck

Adresse der Deutschen Geodätischen Kommission:



Deutsche Geodätische Kommission

Alfons-Goppel-Straße 11 • D – 80 539 München

Telefon +49 – 89 – 23 031 1113 • Telefax +49 – 89 – 23 031 - 1283/ - 1100

e-mail hornik@dgfi.badw.de • <http://www.dgk.badw.de>

Vorsitzender: Prof. Dr.rer.nat. Thomas Kolbe
1. Gutachter: Prof. Dr.-Ing. Dr. h.c. Lothar Gründig
2. Gutachter: Prof. Dr.-Ing.habil. Frank Gielsdorf
3. Gutachter: Prof. Dr.-Ing. Wolfgang Huhnt

Tag der wissenschaftlichen Aussprache: 04. September 2009

© 2010 Deutsche Geodätische Kommission, München

Alle Rechte vorbehalten. Ohne Genehmigung der Herausgeber ist es auch nicht gestattet,
die Veröffentlichung oder Teile daraus auf photomechanischem Wege (Photokopie, Mikrokopie) zu vervielfältigen

ISSN 0065-5325

ISBN 978-3-7696-5059-4

Zusammenfassung

Digitale Gebäudemodelle bilden die Grundlage für die Planung, den Bau und Umbau sowie für die Bewirtschaftung von Gebäuden. Die Geometrische Information wird in der Regel mit CAD-Software bearbeitet und als Zeichnungsdatei in einer Layerstruktur gespeichert. Maße (Relativgeometrie) werden aus der Zeichnung (Absolutgeometrie) von der CAD-Software assoziativ berechnet. Die alphanumerischen Sachdaten werden getrennt von der CAD-Datei gespeichert. Diese Vorgehensweise widerspricht in vielen Punkten einer *Ingenieurgeodätischen Denkweise*: Zunächst ist „Geometrie“ mehr als die Lage eines Zeichnungselementes. Geometrie beschreibt Lage und Form einzelner *Objekte* und bildet darüberhinaus einen geeigneten *Ordnungsrahmen* für alle weiteren Informationen des gesamten Gebäudemodells. Für eine technisch belastbare Bestandsdokumentation ist die Datenerfassung besonders wichtig, weil sie die Datengrundlage für alle weiteren Berechnungen und Analysen bildet. Die Vermessung eines Gebäudes beruht auf der Menge der gemessenen (nicht berechneten) Relativmaße, impliziten oder expliziten Annahmen (Rechtwinkligkeit, Parallelität...) und der Datumsfestlegung.

In der vorliegenden Arbeit wird eine neuartige Methode vorgestellt, die den traditionellen *Arbeitsprozess* umkehrt und eine ingenieurgeodätische Denkweise sowohl im konzeptionellen und logischen Datenmodell, als auch in der entwickelten Software umsetzt: Der Prozess beginnt mit der Festlegung der topologischen Gebäudestruktur mit einer dreidimensionalen Skizze. Relativmaße, die aus stochastisch modellierten Beobachtungen (Relativgeometrie) entstehen und an die Skizze „geheftet“ werden, sind die Grundlage für die Berechnung der Geometrie des darzustellenden Objektes (Absolutgeometrie). Für diese Berechnung wird das Verfahren der Ausgleichsrechnung verwendet. Die Festlegung des endgültigen Koordinatensystems erfolgt als letzter Arbeitsschritt.

Ausgehend von den Forschungsarbeiten von Dr. -Ing. habil. Frank Gielsdorf am *Institut für Geodäsie und Geoinformationstechnik der Technischen Universität Berlin* wird in dieser Arbeit ein neues konzeptionelles und logisches Datenmodell entwickelt. Die Leitidee dieses Ansatzes besteht darin, dass die Vermessung nicht nur in einem Vorverarbeitungsschritt, sondern gemeinsam mit der Gebäudestruktur über den gesamten Lebenszyklus des Gebäudes gespeichert wird. Im Datenmodell werden Relativgeometrie und Absolutgeometrie über die topologischen Elemente der Gebäudestruktur verknüpft. Mit dem Ziel der Minimierung der Anzahl der Geometrieparameter werden zur Beschreibung der Absolutgeometrie keine Punktkoordinaten, sondern *Flächenparameter* verwendet. Diese können zu jedem Zeitpunkt mittels Ausgleichsrechnung berechnet werden. Die *Ausgleichsrechnung*, als zentraler Algorithmus der Methode, wird als Gauß-Helmert Modell funktional modelliert und in einer Software implementiert und getestet. Zum Zweck der Linearisierung der Beobachtungs- und Bedingungsgleichungen wird das leistungsfähige Verfahren der *Differentiationsarithmetik* verwendet, das eine generische Programmierung der Parameterschätzung gut unterstützt. Für den praktischen Arbeitsprozess ist es von hoher Bedeutung, die topologischen und geometrischen *Konsistenzverletzungen* in der 3d-Zeichnung frühzeitig und automatisiert zu detektieren und die Zeichnungsfehler für den Benutzer zu visualisieren. Die vorliegende Dissertationsschrift klassifiziert mögliche Konsistenzverletzungen und zeigt, wie diese in verschiedenen Softwaremodulen visualisiert werden können.

Das zeitraubende Problem der Geoemtriedatenerfassung zur Bestandsdokumentation kann mit dem hier entwickelten Verfahren effizienter und zuverlässiger durchgeführt werden. Darüber hinaus unterstützt das entwickelte System auch eine stochastische Validierung vorhandener Bestandsdaten.

Abstract

Digital Building Models provide the base information for planning, constructing, renovating and maintenance of buildings. In practice, the geometric information is edited within a CAD-software which stores the digital drawing in a layered structure. Measures (relative geometry) are derived from the drawing (absolute geometry) automatically as a computational result. Alphanumeric data are stored separately. This procedure contrasts with an engineering surveyor's way of thinking: *Geometry* is more than just the position of a drawing element. Geometry depicts position and shape of an object and provides an appropriate reference frame for any other type of building information. For technically reliable *as-built* documentations the (geometric) building survey is very important. The survey of a building consists of observed, not calculated *measures*, implicit or explicit *constraints* such as parallelism and perpendicularity and the *datum* that provides the relation to a coordinate system.

In the thesis at hand a novel method is presented, that reverses the traditional work flow and an engineering surveyor's way of think is expressed in both, data model and software. The process starts with a three-dimensional sketch that defines the topological building structure. Observations, measured between topological elements, are modelled stochastically and are attached to the 3d-modell in the sketch. These observations provide the basis for the calculation of the building's geometry (absolute geometry). This computation uses **Least Squares Adjustment (LSA)**. In a last step the datum, a fixation to the reference frame is specified.

Based on the research of Dr.-Ing. habil. Frank Gielsdorf at the *Chair of Engineering Surveying and Adjustment Techniques* (Technische Universität Berlin, Germany) the thesis at hand constitutes a new conceptual and logical datamodel. The central idea of this approach is, that the survey is not only stored for data capture but is an integral part of the building model throughout the whole life-cycle. The datamodel connects relative and absolute geometry via the topological elements of the building structure. Resulting in fewer parameters for depicting the absolute geometry, not point-coordinates but surface-parameters parameterises the geometry. These surface-parameters are calculated on demand (estimated) with a LSA at any time of the workflow. The LSA, as the central algorithm, is designed as a Gauss-Helmert Model and implemented and tested in newly developed software. For the differentiation of the non-linear observation equations and constraints the software uses the powerful technique of Automatic Differentiation, which suits generic programming very good. For the practical work flow it is important to automatically detect invalid elements or relations in the 3d-model and visualise the located inconsistencies to the user. The thesis classifies possible inconsistencies and shows how to present these in diverse software modules.

With the novel method the time consuming work of data acquisition for an as-built documentation can be achieved more efficiently and reliable. Moreover it supports the validation of existing 3d-building models with stochastic analysis.

Inhaltsverzeichnis

Zusammenfassung	3
Abstract	4
Inhaltsverzeichnis	5
1 Einleitung	7
1.1 Motivation und Zielsetzung	7
1.2 Gliederung der Arbeit	8
2 Datenmodell	10
2.1 Grundlagen der Informationsmodellierung	10
2.2 Eingrenzung des Diskursbereiches	12
2.3 Getrennte Spezifikation von Geometrie und Topologie	14
2.4 Parametrisierung der Geometrie durch Ebenen, nicht Punkte	15
2.5 Messwerte und Bedingungen sind Primärdaten	15
2.6 Redundanz	16
2.7 Modellierung von Körpern	17
2.8 Topologie	19
2.9 Topologische Primitive	20
2.10 Topologische Normalisierung	20
2.11 Geometrische Normalisierung	24
2.12 Redundanzvergleich bei Minimalkonfiguration	28
2.13 Beobachtungen und Bedingungen	29
2.13.1 Inverse Darstellung durch Beobachtungen	29
2.13.2 Abstände zwischen topologischen Primitiven	30
2.13.3 Lokale Polarkoordinaten auf topologischen Primitiven	31
2.13.4 Winkelbedingungen zwischen topologischen Primitiven	32
2.13.5 A priori Differenzen zwischen Abständen und Normalenvektoren	32
3 Funktionales Modell	34
3.1 Mathematisch-Geometrische Bedingungen	34
3.2 Abstände zwischen topologischen Primitiven	36
3.3 Lokale Polarkoordinaten auf topologischen Primitiven	39
3.3.1 Inkrementelle Nährungsberechnung	39
3.3.2 Gemeinsame Ausgleichung	40
3.4 Winkelbedingungen zwischen topologischen Primitiven	41
3.5 A priori Differenzen zwischen Abständen und Normalenvektoren	42
3.6 Datumsfestlegung	42
3.6.1 Teilspurminimierung auf Objektgeometrie (freies Datum)	43
3.6.2 Regularisierung mit Identitätsbeobachtungen (weiches Datum)	45
3.6.3 Festhalten einzelner Parameter (starkes Datum)	45

4	Automatisches Differenzieren	46
4.1	Differentiationsarithmetik	46
4.2	Beispiel	48
4.3	Vorwärts oder Rückwärts?	48
4.4	Erweiterung der Differentiationsarithmetik	51
4.5	OO-Implementierung	52
5	Ausgleichsrechnung	54
5.1	Parameterschätzung mit dem Gauß-Helmert Modell	54
5.2	Effiziente Berechnung der Parameter und statistischen Größen	56
5.2.1	Parameternetz und indizierte Gradienten	56
5.2.2	Sequentielle Berechnung der Normalgleichungsanteile und Redundanzanteile	57
6	Validierung der Daten	59
6.1	Begründung und Anwendung der Axiome	59
6.2	Syntaktische Integritätsbedingungen	60
6.3	Konsistenz mit dem Datenmodell	61
6.4	Topologische Integritätsbedingungen	61
6.5	Geometrische Integritätsbedingungen	64
6.6	Integritätsbedingungen für die Ausgleichsrechnung	65
7	Validierung der Methode	67
7.1	Anwendungsfall Ersterfassung	67
7.1.1	Vorbedingungen	67
7.1.2	Skizze (Schritt 1)	67
7.1.3	Skizze validieren (Schritt 2)	68
7.1.4	Exportieren und Normalisieren (Schritt 3)	69
7.1.5	Messen (Schritt 4)	70
7.1.6	Ausgleichsrechnung (Schritt 5)	70
7.1.7	Bewertung des Anwendungsfalls	72
7.2	Anwendungsfall Sekundärdatenerfassung	74
7.2.1	Transformation CityGML \rightarrow Popa3d (Schritt 1)	74
7.2.2	Ausgleichsrechnung (Schritt 2)	76
7.2.3	Transformation Popa3d \rightarrow CityGML (Schritt 3)	77
7.2.4	Bewertung des Anwendungsfalls	77
8	Zusammenfassung und Ausblick	79
	Literaturverzeichnis	81
	Abbilungsverzeichnis	83
	Tabellenverzeichnis	85
	Anhang	87
A	UML-Klassendiagramme	87
A.1	Abstände zwischen topologischen Primitiven	87
A.2	Lokale Polarkoordinaten auf topologischen Primitiven	88
A.3	Winkelbedingungen zwischen topologischen Primitiven	89
A.4	A priori Differenzen zwischen Abständen und Normalenvektoren	90
B	Liste aller Gleichungsobjekte	91
	Lebenslauf	92

1 Einleitung

1.1 Motivation und Zielsetzung

Die herkömmlichen Methoden und Datenmodelle der in der praktischen Arbeit verwendeten CAD Software (Computer Aided Design) widersprechen einer ingenieurgeodätischen Denkweise: Die Arbeit in einer CAD Software beginnt mit der Definition eines Koordinatensystems. Anschließend wird die Geometrie des darzustellenden Objekts gezeichnet (konstruiert). Mittels „assoziativer“ Bemaßung können Relativmaße von der CAD-Software berechnet und im Plan angezeigt werden. Durch Gruppenbildung aus Zeichnungselementen kann eine Topologie erzeugt und damit Nachbarschaftsbeziehungen explizit spezifiziert werden.

In CAD werden die geometrischen Eingangswerte stets deterministisch, also ohne stochastisches Modell verstanden. Maße können sich nicht gegenseitig kontrollieren, weil redundante Beobachtungen bzw. Bedingungen nicht erlaubt sind. Das CAD Programm speichert das Ergebnis der Vermessung, nicht die Vermessung selbst. Parameterschätzung und Qualitätskontrolle einer Vermessung wird nicht im CAD Program ausgeführt. Software, die speziell für Vermessungszwecke entwickelt ist, bietet zwar viele Möglichkeiten, das Aufmaß auszuwerten, unterstützt aber nicht eine geeignete Modellierung der topologisch-geometrischen Gebäudestruktur.

In der vorliegenden Arbeit wird eine neuartige Methode vorgestellt, die den Arbeitsprozess umkehrt und eine ingenieurgeodätische Denkweise verwirklicht: Zunächst wird die dreidimensionale topologische Gebäudestruktur mit einer Skizze festgelegt. Relativmaße, die aus stochastisch modellierten Beobachtungen (Relativgeometrie) entstehen und an die Skizze „geheftet“ werden, sind die Grundlage für die Berechnung der Geometrie des darzustellenden Objektes (Absolutgeometrie). Für diese Berechnung wird das Verfahren der Ausgleichsrechnung verwendet. Die Festlegung des endgültigen Koordinatensystems erfolgt als letzter Arbeitsschritt.

Das digitale Modell eines Gebäudes ist ein zweckmäßig vereinfachtes Abbild der geplanten oder gebauten Situation. Der Zweck eines Informationsmodells ist die Entscheidungsunterstützung des Nutzers. Gebäudeinformation wird vielseitig genutzt: Ingenieure aus Industrie, Verwaltung und Wissenschaft entwerfen, berechnen, bauen und bewirtschaften Gebäude mit Hilfe anwendungsbezogener Computerprogramme, die gebäuderelevante Daten erzeugen, verwalten, analysieren und präsentieren. Dreidimensionale Stadtmodelle, die zunehmend auch den Innenraum einzelner Gebäude beschreiben, werden von Stadtplanern, Touristen oder im Katastrophenschutz verwendet. Die Vielseitigkeit der Anwendungsgebiete spiegelt sich in der Vielseitigkeit der Mathematischen Modelle, Algorithmen, Methoden, Computerprogramme und Softwarearchitekturen.

In der Vielseitigkeit der Anwendung von Gebäudeinformationsmodellen lässt sich ein gemeinsames Problem identifizieren: Die Beschaffung und Fortführung der Daten. Besonders aufwendig ist die Erfassung, Validierung und Fortführung von Daten zur Topologie und Geometrie der Gebäudestruktur. Die Bedeutung der Datenerfassung für das Computer Aided Facility Management (CAFM) betont [Opić 2006, S. 198]: „Die Erfassung der Gebäudegrundrisse und der Ausstattungsdaten stellt den größten Kostenblock bei der Einführung von CAFM dar. Bis zu 75% der Kosten können hierauf entfallen“.

Warum ist die Bestandsdokumentation der Gebäudestruktur so teuer? Warum wird häufig auf die Innenraumvermessung verzichtet? Die beim Aufmaß verwendete Software unterstützt das Aufmaß nur unzureichend. Inkonsistenzen zwischen den Zeichnungselementen werden erst beim nachträglichen Zeichnen festgestellt, Nachmessungen müssen erfolgen, oder Fehler werden im schlimmsten Fall gar nicht aufgedeckt. Typische Fehler sind Widersprüche in Kontrollmaßen, fehlende Bestimmungsmaße, Etagen liegen nicht übereinander oder Teilmodelle passen nicht ins Ganze (Abbildung 1.1). Konstruktion und Nachbearbeitung sind deshalb zeitintensiv und damit teuer. Bei der Sekundärdatenerfassung werden vorhandene, analoge und digitale Pläne sowie Gebäudesachdaten erfasst und aufbereitet. Aufgrund veralteter Unterlagen, mangelhafter Zeichnungsqualität und ungeeigneter Datenformate treten bei diesem Arbeitsprozess Integrationsprobleme auf.

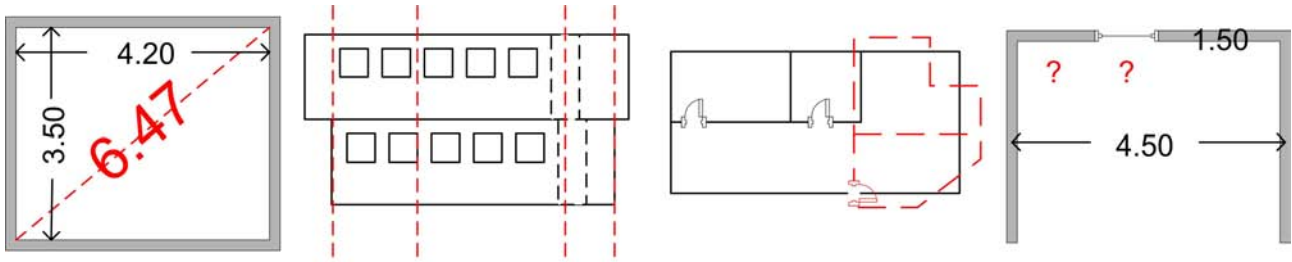


Abbildung 1.1: Typische Fehler bei der Bestandsdokumentation

Motiviert durch die zunehmende Bedeutung technisch belastbarer Gebäudeinformation und die grundsätzlichen Mängel existierender Software wurde in [Gründig und Gielsdorf 2002] ein erstes Datenmodell veröffentlicht, das eine effiziente Datenerfassung und Fortführung der dreidimensionalen Gebäudestruktur erlaubt. Die Leitidee dieses Ansatzes besteht darin, dass die Vermessung nicht nur in einem Vorverarbeitungsschritt, sondern gemeinsam mit der Gebäudestruktur über den gesamten Lebenszyklus des Gebäudes gespeichert wird. Die Parameter der Gebäudegeometrie sind keine Punktkoordinaten sondern Flächenparameter und können zu jedem Zeitpunkt mittels Ausgleichsrechnung berechnet werden. Die Habilitationsschrift „*Ausgleichsrechnung und raumbezogene Informationssysteme*“ [Gielsdorf 2007] entwickelt die theoretischen Grundlagen weiter und bildet die Grundlage der vorliegenden Arbeit.

Zum Zweck der einfacheren Kommunikation und Herausstellung der methodischen Besonderheit bekommt das in dieser Arbeit erweiterte Informationsmodell den Namen *Popa3d*, der sich aus den in [Gielsdorf 2007] entwickelten Forderungen der Flächenparametrisierung, Primärdatenhaltung und Ausgleichsrechnung herleitet. *Popa3d* ist die Abkürzung für „**P**lanes from **O**bservations in a **P**robabilistic data model made for the **A**djustment of **3d** building models“.

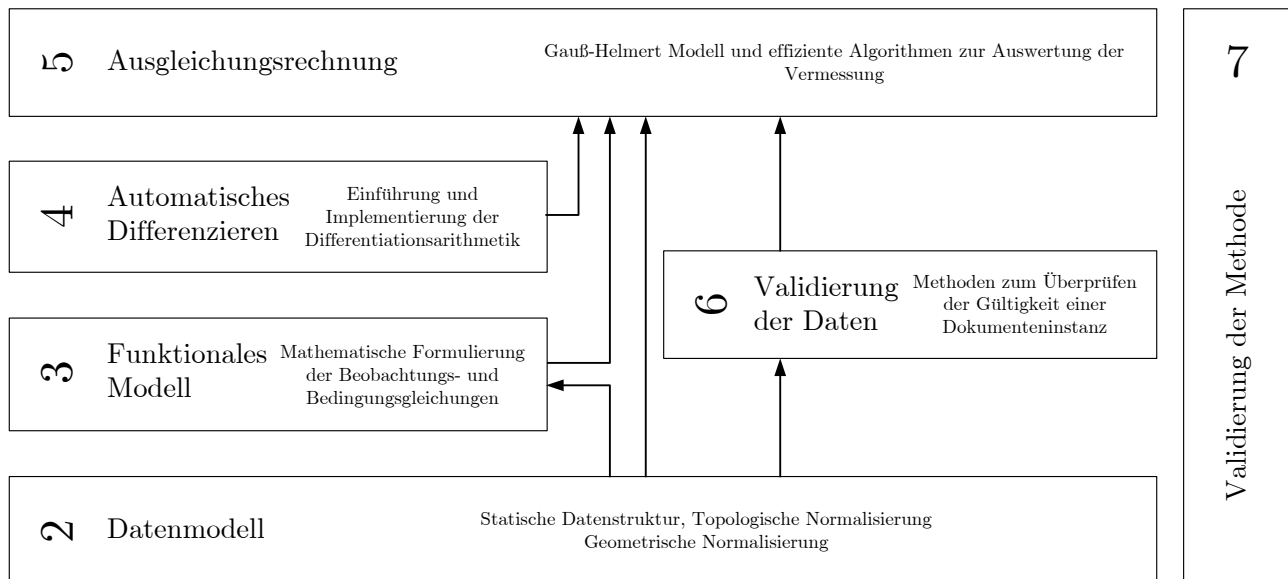
Der methodische Schwerpunkt der vorliegenden Arbeit liegt darin, ein theoretisch fundiertes Modell zu verwirklichen. Das Konzept wird realisiert, indem Datenschnittstellen, Berechnungsalgorithmen und graphische Benutzerschnittstellen entworfen und implementiert werden, das ingenieurgeodätische Wissen in einen informationstechnischen Zusammenhang gestellt und das Erreichte an realen Daten und Arbeitsprozessen getestet wird. Im Sinne eines „proof of concept“ beweisen die Ergebnisse dieser Arbeit die praktische Relevanz der diskutierten Forschungsarbeiten.

Die motivierenden Anwendungsfälle (Kapitel 7) dieser Arbeit sind die *Ersterfassung* und die *Sekundärerfassung* einer Gebäudestruktur. Die *Ersterfassung* einer Gebäudestruktur ist die typische Aufgabe eines Vermessungsingenieurs. *Popa3d* beschleunigt die Vermessung vor Ort und ermöglicht, aufgrund der zentralen Rolle der Ausgleichsrechnung, eine Kontrolle bzw. Bewertung des Vermessungsergebnisses. Der Anwendungsfall *Sekundärerfassung* zeigt, wie die in dieser Arbeit entwickelten Modelle, Methoden und Softwaremodule auch im Zusammenhang mit semantischer Information genutzt werden können.

Die grundlegende neue Arbeitsweise von *Popa3d* ist darüberhinaus auch für andere Anwendungsgebiete geeignet, um verbesserte Methoden zu entwickeln: Die Reihenfolge Topologie → Relativbemaßung → Ausgleichsrechnung → Objektgeometrie kann zum Beispiel auch beim Entwurf und der Planung von Bauwerken angewendet werden. Insbesondere bei bauteilorientierten Datenmodellen im Bauwesen ist ein gemeinsames topologisches Grundgerüst entscheidend für eine erfolgreiche Verbindung unterschiedlicher Fachmodelle [Krämer 2007b]. Bei der Planung der Objektgeometrie spielen Relativmaße, zum Beispiel die Seiten eines Fertigbauteils, eine wichtige Rolle und könnten auch hier als Primärdaten für die Berechnung der Absolutgeometrie gespeichert werden.

1.2 Gliederung der Arbeit

Abbildung 1.2 zeigt die Gliederung der Arbeit. In **Kapitel 2** wird das *Popa3d* Datenmodell motiviert und entwickelt. Dabei werden zunächst die Anforderungen an ein konzeptionelles Datenmodell formuliert, das eine effiziente Erfassung und Validierung von Gebäudemodellen ermöglicht. Zur Vermeidung einer redundanten Datenhaltung und in Analogie zur Datenbanktechnik wird die Abbildung der Geometrie und der Topologie

Abbildung 1.2: *Inhalt der Arbeit*

schrittweise normalisiert.

Die statische Datenstruktur des *Popa3d* Informationsmodells umfasst die Modellierung der Gebäudetopologie, Gebäudegeometrie und Beobachtungstopologie und bildet die Beziehungen zwischen den Objekttypen ab. Das Funktionale Modell der Beobachtungs- und Bedingungsgleichungen, die in **Kapitel 3** definiert werden, beschreibt die geometrischen Beziehungen zwischen den Werten Beobachtungen und der Objektgeometrie. **Kapitel 4** stellt die allgemeine Methode der Differentiationsarithmetik vor, mit der in generischer Software schnell und robust differenziert werden kann. Die mit dem Verfahren des Automatischen Differenzierens gebildeten Ableitungen werden in den Funktionalmatrizen des Gauß-Helmert Modells für die Ausgleichsrechnung benötigt. In **Kapitel 5** werden kurz die wichtigsten Formeln der Ausgleichung mit dem Gauß-Helmert-Modell angegeben und dann gezeigt, wie die mit *Popa3d* modellierte Gebäudeinformation rechentechnisch effizient ausgeglichen werden kann. Axiome für die Validierung einer Dokumenteninstanz des Gebäudemodells werden in **Kapitel 6** entwickelt. Mittel der Integritätsprüfung sind neben allgemein informationstechnischen Verfahren und den Verfahren Geographischer Informationssysteme (GIS) auch die Ausgleichung selbst, die mit statistischen Methoden die geometrische und vermessungstechnische Gültigkeit eines Dokuments prüft. **Kapitel 7** beschreibt und bewertet die neu entwickelte Methode an den zwei Anwendungsbeispielen *Ersterfassung* und *Sekundärdatenerfassung*. Die Arbeit schließt mit einer Zusammenfassung und einem Ausblick in **Kapitel 8**.

2 Datenmodell

2.1 Grundlagen der Informationsmodellierung

Die Informationsmodellierung wird mit einer (Meta-) Sprache, dem Datenmodell, durchgeführt. Das Datenmodell legt Konstrukte fest, mit denen der Diskursbereich beschrieben werden soll. Datenmodelle stellen als Sprache für die Modellierung, *Datentypen* für die Beschreibung von Sachverhalten, *Operatoren* für die Veränderung von Daten und Möglichkeiten zur Definition von *Integritätsbedingungen* zu Verfügung. Die mit Hilfe eines bestimmten Datenmodells vorgenommene Modellierung des Diskursbereiches wird Informationsmodell, oder kurz: Modell, genannt. Diese Arbeit folgt dem allgemeinen Sprachgebrauch, der den Begriff „Datenmodell“ sowohl für das Informationsmodell als auch für die Metasprache, das Datenmodell im eigentlichen Sinne, benutzt.

Die klassische Informationsmodellierung ist in drei Entwürfe gegliedert: Konzeptionelles, logisches und physikalisches Datenmodell.

Nach der Eingrenzung des Diskursbereiches wird in der ersten Phase des Entwurfs ein möglichst implementierungsunabhängiges **konzeptionelles Modell** (auch semantisches Modell) erstellt. Die Aufgabe des konzeptionellen Modells ist die Benennung semantischer Einheiten und deren Strukturierung. *ER-Diagramme* (engl. *Entity-Relationship*, auch *Chen-Diagramme*) beschreiben das konzeptionelle Modell mit den graphischen Zeichenelementen Kreis, Rechteck, Raute, Linie und Pfeil einfach und sind intuitiv, auch für Nichtinformatiker, verständlich (Abbildung 2.1).

Moderner, ausdrucksstärker und formalisierter ist die Modellierungssprache *UML* (engl. *Unified Modeling Language*). Die UML kennt Verhaltensdiagramme (z.B. Aktivitäts-, Use-Case-, Zustands- oder Sequenzdiagramme) und Strukturdiagramme zur Abbildung der statischen Systemstruktur (z.B. Klassen-, Komponenten- oder Objektdiagramme). UML-Klassendiagrammen werden zur graphischen Darstellung der statischen Struktur eines objektorientierten Entwurfs verwendet.

UML Klassendiagramme (Abbildung 2.2) beschreiben:

- eine Klasse mit: Klassenname (erstes Feld), Attribute mit Name, Typ und Sichtbarkeit (zweites Feld) sowie Methodensignatur (drittes Feld) mit Modifikationen (Sichtbarkeit, Eigenschaft).
- Spezialisierungsbeziehungen zwischen Klassen (Implementierungs- und Schnittstellenvererbung)
- Relation vom Typ „Assoziation“ als allgemeine fachliche Bindung zwischen Objekten bestimmter Klassen
- Relation vom Typ „Aggregation“ als Spezialfall der Assoziation (Teil-Ganzes-Beziehung ohne Abhängigkeit des Teils vom Lebenszyklus des Ganzen)

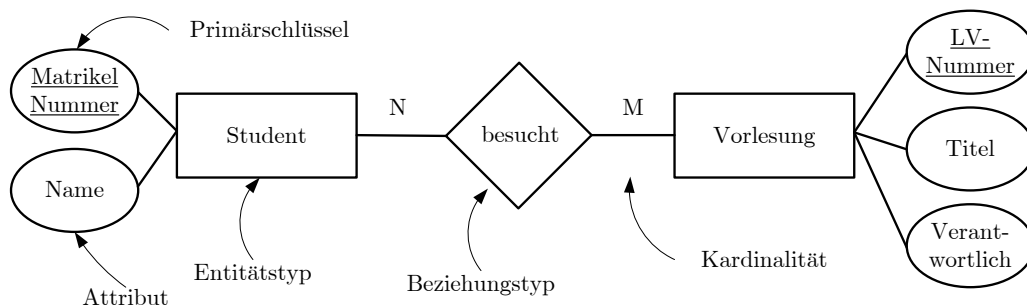


Abbildung 2.1: Entity Relationship Diagramm

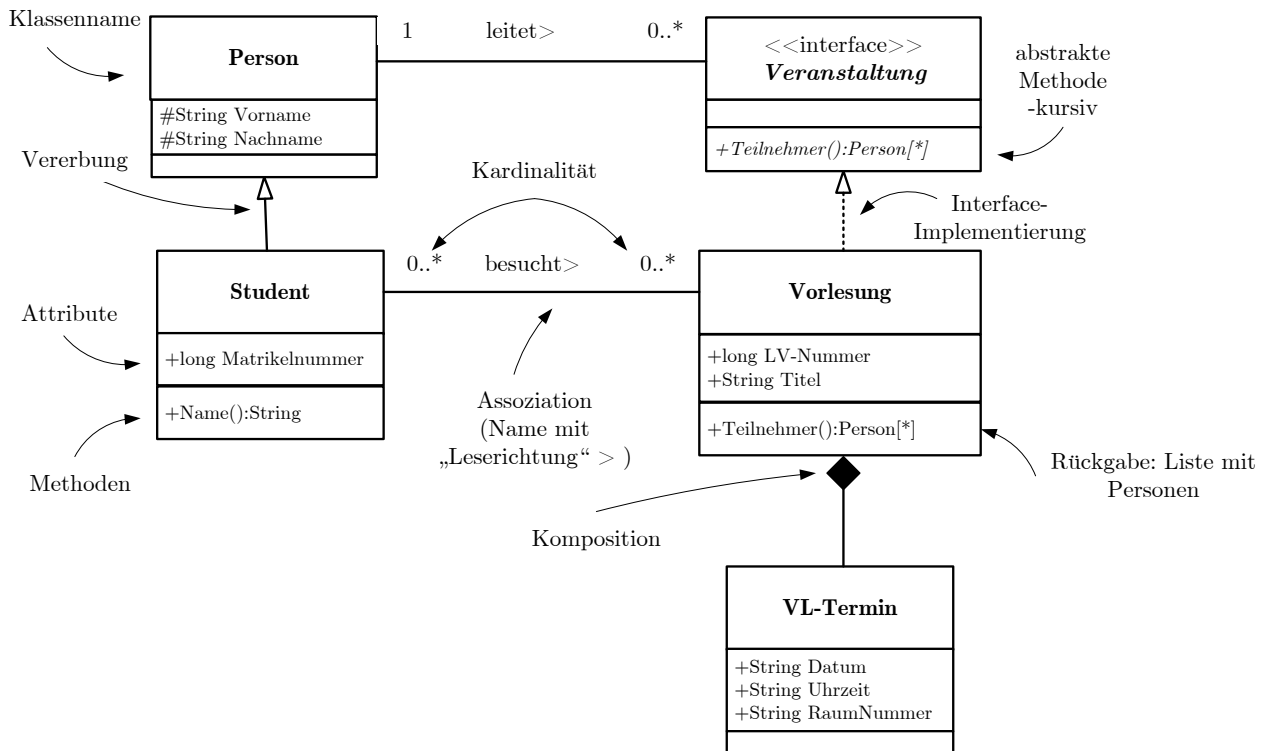


Abbildung 2.2: UML (Unified Modeling Language) Beispiel Diagramm mit Erläuterung

- Relation vom Typ „Komposition“ als Spezialfall der Aggregation (Teil-Ganzes-Beziehung mit Abhängigkeit des Teils vom Lebenszyklus des Ganzen)
- Kardinalität (optionale, einfache oder mehrfache Bindung)
- Schnittstelle (Sammlung von Methoden und statischen Konstanten)

Die zweite Entwurfsphase der Informationsmodellierung ist die Überführung des konzeptionellen Modells in ein **logisches Datenmodell**. Logische Datenmodelle sind zum Beispiel relationale Modelle für Datenbanken, objektorientierte Modelle oder XML-DTD bzw. XML-Schema. Die Wahl des logischen Datenmodells ist applikationsabhängig. In dieser Arbeit wird XML-Schema verwendet, weil die verschiedenen Softwareprogramme mit XML-Dokumenten Daten austauschen und mittels XML-Schema hierbei eine automatisierte Validierung der Dokumentenstruktur (siehe Kapitel 6) vom XML-Parser durchgeführt werden kann.

Da XML-Schema relativ komplex ist, empfiehlt [Kazakos u.a. 2002] XML-Schema in vier Schichten (Abbildung 2.3) einzuteilen:

Das Schichtenmodell ist in diesem Zusammenhang besonders geeignet, weil es ausdrückt, dass die obere nicht ohne die untere Schicht bestehen kann, aber die untere Schicht auch ohne die obere auskommen kann. Zum Beispiel können Datentypen, die in einem XML-Schema definiert werden, auch in anderen Zusammenhängen benutzt werden aber Fremdschlüssel nur auf Instanzen bekannter Typen verweisen.

XML-Schema stellt eine Menge von vordefinierten einfachen **Datentypen** zu Verfügung. Die wichtigsten sind: `boolean`, `String`, `anyUri` (z.B. URL), `long`, `short`, `negativeInteger`, `float`, `double`, `date`, `time`, `gMonth`, `hexBinary`, `base64binary`. Die einfachen Datentypen können genutzt werden, um neue Datentypen zu definieren. Dabei sind diese neuen Datentypen Einschränkungen (**restriction**) der Basistypen. Die Einschränkung kann bei allen Datentypen so vorgenommen werden, dass der Wertebereich als feste Menge angegeben wird (**enumeration**), die maximale/minimale Länge einer Zeichenkette angegeben wird (`minLength`, `maxLength`), der Wertebereich von numerischen Werten durch Ober- und Untergrenzen festgelegt wird (`maxInclusive`, `minInclusive`...) oder die Anzahl der Nachkommastellen bestimmt wird (`fractionDigits`).

Wiederverwendung	Schemadokumentation Modularisierung Erweiterung
Konsistenz	Eindeutigkeit Schlüssel Fremdschlüssel
Struktur	Komplexe Datentypen Kardinalität Elementdeklaration
Datentypen	Einfache Datentypen

Abbildung 2.3: XML-Schema Schichtenmodell. Nach [Kazakos u.a. 2002]

Zur Beschreibung der **Struktur** des logischen Datenmodells werden in XML-Schema Datentypen deklariert und Beziehungstypen zwischen bestimmten Objekten eines Datentyps definiert. Jedes Datenelement besteht aus Name, Typ und Wert. Der Wert eines Elementes wird im Regelfall in der *Dokumenteninstanz* beschrieben, während Name und Typ in der *Schemadefinition* einander zugeordnet werden. Das gleiche gilt für Attribute. Will man Datentypen mehrfach verwenden, ist es sinnvoll, zunächst einen komplexen Datentyp zu deklarieren und dann in einem zweiten Schritt ein Element dieses Datentyps zu definieren. Die Trennung von Definition und Deklaration trägt auch zur Übersichtlichkeit einer Schemadefinition bei.

Bei der Typkonstruktion können die Kindelemente auf verschiedene Arten geschachtelt werden. Der Typ **sequenzen** gibt an, dass alle Elemente in genau der angegebenen Reihenfolge im Instanzdokument aufgeführt werden müssen. Alternativ können die Konstruktoren **all** oder **choice** verwendet werden. Bei **all** müssen alle Elemente vorkommen, wobei die Reihenfolge egal ist. Der Konstruktor **choice** gibt an, dass im Instanzdokument nur eines der angegebenen Kindelemente auftauchen darf.

Die *Kardinalität* wird über die XML-Schema Attribute **minOccurs** und **maxOccurs** festgelegt. Standardmäßig sind die Standardwerte mit **minOccurs=1** und **maxOccurs=1** gesetzt, so dass jedes Attribut genau einmal im Datenelement erscheint. Mit XML-Schema können, wie bei objektorientierten Programmiersprachen Typen von Typen abgeleitet (**extension**) werden. Um zu garantieren, dass ein Typ nur in seiner spezialisierten Form, und nicht der Elterntyp instanziiert werden kann, wird er als **abstract** definiert.

Zur Spezifikation von **Konsistenz**bedingungen verwendet XML-Schema die Sprachelemente **unique**, **key**, **keyref**. Soll sichergestellt werden, dass ein Element nur einmal vergeben wird, muss es als **unique** definiert werden. Das aus dem Datenbankumfeld bekannte Begriffspaar Schlüssel (**key**) und Fremdschlüssel (**keyref**) sind in XML-Schema aufgenommen worden. Die Definition eines Schlüssels enthält implizit die **unique** Definition des entsprechenden Elementes. Zusätzlich wird der Parser aber angewiesen zu überprüfen, ob der durch **keyref** referenzierte **key** auch wirklich vorhanden ist.

Die dritte und letzte Entwurfsphase der Informationsmodellierung befasst sich mit dem **physikalischen Datenmodell**. Ziel dieser produktabhängigen Entwurfsphase ist eine hohe Performanz des Informationssystems. Es wird zum Beispiel über die Art des Datenbanksystems (relationale-, objektrelationale oder XML-Datenbank) entschieden und Indexstrukturen festgelegt.

2.2 Eingrenzung des Diskursbereiches

Jede *Informationsmodellierung* beginnt mit der Festlegung des *Diskursbereiches*. Die Festlegung des Diskursbereiches kann im Laufe eines Projekts noch geändert werden. Wichtig ist zu Beginn die Frage, ob man eine statische (was bleibt gleich?) oder dynamische Sicht (was ändert sich?) auf den Diskursbereich einnimmt. Die

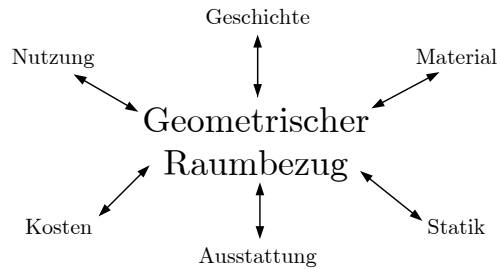


Abbildung 2.4: Objektgeometrie als Ordnungsrahmen [Weferling 2002]

vorliegende Arbeit behandelt die statische Datenstruktur, die in allen Dokumentinstanzen gleich ist. Der Diskursbereich des in dieser Arbeit diskutierten Informationsmodells ist die **Erfassung eines dreidimensionalen geometrisch-topologischen Ordnungsrahmens** für ein *Gebäudemodell*. Ein Gebäudemodell ist ein digitales Abbild eines konkreten Gebäudes. Gebäudemodelle werden von *Gebäudeinformationssystemen* verwaltet. Gebäudeinformationssysteme sind Softwarepakete zur Erfassung, Verwaltung, Analyse und Präsentation von gebäudebezogenen Daten. Verschiedene Fachanwendungen (Facility Management, Stadtmarketing, Innenraumnavigation, Denkmalpflege, ...) definieren einen bestimmten Diskursbereich. Allen gemeinsam ist der geometrische Raumbezug (Abbildung 2.4).

Das Modell des geometrischen Ordnungsrahmens wird durch Datentypen und Datenstrukturen spezifiziert. Diese Datentypen beschreiben die Lage, Größe und Ausrichtung von Objekten (*Geometrie*) und Nachbarschafts- oder Hierarchiebeziehungen von Objekten untereinander (*Topologie*). Die Informationsmodelle der einzelnen Anwendungsdomänen sind nicht Gegenstand dieser Arbeit und werden daher allgemein als *Sachdaten* bezeichnet.

[Zlatanova 2004] beschreiben eine Vielzahl topologischer 3D-Datenmodelle, die in Verwaltung, Industrie und Forschung als konzeptionelle Grundlage räumlicher Software angewendet werden:

- 3DFDS zur Datenhaltung von 3d-Stadtmodellen [Molenaar 1990]
- TEN für geologische Anwendungen [Pilouk 1996]
- SSM für Visualisierungszwecke [Zlatanova 2000]
- UDM zur Speicherung in relationalen Datenbanken [Cooors 2003]
- OO3D als objektorientiertes Modell [Shi u.a.2003]
- u.v.m

Gemeinsam ist allen Modellen, **dass** die räumliche Ausprägung der Objekte (Stadt, Gebäude, Gelände, Erdschichten) der zentrale Ordnungsrahmen ist. Die verschiedenen Anwendungen unterscheiden sich allerdings sehr stark in der Art und Weise, **wie** Geometrie und Topologie modelliert sind.

Die Wahl des dreidimensionalen Datenmodells ist also stark abhängig von den Aufgaben, die die Software, das Gebäudeinformationssystem, vorrangig erfüllen soll. Zum Zweck der Visualisierung muss zum Beispiel eine Datenstruktur verwendet werden, die beim Rendern ein schnelles Navigieren durch Polygone ermöglicht. CAFM-Systeme (engl. *Computer Aided Facility Management*) hingegen benötigen anwendungsabhängige Verknüpfungen zwischen Objektgeometrie und alphanumerischen Sachdaten. Ein Modell, das in einem Aufmaßsystem verwendet wird, spezifiziert Datentypen zum Speichern von geometrischen Messwerten und Bedingungen.

Ein **zweckmäßiges geometrisch-topologisches Informationsmodell** einer Software, die vorrangig der *Erfassung und Validierung* dreidimensionaler Gebäudestrukturen dient, hat nach [Gründig und Gielsdorf 2002] drei Eigenschaften:

1. Getrennte Spezifikation von Geometrie und Topologie (Kapitel 2.3)
2. Parametrisierung der Geometrie durch Ebenen, nicht Punkte (Kapitel 2.4)
3. Messwerte und Bedingungen sind Primärdaten und bilden die Datengrundlage für die Berechnung der Parameter der Objektgeometrie. (Kapitel 2.5)

Im Folgenden werden diese drei, den Diskursbereich definierenden, Forderungen an ein 3D Modell genauer erläutert und deren Zweckmäßigkeit gezeigt.

2.3 Getrennte Spezifikation von Geometrie und Topologie

Die Forderung nach getrennter Spezifikation bedeutet zunächst, dass eine explizite Spezifikation topologischer Information notwendig ist. Das Datenmodell muss demnach die topologischen Primitive Knoten (engl. *node*), Kante (engl. *edge*), Masche (engl. *face*), Körper (engl. *solid*) enthalten und diese in einer geeigneten Datenstruktur miteinander verbinden. Eine explizite Spezifikation der Topologie wird aus folgenden vier Gründen vorgenommen:

- Algorithmische Performanz
- Identifikation einzelner topologischer Elemente der Gebäudestruktur
- Konsistenzsicherung
- Berechnung der Geometrie

Nach [ISO19107:2003] dient eine explizite Spezifikation der Topologie vor allem **der algorithmischen Performanz**:

„The most productive use of topology is to accelerate computational geometry. The method by which this is accomplished is to associate explicitly feature instances and geometric object instances in a manner consistent with and derived from their implicit geometric relations.[...] Geometric calculations such as containment (point-in-polygon), adjacency, boundary, and network tracking are computationally intensive. For this reason, combinatorial structures known as topological complexes are constructed to convert computational geometry algorithms into combinatorial algorithms. Another purpose is, within the geographic information domain, to relate feature instances independently of their geometry. For the first purpose, topology definitions in this clause parallel the structure of the geometric definitions [...]. For the second purpose, the classes in these packages are specified so that they can be used independently of the geometry.“

Eine explizite Spezifikation der Topologie ermöglicht die **Identifikation einzelner topologischer Elemente der Gebäudestruktur** beliebiger Dimension. Dies ist dann besonders wichtig, wenn das Objekt bei einer strukturierten Vermessung (im Gegensatz zur „unstrukturierten“ Vermessung mit Laserscanner) mit geodätischen Messinstrumenten „angezielt“ wird und die Beobachtungstopologie gespeichert wird.

Zur **Konsistenzsicherung** werden die referentielle Integrität und Kardinalität der topologischen Primitive untereinander geprüft. Jedes Einfügen, Ändern oder Löschen von topologischen Elementen muss wieder einen topologisch konsistenten Zustand erzeugen. Weiterhin kann die Konsistenzsicherung mit der Euler-Poincaré-Gleichung (siehe Kapitel 6.4) durch einfaches „Zählen“ aller topologischen Primitive unterstützt werden.

Aus geodätischer Sicht darf die Änderung der Koordinate eines Punktes durch Neuausgleichung, Datumstransformation oder Projektionswechsel nicht zur Verletzung von topologischer Integritätsbedingungen (z.B.: „Flächen berühren sich“) führen. Die explizite Spezifikation der Topologie ermöglicht die Integrität des Datenbestandes im Falle der Fortführung. Explizite topologische Informationen können aus bautechnischer Sicht auch ohne geometrisches Äquivalent für Bauplanung und Kennzeichnung genutzt werden [Huhnt und Gielsdorf 2006].

Als besondere Eigenschaft des *Popa3d* Modells wird bei einer flächenparametrisierten Geometrie (Kapitel 2.4) die Topologie außerdem benötigt, um Schnittbeziehungen zwischen Ebenen anzugeben. In einem flächenpara-

metrisierten Modell ist eine **Berechnung der Geometrie** nur mit Hilfe der topologischen Struktur möglich (Kapitel 2.11). Aus der Topologie wird abgeleitet, welche Ebenen sich in einem bestimmten Knoten schneiden.

2.4 Parametrisierung der Geometrie durch Ebenen, nicht Punkte

In herkömmlichen Informationsmodellen zur Abbildung dreidimensionaler Gebäudestrukturen wird die Objektgeometrie durch Koordinaten von Punkten festgelegt. Solche Modelle werden im Folgenden „punktparametrisiert“ genannt. Es kommt in punktparametrisierten Modellen oft vor, dass eine Ebene über vier oder mehr Punkte beschrieben wird, obwohl zur Definition einer Ebene nur drei nicht in einer Gerade liegende Punkte nötig sind. Durch diese Art der Datenhaltung sind geometrische Aussagen hoch redundant und die Datenbearbeitung (z.B.: Verschieben einer Wand) anfällig für Fehler. Die in dieser Arbeit angewendete Modellierung geht einen anderen Weg. *Popa3d* ist ein „flächenparametrisiertes“ Modell. Die geometrische Information wird an die Ebene „geheftet“ und diese durch einen *Normalenvektor* \vec{n} (Ausrichtung der Ebene) und einen Abstand d zum Koordinatenursprung (Lage der Ebene) festgelegt. Eine Ebene kann mehrere Maschen (Decke, Wand, Seite eines Fensterrahmens,...) tragen. Dadurch werden zum Beispiel Wände, die durch mehrere Stockwerke eines Gebäudes laufen, nur durch ein geometrisches Element, nämlich die Ebene, beschrieben. Eine weitere Reduzierung der Parameterzahl wird durch die Mehrfachverwendung von Parametern erreicht. Ein geeignetes geometrisches Modell wird in den Kapiteln 2.10 „Topologische Normalisierung“ und 2.11 „Geometrische Normalisierung“ entwickelt. Ziel der Normalisierung ist es, das Gebäude mit einer möglichst geringen Anzahl von Geometrieparametern zu beschreiben, denn erst so wird die Anwendung der Ausgleichsrechnung ermöglicht und damit die neue Arbeitsmethodik (Topologie \rightarrow Relativbemaßung \rightarrow Ausgleichsrechnung \rightarrow Objektgeometrie) anwendbar.

Ebenen-Parametrisierte Modelle sind aktueller Forschungsgegenstand: [Thompson 2007] führt die „*regular polytope*“ zur Begründung einer geschlossenen Körperalgebra und deren digitaler Repräsentation ein. Das Modell ist für Abfragen auf großen dreidimensionalen geometrischen Datenbeständen geeignet. Die „*half-space-representation*“ enthält keine explizite B-Rep Darstellung, wie sie für Vermessungszwecke nötig ist. [Tse u.a. 2008] generieren Ebenenparameter aus LIDAR-basierten Voronoi-Diagrammen und nutzen Cluster gleicher Normalenvektoren zur Extraktion von Dachstrukturen in ein CAD-geeignetes Datenmodell.

Die zunehmende Anwendung flächenhaft parametrisierter Modelle wird im Kontext der ingenieurgeodätischen Auswertung von Laserscandaten [Heunecke u.a. 2004] als „*Paradigmenwechsel der Geodäsie*“ bezeichnet.

2.5 Messwerte und Bedingungen sind Primärdaten

Neben der Parametrisierung der Geometrie über Ebenen ist das in dieser Arbeit entwickelte Informationsmodell ein *empirisches* Modell: Es unterscheidet sich von anderen Informationsmodellen zur Abbildung von dreidimensionalen Gebäudestrukturen in der Hinsicht *welche* Daten als *Primärdaten* aufgefasst werden. In GIS/CAD Systemen wird die absolute Geometrie (Koordinaten) als Datengrundlage angenommen. Alle relativen Maße (z.B. Abstand von zwei Mauern) werden daraus berechnet oder als redundante, zusätzliche Aussage gespeichert. [Gielsdorf 2007, S. 45ff] weist darauf hin, dass die Abbildung des Beobachtungsraumes auf den Parameterraum im Falle der Ausgleichung injektiv, nicht aber surjektiv, also nicht umkehrbar ist. In konventionellen CAD-Systemen werden nur die *Koordinaten als Ergebnis der Vermessung* gespeichert und deshalb auf die Modellierung der Beobachtungen verzichtet, da sie sowohl untereinander als auch zu den ausgeglichenen Geometrieparametern redundant sind (Abbildung 2.5). Aus den in einer Zeichnungsdatenbank vorhandenen Parametern der absoluten Geometrie lassen sich zwar „ausgeglichen“ Maße und Maßketten berechnen, nicht aber umgekehrt, die originären Messwerte.

In einem *empirischen Datenmodell* werden, wie in der Geodäsie üblich, die Beobachtungen als Primärdaten aufgefasst. Beobachtungen stellen eine relative geometrische Beziehung zwischen Entitäten des 3d-Gebäudemodells dar. Beobachtungen sind nicht beliebig genau und werden deshalb stochastisch, mittels Residuen, modelliert. Die Einführung redundanter Zufallsgrößen (Beobachtungen) ist eine Voraussetzung, um die Geometrieparameter des Gebäudemodells optimal zu schätzen. Die absolute Geometrie wird als „Sicht“ auf die Beobachtungen verstanden und durch *Ausgleichsrechnung* bestimmt. Dieses streng empirische Verfahren hat folgende praktische

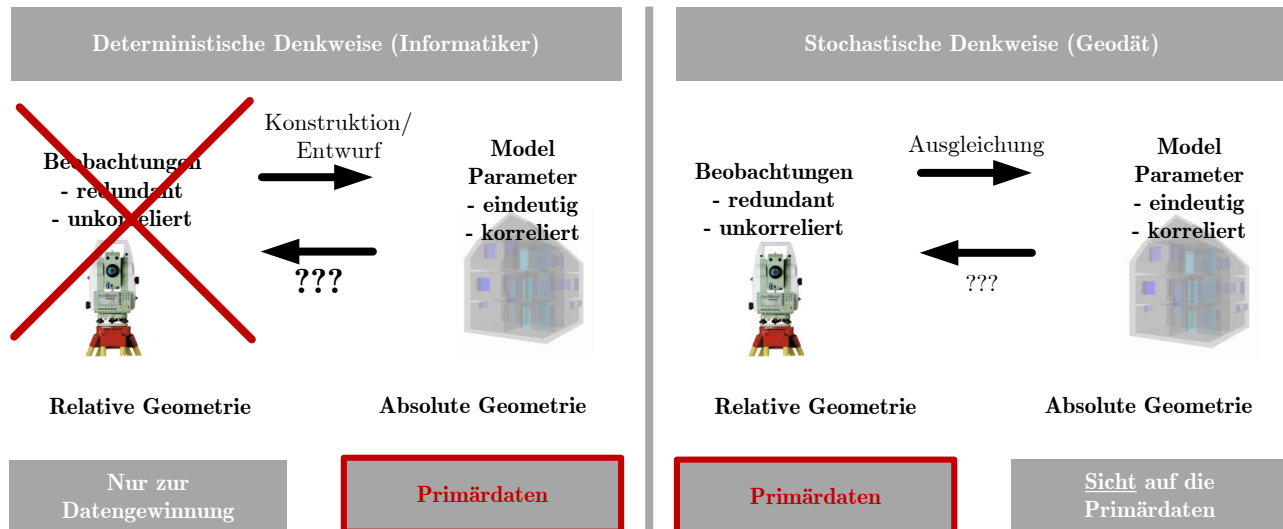


Abbildung 2.5: Beobachtungen als Primärdaten - Die Geodätische Sicht

Vorteile:

- Originale Messdaten gehen nicht verloren
- Die Qualität (Genauigkeit) der Messdaten kann berücksichtigt werden
- Redundante Messungen erhöhen die Genauigkeit und Zuverlässigkeit der absoluten Geometrie
- Fehlerhafte Messungen können detektiert werden
- Fortführungsmessungen bei Veränderung (Verschieben einer Wand) erfordern keinen großen Bearbeitungsaufwand.
- Aussagen über die Qualität der absoluten Geometrie sind möglich

In einigen Arbeiten [Thurrow 2004] des DFG-Sonderforschungsbereichs 524 „Werkstoffe und Konstruktion für die Revitalisierung von Bauwerken“ wird die Bedeutung der geodätischen Messdaten als Primärdaten bereits erkannt und die Ausgleichungsrechnung zur Gewinnung einer dreidimensionalen Objektgeometrie angewendet. Die Objekte werden hier punktparametrisiert dargestellt, während *Popa3d* darüberhinaus die Vorteile der Flächenparametrisierung und Objekttopologie entwickelt und in einem gemeinsamen Datenmodell integriert und nutzbar macht.

2.6 Redundanz

Speichert ein System mehr Information als unbedingt notwendig, spricht man von redundanter Information. In der Geoinformationstechnik wird der Begriff „Redundanz“ sowohl im Kontext von *räumlichen Datenbanken* (SDBMS, engl. *Spatial Data Base Management System*) als auch im Kontext der *Ausgleichungsrechnung* verwendet. Beim Entwurf eines Datenbankschemas versucht man Redundanz zu vermeiden, um beim Einfügen, Ändern und Löschen von Daten stets ein konsistentes Modell zu erzeugen. Empirische Wissenschaften, wie Geodäsie, Photogrammetrie und Ingenieurgeodäsie fordern hingegen redundante Beobachtungen, um unbekannte Parameter zu schätzen und die Vermessung präzise und zuverlässig auswerten zu können. Die unterschiedliche Konnotation des Begriffs „Redundanz“ – negativ beim Entwurf eines Informationsmodells, positiv im Kontext der Ausgleichungsrechnung – ergibt sich aus unterschiedlichen Typen von Information, die das System speichert.

Beim Datenbankentwurf geht man in der Regel davon aus, dass die zu speichernde Information *deterministisch* ist. Unter der Annahme, dass die Daten fehlerfrei und vollständig präzise sind, erwartet das Modell und die Software eine stets widerspruchsfreie Repräsentation der Realwelt. Deterministische Variablen sollten tatsächlich nur einmal gespeichert werden. Im Kontext der deterministischen Information meint Redundanz „mehrfach

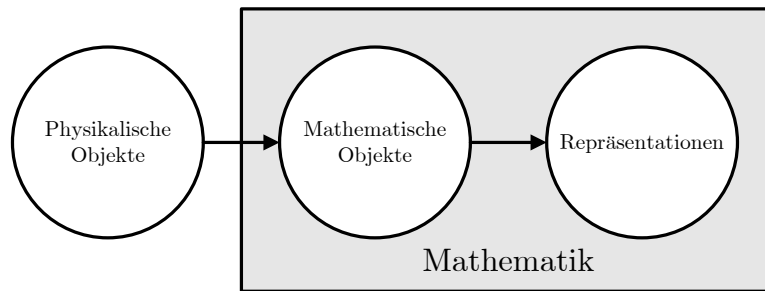


Abbildung 2.6: Drei-Schichten Modellierung nach [Mäntylä 1976, S. 31]

das Selbe speichern“. Modelle, die aus einer Vermessung entstehen, zeichnen sich hingegen durch eine Berücksichtigung der *stochastischen Eigenschaften der geometrischen Information* aus. Aufgrund der einer Messungen innewohnenden Stochastik muss das Modell neben der Messgröße auch Informationen bezüglich der Genauigkeit einzelner Parameter und der Korrelation zwischen Parametern speichern. Auch wenn verschiedene Messwerte die gleiche physikalisch-geometrische Größe beschreiben, *wird jeder redundante Messwert gespeichert*, weil jeder Messwert für sich Gegenstand stochastischer Analyse sein kann. Der Begriff *Redundanz* wird als *statistischer Freiheitsgrad* (engl. *statistical degree of freedom*) verstanden [Mikhail und Ackerman 1976, S. 102].

Die Leitidee dieser Arbeit ist eng mit den beiden Bedeutungen des Begriffs „Redundanz“ verbunden. Das entworfene Datenmodell soll die Integration *redundanter Beobachtungsgrößen* in ein 3d-GIS/CAD Modell ermöglichen. Das „Werkzeug“ Ausgleichsrechnung dient der Parameterschätzung und der Qualitätssicherung. Mittels topologischer (Kapitel 2.10) und geometrischer Normalisierung (Kapitel 2.11) wird eine redundanzarme Repräsentation der Gebäudegeometrie erreicht. Eine redundanzarme Repräsentation der unbekannten Parameter der absoluten Geometrie ist für die Algorithmen der Ausgleichsrechnung geeignet, weil so die Anzahl der zu schätzenden Parameter reduziert, und damit die Dimension des zu lösenden Gleichungssystems verkleinert wird.

2.7 Modellierung von Körpern

Die Begriffe *Geometrie* und *Topologie* sind in raumbezogenen Informationsmodellen miteinander verknüpft. Geometrische Daten beschreiben die Lage, Ausrichtung und Form von Objekten. Topologische Daten spezifizieren Nachbarschaftsbeziehungen, die invariant gegenüber geometrischen Transformationen sind. Spricht man von einem *Geometriemodell*, meint man meistens das geometrisch-topologische Informationsmodell. Die Modellierung erfolgt nach [Mäntylä 1976, S. 30f] in drei Schritten (Abbildung 2.6).

Diese physikalischen Objekte der realen Welt sollen in einer computergeeigneten Datenstruktur repräsentiert werden. Dafür werden sie zunächst in einen mathematisch definierten Objektraum (Punktmengentopologie, algebraische Topologie, Geometrie) überführt.

In [Gielsdorf 2007, S. 8] wird die Notwendigkeit topologischer Modellierung aus folgender Eigenschaft des dreidimensionalen Euklidischen Raums \mathbb{R}^3 abgeleitet: Der Definitionsbereich der Merkmalsklassen (x, y, z) des metrischen Raums ist unendlich, überabzählbar und offen. Es ist unmöglich alle Punkte dieses Raums unstrukturiert abzubilden. Eine Untermenge des \mathbb{R}^3 , die ein Objekt beschreibt, kann nur mit Hilfe einer Abbildungsvorschrift gespeichert werden. Geometrisch-topologische Informationsmodelle unterscheiden sich in der geometrischen Parametrisierung und topologischen Struktur der Abbildungsvorschrift.

[Mäntylä 1976, S. 56] gliedert die impliziten Repräsentationsformen unendlicher Punktmengen in folgende Klassen:

Dekompositionsmodelle. Die Punktmengen werden durch eine Sammlung einfacher Objekte und „Verklebungsvorschriften“ (engl. *gluing operations*) repräsentiert. Beispielsweise verwendet das *Voxel* Modell Würfel als einfaches Objekt zur Zerlegung des Raums und definiert die „Verklebungsvorschrift“ implizit über eine abgeschlossene Aufzählung (engl. *exhaustive enumeration*). Beim *Octree* Modell wird der Raum in Würfel ungleicher Größe zerlegt und kann daher Regionen mit gleicher Eigenschaft zusammengefasst speichern. BSP Modelle (*Binary Space Partitioning*) zerlegen den \mathbb{R}^3 rekursiv in Halbräume. Octree und BSP Modelle sind besonders für

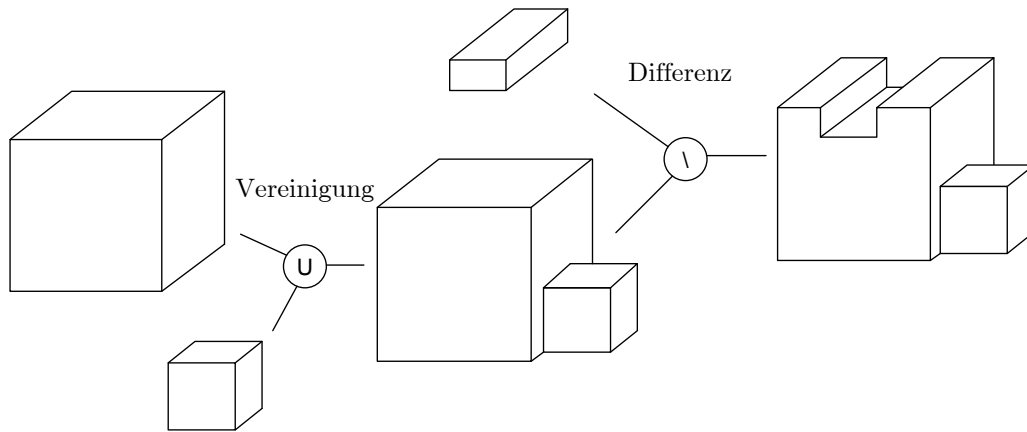


Abbildung 2.7: *Constructive Solid Geometry (CSG) Modell*

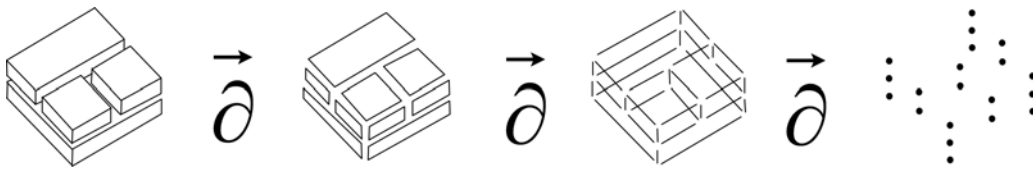


Abbildung 2.8: *Randrepräsentation nach [Bradley und Paul 2009])*

geometrische Suchen und Indizierung geeignet.

Konstruktive Modelle. Die Punktmengen werden ebenfalls durch einfache Objekte zerlegt. Konstruktive Modelle spezifizieren komplexe Konstruktionsoperationen. Man spricht von CSG-Modellen (Constructive Solid Geometry), wenn die Struktur des 3D-Körpers (hier das Bauwerk) durch Verschneidung einfacher, dreidimensionaler Grundkörper beschrieben wird. Die Datenprimitive sind entweder analytische Grundkörper (Quader, Kugel, Zylinder, Kegel, Keil, Torus) oder Profilkörper. Profilkörper entstehen durch Rotation oder Extrusion („Hochziehen“) von Flächen. Komplexe Körper entstehen durch mengentheoretische Operationen (Vereinigung, Differenz, Durchschnitt). Die Reihenfolge der Operationen wird in einer Baumstruktur gespeichert (Abbildung 2.7).

Die Baumstruktur des CSG-Modells führt zu dem Problem, dass die „Entstehungsgeschichte“ immer gemeinsam mit den Grundkörpern gespeichert werden muss. Ändert man zum Beispiel die Breite eines Quaders, müssen alle Verschneidungsoperationen des CSG-Baum neu berechnet werden. Ein weiterer Nachteil ist, dass komplizierte Flächen nicht modelliert werden können. Der Vorteil von CSG-Modellen liegt in der kompakten Formulierung von komplexen Körpern, wenn diese aus wenigen Grundkörpern zusammengebaut werden können. Die CSG-Modellierung korrespondiert beim Entwurf von Bauwerken mit dem menschlichen Denkprozess und ist deshalb zur Abbildung von Bauwerken, und -teilen geeignet, wenn deren Dimension vollständig bekannt ist. CSG-Modelle können beim zerstörungsfreien Aufmaß nicht verwendet werden, weil hier nur sichtbare Teile des Gebäudes mit den Messinstrumenten beobachtet werden können.

Randrepräsentation. Die vektorbasierten Modelle der Kategorie „Randrepräsentation“ (Boundary Representation, B-Rep) strukturieren das 3D-Objekt mit dem Beziehungstyp „Rand“ ∂ (Abbildung 2.8). Komplexe n -dimensionale Gebilde werden indirekt über ihre $(n - 1)$ -dimensionalen Randobjekte beschrieben. Die Struktur, die angibt, wie Objekte niedriger Dimension kombiniert werden, nennt man Topologie.

Die explizite Spezifikation der Topologie ist keine Voraussetzung zur Berechnung topologischer Prädikate. Nachbarschaftsbeziehungen können auch rein geometrisch berechnet werden. Allerdings beschleunigt die topologische Modellierung die Berechnungsgeschwindigkeit wesentlich, da zum Beispiel aufwendige Schnittberechnungen entfallen. Gegenstand der vorliegenden Arbeit sind mathematische Modelle, die eine explizite Spezifikation der Topologie erfordern.

2.8 Topologie

Mathematische Definition. „Eine Menge M kann strukturiert werden, indem bestimmte Teilmengen T von M gegenüber den anderen Teilmengen von M bevorzugt werden. Die Menge dieser bevorzugten Teilmengen T heißt Topologie auf M . Das Gebilde $(M; T)$ heißt topologischer Raum“ [Pahl und Damrath 2000, S. 171]

Definition [Pahl und Damrath 2000, S. 174] : Die Potenzmenge $P(T)$ enthält jede Teilmenge, die in M gebildet werden kann. Eine Teilmenge T der Potenzmenge $P(T)$ heißt **Topologie** auf der Trägermenge M , wenn für die Elemente von T folgendes gilt:

- T1. Die Topologie T enthält die Nullmenge \emptyset und die Trägermenge M $\emptyset \in T \wedge M \in T$
- T2. Jede Schnittmenge von je zwei Elementen A und B von T ist ebenfalls ein Element von T . $A \in T \wedge B \in T \Rightarrow A \cap B \in T$
- T3. Die Vereinigung einer beliebigen Anzahl von Elementen A, B, \dots aus T ist ebenfalls ein Element von T . $A, B, \dots \in T \Rightarrow A \cup B \cup \dots \in T$

In raumbezogenen Informationsmodellen ist M zunächst die Menge aller Punkte des Euklidischen Raums. Teilmengen von M heißen Form. Da M unendlich, überabzählbar und offen ist, kann diese *Punktmengentopologie* nicht im Rechner abgebildet werden. Die Beschreibung der Punktmengen erfolgt daher implizit. Hieraus ergibt sich die Notwendigkeit, die Zerlegung des Euklidischen Raums strukturiert zu beschreiben. „Unter einer Zerlegung einer Menge M [...] versteht man eine Menge paarweise disjunkter nichtleerer Teilmengen von M , deren Vereinigung ganz M ist“ [Jähnich 2005, S. 115]. Teilmengen dürfen sich also nicht überschneiden.

Nach welchen Regeln wird der Raum zerlegt? Zur Beschreibung dieser Regeln wird zunächst der Begriff des „Hausdorff-Raums“ benötigt. Ein topologischer Raum heißt Hausdorff-Raum (oder T_2 -Raum), wenn es für beliebige Punkte $x_1 \neq x_2$ des Raums disjunkte offene Mengen S_1 und S_2 gibt, so dass $x_1 \in S_1$ und $x_2 \in S_2$. Der \mathbb{R}^3 ist ein Hausdorff-Raum, da man für jeden Punkt x eine offene Kugel $\sum_{i=1}^3 x_i^2 < \epsilon$ finden kann, die sich nicht mit einer benachbarten offenen Kugel schneidet. Eine offene Kugel ist eine Kugel ohne Rand. Damit ist die offene Kugel (offener Intervall) eine Umgebung von *jedem* (!) ihrer Punkte.

Eine n -Zelle ist ein Raum, der zu einer offenen Vollkugel im \mathbb{R}^n homöomorph ist. Anschaulich gesprochen bedeutet Homöomorphismus, dass Urbild (n -Zelle) und Bild (Vollkugel) durch eine stetige Abbildung (z.B. Deformation eines Gummituchs ohne Reißen, Spiegeln) umkehrbar ineinander abgebildet werden können. Bei der Änderung der Form dürfen weder Löcher noch Kreuzungspunkte entstehen.

Definition: Ein Hausdorff-Raum X heißt *CW-Komplex* (oder Zell-Komplex), wenn er folgende Bedingungen erfüllt [Kruschwitz 1996, S. 16]:

- CW1: Jeder Punkt des Zellkomplexes gehört zu genau einer Zelle (Zerlegung).
- CW2: Es gibt eine stetige Abbildung, die die offene p -dimensionale Vollkugel auf die p -Zelle abbildet. Das Ergebnis dieser Abbildung ist eine Vereinigung endlich vieler offener Zellen des *CW*-Komplexes mit kleinerer Dimension als p .
- CW3: Eine Teilmenge des *CW*-Komplexes ist abgeschlossen, wenn ihr Durchschnitt mit der abgeschlossenen Hülle einer Zelle jeweils abgeschlossen ist. CW3 ist bei endlicher Zellenanzahl immer erfüllt.

CW1 fordert Durchdringungsfreiheit: Linien dürfen sich nicht schneiden, Flächen sind überlappungsfrei und Körper dürfen sich nicht gegenseitig durchdringen.

Definition: Ein topologischer Raum heißt n -dimensionale **Mannigfaltigkeit** M^n , wenn jeder Punkt von M^n eine zur offenen n -dimensionalen Vollkugel homöomorphe Umgebung besitzt.

Zur Unterscheidung von *CW*-Komplex und Mannigfaltigkeit schreibt [Kruschwitz 1996, S. 16]: „Die Definition der Mannigfaltigkeit fordert Homöomorphie zur Einheits-„kugel“ der jeweiligen Dimension für die Umgebung des Punktes. Die Definition einer Zelle fordert Homöomorphie zur Einheits-„kugel“ für den ganzen topologischen Raum und ist damit strenger als die Definition einer Mannigfaltigkeit“. Abbildung 2.9 verdeutlicht den Unterschied zwischen *CW*-Komplex und Mannigfaltigkeit: Die Homöomorphie zur Einheitskugel gilt in Abbildung 2.9a auf einer 2-Zelle (Masche, engl. face) in Abbildung 2.9b auf dem ganzen Körper, also auch auf den

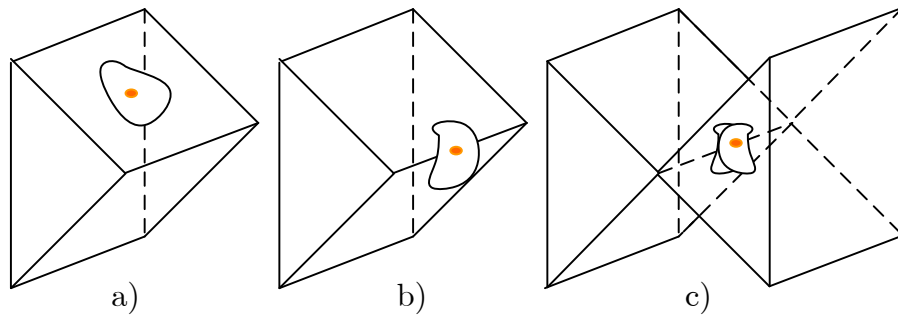


Abbildung 2.9: Homöomorphie und 2-Mannigfaltigkeit nach [Foley u.a. 1995, S. 566]

Knoten und Kanten. Dagegen zeigt Abbildung 2.9c eine Kante, auf der die Homöomorphie zur Einheitskugel nicht bezüglich der Kante gilt.

2.9 Topologische Primitive

Topologische Primitive sind Topologieobjekte, die einzelne, nicht teilbare Elemente eines topologischen Komplexes darstellen (ISO 19107). Topologische Primitive sind offene zusammenhängende Punktmenge. Sie können in den Dimensionen 0 bis 3 mit folgenden Konzepten ausgedrückt werden.

Knoten (engl. *Node*) sind 0-dimensionale Topologieobjekte. Ein Knoten hat keinen Rand. Das Innere eines Knotens ist zugleich offen und abgeschlossen.

Kanten (engl. *Edge*) sind 1-dimensionale Topologieobjekte. Eine Kante wird durch zwei Knoten berandet. Das Innere einer Kante ist eine offene, zusammenhängende Punktmenge. Eine Kante ist eine 1-Mannigfaltigkeit.

Maschen (engl. *Face*) sind 2-dimensionale Topologieobjekte. Eine Masche wird durch eine endliche Menge von Kanten berandet. Der Rand ist geschlossen und selbstüberschneidungsfrei. Das Innere einer Masche ist eine offene, zusammenhängende Punktmenge. Maschen sind 2-Mannigfaltigkeiten. Ist die Masche homöomorph zum offenen Vollkreis, hat sie also keine Löcher, spricht man von einer 2-Zelle.

Topologische Körper (engl. *Topological Solid*) sind 3-dimensionale Topologieobjekte. Ein topologischer Körper wird von Maschen berandet. Die Oberfläche (Rand) ist geschlossen, orientierbar, ohne Selbstüberschneidung und zusammenhängend. Das Innere eines topologischen Körpers ist eine offene, zusammenhängende Punktmenge. Ist der topologische Körper homöomorph zur offenen Vollkugel, spricht man von einer 3-Zelle.

In einer Dokumenteninstanz eines topologischen Datenmodells sind

- die Identifikatoren,
- die Anzahl der topologischen Primitive und
- die Struktur der topologischen Primitive

invariant gegenüber stetigen Abbildungen.

Abbildung 2.10 zeigt dreimal den gleichen topologischen Zustand. Eine unterschiedliche Geometrie ergibt sich zum Beispiel, wenn zunächst eine Skizze (links) angefertigt wird und dann aufgrund einer Vermessung Werte der Objektgeometrie neu berechnet werden (rechts).

2.10 Topologische Normalisierung

Der Begriff *Topologische Normalisierung* wird in dieser Arbeit als *Analogie* zur Datenbanknormalisierung entwickelt: Ziel der Normalisierung im Kontext von Datenbanken ist die Vermeidung von Redundanz. Eine redundanzfreie Datenhaltung ermöglicht eine konsistente Fortführung der Daten, da funktionale Abhängigkeiten

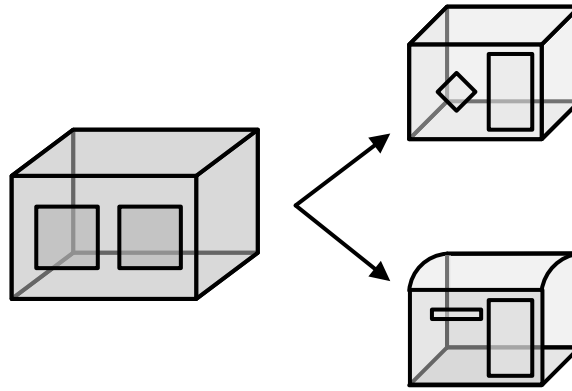


Abbildung 2.10: *Topologie ist invariant gegenüber stetigen Abbildungen*

und Integritätsbedingungen implizit im Informationsmodell spezifiziert sind und durch formalisierte Mechanismen des Datenbankmanagementsystems (DBMS) garantiert werden können. Die hier vorgestellte schrittweise Normalisierung beschreibt den Weg vom topologiefreien Geometriemodell zu einem punktparametrisierten Topologiemodell. Dieser Weg ist sowohl als konzeptionelle Begründung, als auch als Algorithmus zu verstehen.

Die Topologische Normalisierung beginnt mit sogenannten topologiefreien¹ 3d-Modellen. Diese Modelle werden auch „Spaghetti-Modelle“ oder „Polygonsuppe“ genannt. Topologiefreie 3d-Modelle (Abbildung 2.11) speichern zu jeder planaren Masche (engl. *face*) eine Liste mit Koordinatentrippeeln.

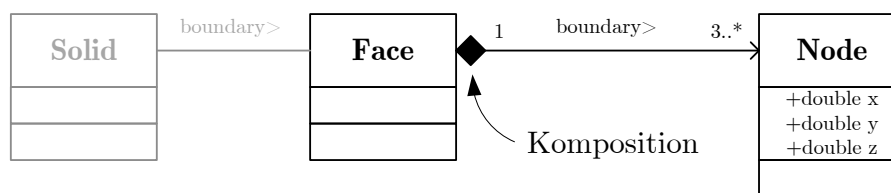


Abbildung 2.11: *Topologiefreies 3D Modell*

Diese Modellierung ist besonders für Visualisierungszwecke geeignet, weil beim Rendern der Polygone auf der Graphikkarte keine geometrischen oder kombinatorischen Berechnungen durchgeführt werden müssen. Auch in Datenmodellen zum Austausch und Verwalten von räumlichen Geodaten werden topologiefreie Modelleelierungen verwendet: Die Auszeichnungssprache Geography Markup Language (GML) [GML 2007, S. 78ff] verwendet die topologiefreie Modellierung raumbezogener Objekte zum Beispiel im Typ `<gml::LinearRing>`. Die objektrelationale Datenbank Oracle-Spatial verwendet Spaghetti-Modelle im `SDO_GEOMETRY` Typ [Oracle 2009, S. (2-5)ff].

Die geometrische Beschreibung ist bei topologiefreien 3d-Modellen augenscheinlich hoch redundant. Es können vier Typen von Redundanz identifiziert werden (Abbildung 2.12):

- Knotenredundanz (Der gleiche Punkt wird mehrfach gespeichert)
- Kantenredundanz (Die gleiche Knotenverbindung (Kante) wird mehrfach gespeichert)
- Loopredundanz (Die gleiche (Loop) Kantenverbindung wird mehrfach gespeichert)
- Maschenredundanz (Die gleiche Masche wird von beiden anliegenden Körpern (engl. *solid*) mehrfach gespeichert)

Zur Vermeidung einer Mehrfachspeicherung der gleichen Koordinate wird jeder Knoten nur einmal in einer Liste oder in einer anderen Containerstruktur gespeichert, identifizierbar gemacht und von der Masche referenziert (Abbildung 2.13).

¹Der Ausdruck *topologiefrei* ist mathematisch nicht korrekt, weil der metrische Raum selber ein topologischer Raum ist. Im Kontext von räumlichen Informationssystemen bezeichnet *topologiefrei* lediglich die Abwesenheit einer explizit spezifizierten Topologie.

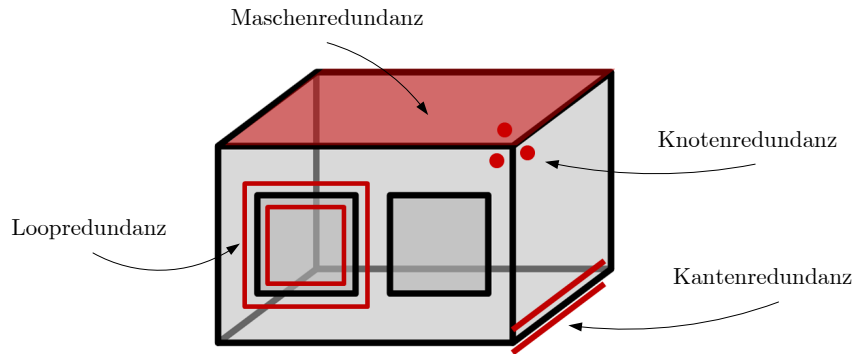


Abbildung 2.12: vier Ebenen der Topologischen Redundanz

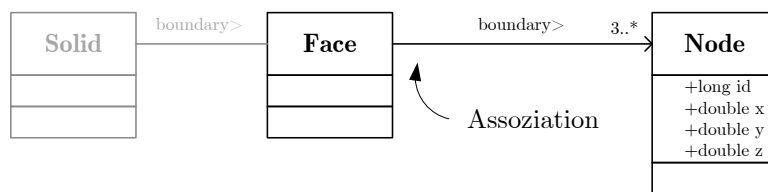


Abbildung 2.13: Vermeidung einer Mehrfachspeicherung von Punktkoordinaten

Der nächste Schritt der Normalisierung beseitigt die Kantenredundanz (Abbildung 2.14). Die Kante wird identifizierbar gemacht und von der Masche als Rand referenziert. Zugleich muss berücksichtigt werden, dass eine Kante gerichtet ist, also muss zwischen Start- und Endknoten unterschieden werden.

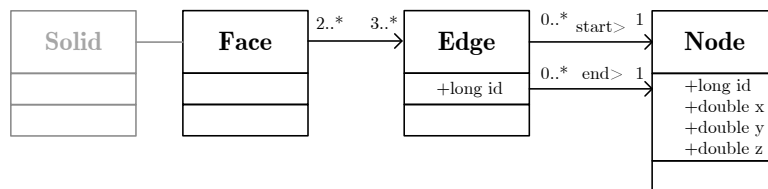


Abbildung 2.14: Vermeidung von Kantenredundanz

Die Kante ist jetzt orientiert und kann referenziert werden. Allerdings wird (unter der Annahme einer 2-Mannigfaltigkeit) jede Kante zweimal gespeichert. Erst mit dem Konzept der Halbkante (Abbildung 2.15) kann die Kantenredundanz unter Beibehaltung gleichsinnig orientierter Kanten beseitigt werden.

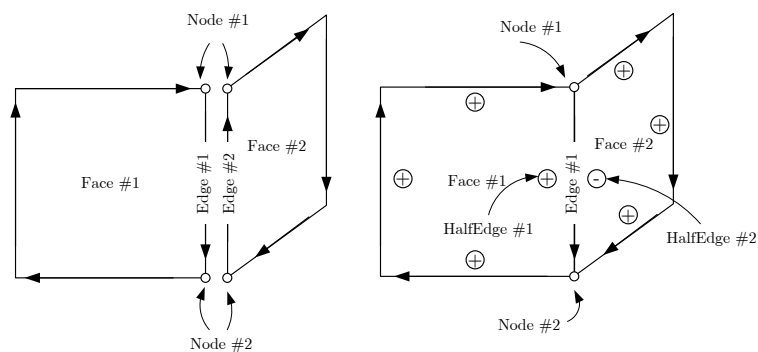


Abbildung 2.15: Das Konzept der Halbkante dient der Beseitigung der Kantenredundanz

Eine Halbkante ist ein Verweis auf eine Kante und eine Vorzeicheninformation, die beschreibt, ob die Halbkante gleich (+) oder entgegengesetzt (-) der Kante orientiert ist (Abbildung 2.16).

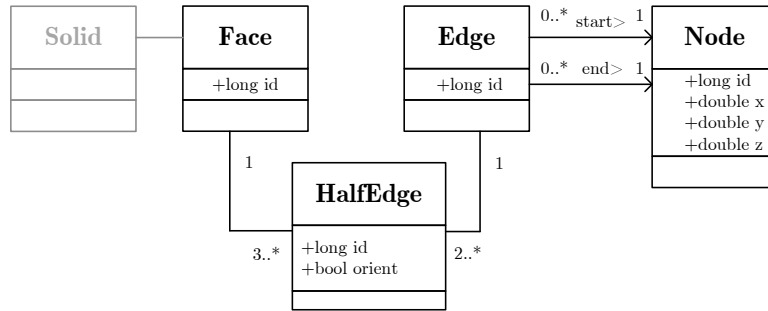


Abbildung 2.16: Eine Halbkante ist ein Verweis auf eine Kante und ein Vorzeichen zur Orientierung

Bis jetzt kann eine Masche lediglich eine (1!) Liste mit Halbkanten enthalten, die nicht als eigener Loop gekennzeichnet und daher nicht referenziert werden kann. Mit dem nächsten Schritt der topologischen Normalisierung wird berücksichtigt, dass ein Loop mehrfach referenziert werden kann. Durch das Einführen des Konzepts „Loop“ kann die doppelte Speicherung von Halbkanten-Sequenzen (Loopredundanz) beseitigt werden. Es wird ebenfalls zwischen innerem und äußerem Loop unterschieden (Abbildung 2.17).

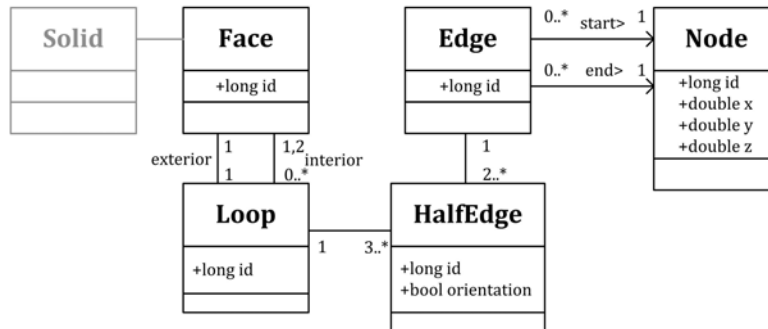


Abbildung 2.17: Das Konzept des Loop

Der letzte Schritt der topologischen Normalisierung berücksichtigt die dritte Dimension in der topologischen Modellierung, den topologischen Körper (engl. *solid*). Zwei Körper (Bauteil, Raum, Außenraum) werden durch eine Masche getrennt. Die Referenzierungsrichtung von Masche nach Körper vermeidet die redundante Speicherung der gleichen Masche in verschiedenen Körpern (Abbildung 2.18).

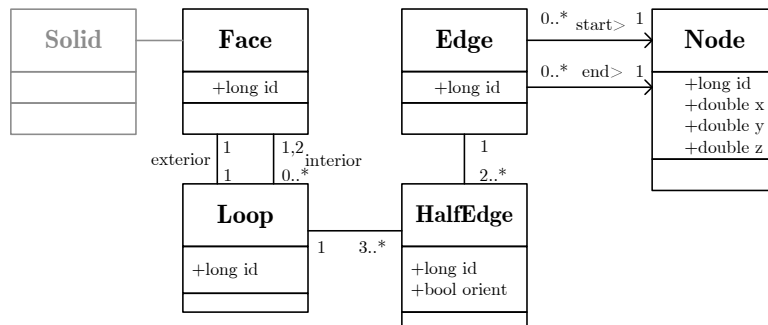


Abbildung 2.18: Das topologisch normalisierte Modell mit Loops

Das Konzept „Loop“ wird im Kern des entwickelten topologischen Modells von *Popa3d* nicht berücksichtigt, weil die Loops einer Masche jederzeit mit kombinatorischen Algorithmen bestimmt werden können. Die Unterscheidung ob ein Loop den äußeren oder inneren Rand einer Masche beschreibt kann durch Flächenvergleich ermittelt werden. Das flächenmäßig größte Loop ist stets der äußere Rand.

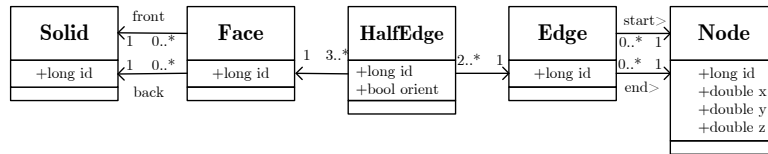


Abbildung 2.19: Topologisch Normalisiertes 3D Modell

2.11 Geometrische Normalisierung

Die topologische Normalisierung behandelt die gegenüber stetigen geometrischen Transformationen invarianten Eigenschaften des 3D-Modells. Die Analogie der Normalisierung wird nun auch auf geometrischen Eigenschaften des Modells angewendet. Bis jetzt wurde die Lage und Form des modellierten Körpers durch den Ortsvektor $(x, y, z)^T$ des Knotens (engl. *node*) festgelegt. Der zentrale Ansatz von (Gründig & Gielsdorf, 2002), der in dieser Arbeit untersucht, erweitert und implementiert wird, ist die Parametrisierung der Geometrie über Oberflächen. Unter „Parametrisierung“ ist in einem deterministischen Modell die Struktur der geometriebeschreibenden Variablen zu verstehen. In einem *empirischen Modell* ist die Parametrisierung darüberhinaus die „Wahl der Unbekannten“, also die Wahl der zu schätzenden Parameter.

[Gielsdorf 2007, S. 21ff] stellt folgende sechs Thesen zur Abbildungsvorschrift topologischer Primitive² auf einen Metrischen Raum (Geometrie) auf:

- T1. Eine Abbildungsfunktion kann für Zellen einer beliebigen Dimension definiert werden.
- T2. Eine Abbildungsfunktion in der Dimension q setzt voraus, dass Hyperflächen mit einer Dimension $q + i$ nicht gekrümmt sind.
- T3. Der Rand von Hyperflächen einer Dimension $q + i$ wird durch Hyperflächen der Dimension $q + i - 1$ definiert.
- T4. Hyperflächen einer Dimension $q - 1$ werden durch Hyperflächen der Dimension $q - i - 1$ begrenzt, die durch Schnitt der Hyperflächen der Dimension $q - 1$ entstehen.
- T5. Verschiedene Zellen gleicher Dimension können sich eine Abbildungsvorschrift teilen.
- T6. Verschiedene Abbildungsvorschriften können auf gleiche Parameter verweisen.

Diese sechs Thesen bilden die Grundlage der Geometrischen Normalisierung. Bei Maschen regelmäßiger Gebäude treten folgende der vier geometrischen Eigenschaften häufig auf (Abbildung 22):

- Eine Masche ist planar: Alle Kanten und Knoten einer Masche liegen in einer Ebene.
- Mehrere Maschen sind koplanar: Die Maschen liegen in einer Ebene.
- Mehrere Maschen sind parallel: Die Maschen haben den gleichen oder entgegengesetzten Normalenvektor.
- Zwei Maschen sind orthogonal: Das Skalarprodukt der Normalenvektoren ist 0.

Wird die Geometrie mit den Koordinaten der Knoten parametrisiert, müssen diese vier Eigenschaften entweder explizit als „Bedingungsgleichungen zwischen Punkt-Koordinaten“ garantiert oder zumindest auf Gültigkeit geprüft werden. Die schrittweise geometrische Normalisierung hat zwei Ziele:

- Die geometrischen Bedingungen (Planarität, Koplanarität, Parallelität und Orthogonalität) werden implizit über das geometrisch-topologische Datenmodell garantiert.
- Die Anzahl der zu schätzenden Geometrieparameter wird stark reduziert

²[Gielsdorf 2007] verwendet den Begriff der Zelle, der im Kontext dieser Arbeit als topologisches Primitiv verstanden werden kann.

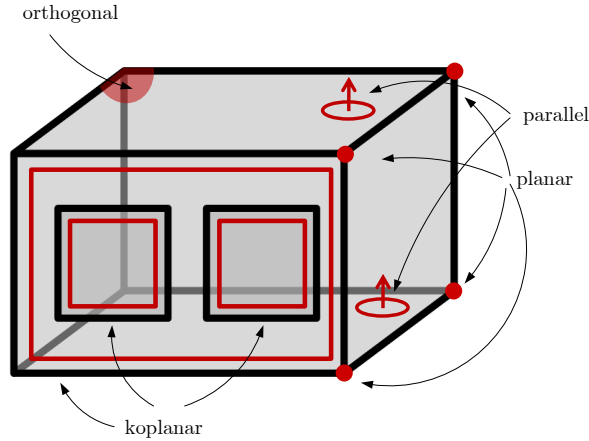


Abbildung 2.20: vier Ebenen der Geometrischen Redundanz

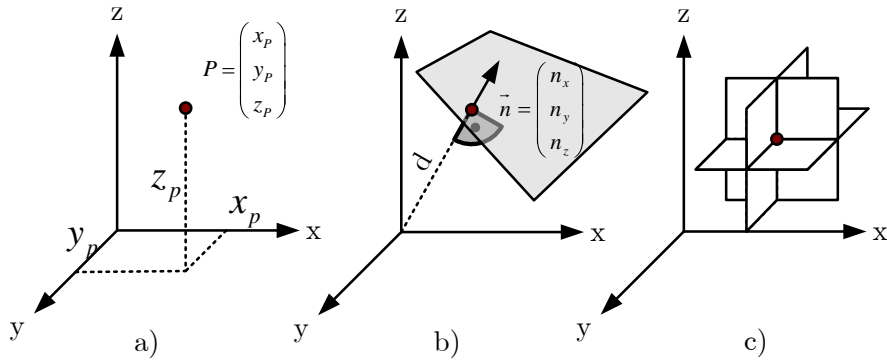


Abbildung 2.21: a) Punkt-Repräsentation b) Flächen-Repräsentation c) Punkt als Ebenenschnitt

Gemäß $T1$ und $T2$ wird in einem ersten Schritt die Geometrie nicht über die Ortsvektoren

$$P = (x_p, y_p, z_p)^T \quad (2.1)$$

der Knoten sondern mit dem Normalenvektor

$$\vec{n} = (n_x, n_y, n_z)^T \quad (2.2)$$

und orthogonalem Abstand zum Ursprung d der Ebenen beschrieben (Abbildung 2.21). In einem flächenparametrisiertem 3d-Modell kommt der topologischen Struktur deswegen eine weitere Bedeutung zu: Die explizite Spezifikation der Topologie ist nach $T3$ und $T4$ notwendig, um die Geometrie der Kanten und Knoten abzuleiten. Die Geometrie der Gerade, auf der sich eine Kante befindet, ergibt sich durch den Schnitt zweier Ebenen. Die Lage eines Knoten ergibt sich durch den Schnitt von mindestens drei nichtparallelen Ebenen. Die Topologie bestimmt dabei, welche Ebenen sich in einem Knoten schneiden.

Treffen sich drei nichtparallele Ebenen in einem Knoten, wird die Punktkoordinate mit $P = (x_p, y_p, z_p)^T$ mit dem linearen Gleichungssystem

$$\begin{bmatrix} n_{ix} & n_{iy} & n_{iz} \\ n_{jx} & n_{jy} & n_{jz} \\ n_{kx} & n_{ky} & n_{kz} \end{bmatrix} \begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix} = \begin{bmatrix} d_i \\ d_j \\ d_k \end{bmatrix} \quad (2.3)$$

berechnet. Wenn sich in einem Punkt mehr als drei Ebenen schneiden, wird das Normalgleichungssystem

$$A^T A x = A d \text{ mit } A = \begin{bmatrix} n_{1x} & n_{1y} & n_{1z} \\ n_{2x} & n_{2y} & n_{2z} \\ \dots & \dots & \dots \\ n_{nx} & n_{ny} & n_{nz} \end{bmatrix}, x = \begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix} \text{ und } d = \begin{bmatrix} d_1 \\ d_2 \\ \dots \\ d_n \end{bmatrix} \quad (2.4)$$

gelöst³. Die Integritätsbedingung der Planarität einer Masche darf allerdings durch die Lösung des Normalgleichungssystems nicht verletzt werden.

Mit dem Konzept der Ebenen-Parametrisierung „wandert“ die Verbindung zwischen Geometrie und Topologie von der nullten Dimension (Knoten-Punkt) in die zweite Dimension (Masche-Ebene).

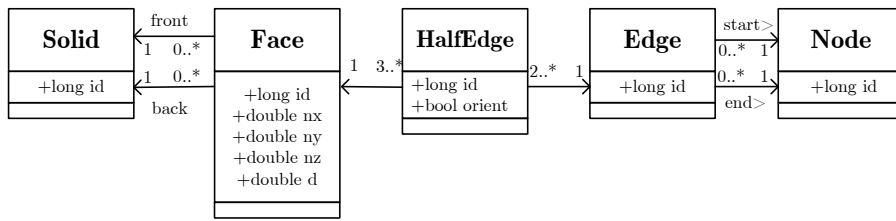


Abbildung 2.22: Erste geometrische Normalisierung: Ebenen-parametrisiertes 3d-Modell

In diesem Modell (Abbildung 2.22) ist die **Planarität** einer Masche durch referenzielle Integrität garantiert und muss nicht mit geometrischen Algorithmen überprüft werden. Der zweite Schritt der geometrischen Normalisierung nutzt die Eigenschaft der **Koplanarität**. Durch Einführung der Ebene (engl. *plane*) als eigenen Entitätstyp kann eine Ebene von mehreren Maschen (engl. *face*) referenziert werden.

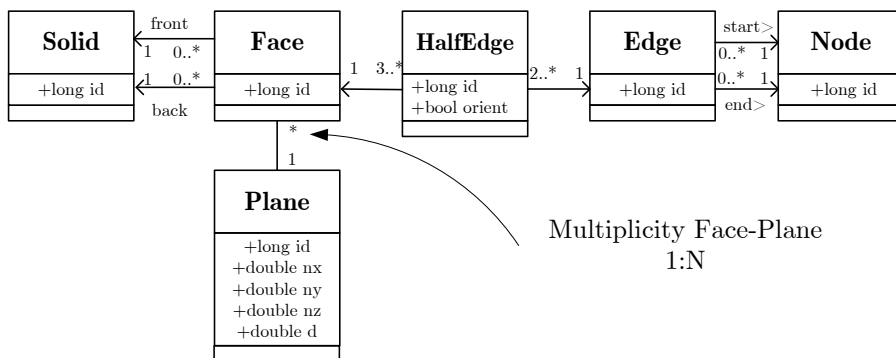


Abbildung 2.23: Zweite Geometrische Normalisierung: Ebene wird referenziert

Der dritte Schritt der geometrischen Normalisierung trennt die Translationskomponente d von der Rotationskomponente $\vec{n} = (n_x, n_y, n_z)^T$ der Ebene (T_6). Der Normalenvektor wird instanziiert und kann von unterschiedlichen Ebenen referenziert werden (Abbildung 2.24). Das Datenmodell unterstützt damit das Konzept der **Parallelität** mit kombinatorischen Mitteln.

³Bei strenger Einhaltung der topologischen Bedingungen einer 2-Mannigfaltigkeit schneiden sich immer genau drei Ebenen in einem Knoten

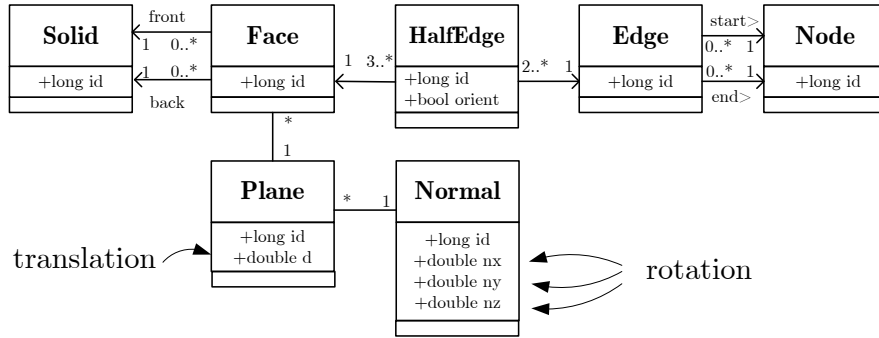


Abbildung 2.24: Dritte geometrische Normalisierung: Normalenvektor wird referenziert

Der vierte Schritt der geometrischen Normalisierung trennt die Parameter des Normalenvektors. Ein Parameter kann von verschiedenen Normalenvektoren referenziert werden.

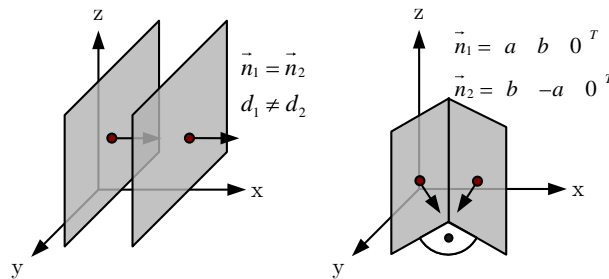


Abbildung 2.25: Parallelität und Orthogonalität

(Abbildung 2.25) zeigt zwei zueinander orthogonale Ebenen. Ist die gleiche Komponente eines Normalenvektors 0 (hier: $n_{1z} = n_{2z} = 0$) kann die Orthogonalität durch Parametertausch und einen Vorzeichenwechsel ausgedrückt werden. Durch Einführung des Parameters als eigenen Entitätstyp, durch algebraisches Vorzeichen qualifizierte Assoziation und einer 1 : N Kardinalität zwischen Normalenvektor und Parameter wird die Orthogonalität mit kombinatorischen Mitteln ausgedrückt (T6).

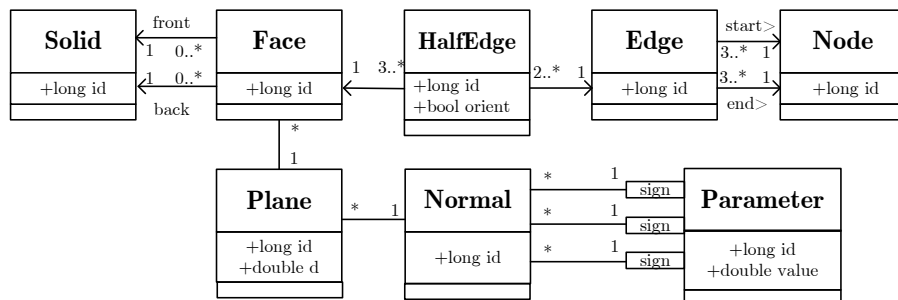


Abbildung 2.26: Vierte Normalisierung: Der Parameter wird Referenziert

2.12 Redundanzvergleich bei Minimalkonfiguration

Man spricht von redundanter Information, wenn mehr Information als unbedingt nötig zur Beschreibung eines geometrischen Zustands vorhanden ist. Für den Redundanzvergleich wird angenommen, dass

- Information „stückweise“ gebildet wird
- positive Information in deterministischen Modellen gespeichert wird
- negative Information in deterministischen Modellen überprüft werden muss
- positive Information in stochastischen Modellen eine Bedingung bzw. Beobachtung ist.
- negative Information in stochastischen Modellen ein unbekannter Parameter ist
- die Redundanz die Summe der (negativen und positiven) Information ist
- die Minimalkonfiguration eine Redundanz von 0 hat.

Zur Darstellung eines Quaders werden 9 „Stück“ Information benötigt. Mit 3 Translationen, 3 Rotationen und 3 Maßen (Länge, Breite, Höhe) kann ein Quader beschrieben werden.

Bei einer **Punkt-Parametrisierung** müssen 8 Koordinatentripel gespeichert werden. Da jede Koordinate 3 Parameter enthält, werden insgesamt (+24) Parameter gespeichert. Die Redundanz liegt also bei $24-9 = 15$. Es müssen also 15 geometrische Konsistenzprüfungen durchgeführt werden:

- Sind alle 6 Maschen planar? (-6)
- Sind die Normalenvektoren gegenüberliegender Maschen parallel und entgegengesetzt (Skalarprodukt der Normalenvektoren -1) ? (-3)
- Sind die Paare paralleler Maschen gegenseitig rechtwinklig? (-3)
- Sind die Maschen gleich orientiert (z.B. Jede Normale nach außen) (-3)

Bei einer **Ebenen-Parametrisierung** (ohne Normalisierung der Normalenvektoren und Parameter) werden je 4 Parameter (n_x, n_y, n_z, d) für jede der 6 Maschen gespeichert. Die Redundanz liegt also ebenfalls bei $24-9 = 15$. Es müssen also 15 geometrische Konsistenzprüfungen durchgeführt werden:

- Sind alle 6 Normalenvektoren normiert ? (-6)
- Sind die Normalenvektoren gegenüberliegender Maschen parallel und entgegengesetzt (Skalarprodukt der Normalenvektoren -1) ? (-3)
- Sind die Paare paralleler Maschen gegenseitig rechtwinklig? (-3)
- Sind die Maschen gleich orientiert (z.B. Jede Normale nach außen) (-3)

Bei einer **geometrisch normalisierten Ebenen-Parametrisierung** (mit Normalisierung der Normalenvektoren und Parameter) werden nicht 6x3 Komponenten eines Normalenvektors gespeichert, sondern 6 Parameter ($\alpha_1=0, \alpha_2=0, \alpha_3=1, \alpha_4=a, \alpha_5=b, \alpha_6=0$), die von den Normalenvektoren referenziert werden.

Mit 6 Parametern und 6 Ebenenabständen d werden insgesamt 12 Parameter gespeichert. Die Redundanz ergibt sich zu $12-9=3$. Es müssen also nur 3 Konsistenzbedingungen überprüft werden:

- Norm $\sqrt{n_1^2 + n_2^2 + n_3^2} = 1$
- Norm $\sqrt{n_4^2 + n_5^2 + n_6^2} = 1$
- Wegen Vorzeichentausch $\alpha_6 = 0$

(Abbildung 2.28) zeigt ein Wohngebäude, das aus 412 Knoten besteht. Bei einer Punktparametrisierung wären 1236 Koordinaten zu berechnen. Das Gebäude wurde im 3d Modellierungsprogramm *Google SketchUp* erzeugt und anschließend mit der in dieser Arbeit entwickelten Software topologisch und geometrisch normalisiert und im *Popa3d* dargestellt. Es setzt sich dann aus 100 Ebenen und 14 Parametern zusammen. Dies entspricht einer *Parameterreduktion um 91%!*

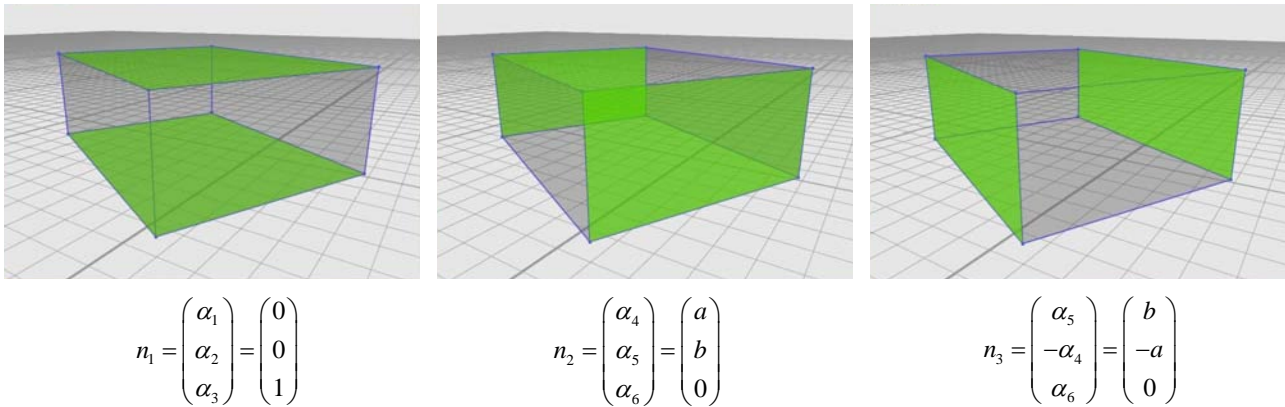
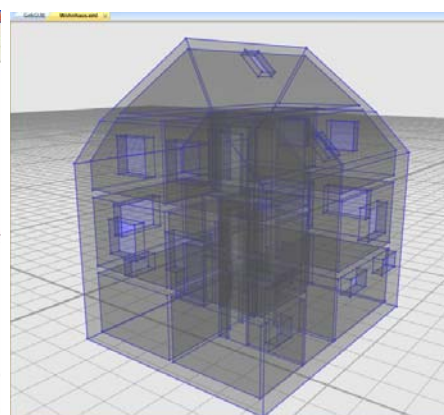
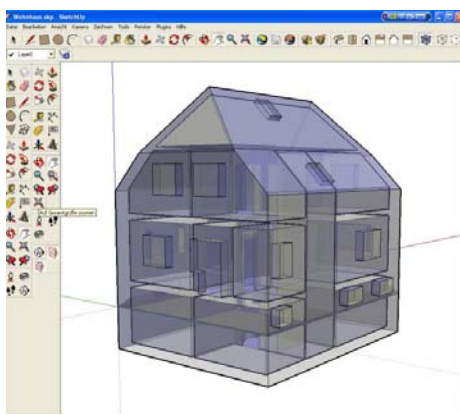


Abbildung 2.27: Normalenvektor #1

**Punkt-Parametrisierung:**

412 Knoten

1236 Koordinatenwerte

1236 unbekannte Parameter**Popa3d:**

9 Normalenvektoren mit

14 Parametern

100 Ebenen

114 unbekannte Parameter→ **91 % weniger Parameter**

Abbildung 2.28: Minimierung der Parameterzahl um 91% am Beispiel eines Wohnhauses

2.13 Beobachtungen und Bedingungen

2.13.1 Inverse Darstellung durch Beobachtungen

Deterministische Modelle speichern die Parameter der absoluten Geometrie. Relativmaße oder Maßketten werden in CAD-Programmen aus diesen Werten berechnet – man spricht von direkten Problemen. Der Vermessungsingenieur denkt hingegen in *inversen Problemen*. Anhand von relativen Beobachtungen zwischen zunächst unbekannten Größen soll die absolute Geometrie berechnet werden. Mittels Funktionalem Modell werden die Parameter der absoluten Geometrie auf die beobachteten Parameter der relativen Geometrie abgebildet. Die Summe aller Gleichungen wird in einem linearen oder linearisierten Gleichungssystem gelöst. Da häufig mehr Beobachtungen oder Bedingungen zur Lösung des inversen Problems in das Modell eingeführt werden, sind diese aufgrund unvermeidlicher Messabweichungen nicht gegenseitig widerspruchsfrei. Das „Werkzeug“ der Ausgleichsrechnung (Kapitel 5) generiert das wahrscheinlichste, widerspruchsfreie Ergebnis für die „verbesserten“ Beobachtungen und die geschätzten Parametern der absoluten Geometrie.

Eingangsgrößen der Ausgleichsrechnung sind:

- **Beobachtungen** sind Messwerte, die mit ingenieurgeodätischen Messinstrumenten (Totalstation, Laserdistanzmessgerät, Nivelliergerät, GNSS, Neigungssensor) gewonnen werden und als kontinuierliche Größe (Dezimalzahl) gespeichert werden. Messwerte sind stochastische Größen und unterliegen damit groben, systematischen und zufälligen Abweichungen. Sie sind deshalb Gegenstand der Optimierung (Ausgleichung) und statistischen Analyse. Messwerte werden in einer geometrischen oder physikalischen Einheit beschrieben.

- **Bedingungen** ergeben sich aus zum einen aus *mathematischen-geometrischen Gesetzen* (Winkelsumme im Dreieck, Normierung eines Normalenvektors, Normierung von Quaternionen) oder ergeben sich aus der *Objektstruktur* (Orthogonalität, Parallelität). Sie können als sogenannte *Pseudobeobachtungen* eingeführt werden, auch wenn sie nicht originär mit einem Messinstrument gewonnen werden.
- **Beobachtungstopologie.** Die Größe des Messwertes (10,23m) ist zum Lösen eines Gleichungssystems alleine nicht ausreichend. Es muss bekannt sein, welche unbekannten Parameter der absoluten Geometrie mittels des funktionalen Modells auf die relative Beobachtung (oder Beobachtungsgruppe) abgebildet wird. Eine fehlerhafte Beobachtungstopologie wirkt sich typischerweise als *grober Fehler* aus.
- **Objektopologie.** Zur Bestimmung der Beobachtungstopologie kann die Objektopologie nötig sein. Eine Beobachtung zwischen zwei topologischen Primitiven der Objektstruktur (z.B. Abstand Masche-Knoten) kann in einem flächenparametrisierten nur mit Hilfe der in der Objektopologie spezifizierten Referenzen zwischen Knoten, Kanten und Maschen funktional ausgewertet werden. Die Objektopologie stellt eine Verknüpfung zwischen den geometrischen Elementen Ebene, Normalenvektor, Parameter und den Messungen her.
- **Struktur der Objektgeometrie.** Zur Bestimmung der Beobachtungstopologie wird die Struktur der Objektgeometrie benötigt. Eine Beobachtung zwischen zwei topologischen Primitiven der Objektstruktur (z.B. Abstand Masche-Knoten) kann in einem flächenparametrisierten nur mit Hilfe der in der Objektopologie spezifizierten Referenzen zwischen Ebenen, Normalenvektoren und Parametern bestimmt werden. Die *Werte der Objektgeometrie* müssen nicht (bei linearen Problemen) oder nur näherungsweise (bei nicht-linearen Problemen) bekannt sein.

Die Ausgleichungsrechnung erzeugt:

- Die beste Schätzung für die unbekannten **Parameter der absoluten Geometrie** als das „eigentliche“ Ziel der Vermessung.
- **Verbesserungen** (optimierte Residuen) für alle stochastischen Eingangsgrößen. Die verbesserten Beobachtungen stimmen widerspruchsfrei mit den geschätzten Parametern der absoluten Geometrie überein und sind gemäß der Methode der kleinsten Quadrate optimiert.
- Eine **stochastisch begründete Entscheidungsgrundlage** zum Auffinden von Beobachtungs- oder Modellfehlern sowie eine Grundlage zur Überprüfung, Generalisierung oder Verfeinerung der Objekte.

2.13.2 Abstände zwischen topologischen Primitiven

Der für die Vermessung wichtigste Beobachtungstyp ist die Abstandsmessung. Sie wird in der Regel mit einem Laser-Entfernungsmessgerät zwischen topologischen Primitiven (Masche, Kante, Knoten) der Objektopologie durchgeführt (Abbildung 2.29).

Der Laserentfernungsmesser wird über eine BluetoothTM Schnittstelle mit *Popa3d* verbunden. Die Spezifikation der Beobachtungstopologie erfolgt interaktiv in der graphischen 3D-Benutzerschnittstelle. Bei einem „Klick“ mit der Maus wird das gewählte Element farblich gekennzeichnet. Sobald zwei topologische Primitive selektiert sind, kann der Benutzer ein Objekt der Beobachtungstopologie erzeugen. Die gemessene Distanz wird vom Laserentfernungsmesser an die Software gesendet. Dort wird die empfangene Distanz als Attribut „Wert“ auf das neu erzeugte Element der Beobachtungstopologie übertragen. Die Elemente des objektorientierten Codemodells werden in einer XML-Datei oder in einer relationalen Datenbank gespeichert. Die Beobachtungsgleichungen für die Ausgleichungsrechnung sind in Kapitel 3.2

Abbildung A.1 (UML Diagramm im Anhang) visualisiert das konzeptionelle Modell der Abstandbeobachtungen und gliedert den Diskursbereich in die Module Ausgleichung (engl. *Adjustment*), Beobachtungstopologie (engl. *Observation-Topology*), Objektopologie (engl. *Object Topology*) und Objektgeometrie (engl. *Object Geometry*).

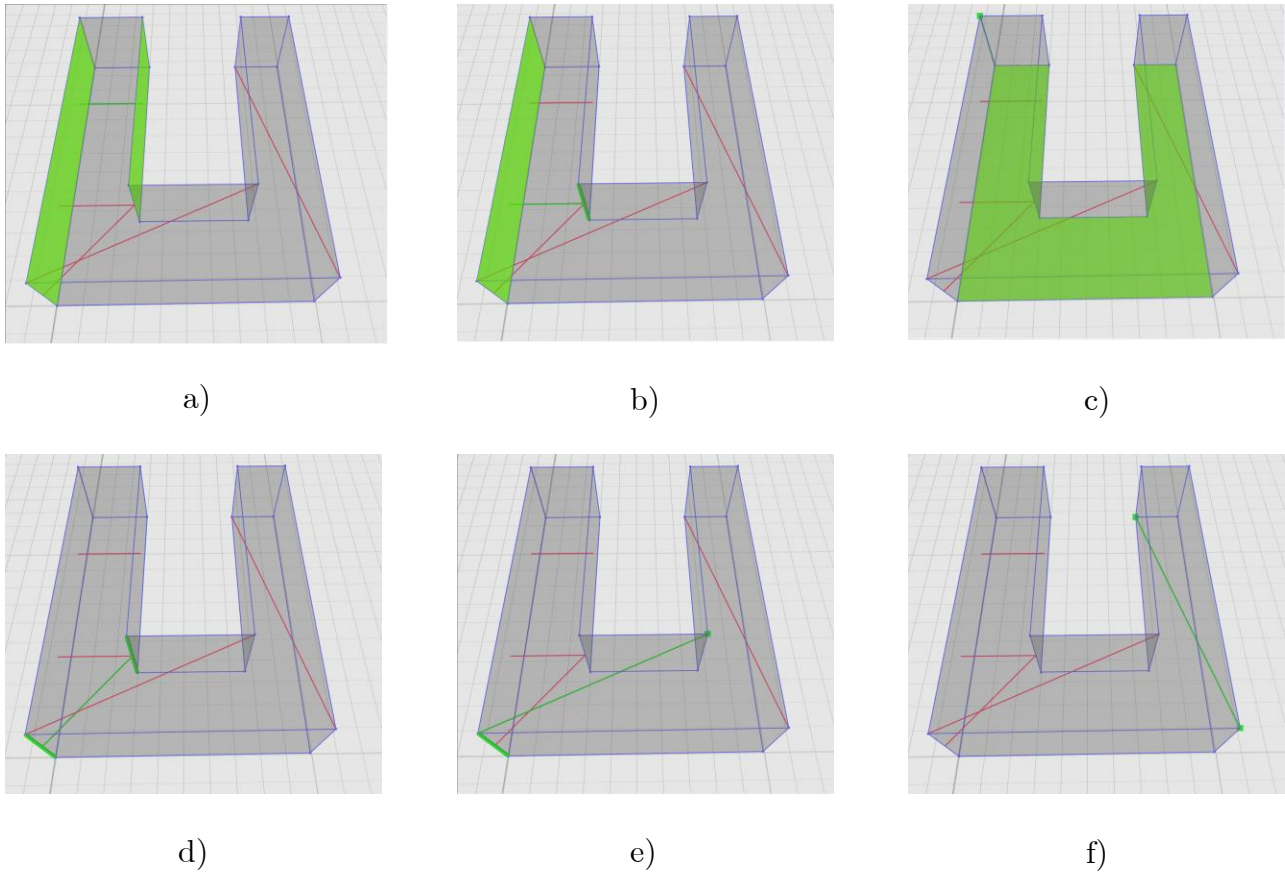


Abbildung 2.29: *Abstandsbeobachtungen zwischen topologischen Primitiven a)Face-Face b)Face-Edge c)Face-Node d)Edge-Edge e)Edge-Node f)Node-Node*

2.13.3 Lokale Polarkoordinaten auf topologischen Primitiven

Der für die berührungslose Vermessung wichtige Beobachtungstyp der lokalen Polarkoordinate wird mit einem (reflektorlosen) Tachymeter durchgeführt. Das Beobachtungsaggregat „lokaler Punkt“ setzt sich aus drei nicht korrelierten elementaren Messgrößen zusammen:

- Horizontalrichtung (Hz)
- Vertikalwinkel (V)
- Schrägstrecke (s)

Die Beobachtung wird auf einem Standpunkt (engl. *setup*) mit einer bestimmten Instrumentenhöhe i durchgeführt. Zwischen Festpunkt und Standpunkt besteht eine $1 : n$ Beziehung. Auf dem gleichen Festpunkt kann der Tachymeter mehrfach aufgestellt werden (Abbildung A.2).

Zur eindeutigen geometrischen Bestimmung einer Ebene muss eine Masche mindesten dreimal, eine Kante zweimal und ein Knoten einmal angezielt werden. Die Transformation ist in Ihrer Translationskomponente dem Punkt und in ihrer Rotationskomponente dem Standpunkt zugeordnet (UML Klassen-Diagramm im Anhang, Abbildung A.2, Seite 88). Die Rotation ist mit einem Quaternion parametrisiert (siehe Abschnitt 3.1) und hat drei Freiheitsgrade. Das Instrument muss nicht unbedingt horizontiert werden, da bei ausreichend konfigurierten Beobachtungen alle drei Freiheitsgrade der Rotation berechnet werden können.

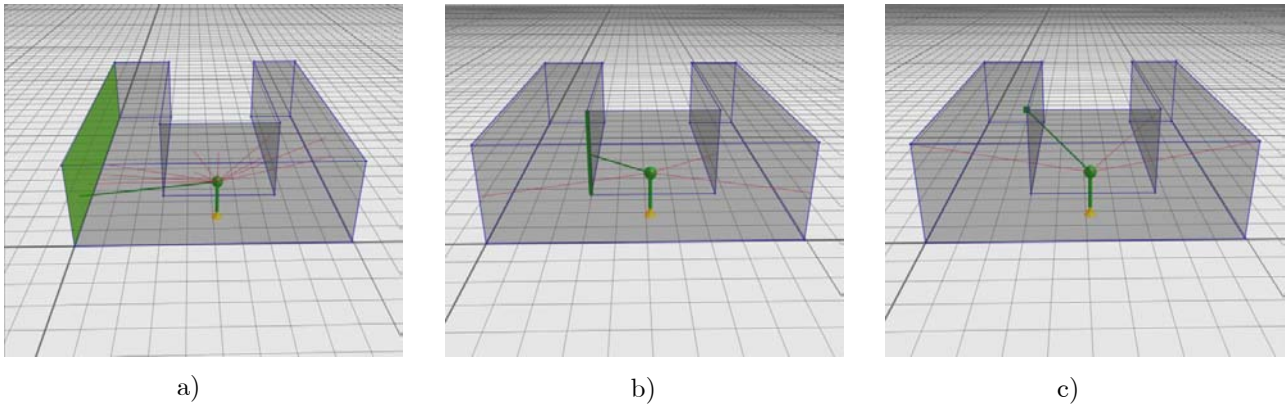


Abbildung 2.30: a) *PointOnFace* b) *PointOnEdge* c) *PointOnNode*

2.13.4 Winkelbedingungen zwischen topologischen Primitiven

Winkelbedingungen zwischen topologischen Primitiven ergeben sich aus der Objektgeometrie und der beobachtungsinduzierten Integrität.

Die **Objektgeometrie** wird in *Popa3d* insofern generalisiert dargestellt, dass Planarität, Koplanarität und Orthogonalität zwischen topologischen Primitiven implizit (kombinatorisch) modelliert werden kann. Winkelbedingungen zwischen topologischen Primitiven ermöglichen darüberhinaus eine *explizite* Spezifikation von Winkelbeziehungen. Diese wird als stochastische Pseudobeobachtung in die Ausgleichung eingeführt und kann mit statistischen Tests überprüft werden.

Beobachtungsinduzierte Integritätsbedingungen ergeben sich aus den Abstandbeobachtungen Masche-Masche, Masche-Kante und Kante-Kante. Diese Abstandsbeobachtungen werden nicht (wie in der Ingenieurgeodäsie üblich) zwischen diskretisierten Objektpunkten gemessen, sondern zwischen *Punktmengen*. Für jeden Punkt der Punktmenge Masche oder Kante muss die Abstandsbeobachtung gelten. Daher ist die Parallelität der Maschen bzw. Kanten eine Winkelbedingung zwischen topologischen Primitiven, die die Lösbarkeit von Beobachtungsgleichungen garantiert. Die Winkelbedingungen werden von der Software automatisch entsprechend der Abstandsmessungen in die Ausgleichung eingeführt.

Das konzeptionelle Modell der Winkelbedingungen (UML Klassen-Diagramm im Anhang, Abbildung A.3, Seite 89) zeigt die zentrale Bedeutung einer expliziten Spezifikation topologischer Primitive, da die Winkelbeziehung nicht zwischen Ebenen, sondern Maschen erfolgt.

2.13.5 A priori Differenzen zwischen Abständen und Normalenvektoren

Die Objektgeometrie soll auch bei unvollständiger Messung berechenbar sein. Dies ermöglicht einen sofortigen Vergleich zwischen Messung und 3d-Skizze - sowohl während der Ersterfassung vor Ort als auch bei einer Qualitätsprüfung von 3d-Modellen.

Aus der topologischen Skizze können Näherungswerte der Objektgeometrie gewonnen werden. Das mathematische Verfahren der Regularisierung nutzt diese Werte mittels Identitätsgleichungen, die allerdings datumsabhängig sind. *Popa3d* kann alternativ datumsinvariante a priori Differenzen in das mathematische Modell der Ausgleichung einführen. Diese werden gebildet aus:

- Relativdistanzen Δd_{ij} von Ebenenabständen d innerhalb Gruppen paralleler Normalenvektoren
- Relativwinkel Δn_{ij} zwischen k Normalenvektoren in allen Kombinationen.

Die Spezifikation der **Relativdistanzen** erfolgt in vier Schritten:

1. Die k Normalenvektoren werden richtungsnormiert, so dass für alle gilt:

$$\begin{aligned} n_x &\geq 0 && \text{oder} \\ n_y &\geq 0 && \text{mit } n_x = 0 \text{ oder} \\ n_z &\geq 0 && \text{mit } n_x = 0 \text{ und } n_y = 0 \end{aligned} \quad (2.5)$$

Die Vorzeichen der Ebenenparameter d werden entsprechend geändert.

2. m Gruppen gleicher Normalenvektoren werden gebildet.
3. Die p Ebenen jeder Gruppe werden nach dem Ebenenabstand d_i geordnet.
4. Die $p - 1$ Relativedifferenzen Δd_i werden in jeder der m Gruppen gebildet.

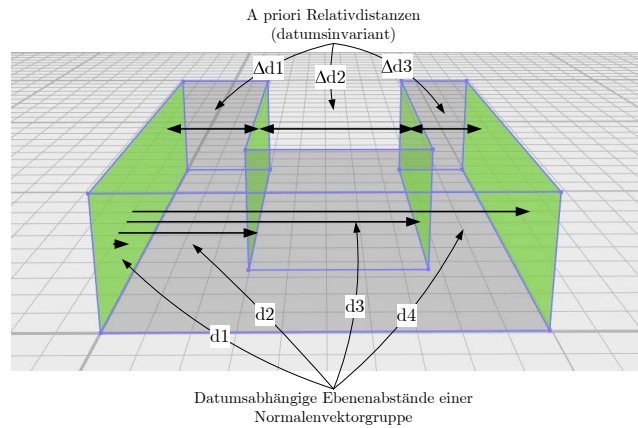


Abbildung 2.31: *A priori Relativedistanzen*

Für die Spezifikation der $\binom{k}{2}$ **Relativwinkel** Δn_{ij} wird das Skalarprodukt der k Normalenvektoren in allen Kombinationen gebildet und als Pseudobeobachtung in die Ausgleichung eingeführt.

Das konzeptionelle Modell der Skizzengeometrie (UML Klassen-Diagramm im Anhang, Abbildung A.3, Seite 89) zeigt, dass die skizzierte Objektgeometrie unabhängig von den topologischen Primitiven (grau) in das konzeptionelle Modell und Ausgleichungsrechnung integriert sind. Der direkte Verweis Geometrie \leftrightarrow Bedingung unter der „Umgehung“ topologischer Primitive dient auch der Reduktion der Anzahl der Bedingungsgleichungen.

3 Funktionales Modell

Der sogenannte Allgemeinfall der Ausgleichung (Gauß-Helmert Modell mit Restriktionen zwischen den Unbekannten) kennt zwei Typen von Funktionalen Modellen. Eine **stochastische Bedingungsgleichung** (kurz: Bedingungsgleichung) enthält eine oder mehrere Beobachtungen l und eine oder mehrere Unbekannte x . Im Allgemeinen wird sie in der Form

$$f(x, l + v, c) = 0 \quad (3.1)$$

angegeben. Dabei sind die unbekannten Parameter x und die Residuen v die zu schätzenden Größen. Die Beobachtungen l und Konstanten c bleiben bei der Parameterschätzung unverändert. Ist an der stochastischen Bedingungsgleichung nur eine Beobachtung beteiligt, kann sie auch, wie im Gauß-Markow-Modell üblich, als Beobachtungsgleichung $l + v = f(x)$ angegeben werden. Eine **deterministische Bedingungsgleichung** (kurz: Restriktionsgleichung) enthält nur unbekannte Parameter und Konstanten aber keine stochastisch modellierte Beobachtung.

$$g(x, c) = 0 \quad (3.2)$$

Im Gegensatz zu den stochastischen Bedingungsgleichungen müssen deterministische Bedingungsgleichungen gegenseitig linear unabhängig sein.

Die methodische Begründung und die Datenmodellierung der Beobachtungen und Bedingungen wurde in Kapitel 2 (S. 10ff) beschrieben. Damit können die einzelnen Information im Rechner gespeichert und bearbeitet werden. Für eine Ausgleichungsrechnung nach dem Gauß-Helmert Modell muss jedem Datentyp der Beobachtungsdomäne noch eine Gleichungen vom Typ (3.1) oder (3.2), das Funktionale Modell, zugeordnet werden.

Die Beobachtungs- bzw. Bedingungstypen lassen sich wie folgt klassifizieren:

- Konsistenzregeln, die sich aus mathematisch-geometrischen Bedingungen ergeben. (Funktionales Modell in Kapitel 3.1)
- Der in Kapitel 2.13.2 beschriebene Beobachtungstyp „Abstand“, z.B. mit einem Laserdistanzmessgerät bestimmt. (Funktionales Modell in Kapitel 3.2).
- Der in Kapitel 2.13.3 beschriebene Beobachtungstyp „Polarkoordinate“, z.B. mit einer reflektorlos messenden Totalstation bestimmt. (Funktionales Modell in Kapitel 3.3)
- Der in Kapitel 2.13.4 beschriebene Bedingungstyp „Winkel“, z.B. aufgrund von Konsistenzbedingungen. (Funktionales Modell in Kapitel 3.4)
- Der in Kapitel A.4 beschriebene Bedingungstyp „Relativedistanz“, zur „weichen“ Modellierung der Skizzengeometrie. (Funktionales Modell in Kapitel 3.5)
- Bedingungstyp „Datumsfestlegung“, zur Spezifikation der Beziehung Objekt \leftrightarrow Koordinatensystem. (Funktionales Modell in Kapitel 3.6)

3.1 Mathematisch-Geometrische Bedingungen

Betrag des Normalenvektors. Das geometrisch normalisierte 3d-Modell *Popa3d* parametrisiert die Geometrie der Objektstruktur mit Hilfe von Ebenenparametern $\vec{n} = (n_x, n_y, n_z)^T$ und dem orthogonalen (kürzesten) Abstand d der Ebene zum Ursprung.

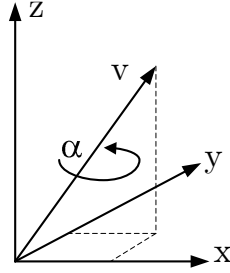


Abbildung 3.1: Geometrische Interpretation eines Quaternions

Eine Ebene ist eine Punktmenge, die unbegrenzt ausgedehnt und nicht gekrümmt ist. Alle Punkte $\vec{x}_p = (x_p, y_p, z_p)^T$ der Ebene erfüllen die Hessesche Normalform

$$\langle \vec{n}_0, \vec{x}_p \rangle - d = n_x x_p + n_y y_p + n_z z_p - d = 0 \quad (3.3)$$

nur unter der Bedingung, dass der Normalenvektor der Ebene normiert ist, also die Länge 1 hat.

$$|\vec{n}_0| = \sqrt{\langle \vec{n}_0, \vec{n}_0 \rangle} = \sqrt{n_x n_x + n_y n_y + n_z n_z} = 1 \quad (3.4)$$

Bei der Lösung der Normalgleichung darf diese Bedingung nicht verletzt werden. Es wird entweder eine „stark“ gewichtete Pseudobeobachtung

$$l + v = f(x) = \sqrt{n_x n_x + n_y n_y + n_z n_z} \quad \text{mit } l = 1.0 \quad (3.5)$$

eingeführt oder eine deterministische Bedingung eingeführt:

$$g(x, c) = \sqrt{n_x n_x + n_y n_y + n_z n_z} - 1 = 0 \quad (3.6)$$

Einheitsquaternion. *Popa3d* parametrisiert die Rotation der Drehachse des Tachymeters mit den sogenannten „hyper-komplexen“ Zahlen, die Hamilton im Jahr 1843 entdeckte und „Quaternione“ nannte [Kuipers 1999, S. 12]. Ähnlich wie bei den Komplexen Zahlen wird durch die algebraische Regel

$$\mathbf{i}^2 + \mathbf{j}^2 + \mathbf{k}^2 = \mathbf{ijk} = -1 \quad (3.7)$$

mit

$$\mathbf{i} = (1, 0, 0)^T \quad \mathbf{j} = (0, 1, 0)^T \quad \mathbf{k} = (0, 0, 1)^T$$

ein spezieller Vektorraum geschaffen, in dem man Vektoren multiplizieren kann. Ein Quaternion besteht aus einem skalaren und einem vektoriellen Teil:

$$Q = q_0 + \mathbf{q} = \left[q_0, (q_1, q_2, q_3)^T \right] = \left[\cos\left(\frac{\alpha}{2}\right), \mathbf{v} \sin\left(\frac{\alpha}{2}\right) \right] \quad (3.9)$$

Ein Quaternion kann mit Drehvektor \mathbf{v} und Drehwinkel α geometrisch gedeutet werden. Die Zeilen- und Spaltenvektoren der Rotationsmatrix

$$R_Q = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (3.10)$$

sind orthonormiert:

$$\sum_{k=1}^{k=3} r_{1k} r_{2k} = 0, \sum_{k=1}^{k=3} r_{1k} r_{3k} = 0, \sum_{k=1}^{k=3} r_{2k} r_{3k} = 0, \sum_{k=1}^{k=3} r_{1k} r_{1k} = 1, \sum_{k=1}^{k=3} r_{2k} r_{2k} = 1, \sum_{k=1}^{k=3} r_{3k} r_{3k} = 1 \quad (3.11)$$

Es findet keine Spiegelung (Wechsel von Rechts- auf Linkssystem) statt:

$$\det(R_Q) = +1 \quad (3.12)$$

Die Werte einer Rotationsmatrix $R_Q = R(q_0, q_1, q_2, q_3)$ werden mit den vier Parametern q_0, q_1, q_2, q_3 bestimmt:

$$R_Q(q_0, q_1, q_2, q_3) = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 + q_3 q_0) & 2(q_1 q_3 + q_2 q_0) \\ 2(q_1 q_2 + q_3 q_0) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 + q_1 q_0) \\ 2(q_1 q_3 + q_2 q_0) & 2(q_2 q_3 + q_1 q_0) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \quad (3.13)$$

Mit der Bedingung $|Q| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} = 1$ sind die Bedingungen für eine Rotationsmatrix erfüllt und der überschüssige vierte Freiheitsgrad ist beseitigt.

Für jeden Standpunkt (siehe Funktionales Modell Gleichung 3.43) wird eine **Restriktion** zur Normierung des Quaternions in das Gleichungssystem eingeführt:

$$g(x, c) = q_0^2 + q_1^2 + q_2^2 + q_3^2 - 1 = 0 \quad (3.14)$$

3.2 Abstände zwischen topologischen Primitiven

Popa3d parametrisiert die Geometrie über Ebenengleichungen. Daher müssen die Beobachtungsgleichung als unabhängige Variable der Funktion die unbekannten Parameter $\vec{n} = (n_x, n_y, n_z)^T$ und d enthalten.

Es muss weiterhin bekannt sein, aus welchen Ebenen die topologischen Primitive gebildet werden. Für alle Abstandsbeobachtungen, die keine Knoten der Objekttopologie referenzieren, sind zusätzliche Integritätsbedingungen bezüglich der beteiligten Normalenvektoren aufzustellen. Diese werden über Winkelbedingungen in das Normalgleichungssystem der Ausgleichung eingeführt.

Abstand Masche-Masche (engl. *distance face-face*). Eine Abstandsbeobachtung l zwischen zwei parallelen Maschen i und j ist die positive Differenz der Ebenenparameter d_i und d_j (Abbildung 3.2a). Die Fallunterscheidung ergibt sich aus der Orientierung der beteiligten Masche.

$$l + v = \begin{cases} d_j - d_i & , \text{wenn } \langle \vec{n}_i, \vec{n}_j \rangle = 1 \quad \cap \quad |d_j| \geq |d_i| \\ d_i - d_j & , \text{wenn } \langle \vec{n}_i, \vec{n}_j \rangle = 1 \quad \cap \quad |d_i| > |d_j| \\ -d_j - d_i & , \text{wenn } \langle \vec{n}_i, \vec{n}_j \rangle = -1 \quad \cap \quad |d_j| \geq |d_i| \\ d_i + d_j & , \text{wenn } \langle \vec{n}_i, \vec{n}_j \rangle = -1 \quad \cap \quad |d_i| > |d_j| \end{cases} \quad (3.15)$$

Es muss weiterhin durch die Ausgleichungsrechnung garantiert werden, dass die Ebenen i und j parallel und sind.

$$\langle \vec{n}_i, \vec{n}_j \rangle = \pm 1 \quad (3.16)$$

Deshalb wird für jede Abstandsbeobachtung eine beobachtungsinduzierte Winkelbedingung (3.48) als Bedingungsgleichung in die Ausgleichungsrechnung (und die Dokumenteninstanz) eingeführt.

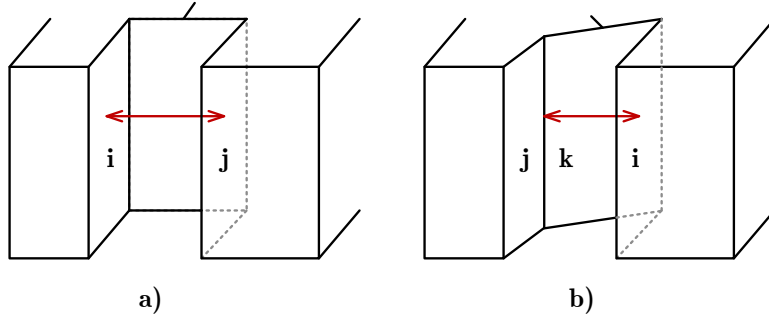


Abbildung 3.2: a) Abstand Masche-Masche b) Abstand Masche-Kante

Abstand Masche-Kante (engl. *distance face-edge*). Eine Abstandsbeobachtung l (Abbildung 3.2b) zwischen einer Masche i und einer Kante, die durch Ebenenschnitt $j \cap k$ entsteht, wird durch:

$$l + v = \begin{cases} d_i - \langle \vec{n}_i, \vec{p}_{jk} \rangle & , \text{wenn } \langle \vec{n}_i, \vec{p}_{jk} \rangle \geq 0 \\ -d_i + \langle \vec{n}_i, \vec{p}_{jk} \rangle & , \text{wenn } \langle \vec{n}_i, \vec{p}_{jk} \rangle < 0 \end{cases} \quad (3.17)$$

mit

$$\vec{p}_{jk} = \left[(\vec{n}_j, \vec{n}_k, \vec{n}_j \times \vec{n}_k)^T \right]^{-1} (d_j, d_k, 0)^T \quad (3.18)$$

modelliert. Es muss weiterhin beachtet werden, dass die Beobachtungsgleichung nur dann für die beiden Punkt-mengen i und $j \cap k$ gültig ist, wenn Ebene i und Kante $j \cap k$ parallel sind.

$$\langle \vec{n}_i, \vec{n}_j \times \vec{n}_k \rangle = 0 \quad (3.19)$$

Um diese zu garantieren wird pro *Abstandsbeobachtung Masche-Kante* eine beobachtungsinduzierte Bedingungs-gleichung *Winkel Masche-Kante* (3.49) in die Ausgleichsrechnung eingeführt.

Abstand Masche-Knoten (engl. *distance face-node*). Eine Abstandsbeobachtung l (Abbildung 3.3a) zwischen einer Masche i und einem Knoten, der durch Ebenenschnitt $j \cap k \cap l$ entsteht, wird durch die Beobachtungsglei-chung

$$l + v = d_i - \langle \vec{n}_i, \vec{p}_{jkl} \rangle \quad (3.20)$$

mit

$$\vec{p}_{jkl} = \left[(\vec{n}_j, \vec{n}_k, \vec{n}_l)^T \right]^{-1} (d_j, d_k, d_l)^T \quad (3.21)$$

modelliert. Zur Vermeidung negativer Abstände gilt $d_i \geq \langle \vec{n}_i, \vec{p}_{jkl} \rangle$.

Abstand Kante-Kante (engl. *distance edge-edge*). Eine Abstandsbeobachtung l (Abbildung 3.3b) zwischen zwei Kanten, die durch die Ebenenschnitte $i \cap j$ und $k \cap l$ gebildet werden, wird durch die Beobachtungsgleichung

$$l + v = |\vec{p}_{ij} - \vec{p}_{kl}| \quad (3.22)$$

mit

$$\vec{p}_{ij} = \left[(\vec{n}_i, \vec{n}_j, \vec{n}_i \times \vec{n}_j)^T \right]^{-1} (d_i, d_k, 0)^T \quad (3.23)$$

und

$$\vec{p}_{kl} = \left[(\vec{n}_k, \vec{n}_l, \vec{n}_k \times \vec{n}_l)^T \right]^{-1} (d_k, d_l, 0)^T \quad (3.24)$$

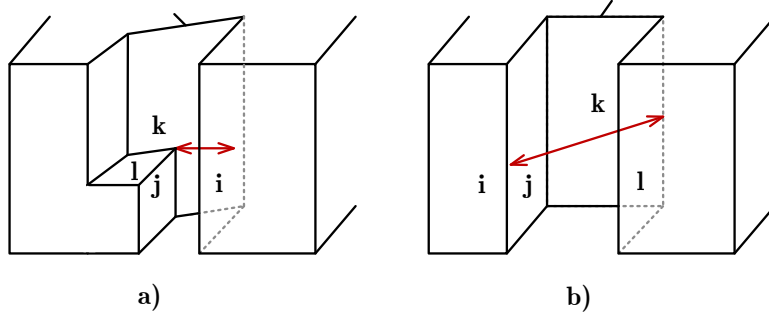


Abbildung 3.3: a) Abstand Masche-Knoten b) Abstand Kante-Kante

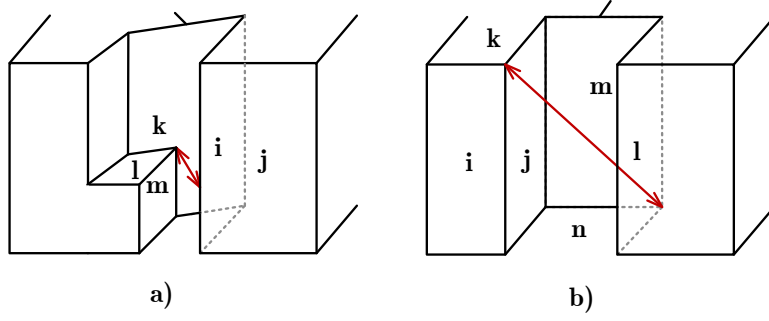


Abbildung 3.4: a) Abstand Kante-Knoten b) Abstand Knoten-Knoten

modelliert. Es muss beachtet werden, dass die Beobachtungsgleichung nur dann für die beiden Punktmengen $i \cap j$ und $k \cap l$ gültig ist, wenn Kante $i \cap j$ und Kante $k \cap l$ parallel sind.

$$\langle \vec{n}_i \times \vec{n}_j, \vec{n}_i \times \vec{n}_j \rangle = 1 \quad (3.25)$$

Deshalb wird für jede Abstandsbeobachtung eine beobachtungsinduzierte Winkelbedingung (3.50) als Bedingungsgleichung in die Ausgleichsrechnung (und die Dokumenteninstanz) eingeführt.

Abstand Kante-Knoten (engl. *distance edge-node*). Eine Abstandsbeobachtung l (Abbildung 3.4a) zwischen einer Kante, die durch Ebenenschnitt $i \cap j$ und einem Knoten, der durch Ebenenschnitt $k \cap l \cap m$ entsteht, wird durch die Beobachtungsgleichung

$$l + v = |\vec{p}_{ij} - \vec{p}_{klm}| \quad (3.26)$$

mit

$$\vec{p}_{ij} = \left[(\vec{n}_i, \vec{n}_j, \vec{n}_i \times \vec{n}_j)^T \right]^{-1} (d_i, d_j, \langle \vec{n}_i \times \vec{n}_j, \vec{p}_{klm} \rangle)^T \quad (3.27)$$

und

$$\vec{p}_{klm} = \left[(\vec{n}_k, \vec{n}_l, \vec{n}_m)^T \right]^{-1} (d_k, d_l, d_m)^T \quad (3.28)$$

modelliert.

Abstand Knoten-Knoten (engl. *distance node-node*). Eine Abstandsbeobachtung l (Abbildung 3.3b) zwischen zwei Knoten, die durch Ebenenschnitt $i \cap j \cap k$ und $l \cap m \cap n$ gebildet wird, wird durch die Beobachtungsgleichung

$$l + v = |\vec{p}_{ijk} - \vec{p}_{lmn}| \quad (3.29)$$

mit

$$\vec{p}_{ijk} = \left[(\vec{n}_i, \vec{n}_j, \vec{n}_k)^T \right]^{-1} (d_i, d_j, d_k)^T \quad (3.30)$$

und

$$\overrightarrow{p_{klm}} = \left[(\vec{n}_l, \vec{n}_m, \vec{n}_n)^T \right]^{-1} (d_l, d_m, d_n)^T \quad (3.31)$$

modelliert.

3.3 Lokale Polarkoordinaten auf topologischen Primitiven

Lokale Polarkoordinaten (Hz, V, s) werden mit einem reflektorlosen Tachymeter gemessen. Zur Beschreibung der stochastischen Bedingungsgleichung werden die lokalen Beobachtungen in das globale System transformiert.

$$(Hz, V, s)_{Stdpkt}^T \xrightarrow{polar \rightarrow kart.} (x, y, z)_{Stdpkt}^T \xrightarrow{Rot.} (x, y, z)_{Stdpkt_{ori}}^T \xrightarrow{Trans.} (x, y, z)_{global}^T \quad (3.32)$$

Es werden nacheinander folgende Transformationen durchgeführt:

1. Koordinatenumformung von polarer Darstellung in kartesische Darstellung

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{Stdpkt} = \begin{bmatrix} s \cdot \sin(V) \cdot \sin(Hz) \\ s \cdot \sin(V) \cdot \cos(Hz) \\ s \cdot \cos(V) \end{bmatrix} \quad (3.33)$$

2. Rotation des Standpunktsystems in ein zum globalen System achsenparalleles Koordinatensystem

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{Stdpkt_{ori}} = R_Q(q_0, q_1, q_2, q_3) \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{Stdpkt} \quad (3.34)$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{Stdpkt_{ori}} = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 + q_3q_0) & 2(q_1q_3 + q_2q_0) \\ 2(q_1q_2 + q_3q_0) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 + q_1q_0) \\ 2(q_1q_3 + q_2q_0) & 2(q_2q_3 + q_1q_0) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{Stdpkt} \quad (3.35)$$

3. Translation des orientierten Standpunktsystems in das globale System

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{global} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{Stdpkt_{ori}} + \begin{bmatrix} t_x \\ t_y \\ t_z + iH \end{bmatrix} \quad (3.36)$$

Die Komponenten der Translation ergeben sich aus der Standpunktkoordinate (t_x, t_y, t_z) und der Instrumenten= höhe i_H . Eine Instrumentenhöhe kann nur bei horizontiertem Instrument angegeben werden. Die Horizontierung (Drehachse des Instruments steht senkrecht) kann rechnerisch mit zwei Restriktionsgleichungen $g_1(x)$ und $g_2(x)$ „erzwungen“ werden:

$$g_1(x) = q_1 = 0 \quad \text{und} \quad g_2(x) = q_2 = 0 \quad (3.37)$$

3.3.1 Inkrementelle Nährungsberechnung

Die Bedingungsgleichungen (3.42), (3.44) und (3.46) der lokalen Polarkoordinaten sind nicht linear: Bevor die Parameter der Rotation und Translation gemeinsam mit der Objektgeometrie berechnet werden, müssen deshalb Nährungswerte für diese standpunktbezogenen Parameter bestimmt werden. Zu diesem Zweck werden alle Beobachtungen vom Typ *lokale Polarkoordinate auf topologischem Primitiv* mit der Hesseschen Normalform als Pseudobeobachtung

$$l + v = 0 = \langle \vec{x}_i, \vec{n}_j \rangle - d_j \quad (3.38)$$

vom Typ *lokale Polarkoordinate i auf Ebene j* dargestellt.

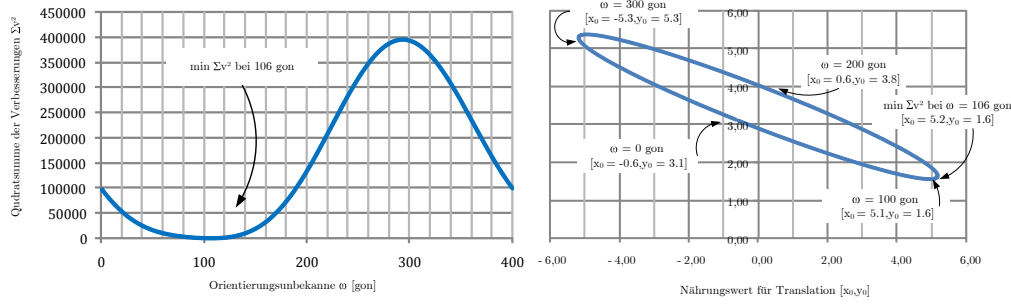


Abbildung 3.5: Inkrementelle Suche nach $\Sigma v^2 = \min$ (Inkrement = 1 gon)

Die Anzahl der Pseudobeobachtungen ergibt sich aus der Anzahl der (Schnitt-)ebenen am topologischen Primitiv:

- Eine Pseudobeobachtung pro lokaler Polarkoordinate auf Masche
- Mindestens zwei Pseudobeobachtungen pro lokaler Polarkoordinate auf Kante
- Mindestens drei Pseudobeobachtung pro lokaler Polarkoordinate auf Knoten

Die Beobachtungsgleichung ist:

$$l + v = 0 + v = \langle \vec{p}_i, \vec{n}_j \rangle - d_j \quad (3.39)$$

mit

$$\vec{p}_i = R_Q(q_0, q_1, q_2, q_3) \begin{bmatrix} s \cdot \sin(V) \cdot \cos(Hz) \\ s \cdot \sin(V) \cdot \sin(Hz) \\ s \cdot \cos(V) \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z + iH \end{bmatrix} \quad (3.40)$$

Werden ausschließlich die Translationskomponenten (t_x, t_y, t_z) als Unbekannte und Polarkoordinaten und Rotationsparameter als konstant in die Ausgleichung eingeführt, ergibt sich ein lineares Ausgleichungsproblem. Unter der Annahme, dass das Instrument horizontiert ist, kann die Rotation mit einem Freiheitsgrad (Rotation um z-Achse) ausgedrückt werden:

$$R_Q(i) = \left(q_0 = \cos\left(\frac{i}{2}\right), q_1 = 0, q_2 = 0, q_3 = \sin\left(\frac{i}{2}\right) \right) \quad (3.41)$$

Das Normalgleichungssystem der linearen Ausgleichung hat die Dimension 3×3 und wird mit einem Winkelinkrement von 1 gon für $i = 0 \text{ gon}, 1 \text{ gon}, \dots, 399 \text{ gon}$ berechnet (Abbildung 3.5). Das Ausgleichungsergebnis i mit der minimalen Verbesserungsquadratsumme $\sum v^2$ wird gewählt und die berechneten Parameter als Näherungswerte für die gemeinsame Ausgleichung verwendet (Abbildung 3.6).

3.3.2 Gemeinsame Ausgleichung

In der gemeinsamen Ausgleichung werden die Polarkoordinaten gemeinsam mit den anderen Beobachtungen und Bedingungen sowie der Objektgeometrie berechnet. Die Verbesserungen werden an den unkorrelierten Polarkoordinaten angebracht. Als Näherungswerte für die unbekannten Rotations- und Translationsparameter dienen die in der inkrementellen Näherungswertberechnung ermittelten Ergebnisse.

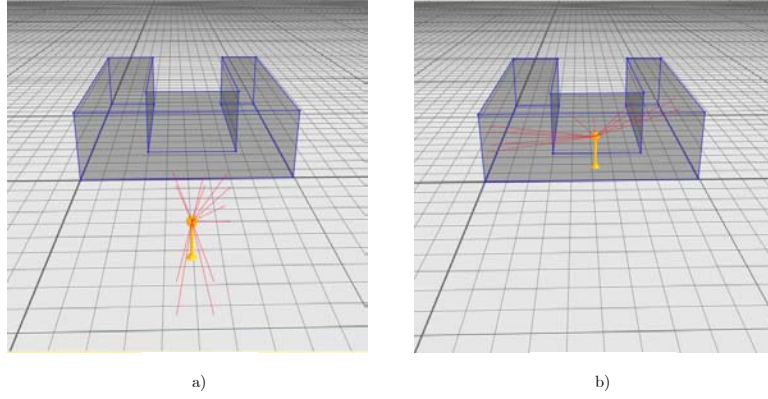


Abbildung 3.6: Startwert a) und Ergebnis b) der inkrementellen Suche

Lokale Polarkoordinate auf Masche. Die Beobachtungsgruppe i $(Hz, V, s)_{Stdpkt}^T$ wird von einem Standpunkt mit der Orientierung (q_0, q_1, q_2, q_3) auf dem Festpunkt (t_x, t_y, t_z) mit der Instrumentenhöhe iH aufgenommen. Die verbesserten Beobachtungen $(Hz + v_{Hz}, V + v_V, s + v_s)_{Stdpkt}^T$ müssen die Bedingung „auf Masche“ erfüllen. Die Masche liegt auf der Ebene j .

$$f(x, l + v) = \langle \vec{p}_i, \vec{n}_j \rangle - d_j = 0 \quad (3.42)$$

mit

$$\vec{p}_i = R_Q(q_0, q_1, q_2, q_3) \begin{bmatrix} (s + v_s) \cdot \sin(V + v_V) \cdot \cos(Hz + v_{Hz}) \\ (s + v_s) \cdot \sin(V + v_V) \cdot \sin(Hz + v_{Hz}) \\ (s + v_s) \cdot \cos(V + v_V) \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z + iH \end{bmatrix} \quad (3.43)$$

Lokale Polarkoordinate auf Kante. Die Beobachtungsgruppe i $(Hz, V, s)_{Stdpkt}^T$ wird von einem Standpunkt mit der Orientierung (q_0, q_1, q_2, q_3) auf dem Festpunkt (t_x, t_y, t_z) mit der Instrumentenhöhe iH aufgenommen. Die verbesserten Beobachtungen $(Hz + v_{Hz}, V + v_V, s + v_s)_{Stdpkt}^T$ müssen die Bedingung „auf Kante“ erfüllen. Die Kante wird durch Ebenenschnitt $j \cap k$ gebildet.

$$f(x, l + v) = |\vec{p}_i - \vec{p}_{jk}| = 0 \quad (3.44)$$

mit

$$\vec{p}_{jk} = \left[(\vec{n}_j, \vec{n}_k, \vec{n}_j \times \vec{n}_k)^T \right]^{-1} (d_j, d_k, \langle \vec{n}_j \times \vec{n}_k, \vec{p}_i \rangle)^T \quad (3.45)$$

und Gleichung (3.43).

Lokale Polarkoordinate auf Knoten. Die Beobachtungsgruppe i $(Hz, V, s)_{Stdpkt}^T$ wird von einem Standpunkt mit der Orientierung (q_0, q_1, q_2, q_3) auf dem Festpunkt (t_x, t_y, t_z) mit der Instrumentenhöhe iH aufgenommen. Die verbesserten Beobachtungen $(Hz + v_{Hz}, V + v_V, s + v_s)_{Stdpkt}^T$ müssen die Bedingung „auf Knoten“ erfüllen. Der Knoten wird durch Ebenenschnitt $j \cap k \cap l$ gebildet.

$$f(x, l + v) = |\vec{p}_i - \vec{p}_{jkl}| = 0 \quad (3.46)$$

mit

$$\vec{p}_{jkl} = \left[(\vec{n}_j, \vec{n}_k, \vec{n}_l)^T \right]^{-1} (d_j, d_k, d_l)^T \quad (3.47)$$

und Gleichung (3.43).

3.4 Winkelbedingungen zwischen topologischen Primitiven

Winkelbedingungen zwischen topologischen Primitiven ergeben sich aus der Objektgeometrie und der beobachtungsinduzierten Integrität. Da die Winkelbeziehung in fast allen Fällen entweder rechtwinklig oder parallel ist, wird die Beobachtung l für die Verwendung im Skalarprodukt als Cosinus des Winkels ausgedrückt.

Winkel Masche-Masche. Die Pseudobeobachtung $l = 0$ oder $l = 1$ der Winkelbeziehung zweier Maschen wird als Betrag des Skalarprodukts der Normalenvektoren ausgedrückt.

$$l + v = \left\{ \begin{array}{c} 0 + v \\ 1 + v \end{array} \right\} = |\langle \vec{n}_i, \vec{n}_j \rangle| = \left\{ \begin{array}{c} \dots orthogonal \\ \dots parallel \end{array} \right\} \quad (3.48)$$

Winkel Masche-Kante. Die Pseudobeobachtung $l = 0$ oder $l = 1$ der Winkelbeziehung zwischen einer Masche i und einer Kante $j \cap k$ wird als Betrag des Skalarprodukts des Normalenvektors i mit dem Kreuzprodukt der beiden Normalenvektoren der Kante $j \cap k$ ausgedrückt.

$$l + v = \left\{ \begin{array}{c} 0 + v \\ 1 + v \end{array} \right\} = |\langle \vec{n}_i, \vec{n}_j \times \vec{n}_k \rangle| = \left\{ \begin{array}{c} \dots parallel \\ \dots orthogonal \end{array} \right\} \quad (3.49)$$

Winkel Kante-Kante. Die Pseudobeobachtung $l = 0$ oder $l = 1$ der Winkelbeziehung zwischen zwei Kanten $i \cap j$ und $k \cap l$ wird als Betrag des Skalarprodukts der jeweiligen Kreuzprodukte der Normalenvektoren ausgedrückt.

$$l + v = \left\{ \begin{array}{c} 0 + v \\ 1 + v \end{array} \right\} = |\langle \vec{n}_i \times \vec{n}_j, \vec{n}_k \times \vec{n}_l \rangle| = \left\{ \begin{array}{c} \dots orthogonal \\ \dots parallel \end{array} \right\} \quad (3.50)$$

3.5 A priori Differenzen zwischen Abständen und Normalenvektoren

A priori Relativmaße l zwischen Ebenenabständen und Normalenvektoren werden aus den Nährungswerten der Objektgeometrie berechnet. Sie werden in die Ausgleichsrechnung eingefügt, wenn die tatsächlich gemessenen Beobachtungen das Modell nur teilweise beschreiben. Die Beobachtungen l_i werden *schwach gewichtet* und beeinflussen das Ausgleichsergebnis nur, wenn die stark gewichteten Messwerte diese Teile der Objektgeometrie nicht ausreichend beschreiben.

A priori Relativdistanzen. Für die geordneten Ebenenabstände der Nährungswerte $\{d_{01}, d_{02}, d_{03}, \dots, d_{0N}\}$ mit $d_{0i} \leq d_{0i+1}$ einer Gruppe gleichgerichteter Normalenvektoren werden zunächst die Pseudo-Abstandsbeobachtungen mit

$$l_i = \Delta d_i = d_{0i+1} - d_{0i} \quad (3.51)$$

erzeugt und dann als lineare Beobachtungsgleichung

$$l_i + v_i = \Delta d_i + v_i = f(x) = d_{i+1} - d_i \quad (3.52)$$

mit schwachen Gewicht in die Ausgleichung eingeführt.

A priori Relativwinkel. Die Beobachtung l_{ij} wird als Skalarprodukt der Normalenvektoren i und j aus den Nährungswerten der Parameter nx_0, ny_0, nz_0 mit

$$l_{ij} = \Delta n_{ij} = nx_{0i}nx_{0j} + ny_{0i}ny_{0j} + nz_{0i}nz_{0j} \quad (3.53)$$

berechnet und dann als Beobachtungsgleichung

$$l_{ij} + v_{ij} = \Delta n_{ij} + v_{ij} = f(x) = nx_i nx_j + ny_i ny_j + nz_i nz_j \quad (3.54)$$

mit schwachen Gewicht in die Ausgleichung eingeführt.

3.6 Datumsfestlegung

Relativbeobachtungen (Abstandsmessungen, Polarkoordinaten auf topologischen Primitiven, a priori Relativmaße) stellen den Bezug zwischen den Parametern der unbekannten Objektgeometrie her. Sie stellen *nicht* den Bezug zum Koordinatensystem her. Es gibt unendliche viele gültige Lösungen für eine ausgeglichene Objektgeometrie. Das Normalgleichungssystem hat daher einen Rangdefekt. „Es ist somit erforderlich, über die Lagerung

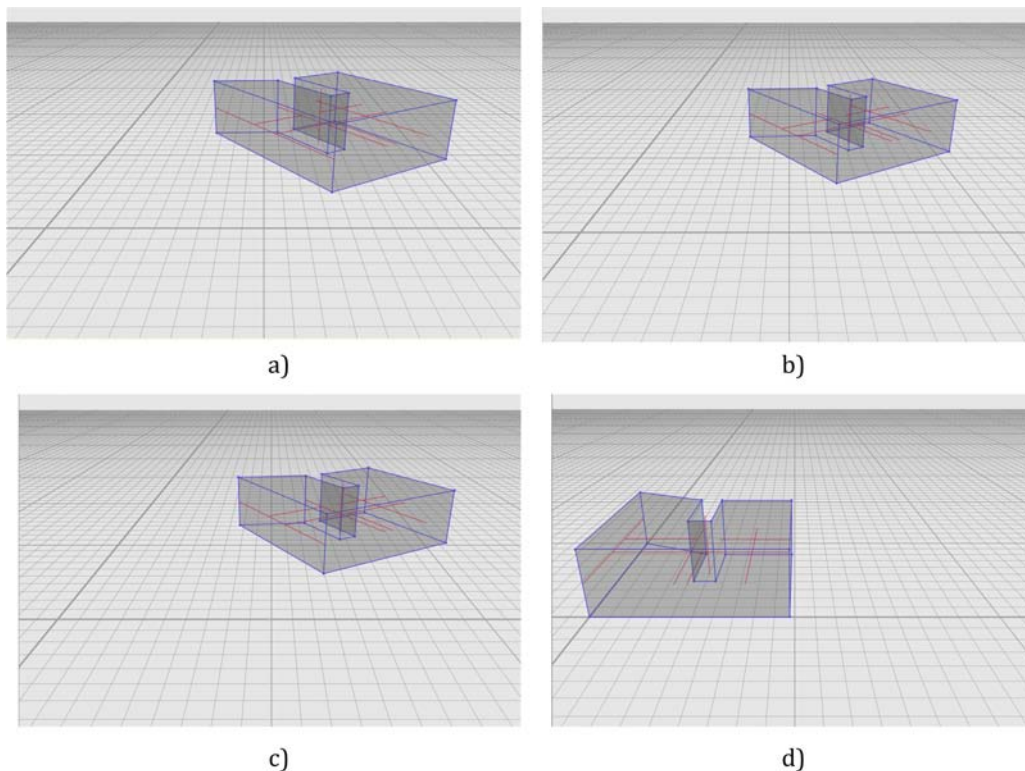


Abbildung 3.7: a) Nährungswerte b) Datum mit Identitätsbeobachtungen c) Datum mit Teilspurminimierung d) Datum mit Festhalten

und Orientierung eines Netzes (hier: der Objektgeometrie) im Koordinatensystem durch einen separaten Schritt, der **Datumsfestlegung**, zu verfügen“ [Niemeier 2002, S. 206].

In einem dreidimensionalen Koordinatensystem mit festem Maßstab müssen 6 Freiheitsgrade festgelegt werden, um das singuläre Normalgleichungssystem zu lösen. Die in dieser Arbeit entwickelte Software unterstützt drei Möglichkeiten der Datumsfestlegung, die in Analogie zur geodätischen Netzausgleichung auch nacheinander bearbeitet werden können:

1. Teilspurminimierung auf der Objektgeometrie (freies Datum)
2. Regularisierung mit Identitätsbeobachtungen (weiches Datum)
3. Festhalten einzelner Parameter (Varianzfreie Rechenbasis oder Ausgleichung unter Zwang)

Die Regularisierung mit Identitätsbeobachtungen hat darüber hinaus noch die Funktion einen ggf. vorhandenen **Konfigurationsdefekt** zu beseitigen. Ein Konfigurationsdefekt liegt vor, wenn nicht alle Elemente der Objektgeometrie durch Messungen oder Bedingungen berechnet werden können– das Modell ist nicht vollständig invers beschrieben. In diesem Fall verwendet die Software die ungenaue Information der 3D-Skizze, die bei der dreidimensionalen Spezifikation der Objekttopologie zur „ungefähren“ Visualisierung der Benutzerinteraktion berechnet wird.

3.6.1 Teilspurminimierung auf Objektgeometrie (freies Datum)

Die **Teilspurminimierung auf der Objektgeometrie** ist eine frei gelagerte Ausgleichung – es wird durch die Datumverfügung kein Zwang auf die Beobachtungen ausgeübt. Im Gegensatz zu einer varianzfreien Rechenbasis [Niemeier 2002, S. 233] kann die Teilspurminimierung ohne eine Festlegung des Benutzers erfolgen. Als eine Variante der „freien“ Ausgleichung ist dieses Verfahren zum Suchen grober Fehler im Beobachtungsmaterial geeignet.

Bei einer Teilspurminimierung nehmen alle (!) *Parameter der Objektgeometrie* an der Datumsfestlegung teil. Geometrisch veranschaulicht bedeuten die eingeführten 6 Restriktionsgleichungen, dass sich die Summe der Translationen und Rotationen zu Null ergibt.

Die differentielle Änderung dd_i eines Ebenenabstandes d_i aufgrund einer differentiellen Verschiebung dt_x, dt_y, dt_z der Datumskomponenten t_x, t_y, t_z ergibt sich aus dem totalen Differential

$$dd_i(dt_x, dt_y, dt_z) = n_{ix}dt_x + n_{iy}dt_y + n_{iz}dt_z \quad (3.55)$$

Die **translatorische** Datumsfestlegung erfolgt durch die Forderung, dass sich diese differentiellen Änderungen in ihrer Summe aufheben:

$$g_{tx}(x) = g_{tx}(dt_x) = \sum_{i=1}^{i=N} n_{0ix}dt_x = 0 \quad (3.56)$$

$$g_{ty}(x) = g_{ty}(dt_y) = \sum_{i=1}^{i=N} n_{0iy}dt_y = 0 \quad (3.57)$$

$$g_{tz}(x) = g_{tz}(dt_z) = \sum_{i=1}^{i=N} n_{0iz}dt_z = 0 \quad (3.58)$$

Bei der Lösung des Normalgleichungssystems wird die Normalgleichungsmatrix mit der Bedingungsmatrix G_T geändert.

$$G_T = \begin{bmatrix} d_1 & d_2 & \cdots & d_N \end{bmatrix} \cdots \begin{bmatrix} n_{1x} & n_{1y} & n_{1z} & \cdots & n_{Nx} & n_{Ny} & n_{Nz} \end{bmatrix} \quad (3.59)$$

$$\left[\begin{bmatrix} n_{01x} & n_{02x} & \cdots & n_{0Nx} \\ n_{01y} & n_{02y} & \cdots & n_{0Ny} \\ n_{01z} & n_{02z} & \cdots & n_{0Nz} \end{bmatrix} \cdots \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 \end{bmatrix} \right]$$

Die differentielle Änderungen dn_i der Komponenten n_{ix}, n_{iy}, n_{iz} eines Normalenvektor \vec{n}_i (**Rotation**) aufgrund einer differentiellen Drehung $d\phi, d\lambda$ und $d\kappa$ der Datumskomponenten ϕ, λ und κ ergibt sich aus den totalen Differentialen

$$\begin{aligned} dn_{ix}(d\phi, d\lambda, d\kappa) &= 0 \cdot d\phi + n_{iz} \cdot d\lambda - n_{iy} \cdot d\kappa \\ dn_{iy}(d\phi, d\lambda, d\kappa) &= -n_{iz} \cdot d\phi + 0 \cdot d\lambda + n_{ix} \cdot d\kappa \\ dn_{iz}(d\phi, d\lambda, d\kappa) &= n_{iy} \cdot d\phi - n_{ix} \cdot d\lambda + 0 \cdot d\kappa \end{aligned} \quad (3.60)$$

Die rotatorische Datumsfestlegung erfolgt durch die Forderung, dass sich diese differentiellen Änderungen in ihrer Summe aufheben:

$$\begin{aligned} g_{rx}(x) &= g_{rx}(d\phi) = \sum_{i=1}^{i=N} (n_{0iy} - n_{0iz}) d\phi = 0 \\ g_{ry}(x) &= g_{ry}(d\lambda) = \sum_{i=1}^{i=N} (n_{0iz} - n_{0ix}) d\lambda = 0 \\ g_{rz}(x) &= g_{rz}(d\kappa) = \sum_{i=1}^{i=N} (n_{0ix} - n_{0iy}) d\kappa = 0 \end{aligned} \quad (3.61)$$

Bei der Lösung des Normalgleichungssystems wird die Normalgleichungsmatrix mit der Bedingungsmatrix G_R geändert.

$$G_R = \frac{\begin{bmatrix} d_1 & d_2 & \cdots & d_N \end{bmatrix} \cdots \begin{bmatrix} n_{1x} & n_{1y} & n_{1z} & \cdots & n_{Nx} & n_{Ny} & n_{Nz} \end{bmatrix}}{\begin{bmatrix} \begin{bmatrix} 0 & 0 & \cdots & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 & \cdots & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 & \cdots & 0 \end{bmatrix} \end{bmatrix} \cdots \begin{bmatrix} 0 & n_{01y} & -n_{01z} & \cdots & 0 & n_{0Ny} & -n_{0Nz} \\ -n_{01x} & 0 & n_{01z} & \cdots & -n_{0Nx} & 0 & n_{0Nz} \\ n_{01x} & -n_{01y} & 0 & \cdots & n_{0Nx} & -n_{0Ny} & 0 \end{bmatrix}} \quad (3.62)$$

3.6.2 Regularisierung mit Identitätsbeobachtungen (weiches Datum)

Eine weiche Lagerung kann mittels einer **Regularisierung mit schwach gewichteten Identitätsbeobachtungen** modelliert werden. Hier werden die Nährungswerte als schwach gewichtete Identitätsbeobachtung in die Ausgleichung eingeführt. Die Identitätsbeobachtung dient zunächst der Datumsverfügung. Darüber hinaus haben die Identitätsbeobachtungen einen Einfluss auf das Ausgleichungsergebnis der Objektgeometrie, wenn die Messung unvollständig ist. Die Regularisierung mit Identitätsbeobachtungen beseitigen auch einen *Konfigurationsdefekt*.

Für alle Parameter der Objektgeometrie wird eine Pseudobeobachtung $l_i = x_{0i}$ in die Ausgleichung eingeführt, die den Bezug zum Koordinatensystem durch direkte „Beobachtung“ der absoluten Geometrie herstellt. Die Identitätsbeobachtung wird im stochastischen Modell schwach gewichtet.

$$l_i + v_i = x_{0i} + v_i = f(x) = x_i \quad (3.63)$$

3.6.3 Festhalten einzelner Parameter (starkes Datum)

Mit dem **Festhalten** einzelner Parameter kann die Objektgeometrie fest mit dem Koordinatensystem verknüpft werden. Das Festhalten wird als Restriktion in die Ausgleichung eingeführt.

$$g(x) = x_i - x_{0i} = 0 \quad (3.64)$$

Zur vollen Beseitigung des Datumsdefekts müssen drei linear unabhängige Ebenenabstände d und drei linear unabhängige Rotationen durch Festgehalten von Normalenvektor-Parametern festgelegt werden. Dabei fixiert das Festhalten eines Normalenvektors (nur) zwei Freiheitsgrade, da sich die dritte Komponente aufgrund der Normierung

$$|\vec{n}_0| = 1 \quad (3.65)$$

stets berechnen lässt. Werden mehr Parameter als zur Beseitigung des Datumsdefekts nötig festgehalten, übt das Festhalten einen Zwang auf das Ausgleichungsergebnis aus.

4 Automatisches Differenzieren

Das Automatische Differenzieren dient dem *Programmierer* als eine Methode zum schnellen und robusten Entwickeln von Algorithmen, die eine Linearisierung erfordern und ist sehr gut für den Einsatz in *generischer Software* geeignet.

Die numerischen Ergebnisse sind die gleichen wie bei der analytischen Methode, die allerdings ein händisches Programmieren der Ableitungen erfordert. Die in Kapitel 3 angegebenen Beobachtungs- und Bedingungsgleichungen enthalten zum Teil Funktionen, die in mehreren Teilschritten (Koordinatentransformation, Ebenenschnitt, Invertieren) modelliert sind. Durch die Verwendung der Differentiationsarithmetik kann eine korrekte Differentiation - unter der Voraussetzung eines korrekten funktionalen Modells - stets garantiert werden.

Differenzieren ist Handwerkszeug. Schon in der Schule lernt man die Ableitungsregeln, mit denen man eine Formel $f'(x)$ für die Ableitung einer Funktion $f(x)$ erzeugt. Numerisches Differenzieren ist ebenfalls ein bekanntes Verfahren zur Berechnung des Wertes der Ableitung an einer Stelle x . Automatisches Differenzieren (AD) - auch Algorithmisches Differenzieren genannt - ist hingegen selten bekannt, obwohl es auf einfachen algebraischen Regeln und einfachen Implementierungen beruht. Beide der erstgenannten bekannten Verfahren haben Schwächen bei der Implementierung von Algorithmen in Computer-Programmen: Das analytische Ableiten, also das Codieren der Formel der Ableitung im Quelltext, erfordert deren *symbolische Darstellung*, die ggf. schwer zu finden ist. Die Codeerzeugung ist *fehleranfällig* und muss stets aufwendig (zum Beispiel durch den Vergleich mit der numerischen Ableitung) geprüft werden. Das numerische Ableiten beruht auf der Angabe eines (kleinen) numerischen Wertes für den Ausdruck Δx des Differenzenquotienten:

$$f'(x) = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x) - f(x)}{\Delta x} \quad (4.1)$$

Beim Verfahren der numerischen Differentiation treten Rundungsfehler aufgrund der Rechengenauigkeit und des Verfahrensfehlers auf, wobei letzterer entsteht, weil man die Sekante und nicht die Tangente berechnet. Das Ergebnis ist stark abhängig von der Größe Δx , die aufgrund der begrenzten Rechengenauigkeit nicht beliebig klein gewählt werden kann. AD bietet die *Möglichkeit auf die Codierung der Ableitungsformel zu verzichten*, ohne dass Rundungsfehler auftreten. Die Differentiationsarithmetik ist nicht auf eindimensionale Funktionen mit einer Variablen beschränkt. Für die Berechnung der Ableitungen beliebiger Ordnung $f^i(x)$, aller partieller Ableitungen einer Funktion, Gradienten, Jakobimatrizen oder Hesse-Matrizen wird *der gleiche Quellcode* wie für die Berechnung des eigentlichen Funktionswertes genutzt.

4.1 Differentiationsarithmetik

Mit der Verwendung der Differentiations-Arithmetik [Rall 1986] ist es möglich, sowohl den Wert einer Funktion $f(x)$, als auch die dazugehörige Ableitung $f'(x)$ in einer Variablen zu vereinen. Dazu werden, ähnlich wie bei den komplexen Zahlen, geordnete Paare von Zahlen verwendet.

$$U = [u, u'] \quad V = [v, v'] \quad U, V \in \mathbb{R}^2 \quad (4.2)$$

Sind für diese Arithmetik die Grundrechenoperationen und Elementarfunktionen definiert, so kann neben der Berechnung eines Wertes simultan die Berechnung der Ableitung(en) ausgeführt werden. Einfache Operationen werden durch das Einführen eines Infinitesimalteil d mit der Rechenregel $d^2 = 0$ begründet.

Die Rechenregeln $+$, $-$, \cdot , \div für $U = [u, u'd]$, $V = [v, v'd]$ mit $U, V \in D$ können einfach aus der Rechenregel $d^2 = 0$ hergeleitet werden:

Summe:

$$U + V = [u, u'd] + [v, v'd] = [(u + v), (u' + v')d] \quad (4.3)$$

Differenz:

$$U - V = [u, u'd] - [v, v'd] = [(u - v), (u' - v')d] \quad (4.4)$$

Produkt:

$$U \cdot V = [u, u'd] \cdot [v, v'd] = uv + uv'd + u'vd + u'v'd^2 = [uv, (uv' + u'v)d] \quad (4.5)$$

Quotient:

$$\frac{U}{V} = \frac{[u, u'd]}{[v, v'd]} = \frac{uv - uv'd + u'vd - u'v'd^2}{v^2 - v'vd + v'vd - v'd^2} = \left[\frac{u}{v}, \frac{u'v - uv'}{v^2} d \right] \quad (4.6)$$

Dabei entspricht der Realteil der reellen Arithmetik und der Infinitesimalteil d der korrespondierenden Ableitungsregel. Jede differenzierbare Funktion $f: \mathbb{R} \rightarrow \mathbb{R}$ kann nach $f: D \rightarrow D$, in die Differentiationsarithmetik erweitert werden. Die reellen Zahlen \mathbb{R} sind in D mit $U = [u, 0]$, $u \in \mathbb{R}$ im Zahlenraum der Differentiationsarithmetik enthalten. Das Konzept, die Ableitungsregel dem Infinitesimalteil d zuzuordnen, kann auf beliebige differenzierbare Funktionen übertragen und mit der bekannten Kettenregel „Innere mal äußere Ableitung“ erweitert werden:

$$f([u, u']) = [f(u), u'f'(u)] \quad f(u, du) = [f(u), du \cdot df(u)] \quad (4.7)$$

Die Elementarfunktionen ergeben sich aus den bekannten“ Ableitungsregeln:

$$e^U = e^{[u, u']} = [e^u, u'e^u] \quad (4.8)$$

$$\sin(U) = \sin([u, u']) = [\sin(u), u' \cdot \cos(u)] \quad (4.9)$$

$$\cos(U) = \cos([u, u']) = [\cos(u), u' \cdot (-1) \cdot \sin(u)] \quad (4.10)$$

$$\arctan(U) = \arctan([u, u']) = \left[\arctan(u), u' \cdot \frac{1}{1 + u^2} \right] \quad (4.11)$$

Da sich der differenzierbare Formelausdruck der Beobachtungs- und Bedingungsgleichungen aus den arithmetischen Operatoren und den definierten Elementarfunktionen zusammensetzt, lässt er sich in eine Sequenz von elementaren Ausdrücken zerlegen. Beim Überführen einer Funktion $f: \mathbb{R} \rightarrow \mathbb{R}$ nach $f: D \rightarrow D$ wird jede Variable x durch $X = [x, 1]$ und jede Konstante c durch $C = [c, 0]$ ersetzt. Dies entspricht den allgemeinen Regeln des Differenzierens:

$$\frac{dx}{dx} = 1 \quad (4.12)$$

und

$$\frac{dc}{dx} = 0 \quad (4.13)$$

4.2 Beispiel

Bei der Auswertung terrestrischer Lagenetze muss bei der Verwendung des Richtungswinkels (Azimuth) häufig die trigonometrische Funktion *arctan* differenziert werden. Als Beispiel sei nur die partielle Ableitung $\frac{\partial f}{\partial x_2}$ der Funktion

$$y = f(x_1, x_2, y_1, y_2) = \arctan\left(\frac{y_2 - y_1}{x_2 - x_1}\right) \quad (4.14)$$

nach der x-Koordinate des Zielpunktes gesucht. Der Formelausdruck für die partielle Ableitung

$$\frac{\partial f}{\partial x_2} = \frac{y_1 - y_2}{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (4.15)$$

wird beim analytischen Differenzieren mit einer „Formelsammlung“ hergeleitet und anschließend im Quelltext codiert. Zur Laufzeit werden die Variablen mit Werten besetzt und die Formelausdrücke ausgewertet

$$y = f(x_1 = 2, x_2 = 13, y_1 = 5, y_2 = 19) = 0.9048 \quad (4.16)$$

$$\frac{\partial f(x_1 = 2, x_2 = 13, y_1 = 5, y_2 = 19)}{\partial x_2} = -0.0442 \quad (4.17)$$

Anders bei der Differentiationsarithmetik: Hier wird die Funktion f nach $f : D \rightarrow D$ überführt, indem die Infinitesimalteile der Konstanten mit $C = [c, 0]$ und die unabhängige Variablen x_2 mit

$$X_2 = \left[x_2, \frac{dx}{dx} \right] = [x_2, 1] = [13, 1] \quad (4.18)$$

initialisiert werden.

$$\begin{aligned} Y &= \arctan\left(\frac{[19, 0] - [5, 0]}{[13, 1] - [2, 0]}\right) = \arctan\left(\left[\frac{14}{11}, \frac{-14}{11^2}\right]\right) = \arctan([1.2727, -0.1157]) \\ &= \left[\arctan(1.2727), \frac{1}{1 + (1.2727)^2} \cdot (-0.1157) \right] = [0.9048, -0.0442] \end{aligned} \quad (4.19)$$

Die Ableitung ist **gemeinsam mit** der Auswertung des Funktionswertes berechnet worden. Ein spezieller Formelausdruck für $\frac{\partial f}{\partial x_2}$ wurde dabei nicht benötigt. Bei der Berechnung der *arctan*-Funktion mit der Differentiationsarithmetik wird die Ableitung numerisch exakt ausgewertet, ohne dass die Ableitung mit eigenem Quellcode programmiert werden muss.

4.3 Vorwärts oder Rückwärts?

Eine Funktion beliebiger Komplexität wird in einem Computerprogramm in eine Sequenz von elementaren Operationen $\{f_i\}_{i=1, \dots, n}$ zerlegt, wobei f_i jeweils lediglich abhängig ist von den Zwischenergebnissen $\tau_1, \dots, \tau_{i-1}$ und $\tau_i = f_i\{\tau_1, \dots, \tau_{i-1}\}$. Dabei können auch bedingte Anweisungen (**if**), Verzweigungen (**switch**) oder Schleifen (**do, for**) Bestandteil einer Methode sein. Einige der Zwischenergebnisse τ_i werden korrespondierend zu den Variablen des Programms an verschiedenen Stellen im Speicher abgelegt und überschreiben dabei einen alten Wert der gleichen Variablen oder es wird freigegebener Speicherplatz verwendet. Andere Variablen müssen für die Verfügbarkeit in mehreren Operationen in einem temporären Register gespeichert werden, bevor der Speicherplatz freigegeben werden kann.

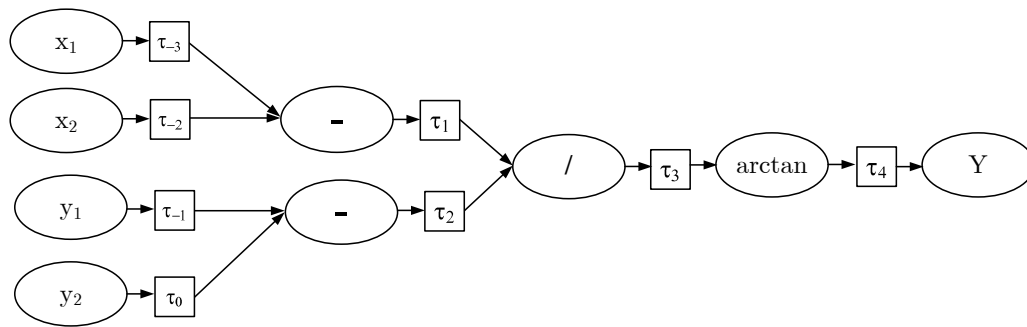


Abbildung 4.1: Berechnungsgraph Funktion

Das obige Beispiel

$$y = f(x_1, x_2, y_1, y_2) = \arctan\left(\frac{y_2 - y_1}{x_2 - x_1}\right) \quad (4.20)$$

mit den Startwerten $x_1=2$, $x_2 = 13$, $y_1 = 5$, $y_2 = 19$ wird für die Berechnung des Funktionswertes y in die folgende Sequenz zerlegt:

Variable	Formel	Nebenrechnung	Ergebnis
τ_{-3}	$= x_1$	$= 2$	
τ_{-2}	$= x_2$	$= 13$	
τ_{-1}	$= y_1$	$= 5$	
τ_0	$= y_2$	$= 19$	
τ_1	$= \tau_0 - \tau_{-1}$	$= 19 - 5$	$= 14$
τ_2	$= \tau_{-2} - \tau_{-3}$	$= 13 - 2$	$= 11$
τ_3	$= \tau_1 / \tau_2$	$= 14/11$	$= 1.2727$
τ_4	$= \arctan(\tau_3)$	$= \arctan(1.2727)$	$= 0.9048$
y	$= \tau_4$	$= 0.9048$	

Tabelle 4.1: Berernungssequenz der Funktion

Die Berechnung kann graphisch mit einem Berechnungsdiagramm (computational graph) dargestellt werden. Eine wichtige Regel dabei ist, dass keine mathematische Variable mehr als einmal deklariert wird. Die Berechnung beginnt mit der Initialisierung der unabhängigen Variablen x_i . Jede Zwischenvariable τ_i ergibt sich unter Anwendung von Elementaroperationen auf vorher definierten Zwischenvariablen.

Damit ist eine Form erreicht, die eine effiziente Berechnung gewährleistet und den Einsatz der Kettenregel sehr begünstigt. Die Kettenregel wird nach folgenden zwei Ansätzen implementiert:

1. Es werden für jede Variable im Laufe der Berechnungskette simultan die partiellen Ableitungen nach den gewünschten Variablen mitgeführt (Forward-Methode).
2. Es wird zunächst der Funktionswert berechnet und dabei der Berechnungsablauf gespeichert. Anschließend kann ausgehend vom Ende der Berechnungen das aufgezeichnete Berechnungsdiagramm von hinten abgearbeitet werden (Backward-Methode).

Forward-Methode. Zur Erläuterung dieser Methode soll wieder beispielhaft die Ableitung $\frac{\partial f}{\partial x_2}$ der Funktion

$$y = f(x_1, x_2, y_1, y_2) = \arctan\left(\frac{y_2 - y_1}{x_2 - x_1}\right)$$

bestimmt werden. Eine Möglichkeit der Lösung besteht darin, für jeden Wert τ_i der Sequenz in Tabelle 4.2 auch den Wert der Ableitung nach x_2 , also $\dot{\tau}_i = \partial \tau_i / \partial x_2$ mitzuführen. Damit ist x_2 die einzige unabhängige Variable und y eine abhängige Variable.

Die Sequenz der Zwischenergebnisse ist in Tabelle 4.2 aufgelistet und in Abbildung 4.2 visualisiert.

Variable	Formel	Nebenrech.	Ergebnis
$\dot{\tau}_{-3}$	dc/dx	$= 0$	$= 0$
$\dot{\tau}_{-2}$	$\mathbf{d}x/\mathbf{d}x$	$= 1$	$= 1$
$\dot{\tau}_{-1}$	dc/dx	$= 0$	$= 0$
$\dot{\tau}_0$	dc/dx	$= 0$	$= 0$
$\dot{\tau}_1$	$(\partial\tau_1/\partial\tau_{-3})\dot{\tau}_{-3} + (\partial\tau_1/\partial\tau_{-2})\dot{\tau}_{-2} = (-1)\dot{\tau}_{-3} + (1)\dot{\tau}_{-2}$	$= 0+1$	$= 1$
$\dot{\tau}_2$	$(\partial\tau_2/\partial\tau_{-1})\dot{\tau}_{-1} + (\partial\tau_2/\partial\tau_0)\dot{\tau}_0 = (-1)\dot{\tau}_{-1} + (1)\dot{\tau}_0$	$= 0+0$	$= 0$
$\dot{\tau}_3$	$(\partial\tau_3/\partial\tau_1)\dot{\tau}_1 + (\partial\tau_3/\partial\tau_2)\dot{\tau}_2 = (1/\tau_2)\dot{\tau}_1 + (-\tau_1/\tau_2^2)\dot{\tau}_2$	$= 0+14/11^2$	$= -0.1157$
$\dot{\tau}_4$	$(\partial\tau_4/\partial\tau_3)\dot{\tau}_3 = (1/(1+\tau_3^2))\dot{\tau}_3$	$= -0.0442$	$= -0.0442$
$\partial f/\partial x_2$	$\dot{\tau}_4$	$= -0.0442$	

Tabelle 4.2: Berernungssequenz der Forward-Methode

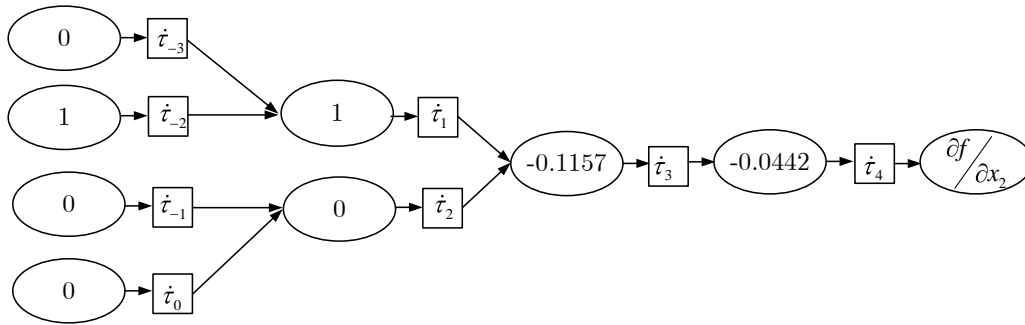


Abbildung 4.2: Berechnungsgraph Forward Methode

Die Vorgehensweise wird *Forward* genannt, da die Ableitungen $\dot{\tau}_i$ für jeden Berechnungsschritt mitgeführt werden. Das Problem ein Programm so zu transformieren, dass die zusätzlich benötigten Variablen und Berechnungsvorschriften zur Verfügung gestellt werden, wird in Kapitel 4.5 behandelt.

Die Backward-Methode. Wurden im ersten Ansatz die *unabhängigen* Variablen ausgewählt und für jede Zwischenvariable die Ableitung nach diesen bestimmt, so werden hier die *abhängigen* Variablen gewählt und die Ableitungen dieser nach den Zwischenwerten berechnet, um den Wert der Ableitung zu erhalten. Die einzige abhängige Variable in dem Beispiel ist y . Für jede Variable τ_i existiert eine Ableitung $\bar{\tau}_i = \partial y / \partial \tau_i$, welche auch als *adjungierte Variable* bezeichnet wird. Um diesen entscheidenden Unterschied zur Forward-Methode zu kennzeichnen, wird für die partiellen Ableitungen statt $\dot{\tau}_i$ nun die Notation $\bar{\tau}_i$ verwendet. Die adjungierten Variablen können wieder mit der Kettenregel und Operatorüberladung automatisch bestimmt werden. Dazu muss der Berechnungsgraph *rückwärts* durchschritten werden. Der Startwert für die einzige abhängige Variable lautet für das Beispiel: $\bar{y} = \bar{\tau}_4 = dy/dy = 1$. τ_4 ist mit $\tau_4(\tau_3) = \arctan(\tau_3)$ abhängig von τ_3 . Die Variable $\bar{\tau}_3$ berechnet sich mittels partieller Ableitung des arctan mit: $\bar{\tau}_3 = \bar{\tau}_4 / (1 + \tau_3^2)$. Abbildung 4.3 gibt die Reihenfolge der Operationen im Backward-Modus wieder. Die Sequenz der Zwischenergebnisse ist in Tabelle (4.3) wiedergegeben.

Das Ergebnis stimmt mit dem aus der Forward-Methode exakt überein, allerdings wurden hier alle partiellen Ableitungen bestimmt.

Vergleich. Die Forward-Methode sollte dann angewendet werden, wenn die Anzahl der abhängigen Variablen (Output) größer ist als die Anzahl der unabhängigen Variablen (Input). Hingegen sollte die Backward Methode gewählt werden, wenn wenige abhängige Variablen aus vielen unabhängigen Variablen berechnet werden. Diese generellen Aussagen bestätigen sich in der Arbeit von [Kersten 2008], die beide Verfahren an beispielhaften Aufgaben der Ausgleichsrechnung und Ingenieurgeodäsie anwendet und vergleicht.

Variable	Formel	Ergebnis
$\bar{\tau}_4$	$= dy/dy$	$= 1$
$\bar{\tau}_3$	$= \bar{\tau}_4 \cdot (\partial\tau_4/\partial\tau_3) = \bar{\tau}_4 \cdot (1/(1+\tau_3^2))$	$= 0.3817$
$\bar{\tau}_2$	$= \bar{\tau}_3 \cdot (\partial\tau_3/\partial\tau_2) = \bar{\tau}_3 \cdot (1/\tau_2)$	$= 0.0347$
$\bar{\tau}_1$	$= \bar{\tau}_3 \cdot (\partial\tau_3/\partial\tau_1) = \bar{\tau}_3 \cdot (-\tau_1/\tau_2^2)$	$= -0.0442$
$\bar{\tau}_0$	$= \bar{\tau}_2 \cdot (\partial\tau_2/\partial\tau_0) = \bar{\tau}_2 \cdot (1)$	$= 0.0347$
$\bar{\tau}_{-1}$	$= \bar{\tau}_2 \cdot (\partial\tau_2/\partial\tau_{-1}) = \bar{\tau}_2 \cdot (-1)$	$= -0.0347$
$\bar{\tau}_{-2}$	$= \bar{\tau}_1 \cdot (\partial\tau_2/\partial\tau_{-2}) = \bar{\tau}_1 \cdot (1)$	$= -0.0442$
$\bar{\tau}_{-3}$	$= \bar{\tau}_1 \cdot (\partial\tau_2/\partial\tau_{-3}) = \bar{\tau}_1 \cdot (-1)$	$= 0.0442$

Tabelle 4.3: Berernungssequenz der Backward-Methode

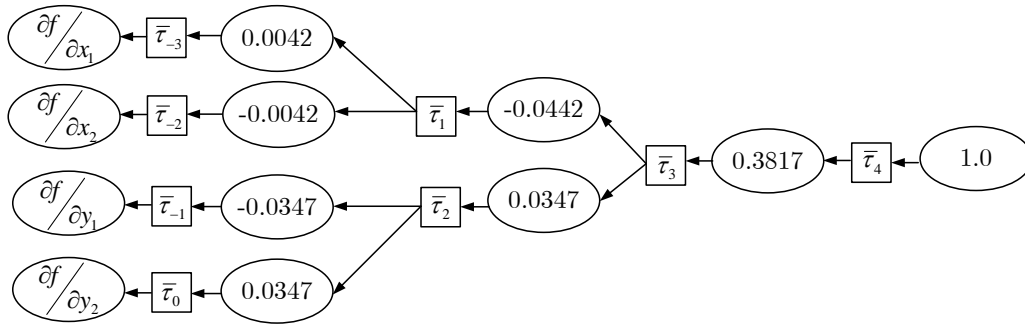


Abbildung 4.3: Berechnungsgraph Backward-Methode

4.4 Erweiterung der Differentiationsarithmetik

Die vorgestellte Arithmetik wird mit zwei Zielen erweitert:

1. Berechnung aller partiellen Ableitungen $\nabla f = \text{grad}(f) = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_u} \end{pmatrix}$ einer Funktion $f : \mathbb{R}^u \rightarrow \mathbb{R}^1$
2. Berechnung beliebiger höherer Ableitungen $f^i(x)$ mit $i > 1$

Durch eine Erweiterung des skalaren Infinitesimalteil d in Gleichung (4.2) zu einem Vektor \vec{d} der Dimension u werden **alle partiellen Ableitungen gleichzeitig berechnet**. Dabei entspricht u der Anzahl der unabhängigen Variablen der Funktion $f(x_1, \dots, x_u)$.

$$V = \begin{bmatrix} v'_1 \\ v \\ \vdots \\ v'_u \end{bmatrix} \quad V = [v, \nabla v] \quad (4.21)$$

mit

Die Operationen und Funktionen der Differentiationsarithmetik werden für jede Dimension des Infinitesimalteil getrennt - nach den oben beschriebenen Regeln - durchgeführt. Alle Elemente des Zahlenraums D^u sind reelle Zahlen \mathbb{R} .

Im Kontext der *Popa3d* Ausgleichungsrechnung wird jedem Gleichungsobjekt ein Gradient (4.21) zugeordnet. Der Gradient ∇v repräsentiert dann die partiellen Ableitungen der Beobachtungs- bzw. Bedingungsgleichung „nach“ den zu schätzenden Unbekannten x und Residuen v . Die Dimension des Gradienten ergibt sich nicht aus der Anzahl der Parameter des Gesamtproblems, sondern aus der kleinen Anzahl der Parameter einer

bestimmten Gleichung. In Abschnitt 5.2.1 (Seite 56) wird die informationstechnische Umsetzung dieses Ansatzes erklärt.

Für die Berechnung der **Ableitungen i-ter Ordnung** sind die Infinitesimalteile nicht reelle Zahlen, sondern selber Elemente des Zahlenraums D^u

$$V = \left[v, \begin{bmatrix} v_1^i \\ \vdots \\ v_u^i \end{bmatrix} \right] = \left[v, \begin{bmatrix} [v_1^i, \nabla v_1^{i-1}] \\ \vdots \\ [v_u^i, \nabla v_u^{i-1}] \end{bmatrix} \right] \quad (4.22)$$

Die Ableitung ist im Falle der 2ten-Ordnung die Hessematrix H . Für die Ableitungen höherer Ordnung ist die Ableitung, ganz allgemein, ein Tensor i-ter Stufe. Interessante Möglichkeiten im Kontext der Ausgleichsrechnung bietet das Automatische Differenzieren wenn als abzuleitende Funktion direkt die Zielfunktion $\Omega = v^t v \rightarrow \min$ angesetzt wird und das Minimum dieser Funktion mittels Newton-Raphson-Verfahren gefunden wird. [Kersten 2008] untersucht die Anwendung der Differentiationsarithmetik für Aufgaben der Ausgleichsrechnung mit dem Newton-Raphson-Verfahren, Haley-Verfahren und zeigt, wie beliebige M-Schätzer generisch und ohne Regewichtung der Gewichtsmatrix programmiert werden können.

4.5 OO-Implementierung

Die in dieser Arbeit erstellte Anwendungssoftware berechnet alle Ableitungen mit der frei verfügbaren¹ C++ Bibliothek FADBAD++ [Bendtsen und Stauing 1996]. Für die Umsetzung in Programmen deklariert FADBAD++ zunächst einen generischen Datentyp für die Forward- und Backward-Methoden und deklariert dann mittels C++ Operatorüberladung und statischen C++ Funktionen die wichtigsten binären und unären Operationen.

Operatorüberladung. Definiert man einen neuen Datentyp (z.B. geordnete Paare von Zahlen), so sind die Operatoren $+$, $-$, \cdot , \div für diesen Typ zunächst nicht definiert und eine Anwendung ist nicht möglich. In der Computersprache C++ hat man deshalb die Möglichkeit der Operatorüberladung, wodurch die Funktionsweise bzw. Bedeutung der Operatoren nach der jeweils geltenden Arithmetik bzw. des verwendeten Typs angepasst werden kann. Für Datentypen von FADBAD++ wird dann die differentielle Arithmetik verwendet. Der Quellcode für die Funktion (Beobachtungs, Bedingungsgleichung) unterscheidet sich lediglich im Datentyp: Statt double oder float wird der Datentyp AD der Differentiationsarithmetik verwendet.

Datentyp. Bei der AD wird, wie oben beschrieben, eine Funktion von $f: \mathbb{R} \rightarrow \mathbb{R}$ nach $f: D \rightarrow D$ erweitert. Bei reellen Funktionen sind die Funktionsparameter zum Beispiel vom Typ double. Die FADBAD++ Bibliothek deklariert den Datentyp der AD im Vorwärtsmodus wie folgt:

```
1  template <class T>
2  class FTypeName
3  {
4  public:
5      T v;                // der Realteil vom Typ T
6      T *g;               // Feld mit infinitesimal Teilen von Typ T
7      int gsize;          // Größe von g = Anzahl der unabhängigen Variablen
8
9      void diff(int idx, int n); // Methode zum Initialisieren der unabhängigen
10                                     // Variablen mit dx/dx=1 bzw. dc/dx=0
11      ...                      // weitere Methoden
12 }
```

Listing 4.1: Typdeklaration für Differentiationsarithmetik

¹<http://www.fadbad.com/fadbad.html>

In der Regel wird diese Klassenschablone mit dem Datentyp `double` spezialisiert. Für die Berechnung höherer Ableitungen kann die Typdefinition wiederum mit der Klassenschablone parametrisiert werden.

```
1 typedef FTypeName<double> AD;    // AD Datentyp für erste Ableitung
2 typedef FTypeName<AD> AD2;      // AD Datentyp für zweite Ableitung
```

Listing 4.2: *Rekursive Typdefinition für die Darstellung Ableitungen höherer Ordnungen*

Operatoren. Die Klasse hat einen Realteil `v` und so viele „infinitesimal Teile“ `g[i]`, wie es unabhängige Variablen gibt, nach denen differenziert wird. Für diesen Zahlentyp wird von der Bibliothek die Arithmetik bereitgestellt. Der notwendige Sprachmechanismus ist die Operatorüberladung. Die Bibliothek deklariert Regeln für die elementaren Operatoren `+`, `-`, `*`, `/` sowie elementare Funktionen. Trifft der Compiler beim Übersetzungsprozess auf Operationen dieses Datentyps, werden nicht die „Standard“-Befehle, sondern, die neu definierten Operatoren aufgerufen. Zum Beispiel ist der binäre `*` Operator wie folgt mit überladen:

```
1 template <class T>
2 FTypeName<T> operator* (const FTypeName<T>& a, const FTypeName<T>& b)
3 {
4     FTypeName<T> c(a.v*b.v);          // Initialisieren mit Multiplikation der Realteile
5
6     c.touchg(a.gsize);                // Allokieren des infinitesimalen Vektors
7
8     for (int i=0;i<a.gsize;i++)       // Produktregel für alle Elemente des
9     {c.g[i]=a.g[i]*b.v+b.g[i]*a.v;} // infinitesimalen Vektors
10
11     return c;}                       // Rückgabe
```

Listing 4.3: *Überladung des `*` Operators für die Differentiationsarithmetik*

Nach der Initialisierung des Realteiles wird der Vektor für die partiellen Ableitungen allokiert. Die Produktregel wird mit einer Schleife über alle Vektorelemente `g[i]` (partiellen Ableitungen) implementiert.

Funktionen. Auch die Standardfunktionen wie unären Funktionen `sin`, `cos`, `sqr`, `exp` oder `pow` werden überladen.

```
1 template <class T>
2 FTypeName<T> sin(const FTypeName<T>& a)
3 {
4     //RealteilFTypeName<T>
5     c(sin(a.v));
6
7     //Dimension des Infinitesimalteils
8     c.touchg(a.gsize);
9
10    //Berechnung der Infinitesimalteile
11    T tmp(cos(a.v));
12    for (int i=0;i<a.gsize;i++) c.g[i]=a.g[i]*tmp;
13
14    //Rückgabewert
15    return c;
16 }
```

Listing 4.4: *Überladung der sinus-Funktion für die Differentiationsarithmetik*

In der `for`-Schleife werden die partiellen Ableitungen gemäß der Differentiationsarithmetik berechnet:

$$\sin(U) = \sin\left(\begin{bmatrix} u, u' \end{bmatrix}\right) = \begin{bmatrix} \sin(u), u' \cos(u) \end{bmatrix}$$

Der Einsatz der *Differentiationsarithmetik* in *Popa3d* hat sich als sehr günstig erwiesen, weil das programmierte funktionale Modell zur Laufzeit immer numerisch korrekt abgeleitet wird und, weil die Objektorientierte Programmierung der *Differentiationsarithmetik* eine schnelle und sichere Softwareentwicklung ermöglicht.

5 Ausgleichungsrechnung

5.1 Parameterschätzung mit dem Gauß-Helmert Modell

Die Parameter der Objektgeometrie und die verbesserten Beobachtungen werden mit einem Gauß-Helmert-Modell geschätzt. Der „Allgemeinfall“ der Ausgleichungsrechnung, das Gauß-Helmert-Modell mit Restriktionen, ist eine Bedingte Ausgleichung mit Unbekannten und Bedingungen zwischen den Unbekannten. Jede Beobachtung und Bedingung liefert ein *Stück* Information, das mathematisch als eine ggf. nichtlineare Gleichung ausgedrückt wird:

$$f_i(x, l + v) = s_f \text{ mit } i = 1 \dots m \quad (5.1)$$

$$g_i(x) = s_g \text{ mit } i = 1 \dots p \quad (5.2)$$

Die Redundanz r ist der Grad der geometrischen Überbestimmtheit und ergibt sich mit der Anzahl u der zu schätzenden Parameter und der Anzahl $m + p$ Gleichungen mit

$$r = m + p - u \quad (5.3)$$

Der Einfluss einer Beobachtung auf das Ausgleichungsergebnis hängt neben dem funktionalen Modell und der geometrischen Konfiguration stark von der angenommenen Genauigkeit der Beobachtung ab. Aus der Kovarianzmatrix C_{ll} aller Beobachtungen l_i und der *Varianz der Gewichtseinheit* σ_0^2 wird die Gewichtsmatrix P aller Beobachtungen berechnet.

$$\underbrace{Q_{ll}}_{n \times n} = \frac{1}{\sigma_0^2} \underbrace{C_{ll}}_{n \times n} \quad (5.4)$$

$$\underbrace{P}_{n \times n} = \underbrace{Q_{ll}^{-1}}_{n \times n} \quad (5.5)$$

Wenn die Bedingungsgleichungen nichtlinear sind, muss das zu bildende Normalgleichungssystem iterativ gelöst werden. In jedem Iterationsschritt k wird der durch Normalgleichung aufzulösende Widerspruch berechnet.

$$w_f = s_f - f(x + \Delta x_{k-1}, l + v_{k-1}, c) \quad (5.6)$$

$$w_g = s_g - g(x + \Delta x_{k-1}, c) \quad (5.7)$$

Für die iterative Lösung des nichtlinearen Normalgleichungssystems werden die ursprünglichen Bedingungsgleichungen in jedem Iterationsschritt k mit den Jacobimatrizen

$$\underbrace{A}_{m \times u} = \frac{\partial f_i}{\partial x_j} \big|_{x_0 + \Delta x_{k-1}} \quad (5.8)$$

$$\underbrace{B}_{m \times n} = \frac{\partial f_i}{\partial l_j} \big|_{l + v_{k-1}} \quad (5.9)$$

$$\underbrace{C}_{p \times u} = \frac{\partial g_i}{\partial x_j} \big|_{x + \Delta x_{k-1}} \quad (5.10)$$

linearisiert. Nach [Böck 1961] erfordert eine strenge Linearisierung der Bedingungsgleichung eine Linearisierung von $B = \partial f / \partial l$ an der Stelle $l + v$, wenn die Verbesserungen v nicht differentiell klein sind. Die in *Popa3d* erzeugten Pseudobeobachtungen, die aus der 3d-Skizze abgeleitet werden, erhalten ein geringes Gewicht P_{ii} und werden deshalb stark „verbessert“, wenn hochwertige Messungen vorliegen. Insofern ist eine korrekte Linearisierung der Bedingungsgleichungen unbedingt erforderlich. Mit A , B und C erhält man die aus linearisierten

Bedingungsgleichungen gebildeten Normalgleichungen in einem Linearen Gleichungssystem.

$$\begin{bmatrix} \underbrace{BQ_{ll}B^T}_{m \times m} & \underbrace{A}_{m \times u} & 0 \\ \underbrace{A^T}_{u \times m} & 0 & \underbrace{C^T}_{u \times p} \\ 0 & \underbrace{C}_{p \times u} & 0 \end{bmatrix} \begin{bmatrix} \underbrace{k_f}_{m \times 1} \\ \underbrace{\Delta x}_{u \times 1} \\ \underbrace{k_g}_{p \times 1} \end{bmatrix} = \begin{bmatrix} \underbrace{w_f + Bv_{k-1}}_{m \times 1} \\ 0 \\ \underbrace{w_g}_{p \times 1} \end{bmatrix} \quad (5.11)$$

Die Addition $w_f + Bv_{k-1}$ kann algebraisch hergeleitet werden [Gründig 2003, S. 104ff] und garantiert die Forderung der Minimalen gewichteten Quadratsumme der Verbesserungen. In [Neitzel und Petrovic 2008] wird gezeigt, dass eine korrekte Linearisierung des Gauß-Helmert-Modells die gleichen Ergebnisse wie eine Total Least Squares (TLS) Ausgleichung liefert. Die Standard-Algorithmen, wie das Gaußsche Eliminationsverfahren zum Invertieren einer Matrix sind $O(n^3)$ komplex und damit sehr rechenaufwendig. Das Normalgleichungssystem wird daher zum Lösen von Δx reduziert. Mit

$$M = (B^T Q_{ll} B) = (B^T P^{-1} B) \quad (5.12)$$

$$N = A^T M^{-1} A = A^T (B^T Q_{ll} B)^{-1} A \quad \text{und} \quad (5.13)$$

$$w_N = A^T (B^T Q_{ll} B)^{-1} (w_f + Bv_{k-1}) \quad (5.14)$$

wird das Normalgleichungssystem zu

$$\begin{bmatrix} \underbrace{N}_{u \times u} & \underbrace{C^T}_{u \times p} \\ \underbrace{C}_{p \times u} & 0 \end{bmatrix} \begin{bmatrix} \underbrace{\Delta x}_{u \times 1} \\ \underbrace{k_g}_{p \times 1} \end{bmatrix} = \begin{bmatrix} \underbrace{w_N}_{u \times 1} \\ \underbrace{w_g}_{p \times 1} \end{bmatrix} \quad (5.15)$$

auf die Dimension $(u + p)$ verkleinert und in jedem Iterationsschritt ohne vollständige Inversion gelöst. Der Vektor der Verbesserung v berechnet sich mit

$$\underbrace{v}_{n \times 1} = \underbrace{Q_{ll}}_{n \times n} \underbrace{B^T}_{n \times m} \underbrace{M^{-1}}_{m \times m} \underbrace{(w_f - A\Delta x)}_{m \times 1} \quad (5.16)$$

Bei Konvergenz wird das Normalgleichungssystem erst nach dem Erreichen eines hinnehmbaren Linearisierungsfehlers vollständig invertiert.

$$\begin{bmatrix} Q_{xx} & Q_{xk_g} \\ Q_{k_g x} & Q_{k_g k_g} \end{bmatrix} = \begin{bmatrix} N & C^T \\ C & 0 \end{bmatrix}^{-1} \quad (5.17)$$

Die Kofaktorenmatrix der Unbekannten Q_{xx} ergibt sich nach der vollständigen Inversion der geränderten Normalgleichungsmatrix. Zum Zweck der Beurteilung des Ausgleichungsergebnisses mittels statistischer Methoden wird die Kofaktorenmatrix der Verdesserungen Q_{vv} berechnet. Die Kofaktorenmatrix $Q_{k_f k_f}$ der Korrelaten der Bedingungsgleichungen $f_i(x, l + v)$ ist:

$$\underbrace{Q_{k_f k_f}}_{m \times m} = \underbrace{M^{-1}}_{m \times m} - \underbrace{M^{-1}}_{m \times m} \underbrace{A}_{m \times u} \underbrace{Q_{xx}}_{u \times u} \underbrace{A^T}_{u \times m} \underbrace{M^{-1}}_{m \times m} \quad (5.18)$$

$$\underbrace{Q_{vv}}_{n \times n} = \underbrace{Q_{ll}}_{n \times n} \underbrace{B^T}_{n \times m} \underbrace{Q_{k_f k_f}}_{m \times m} \underbrace{B}_{m \times n} \underbrace{Q_{ll}}_{n \times n} \quad (5.19)$$

$$\underbrace{Q_{vv}P}_{n \times n} = \underbrace{Q_{ll}}_{n \times n} \underbrace{B^T}_{n \times m} \underbrace{Q_{k_f k_f}}_{m \times m} \underbrace{B}_{m \times n} \quad (5.20)$$

Zur Analyse des Ausgleichungsergebnisses berechnet die Software die Redundanzanteile

$$r_i = (Q_{vv}P)_{ii} \quad (5.21)$$

mit $\sum r_i = \text{spur}(Q_{vv}P) = r$ sowie normierte Verbesserungen nv_i mit

$$nv_i = \frac{|v_i|}{\sigma_0^2 Q_{vv_{ii}}}. \quad (5.22)$$

5.2 Effiziente Berechnung der Parameter und statistischen Größen

Der Benutzer der Software kann während des Aufmaßes vor Ort die schrittweise geometrische Verfeinerung der Objektgeometrie anhand ständig neu eingefügter Beobachtungen kontrollieren. Die Ausgleichung muss daher so schnell berechnet werden, dass der Arbeitsprozess der Vermessung nicht unterbrochen wird. Die algorithmische Implementierung der in Abschnitt 5.1 vorgestellten Gleichungen wird so vorgenommen, dass sie sowohl für neue Gleichungstypen erweiterbar ist als auch mit geringer Komplexität berechnet werden kann. Das Ziel der *Erweiterbarkeit* wird durch generische objektorientierte Entwurfsmethoden und durch die Verwendung der Arithmetik des Automatischen Differenzieren (Kapitel 4) erreicht. Eine hohe algorithmische Performanz wird im Wesentlichen erreicht, indem der Speicherbedarf für Nebenrechnungen minimiert wird. Zur Lösung der Linearen Gleichungssysteme wird das iterative Lösungsverfahren der konjugierten Gradienten (PCG) mit symmetrischer Vorkonditionierung verwendet.

5.2.1 Parameternetz und indizierte Gradienten

Jede Beobachtungs- und Bedingungsgleichung ist als ein Gleichungsobjekt repräsentiert. Das funktionale Modell eines *Gleichungstyps* ist in der Klasse des Gleichungsobjekts implementiert. In [Hau 2000] wird eine Baumstruktur zur Repräsentation der Beobachtungs- und Parameterstruktur vorgeschlagen. Das in Kapitel 2 entwickelte Datenmodell und in Kapitel 3 formelmäßig beschriebene funktionale Modell von *Popa3d* lässt diese Speicherstruktur allerdings nicht zu. Mit dem Ziel der Parameterreduktion wurde eine geometrische Normalisierung (Kapitel 2.11) begründet. Als Strukturtyp zur Repräsentation der Parameter wird daher das *Netz* gewählt. Das Konzept des Parameternetzes ist für die geometrische Parametrisierung von *Popa3d* entwickelt, kann aber auch verallgemeinert werden:

1. Jede symbolische Gleichung eines Gleichungstyps des Gauß-Helmert Modells enthält eine Anzahl u' von unbekannten Parametern ξ und eine Anzahl n' von Beobachtungen λ . ξ und λ sind die *Variablen der symbolischen Gleichung*.
2. Die *unabhängigen Variablen* (nach denen die Gleichungen differenziert werden) sind die unbekannten Parameter x_i und die Beobachtungen l_i . Diese werden in den nur einmal in den Instanzen der Objektgeometrie bzw. Beobachtungsobjekten gespeichert und ansonsten referenziert.
3. Aufgrund der geometrischen Normalisierung besteht zwischen den *unabhängigen Variablen* und den *Variablen der symbolischen Gleichung* eine $n : m$ Beziehung, die durch das Parameternetz aufgelöst wird.
4. Die Anzahl der Variablen einer Gleichung u' und n' sind *vor der Laufzeit* des Programms bekannt. Die Anzahl der unabhängigen Variablen der Objektgeometrie u^* bzw. der Beobachtungen n^* wird *zur Laufzeit* ermittelt. Alle abhängigen und unabhängigen Variablen der Differentiationsarithmetik werden zur Laufzeit mit $u^* + n^*$ dimensioniert.

Abbildung 5.2 zeigt mit dem Beispiel des Beobachtungstyps „Abstand Kante-Kante“ den Nutzen des Parameternetzes: Alle vier z-Komponenten n_z der Normalenvektoren zeigen auf den Parameter p_3 .

Durch die Verwendung der Differentiationsarithmetik werden beim Berechnen der Funktionswerte $w_f = s_f - f(x + \Delta x_{k-1}, l + v_{k-1}, c)$ und $w_g = s_g - g(x + \Delta x_{k-1}, c)$ zeitgleich die partiellen Ableitungen gebildet und in einem indiziertem Gradientenvektor gespeichert.

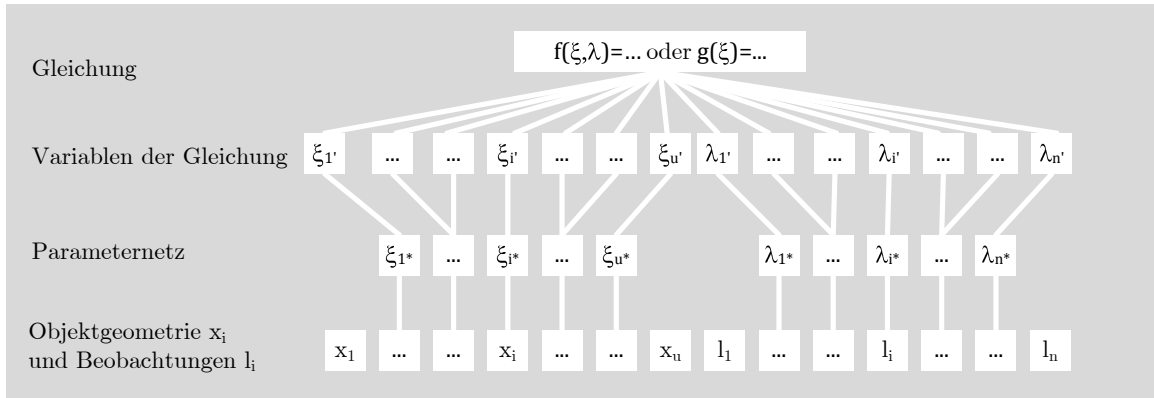


Abbildung 5.1: Parameternetz

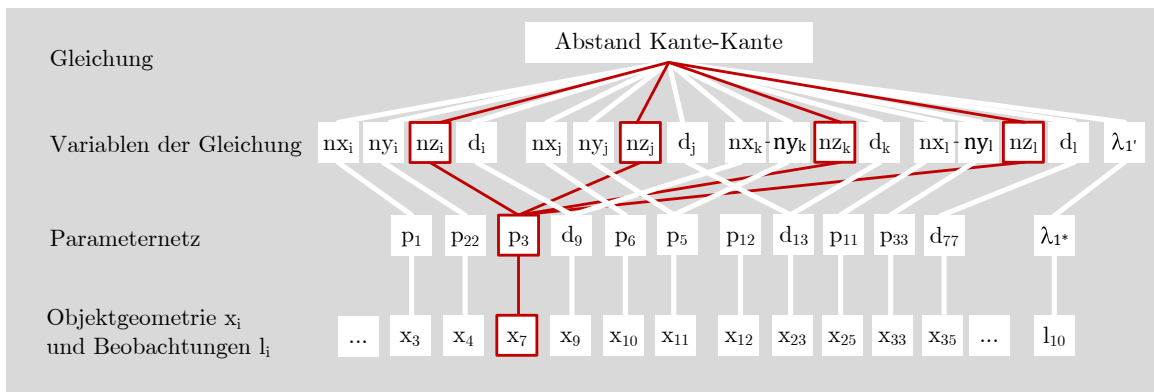


Abbildung 5.2: Parameternetz am Beispiel des Gleichungsobjekts „Abstand Kante-Kante“

$$\nabla f_i = (\nabla f_{iA}, \nabla f_{iB}) = \left(\frac{\partial f_i}{\partial_{1^*}} \cdots \frac{\partial f_i}{\partial_{u^*}}, \frac{\partial f_i}{\partial_{1^*}} \cdots \frac{\partial f_i}{\partial_{n^*}} \right) \quad (5.23)$$

$$\nabla g_i = \left(\frac{\partial g_i}{\partial_{1^*}} \cdots \frac{\partial g_i}{\partial_{u^*}} \right) \quad (5.24)$$

Die Werte der Partiellen Ableitung ergeben sich aus dem Vektor der Infinitesimalwerte der mit Differentiationsarithmetik ausgewerteten *abhängigen Variable* w_f bzw. w_g . Die Speicherung einer partiellen Ableitung erfolgt als Trippel {Zeile Funktionsindex; Spalte Parameterindex; Wert der partiellen Ableitung}. Die Dimension der Gradienten ist $1 \times (u^* + n^*)$.

5.2.2 Sequentielle Berechnung der Normalgleichungsanteile und Redundanzanteile

Die Berechnungszeit zum Aufstellen der der geränderten Normalgleichungsmatrix und der rechten Seite des Linearen Gleichungssystems können wesentlich beschleunigt werden, wenn

- die Normalgleichungsanteile sequentiell berechnet werden und
- dabei berücksichtigt wird, dass einige Matrizen Diagonalmatrizen sind und
- Diagonalmatrizen durch Invertierung der einzelnen Diagonalelemente invertiert werden und
- Nullen in den Funktionalmatrizen nicht gespeichert werden.

Bei der Berechnung von $M = (B^T Q_u B) = (B^T P^{-1} B)$ wird ausgenutzt, dass *Popa3d* ausschließlich unkorrelierte Beobachtungen l_i verwendet und zwischen Beobachtung und Gleichung eine 1:n Beziehung besteht. Hieraus folgt,

dass M eine Diagonalmatrix ist und deren Elemente mit

$$M_{ii} = \sum_{j=1^*}^{j=n^*} (\nabla f_{ijB}) (\nabla f_{ijB})^T Q_{ll_{ij}} \quad (5.25)$$

berechnet und mit dem Kehrwert der Diagonalelemente $M_{ii}^{-1} = 1/M_{ii}$ invertiert werden kann. *Popa3d* fügt viele Pseudobeobachtungen in die Ausgleichung ein, sodass $n \gg u$ und die Funktionalmatrix A viel Speicherplatz beanspruchen würde. Die Berechnung der Normalgleichungsmatrix $N = A^T M^{-1} A$ erfolgt deshalb durch sequentielles Summieren der Normalgleichungsanteile.

$$\underbrace{N}_{u \times u} = \underbrace{N_1}_{u \times u} + \dots + \underbrace{N_i}_{u \times u} \dots + \underbrace{N_m}_{u \times u} \quad (5.26)$$

Die Normalgleichungsanteile N_i einer Beobachtungs- bzw. Bedingungsgleichung f_i werden im Fall $n \gg u$ mit dem *dyadischen Produkt* berechnet.

$$\underbrace{N_i}_{u \times u} = \underbrace{M_{ii}}_{1 \times 1} \cdot \underbrace{(\nabla f_{iA} \otimes \nabla f_{iA})}_{u \times u} = \underbrace{M_{ii}}_{1 \times 1} \cdot \left(\underbrace{(\nabla f_{iA})^T}_{u \times 1} \underbrace{(\nabla f_{iA})}_{1 \times u} \right) \quad (5.27)$$

Es werden nur die Nicht-Nullelemente ausgewertet. Nach dem Lösen des geränderten Normalgleichungssystems werden die Verbesserungen v_i der Beobachtung l_i für jede Gleichung f_j bestimmt.

$$v_i = Q_{ll_{ii}} \nabla f_{jiB} M_{jj}^{-1} (w_{fj} - \nabla f_{jA} \triangle x) \quad (5.28)$$

Nach der iterativen Lösung der nichtlinearen Normalgleichung werden für alle Funktionen f_j die Diagonalelemente der Kofaktorenmatrix der Korrelaten k_f berechnet.

$$Q_{k_f k_f jj} = M_{jj}^{-1} - (M_{jj}^{-1})^2 \sum_{i=1^*}^{i=u^*} \nabla f_{jiA} \cdot \sum_{k=1^*}^{k=u^*} \nabla f_{jiA} Q_{xx_{ik}} \quad (5.29)$$

Der Redundanzanteil r_i einer Beobachtung i ist das Diagonalelement der Matrix $Q_{vv}P$

$$Q_{vv}P = Q_{ll}B^T Q_{k_f k_f} B \quad (5.30)$$

$Q_{vv}P_{ii}$ wird für jede Gleichung j berechnet.

$$r_i = (Q_{vv}P)_{ii} = Q_{ll_{ii}} Q_{k_f k_f jj} (\nabla f_{jiB})^2 \quad (5.31)$$

6 Validierung der Daten

6.1 Begründung und Anwendung der Axiome

Die Struktur des in Kapitel 2 entwickelten Datenmodells muss nicht zwangsläufig zu einem korrekten dreidimensionalen Gebäudemodell führen. In Analogie zu [Gröger 2000, S. 27ff] werden in diesem Kapitel die **Axiome** für ein korrektes *Popa3d* Modell behandelt. Die Axiome und Methoden zur Sicherung der Datenintegrität validieren eine Dokumenteninstanz. Die *Validierung der Daten* ist gegen eine *Validierung der Methode* abzugrenzen. Eine Methodenvalidierung behandelt die Fragestellung der effizienten Problemlösung und wird in der vorliegenden Arbeit im Kapitel 7 behandelt.

In dieser Arbeit stehen die Integritätsbedingungen für raumbezogene Informationsmodelle im Vordergrund, die sich insbesondere aus den Beobachtungen beziehungsweise Bedingungen und der Ausgleichsrechnung ergeben. Abbildung 6.1 gliedert die Axiome thematisch und stellt sie in einem Schichtenmodell gemeinsam mit den allgemeinen, informationstechnischen Regeln dar. Es soll verdeutlicht werden, dass die *höheren* Schichten die Korrektheit der *tieferen* Schichten voraussetzen. Für eine praktikable Anwendbarkeit der Software muss

Ausgleichung	Berechnung	Die Berechnung kann durchgeführt werden (Die Normalgleichungsmatrix ist invertierbar)
	Beobachtung	Alle Beobachtungen und Bedingungen sind gültig
3d GIS/CAD	Geometrie	Die geometrischen Integritätsbedingungen werden nicht verletzt.
	Topologie	Die topologischen Integritätsbedingungen werden nicht verletzt.
Modellierung	Eindeutigkeit, Referenz und Wertebereich	Der Inhalt der Struktur ist im Sinne der allgemeinen Informationsmodellierung formal korrekt.
	Struktur	Alle komplexen Datentypen des Datenmodells können gebildet werden
IT	Syntax	Zahlen und Buchstaben sind korrekt

Abbildung 6.1: Schichtenmodell für die Konsistenzprüfung

stets ein konsistentes Modell vorliegen. Verletzungen der Axiome können bei *Benutzerinteraktion* während des Bearbeitens innerhalb einer Software entstehen. Beim Einlesen von fremderzeugten Dateien (*SketchUp*, *IFC*, *Oracle Spatial*, *CityGML*), die eine Transformation der Datenstruktur erfordern, müssen alle Axiome geprüft werden.

Die Axiome werden in unterschiedlichen Softwaremodulen und zu verschiedenen Zeitpunkten im Arbeitsprozess geprüft. Folgende Prüfinstanzen wurden in dieser Arbeit entwickelt:

Prüfinstanz Skizze. Die Spezifikation der Topologie erfolgt beim Anfertigen der 3d Skizze, die mit dem Entwurfsprogramm „*Google SketchUp*“ durchgeführt wird. Das originäre *SketchUp* Geometriemodell garantiert dabei einige Topologisch-Geometrische Axiome. Zusätzlich Konsistenzprüfungen werden als Plug-in mit der objektorientierten Programmiersprache Ruby implementiert und können währen der Anfertigung der Skizze ausgeführt werden. Konsistenzverletzungen werden dem Benutzer graphisch visualisiert.

Prüfinstanz Datenaustausch. Der Datenaustausch erfolgt mit XML Dokumenten. Die Dokumente werden gegen ein XML-Schema geprüft, dass insbesondere Konsistenzbedingungen vom Typ *Eindeutigkeit* und *referentielle Integrität* prüft.

Prüfinstanz Messung und Berechnung. Die zur Beobachtungseingabe und Ausgleichung entwickelte Software liest die validierten XML Dokumente ein und prüft darüberhinaus topologische, geometrische und beobachtungsinduzierte Axiome.

6.2 Syntaktische Integritätsbedingungen

Die Syntaktische Konsistenz ist die „unterste“ Ebene der technischen Darstellung von Information.“*Typische Beispiele sind unterschiedliche Zahlenformate[...], unterschiedliche Zeichenkodierungen (Unicode versus ASCII) oder unterschiedliche Trennzeichen in Textformaten*“ [Leser und Naumann 2007]. Sind Zahlen und Buchstaben in zwei Systemen (Code-Model, XML, *SketchUp*) unterschiedlich repräsentiert, führt dies beim Datentransfer zu Konsistenzverletzungen.

Bei technisch belastbaren Dokumenten kommt der Genauigkeit der Fließkommazahlen eine besondere Bedeutung zu. Es muss sichergestellt sein, dass die Repräsentation einer Zahl als Zeichenkette im XML Dokument mit der gleichen Anzahl von Nachkommastellen wie in der double Variable während der Berechnung zur Laufzeit dargestellt wird. Eine *double* Variable hat 16 signifikante Stellen [IEEE Std 754-2008].

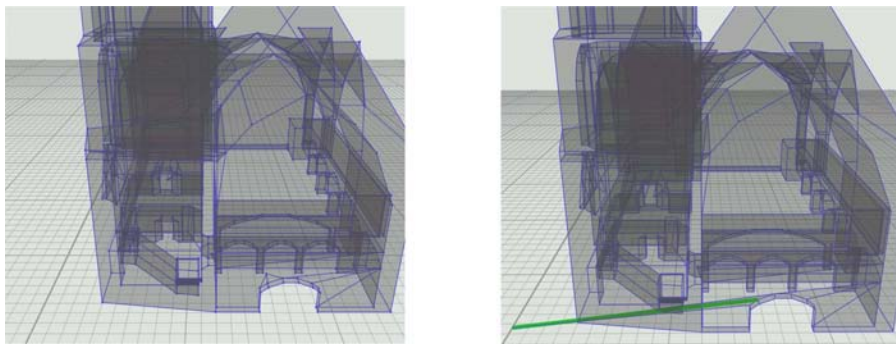


Abbildung 6.2: Auswirkung einer falschen Kodierung von Fließkommazahlen

Abbildung 6.2 zeigt das Ergebnis einer unzureichend genauen Darstellung von Dezimalstellen in der XML Dokumenteninstanz. Die unzureichende genaue Version speichert die Fließkommazahlen mit als Exponential Darstellung „e“ mit einer standardmäßig¹ vergeben Anzahl von 6 Nachkommastellen.

```
<parameter id="12" wert="7.071068e-001" />
<parameter id="12" wert="0.70710678118654857/>
```

¹Das *Popa3d* Softwaremodul verwendet für Fließkommazahlen den primitiven Datentyp `System::Double` des Microsoft .Net Frameworks. Dieser Datentyp implementiert eine Methode `Double::ToString` zur Ausgabe der Zahl als Zeichenfolge.

Die in der Software verwendete Version speichert alle signifikanten Stellen der Fließkommazahlen, die die Werte der Objektgeometrie beschreiben.

6.3 Konsistenz mit dem Datenmodell

Die Axiome, die sich unmittelbar aus dem Datenmodell ergeben, werden als Modell-Axiome bezeichnet und mit dem Präfix MA nummeriert. Sie ergeben sich aus den mathematisch-geometrischen Sachverhalten von 3d Modellen und können mit allgemein-informationstechnischen Mitteln zur Sicherung der Datenintegrität geprüft werden.

Die *Strukturelle Integrität* wird durch die Datenstruktur des Informationsmodells definiert. Als Voraussetzung einer Prüfung der Referentiellen Integrität müssen die komplexen Datentypen der Datenelemente einer Dokumenteninstanz sowie deren Attribute korrekt benannt und strukturiert sein.

Die *Eindeutigkeit* der Datenelemente garantiert, dass jeder Identifikator der Instanz eines Datentyps innerhalb des Nummerierungsbereichs nur einmal vorkommt. Dabei bildet der Datentyp den Nummerierungsbereich. Auf den UUID-Standard zur Vergabe von „weltweit“ eindeutigen Identifikatoren (UUID - *Universally Unique Identifier*) wurde in Popa3d verzichtet, weil die Zeichenlänge der UUID von 16 Byte für den Benutzer das Lesen der XML-Dokumente und die Interaktion mit dem Modell während der Beobachtungseingabe erschwert. Die Eindeutigkeit der Datenelemente ist *einfach zu realisieren*: Im Codemodell werden assoziative Container verwendet, im XML Model wird die Eindeutigkeit durch eine Schlüssel- bzw. Eindeutigkeitsdefinition im XML-Schema festgelegt. Tabelle 6.1 führt in MA01 die Eindeutigkeitsbedingung aufgrund des allgemeinen Charakters als Integritätsbedingung für alle Elemente auf.

Die *Referentielle Integrität* beinhaltet die Eindeutigkeit und korrekte Struktur der Datenelemente und findet darüberhinaus auf der Beziehungsebene statt. Jeder Datensatz, der eine Beziehung zwischen Elementen der Objekttopologie, Objektgeometrie und Beobachtungen beschreibt, muss daraufhin überprüft werden, dass die Referenzen „nicht ins Leere“ zeigen. Die *Referentielle Integrität* kann mit allgemein gebräuchlichen Techniken der Informationstechnologie realisiert werden: Das XML-Schema von *Popa3d* verwendet die XML-Schema Sprachmittel zur Definition von Beziehungen als Schlüssel-Fremdschlüssel Paar. Tabelle 6.1 benennt die Axiome der Referentiellen Integrität von *Popa3d* in den Axiomen MA02 bis MA07.

Der *Wertebereich* ist eine Integritätsbedingung auf Datenfeldebene. Dabei wird der Wertebereich des elementaren Datentyps des Attributs verkleinert. Winkelbedingungen in Popa3d beschreiben Parallelität oder Rechtwinkligkeit mit Hilfe des Skalarprodukts der normierten Normalenvektoren. Die möglichen Werte sind damit auf -1, 0 und 1 beschränkt. Weiter Axiome für Wertebereiche gelten für einige Beobachtungswerte und für die Standardabweichung aller Beobachtung (Tabelle 6.1). Der Wertebereich kann ebenfalls mit einfachen, allgemeingebräuchlichen Programmier Techniken realisiert werden.

6.4 Topologische Integritätsbedingungen

Die Prüfung der topologischen Konsistenz erfolgt ständig während des Anfertigen der 3d-Skizze „vor Ort“ und einmal beim Einlesen von Fremd- und Eigendokumenten. Fehlerhafte topologische Primitive werden farblich visualisiert, damit der Zeichner das Modell bereinigen und ein konsistentes Modell anfertigen kann. Die Axiome der topologischen Integrität von *Popa3d* basieren auf den Untersuchungen von [Thiemann 2001] und [Gröger 2000]. Die topologische Integrität eines Dokuments kann nur geprüft werden, wenn die Datenstruktur, Eindeutigkeit und Referentielle Integrität korrekt sind. Die Axiome gehen über die in der allgemeinen Informationstechnologie üblichen Konsistenzsicherungen hinaus und sind eine typische Eigenschaft räumlicher Informationssysteme.

Abbildung 6.3 zeigt vier typische Beispiele, die beim Anfertigen der 3d-Skizze mit dem dreidimensionalen Entwurfsprogramm *Google SketchUp* auftreten. Die für die Validierung entwickelten Softwarebausteine (Ruby Plug-in) unterstützen den Zeichner bei der Konsistenzsicherungen, indem sie die als fehlerhaft erkannten topologischen Entitäten farblich darstellen.

#	Geometrisches Element	Axiom
MA01	Alle Elemente	Alle Elemente der Topologie, Geometrie und Beobachtungen haben einen eindeutigen Identifikator innerhalb des eigenen Namensbereichs
MA02	Kante	Jede Kante referenziert genau zwei existente Knoten
MA03	Halbkante	Jede Halbkante referenziert genau eine existente Kante.
MA04	Masche	Jede Masche referenziert genau eine existente Ebene. Jede Masche referenziert genau zwei existente Räume.
MA05	Ebene	Jede Ebene referenziert genau einen existente Normalenvektor.
MA06	Normalenvektor	Jeder Normalenvektor referenziert genau drei existente Parameter.
MA07	Alle Beobachtungen und Bedingungen	Die von den Beobachtungen und Bedingungen referenzierten existente Elemente der Objekttopologie und Objektgeometrie
MA08	Parameter	Der Wert eines Parameters ist nicht kleiner als -1.0 und nicht größer als 1.0
MA09	Winkelbedingung	Wert entweder -1, 0 oder 1
MA10	Winkelmessung	Wert größer gleich 0 und kleiner als 400
MA11	Streckenmessung	Wert größer gleich 0.0
MA12	Standardabweichung	Wert größer als 0.0

Tabelle 6.1: Axiome der referentiellen Integrität auf Modellebene

Abbildung 6.3a) zeigt die Seite einer Masche, die das **Axiom TA01** „Jeder Knoten wird von mindestens 3 Kanten referenziert“ verletzt. Der Fehler kann beseitigt werden, indem die in zwei Kanten geteilte Seite zu einer Kante vereinigt wird.

Abbildung 6.3b) zeigt einen Kantenzzyklus, der mit nur einer Masche verknüpft ist. Dieser Fehler tritt auf, wenn eine Menge von Kanten nach der *Extrusion* nicht mit dem Gesamtmodell verschnitten ist.

Abbildung 6.3c) und Abbildung 55 d) zeigen Artefakte, die bei komplexen Skizzen entstehen können. Eine Masche, die nur Teilweise mit dem „Drahtgittermodell“ verbunden ist, ist von Kanten berandet, die nur auf keine weitere Masche verweisen. Die in Abbildung 6.3d) dargestellte Situation entsteht dann, wenn die Masche nicht genau *planar* ist und vom Entwurfsprogramm eine *Triangulierung* (Zerlegung der Polygone in Dreiecke) angestoßen wird.

Das **Axiom TA10** untersucht die Anzahl der topologischen Primitive als eine topologische Invariante und kann mit der *Euler-Poincaré Formel* zur topologischen Konsistenzprüfung genutzt werden. Die *Euler-Poincaré Formel* vergleicht die Anzahl der

- Knoten v ,
- Kanten e ,
- Maschen f ,
- topologischen Körpern s ,
- Komponenten (Zusammenhängende v , e , f , s) c ,
- Durchbohrungen (zum Beispiel eine tragende Säule) t und
- Blasen b im Körper.

Durch Vergleich der Euler-Charakteristik und der Betti-Zahl [Mäntylä 1976, S. 45] ergibt sich nach [Kruschwitz 1996, S. 33] folgende Konsistenzbedingung für Zellkomplexe:

$$v - e + f - s = c - t + b \quad [\text{Kruschwitz}] \quad (6.1)$$

#	Topologische Dimension	Axiom
TA01	Knoten	Jeder Knoten wird von mindestens 3 Kanten referenziert.
TA02	Kante	Jede Kante hat genau zwei unterschiedliche Konten.
TA03	Kante	Jede Kante wird von mindestens zwei Halbkanten referenziert.
TA04	Halbkante	Die Masche-Kanten Paare sind eindeutig.
TA05	Masche	Jede Masche wird von mindestens drei Halbkanten referenziert.
TA06	Masche	Die Anzahl der Kanten einer Masche entspricht der Anzahl der Knoten einer Masche. [h]
TA07	Masche	Innerhalb einer Masche bilden die Kanten einfache Zyklen.
TA08	Masche	Innerhalb einer Masche ist jeder Kante genau einem Zyklus zugeordnet.
TA09	Raum	Jeder Raum wird von mindestens drei Maschen referenziert.
TA10	Menge aller Räume, Maschen, Halbkanten, Kanten und Knoten	Die erweiterte Euler-Poincaré Gleichung ist gültig.

Tabelle 6.2: Axiome der Topologischen Integrität

Zellkomplex (Gl.6.1)					Zellkomplex (Gl.6.2)					2-Mannigfaltigkeit (Gl.6.3)				
		a)	b)	c)			a)	b)	c)			a)	b)	c)
vertex	+	16	16	24	vertex	+	16	16	24	vertex	+	16	16	24
edge	−	20	22	36	edge	−	20	22	36	edge	−	20	22	36
face	+	8	8	16	face	+	8	8	16	face	+	8	8	16
solid	−	1	1	1										
Σ_{Euler}		3	1	3	Σ_{Euler}		4	2	4	Σ_{Euler}		4	2	4
component	+	1	1	1	shell	+2×	1	1	1	shell	+2×	1	1	1
transfixion	−	0	0	0	hole	−2×	0	0	0	hole	−2×	0	0	0
bubble	+	0	0	0						ring	−	2	0	2
Σ_{Betti}		1	1	1	Σ_{Betti}		2	2	2	Σ_{Betti}		4	2	4
$\Sigma_{Euler} = \Sigma_{Betti}$		n	✓	n	$\Sigma_{Euler} = \Sigma_{Betti}$		n	✓	n	$\Sigma_{Euler} = \Sigma_{Betti}$		✓	✓	✓

Tabelle 6.3: Vergleich der unterschiedlichen Euler-Poincaré Formeln

In [Mäntylä 1976, S. 46] wird die Euler-Poincare Formel mit Hilfe der Anzahl der Knoten v , Kanten e , Maschen f und Hüllen s (engl.: shell, das Innere einer Hülle ist das Innere des topologischen Körpers) und Löchern h (engl.: hole, siehe Durchdringung) ausgedrückt:

$$v - e + f = 2(s - h) \quad [\text{Mäntylä 1}] \quad (6.2)$$

[Mäntylä 1] gilt ebenfalls für Zellkomplexe. Durch die Einbeziehung der Anzahl der inneren Polygone r , die ein Loch in einer Masche beranden (engl. ring) wird eine topologische Integritätsbedingung für dreidimensionale B-Rep Modelle, die lediglich 2-Mannigfaltigkeit der Maschen fordern, formuliert:

$$v - e + f = 2(s - h) + r \quad [\text{Mäntylä 2}] \quad (6.3)$$

Abbildung 6.4 zeigt drei topologische Skizzen eines Komplexes. Zunächst sollen nur Fig. a) und b) unterschieden werden. a) begrenzt den topologischen Körper mit 2-Mannigfaltigkeiten, b) ist ein Zellkomplex.

Tabelle 6.3 vergleicht die Anwendung der drei verschiedenen Formeln auf drei unterschiedliche Modelle. Dabei ist Modell c) ein flächenparametrisiertes Beispiel, in dem sich Ebenen in den Kanten schneiden. Zur Prüfung der Integritätsbedingung **TA10** verwendet *Popa3d* daher die Formel [Mäntylä 2] und fordert damit keinen Zellkomplex sondern 2-Mannigfaltigkeit.

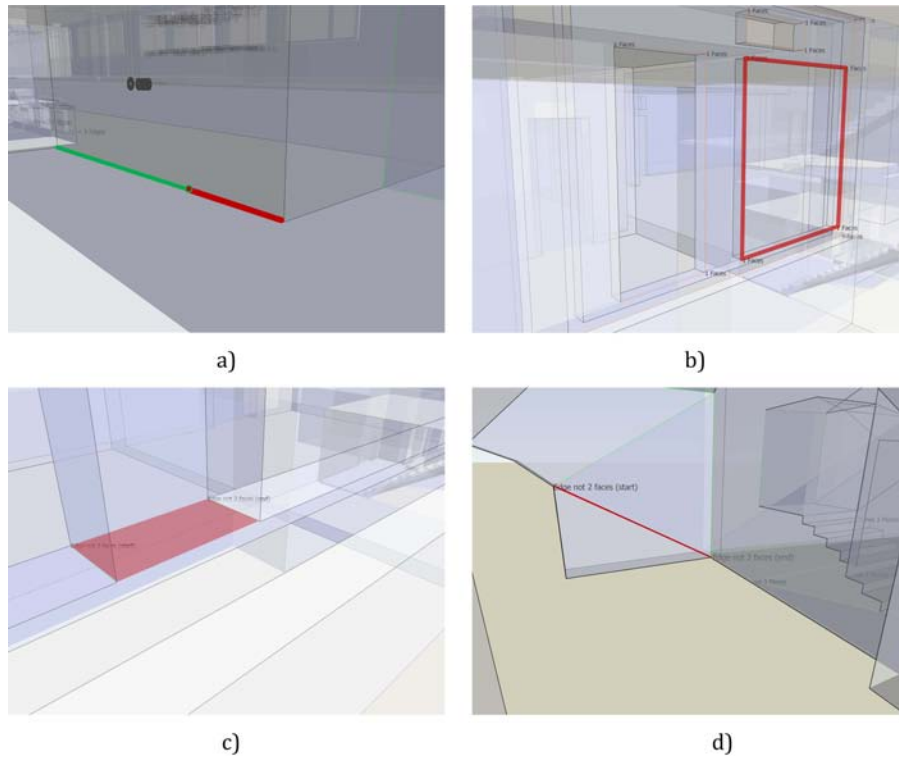


Abbildung 6.3: Verletzung der topologischen Integrität beim Anfertigen der 3d Skizze

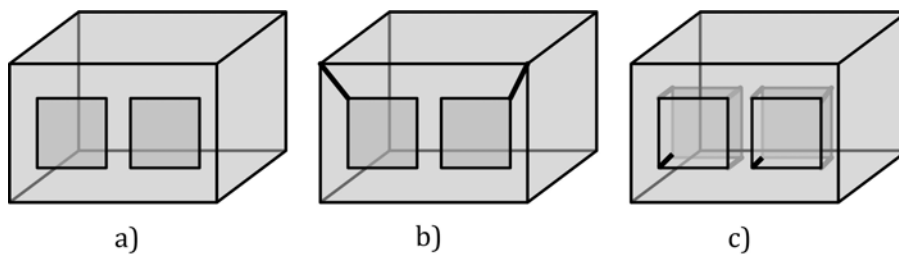


Abbildung 6.4: 3D-Körper mit Löchern a) 2-Mannigfaltigkeit b) Zellkomplex c) Popa3d

6.5 Geometrische Integritätsbedingungen

Das Prüfen der geometrischen Konsistenz erfolgt während der Anfertigung der 3d-Skizze „vor Ort“ und beim Transformieren in das *Popa3d* Modell. Die geometrische Konsistenz eines Dokuments kann nur geprüft werden, wenn das Modell *topologisch konsistent* ist. Die Axiome unterscheiden sich von den in punktparametrisierten räumlichen Informationssystemen üblichen Konsistenzsicherungen und sind damit eine typische Eigenschaft von *Popa3d*. Die geometrischen Integritätsbedingungen beziehen sich auf die *Werte der Objektgeometrie*. Das **Axiom GA01** „Jeder Knoten wird indirekt von mindestens 3 linear unabhängigen Ebenen referenziert“ ist eine Integritätsbedingung, die verletzt wird, wenn sich in einem Knoten 3 Kanten treffen, die koplanare Maschen trennen (Abbildung 6.5). Der Zustand kann beseitigt werden, indem die geometrisch nicht definierte Kante (rot gestrichelt) gelöscht und die beiden angrenzenden Maschen vereinigt werden.

Das **Axiom GA02** „Jede Kante hat genau zwei *geometrisch* unterschiedliche Knoten“ ist in Abbildung 58b) nach einer Ausgleichung mit einer Koplanaritäts-Bedingung (grüne Masche) verletzt. Die Topologie wird durch die Ausgleichungsrechnung nicht verändert. Der nicht konsistente Zustand kann beseitigt werden, indem der Versatz (siehe Abbildung 6.6a) topologisch beseitigt und die vertikalen Maschen der Gebäudeseite topologisch vereinigt werden.

#	Element	Axiom
GA01	Knoten	Jeder Knoten wird indirekt von mindestens 3 linear unabhängigen Ebenen referenziert.
GA02	Kante	Jede Kante hat genau zwei <i>geometrisch</i> unterschiedliche Knoten.
GA03	Masche	Der Rand einer Masche ist selbstüberschneidungsfrei.
GA04	Normalenvektor	Des Normalenvektor hat den Betrag 1.0.

Tabelle 6.4: Axiome der geometrischen Integrität

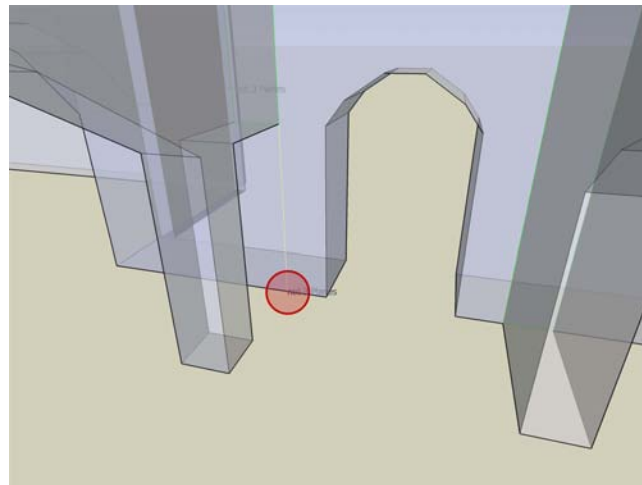


Abbildung 6.5: Ein Knoten wird indirekt von 3 linear abhängigen Ebenen referenziert

6.6 Integritätsbedingungen für die Ausgleichsrechnung

Für die Anwendung der Ausgleichsrechnung muss sowohl das Funktionale Modell jeder *einzelnen* Beobachtungsgleichung als auch die Gesamtheit der Parameter berechenbar sein. Berechenbar ist eine Gleichung bzw. das Gleichungssystem, wenn ein Inversion möglich ist. Hieraus ergeben sich „strengere“ Konsistenzbedingungen, als für Standard-3d-Modelle.

Die Berechenbarkeit der Gerade einer Kante als Ebenenschnitt wurde in Abschnitt 6.5 nicht gefordert. **Axiom AA01** fordert für Kanten, die an einer Beobachtung vom Typ

- Abstand Masche-Kante
- Abstand Kante-Kante
- Abstand Kante-Knoten
- Lokale Polarkoordinate auf Kante

teilnehmen, dass die *einzelne Beobachtungsgleichung* nicht singulär ist. Das funktionale Modell der Beobach-

#	Element	Axiom
AA01	Kanten	Jede Kante, die an einer Beobachtung teilnimmt, wird indirekt mit zwei nichtparallelen Normalenvektoren gebildet
AA02	Menge aller Beobachtungen, Bedingungen und Datumsfestlegungen	Die Objektgeometrie ist vollständig oder überbestimmt mit einer Menge von Funktionen beschreibbar (Inversion)

Tabelle 6.5: Axiome der Integrität für die Ausgleichsrechnung

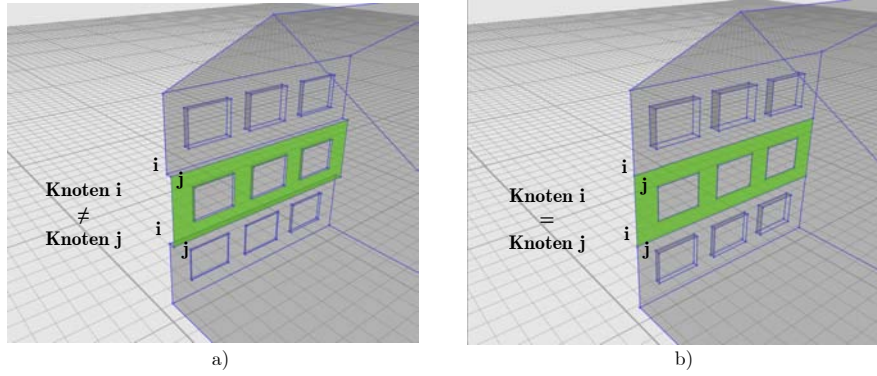


Abbildung 6.6: Die Integritätsbedingung GA02

tungsgleichungen enthält die Berechnung eines Ebenenschnitt $j \cap k$ mit dem Ausdruck

$$\left[(\vec{n}_j, \vec{n}_k, \vec{n}_j \times \vec{n}_k)^T \right]^{-1} \quad (6.4)$$

Parallele Normalenvektoren führen wegen $\vec{n}_j \times \vec{n}_k = 0$ deshalb zu einer singulären Beobachtungsgleichung. Die Software prüft die Bedingung vor der Berechnung der Ausgleichung.

Das **Axiom AA02** fordert, dass ausreichend viele Funktionen das Modell (invers) beschreiben und eine eindeutige Lösung des Normalgleichungssystems gefunden werden kann. Die Funktion beschreiben am Objekt durchgeführte Messungen, Bedingungen, mathematische-geometrische Restriktionen, aus der Skizze abgeleitete Vorinformationen und Datumsverfügungen. Die Software prüft die Vollständigkeit der inversen Beschreibung zunächst algorithmisch und unterstützt den Benutzer bei der Wahl der Funktionalen Modelle.

7 Validierung der Methode

7.1 Anwendungsfall Ersterfassung

Die in dieser Arbeit entwickelten Methoden und Softwaremodule sollen im Folgenden auf Einsatzfähigkeit und Nutzen untersucht werden. Der Anwendungsfall wird mit zwei Software-Programmen durchgeführt: Das um eigene Plug-ins erweiterte 3d Entwurfsprogramm *Google SketchUp* wird als dreidimensionales Feldbuch verwendet und erzeugt die Topologie sowie die Nahrungsgeometrie des Objektes. Die Software *Popa3d* dient der Beobachtungseingabe und der Berechnung bzw. Analyse der Messwerte und der Objektgeometrie. Die komplette Vermessung kann vor Ort durchgeführt und graphisch interaktiv ausgewertet werden. Beide Software-Programme können auf handelsüblichen Laptops ausgeführt werden. Als Messinstrument kommt in diesem Beispiel ausschließlich ein Laserentfernungsmesser zum Einsatz.

Im sechsten Obergeschoss des Hauptgebäudes der Technischen Universität Berlin befinden sich die Räume des Fachgebiets für Geodäsie und Ausgleichsrechnung. Die Südseite des Westflügels besteht aus sechzehn Büroräumen, die Nordseite besteht aus fünf Seminarräumen, vier Büroräumen und vier kleinen Abstellräumen. Die einzelnen Räume werden über den langgestreckten mittleren Flur erschlossen, der am östlichen und am westlichen Ende je einen Zugang zum Treppenhaus des Hauptgebäudes hat. Die Messdaten wurden zeitlich parallel mit der morgendlichen Reinigung erhoben, weil das Reinigungspersonal Zutritt zu allen Zimmern hat. Es wurde ausschließlich mit einem Laserentfernungsmessgerät gearbeitet, da für jeden Raum nur maximal 2-3 Minuten zur Verfügung standen.

Eine Ersterfassung ist notwendig, wenn zu Beginn der Vermessung weder geometrische noch topologische Daten über die Gebäudestruktur vorliegen. Es werden folgende Arbeitsschritte nacheinander durchgeführt und dabei untersucht, ob die entwickelten Methoden für den Einsatzzweck der Ersterfassung geeignet sind:

1. **3d-Skizze der Situation.** Ist *Google SketchUp* ein geeignetes Werkzeug zum Erzeugen der Gebäudetopologie und Nahrungsgeometrie?
2. **Validieren der Skizze.** Sind die in Abschnitt 6 entwickelten Methoden zur Validierung der Daten und die implementierten Plug-ins effizient nutzbar?
3. **Transformation nach Popa3d.** Führen die in Abschnitt 2.10 und 2.11 entwickelten Methoden der Normalisierung zu einer besseren Parametrisierung der Gebäudestruktur ?
4. **Messen.** Können Beobachtungstopologie und Messwerte effizient spezifiziert werden?
5. **Ausgleichung.** Wie verbessert die Ausgleichsrechnung die Arbeitsweise der Ersterfassung?

7.1.1 Vorbedingungen

Es muss genau vereinbart werden, bis zu welchem Detaillierungsgrad die Gebäudestruktur erfasst werden soll, wie Fenster, Türen oder Deckenverkleidungen abstrahiert werden und mit welcher Methode die Qualität des Vermessungsergebnisses geprüft werden kann. Das Personal benötigt möglichst ungehinderten Zugang zu allen Zimmern des Gebäudes sowie einen Ansprechpartner vor Ort.

7.1.2 Skizze (Schritt 1)

Die Skizze wurde mit dem dreidimensionalen Entwurfsprogramm *Google SketchUp* „nach Augenmaß“ angefertigt. Die kostenlose Software *Google SketchUp* ist einfach zu bedienen und beansprucht nur wenig Rechen- und Speicherleistung. Beim Anfertigen der 3d-Skizze werden Polygone gezeichnet und anschließend extrudiert. Durch Anwendung der Fangfunktionen kann SketchUp eine Topologie der Knoten, Kanten und Maschen bilden.

Um die Skizze möglichst schnell anzufertigen, wurden Kanten und Maschen mit Hilfe von *Konstruktionslinien* geometrisch ausgerichtet. Räume, Türen oder Fenster konnten als *Block* kopiert und verschoben werden.

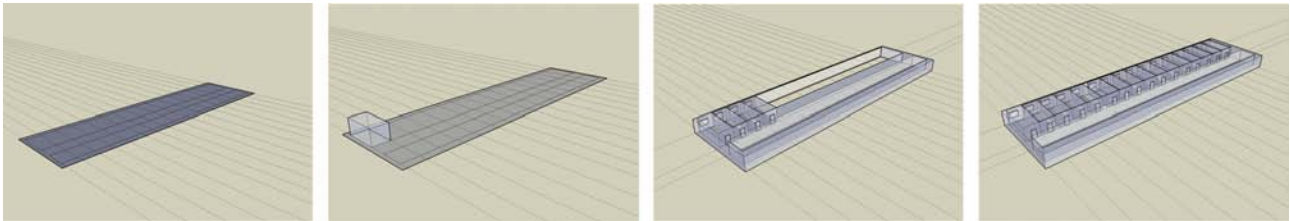


Abbildung 7.1: *Anfertigen der Skizze mit Hilfe von Konstruktionslinien*

Bei der graphischen Darstellung und Interaktion eines 3d Modells besteht grundsätzlich das Problem, dass die dreidimensionale Situation auf eine zweidimensionale Fläche, den Bildschirm, abgebildet wird. Die Zentralprojektion ermöglicht einen guten visuellen Eindruck. Für ein schnelles Arbeiten, insbesondere bei der graphisch-interaktiven Auswahl mit dem Mauszeiger kann der Anwender in die Parallelprojektion wechseln.

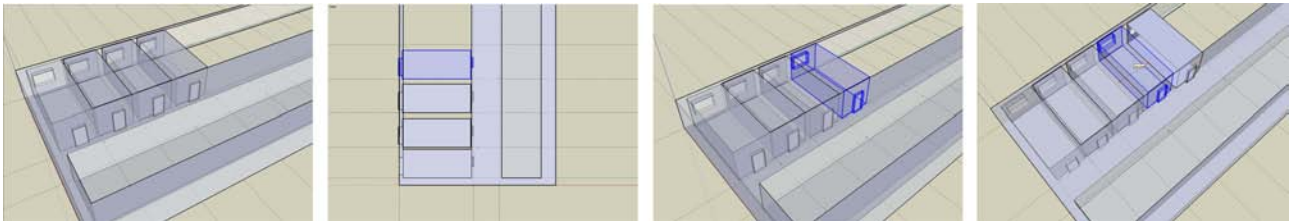


Abbildung 7.2: *Selektieren, Kopieren und Verschieben von Blöcken*

Im Vergleich zu traditioneller und hoch spezialisierter CAD Software lässt sich *Google SketchUp* intuitiv bedienen und ist mit Hilfe von Plug-ins funktional erweiterbar.

7.1.3 Skizze validieren (Schritt 2)

Ziel dieses Arbeitsschrittes ist die Erzeugung einer dreidimensionalen Skizze, die, obwohl geometrisch ungenau, eine gültige Gebäudetopologie spezifiziert. Beim interaktiven Skizzieren im 3d-Feldbuch werden Linien häufig nicht korrekt gestutzt oder ebene geschlossene Polygone nicht als „Masche“, sondern als „Sammlung von Kanten“ gespeichert. Die in dieser Arbeit entwickelten Ruby-Plug-ins zum Validieren der Gebäudegeometrie visualisieren Verletzungen der topologischen Konsistenz so, dass Zeichenfehler und Artefakte während der Anfertigung der Skizze signalisiert werden (Abbildung 7.3).

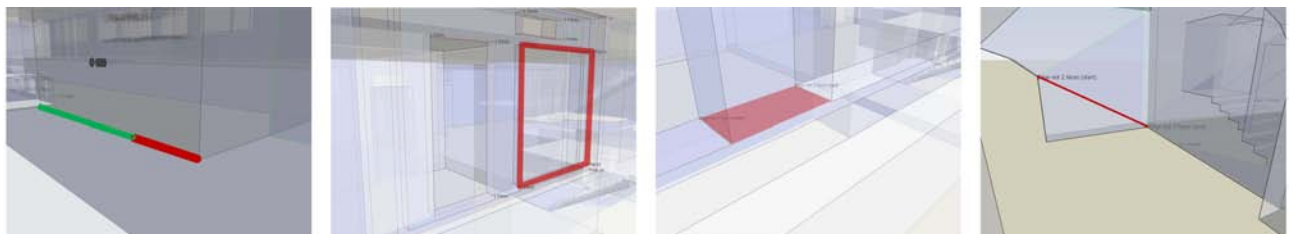


Abbildung 7.3: *Visualisieren von Zeichnungsfehlern (a)*

Insbesondere muss jede Kante von zwei Maschen referenziert werden (Abbildung 7.4) und jeder Knoten durch einen Schnitt von mindestens drei nichtparallelen Ebenen berechenbar sein (Abbildung 7.5 und 7.6). Die Plug-ins analysieren das Modell, protokollieren die Fehler und visualisieren die Konsistenzverletzungen im 3d-Modell durch Einfärben topologischer Elemente oder durch dreidimensionale Platzierung von Fehlertexten.

Der Arbeitsprozess wird durch das Visualisieren der Konsistenzverletzungen wesentlich beschleunigt, da diese frühzeitig und vor Ort beseitigt werden können: Durch „Nachzeichnen“ der Polygone können topologische

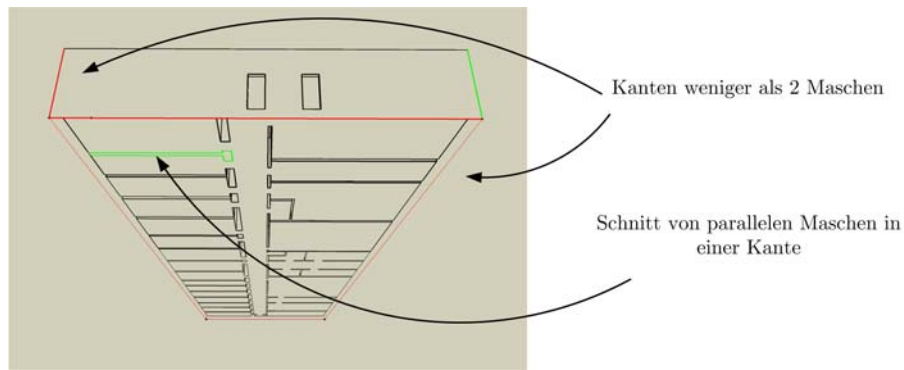


Abbildung 7.4: Visualisieren von Zeichnungsfehlern (b)

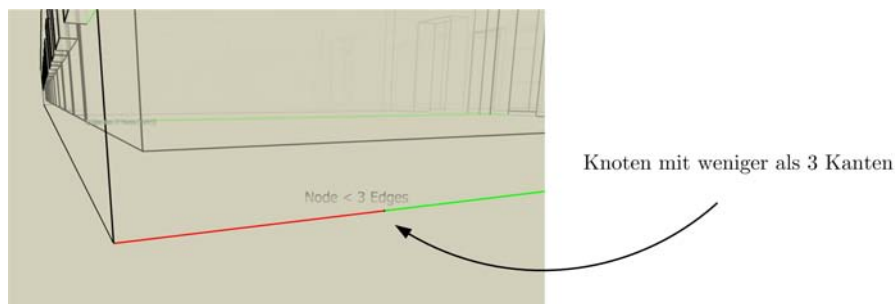


Abbildung 7.5: Visualisieren von Zeichnungsfehlern (c)

Inkonsistenzen beseitigt werden. Ein topologisch korrektes Modell ist die Voraussetzung für den Import der Gebäudestruktur in die Software *Popa3d*.

7.1.4 Exportieren und Normalisieren (Schritt 3)

Das bereinigte, topologisch-geometrisch konsistente Modell wird nach *Popa3d* importiert. Auf der Modellebene erfolgt hier der Übergang zur Flächenparametrisierung und eine Transformation (Normalisierung) der Topologie. Auf der Softwareebene erfolgt der Übergang vom kommerziellen *Google SketchUp* in das in dieser Arbeit entwickelte Programm *Popa3d*, das die binäre Datei im *SketchUp* Format (*.skp) liest und die Daten in das eigene Code-Modell transformiert. Zum Zweck der Datensicherung und Validierung können die Daten als XML-Datei gemäß des *Popa3d* XML-Schemas oder in einer relationalen Datenbank gespeichert werden. Nach der Transformation von einer punktparametrisierten Darstellung in eine flächenparametrisierte Darstellung erfolgt die geometrische Normalisierung in mehreren, voll automatisierten Schritten: Das Modell enthält zunächst eine 1:1 Beziehung zwischen Masche:Normalenvektor. In einem ersten Schritt (Abbildung 7.7) wird die Parallelität der Maschen genutzt indem gleiche Normalenvektoren vereinigt werden. Es entsteht eine 1:1 Beziehung zwischen Masche:Ebene und eine 1:N Beziehung zwischen Ebene:Normalenvektor.

Anschließend werden, unter der Ausnutzung der Koplanarität, gleiche Ebenen vereinigt (Abbildung 7.8). Es entsteht eine 1:N Beziehung zwischen Masche:Ebene.

Die Zusammenfassung (Tabelle 7.1) der **Anzahl** der topologischen und geometrischen Elemente zeigt eine Reduktion der Geometrieparameter von 3x1104 (Koordinaten) auf 232 Ebenenparameter d und 9 Parameter der Normalenvektoren. Das entspricht einer Parameterreduktion um 92%!

Das Model ist nun hinreichend geometrisch und topologisch normalisiert. Damit ist die geometrisch-topologische Beschreibung der Gebäudestruktur für die Ausgleichsrechnung vorbereitet.

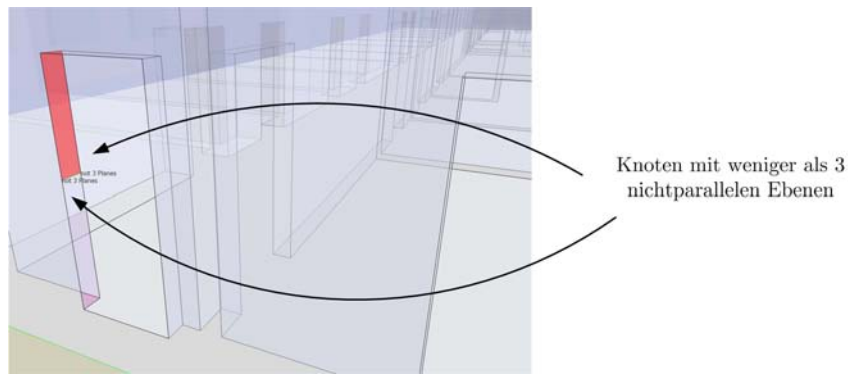


Abbildung 7.6: Visualisieren von Zeichnungsfehlern (d)

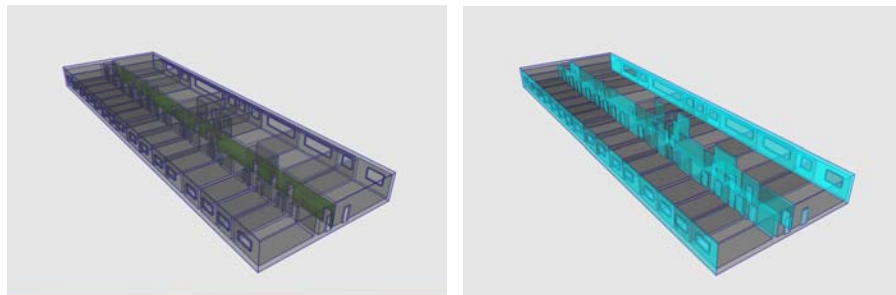


Abbildung 7.7: Modell vor und nach der Vereinigung von gleichen *Normalenvektoren*

7.1.5 Messen (Schritt 4)

Die Software *Popa3d* erfüllt im Arbeitsschritt „Messen“ die Aufgabe eines dreidimensionalen graphischen Feldbuchs zur Spezifikation der Beobachtungstopologie und Beobachtungsgeometrie. Die topologischen Elemente können in der 3d-Benutzerschnittstelle selektiert werden. Die Spezifikation der Beobachtungstopologie erfolgt graphisch-interaktiv im 3d-Modell oder alphanumerisch in den Eingabefeldern. Bei einem „Klick“ mit der Maus auf dem Bildschirm rechnet die *OpenGL*-Graphikkarte die Bildschirmkoordinate in einen Zielstrahl um und sendet eine Liste (z-Buffer) der „getroffenen“ Elemente an die Software zurück. Die beim Mausklick „getroffenen“ Beobachtungen, Masche, Kante, Knoten sind mit den in *Popa3d* vergebenen Identifikatoren bezeichnet. Mit der rechten Maustaste kann der Benutzer durch den z-Buffer entlang der „getroffenen“ Objekte auf dem Zielstrahl navigieren. Das jeweils gewählte Element wird farblich gekennzeichnet.

Sind zwei gültige Elemente selektiert, wird ein Beobachtungsobjekt erzeugt und die Werte der Beobachtungsgrößen kabellos vom Messinstrument an die Software übertragen und angezeigt. Abbildung 7.9 zeigt die Messung mit einem Entfernungsmessgerät. Die beiden grünen Maschen wurden vom Benutzer im Feldbuch selektiert und bilden die Topologie der Distanzmessung.

7.1.6 Ausgleichungsrechnung (Schritt 5)

Die Ausgleichungsrechnung kann zu jedem Zeitpunkt der Vermessung gestartet werden (Abbildung 7.10). Bei einer unvollständigen Beschreibung der Gebäudegeometrie durch Messwerte wird die „nach Augenmaß“ dimensionierte Geometrie der in *SketchUp* erzeugten Skizze zur Berechnung der Gebäudegeometrie verwendet. Die Parameterschätzung schon während der schrittweisen Beobachtungseingabe ermöglicht die Kontrolle und Visualisierung der Messung direkt vor Ort (Abbildung 7.11). So kann die Gebäudegeometrie schrittweise bearbeitet und verbessert werden. Dabei wird die Vollständigkeit und Kontrolliertheit der Messung während jedes Bearbeitungsschrittes berechnet.

Die Qualität der Vermessung wird durch die Angabe der normierten Verbesserungen und der Redundanzanteile dokumentiert. Fehlerhafte Messungen oder Eingaben können sofort automatisiert detektiert und manuell

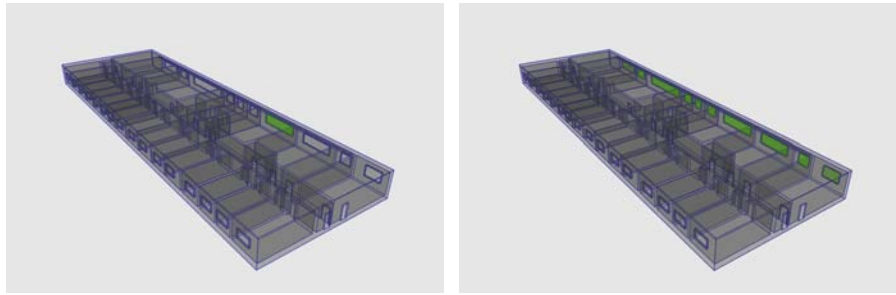


Abbildung 7.8: Modell vor und nach der Vereinigung von gleichen **Ebenen**

Element	SketchUp	Popa3d	Popa3d	Popa3d
Gemetrisches Konzept	Punkt	Ebene planar	Ebene parallel	Ebene koplanar
Node	1104	1104	1104	1104
Edge	1783	1780	1780	1780
Face	734	732	732	732
Plane	X	732	732	232
Normal	X	732	3	3
Parameter	X	2196	9	9
3d-Koordinate	1104	x	x	x
Anzahl der Parameter	100%	88%	22%	7%

Tabelle 7.1: Parameterreduktion durch Normalisierung

geändert werden.

Die Darstellung des Ausgleichungsergebnisses (Abbildung 7.11) erfolgt graphisch im dreidimensionalen Feldbuch und tabellarisch in einer interaktiven Liste. Die Selektion einer Beobachtung kann wahlweise in der 3d Darstellung oder der Liste erfolgen. Zum Beispiel lassen sich die Beobachtungen nach der Größe der Verbesserung (v), der normierten Verbesserung (NV) oder dem Grad der Kontrolliertheit (EV) sortieren.

Die im Rahmen dieser Arbeit entwickelte Software erlaubt das schrittweise Einfügen und Ausgleichen von Messungen. Zur Vermeidung von Konfigurationsdefekten werden Pseudobeobachtungen, die sich aus der Skizze ableiten, in das Gleichungssystem eingeführt. Ein Konfigurationsdefekt entsteht dann, wenn Unbekannte weder direkt noch indirekt aus den Beobachtungen herzuleiten sind. Im vorliegenden Beispiel wurden zunächst einige Türbreiten, eine Raumhöhe sowie eine geschätzte Länge und Breite des Gebäudes als Beobachtung in die Ausgleichung aufgenommen. Das Beobachtungsmaterial kann sofort (ohne Ausgleichung) mit der Skizzengeometrie verglichen werden (Tabelle 7.2).

Gleichungstyp	Genauigkeit (1σ)	Anzahl	max. Widerspruch
Abstand Masche-Masche	($\pm 1 \text{ mm}$)	36	13,00 [m]
Differenzen zwischen Ebenen (Skizze)	($\pm 10 \text{ cm}$)	228	0,00 [m]
Relativwinkel der Normalenvektoren (Skizze)	($\pm 0,001$)	3	0,0 [°]
Normierung der Normalenvektoren	($\pm 0,001$)	3	0,0 [°]
Datumsfestlegung mit Eigenvektoren	deterministisch	6	

Tabelle 7.2: Beispiel: Ausgleichung mit 36 Distanzmessungen und Skizzengeometrie

Die Widersprüche zwischen Messung und Skizze üben in der anschließenden Ausgleichung den Zwang aus, der zur Veränderung der Objektgeometrie führt. Abbildung 7.12 zeigt das Ausgleichungsergebnis. Die eingeführten Abstandsbeobachtungen erhalten aufgrund der höheren Genauigkeit sehr kleine Verbesserungen, während die a priori Differenzen (Skizzengeometrie) große Verbesserungen aufweisen.

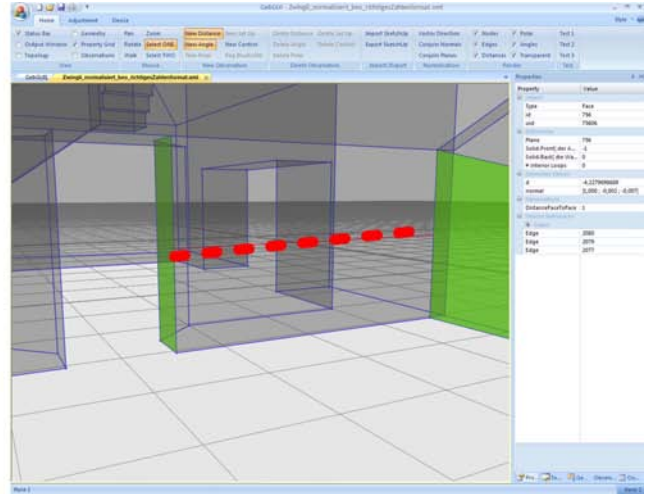


Abbildung 7.9: Dreidimensionale Benutzereingabe bei der Messung mit einem Entfernungsmessgerät



Abbildung 7.10: Visualisierung der Objektgeometrie vor und nach der Ausgleichsrechnung

Die Parameter der Normalenvektoren bleiben unverändert. Die Ebenenabstände d verändern sich in ihrer absoluten Lage um bis zu 6,0 [m]. Im Laufe der Vermessung werden die Abstandsmessungen sukzessive in das Modell eingeführt. Abbildung 7.12 zeigt einen Zwischenschritt in der Bearbeitung. Schon während der Vermessung können die Beobachtungen auf ihre *Zuverlässigkeit* geprüft werden. Große Normierte Verbesserungen lassen auf unzulässige Widersprüche im Beobachtungsmaterial schließen. Um die *Vollständigkeit* der Messung zu prüfen, werden die aus der Skizzengeometrie abgeleiteten Relativmaße auf Ihre Redundanzanteile geprüft. Dabei können die a priori Relativmaße zwischen Ebenen angezeigt werden, die „noch“ einen Einfluss auf das Ausgleichungsergebnis haben (Abbildung 7.13).

7.1.7 Bewertung des Anwendungsfalls

Die Arbeit an diesem Beispiel hat gezeigt, dass der Einsatz der Ausgleichsrechnung zur (schrittweisen) Bestimmung der Objektgeometrie ein effizientes und zuverlässiges Aufmaß beschleunigt, weil der Benutzer sowohl numerisch als auch graphisch über die Qualität seiner Vermessung informiert wird.

Sowohl die Spezifikation der 3d-Topologie und der Nahrungsgeometrie mit *Google SketchUp*, als auch die interaktive Eingabe der Beobachtungstopologie, die Erfassung der Beobachtungswerte mit einer Laserdistanzmessgerät sowie die Ausgleichung der Messwerte und der Objektgeometrie erfordert keine besonderen vermessungstechnischen Kenntnisse und kann von Laien durchgeführt werden. Die Analyse und Bewertung des Ausgleichungsergebnisses erfordert allerdings ingenieurgeodätischen Sachverstand. In einer zukünftigen Weiterentwicklung der

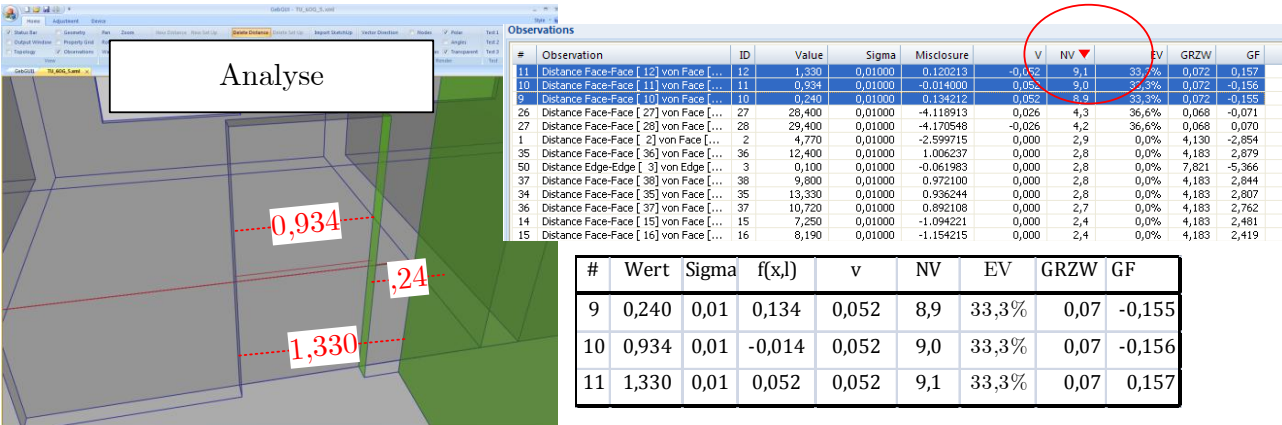


Abbildung 7.11: Analyse der Objektgeometrie während der Vermessung

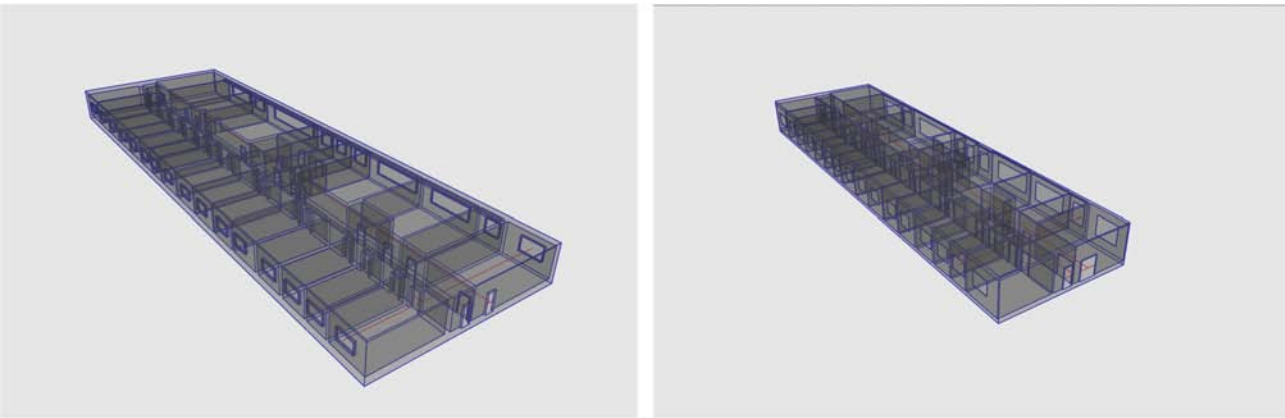


Abbildung 7.12: Veränderung der Objektgeometrie nach einer ersten Ausgleichung

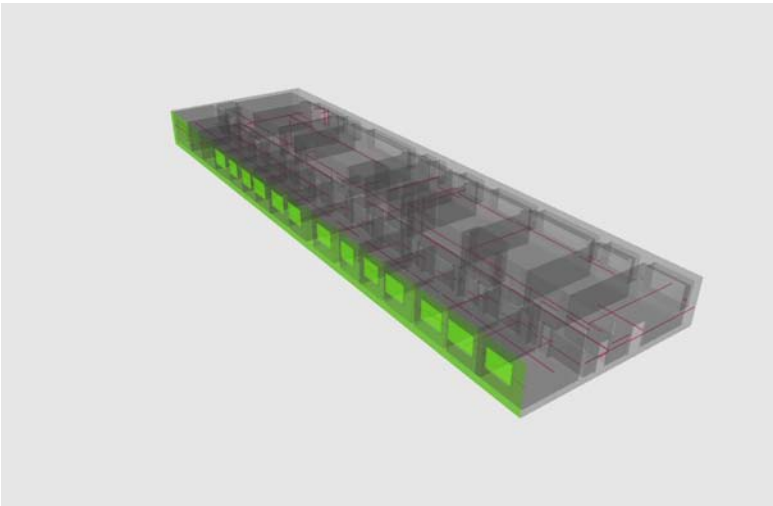


Abbildung 7.13: Visualisierung noch nicht gemessener Elemente

Software müssen daher konkrete Entscheidungsunterstützungen durch die Software erzeugt und dem Anwender angezeigt werden.

Die Reduzierung der zu berechnenden Geometrieparameter (aufgrund der geometrisch-topologischen Normalisierung) macht eine vermessungsbegleitende Ausgleichung erst praktikabel, weil hierfür nur eine geringe Rechenzeit nötig ist. Für den praktischen Einsatz sollte versucht werden, dass die graphische Benutzerschnittstelle

Gleichungstyp	max. Verbesserung	\varnothing Kontrolliertheit (EV)
Abstand Masche-Masche	3 [mm]	1%
Differenzen zwischen Ebenen (Skizze)	58 [cm]	90%
		,wenn korreliert mit Abstandsmessung
		1%
		,wenn unkorreliert mit Abstandsmessung
Relativwinkel der Normalenvektoren (Skizze)	0.00 []	0%
Normierung der Normalenvektoren	0.00 []	0%

Tabelle 7.3: Analyse der Ausgleichung mit 36 Distanzmessungen und Skizzengeometrie

von *Popa3d* die Bemaßung von Standardobjekten (Tür, Fenster, Mauer) besser unterstützt. Eine Darstellung in Parallelperspektive kann die graphische Auswahl von Objekten gegebenenfalls beschleunigen. Der Zwischschritt, erst die Topologie und Nahrungsgeometrie zu spezifizieren und dann anschließend die Geometrie schrittweise zu verfeinern, sollte für den Benutzer in einer graphischen Benutzerschnittstelle zusammengefasst werden. Es wäre sinnvoll, die Funktionalität von *Popa3d* (Datenmodell, Normalisierung, Ausgleichung, Validierung, XML-Serialisierung) als Softwarebibliothek an *Google SketchUp* zu binden, so dass der Benutzer nur eine Software zu bedienen hat.

7.2 Anwendungsfall Sekundärdatenerfassung

Das Beispiel *Sekundärdatenerfassung*¹ zeigt, wie *Popa3d* für den Zweck der geometrisch-topologischen Qualitätsverbesserung eingesetzt werden kann. Im Gegensatz zum Anwendungsfall *Ersterfassung* in Kapitel 7.1 liegen keine geodätischen Messungen vor. Stattdessen ist ein vorhandenes 3d-Modell die alleinige Datengrundlage. Aus dem Zusammenspiel zwischen *Geometrie* und *Semantik* werden Bedingungen erzeugt, die der Ausgleichungsrechnung zugeführt werden, um das geometrische Modell zu verbessern. Im Kontext raumbezogener Informationssysteme meint der Begriff *Semantik*, dass die geometrischen Objekte wissen, was sie darstellen und/oder dass die semantischen Objekte wissen, wo sie sind [Stadler 2007, S. 175]. *Geometrisch verbessern* meint im Kontext der Sekundärdatenerfassung, dass Erfassungsfehler, zum Beispiel doppelte Linien, *overshots* bzw. *undershots* oder vertikale Versätze zwischen Stockwerken, bereinigt werden.

Ausgangssituation: Es liegt ein *CityGML* Dokument [Gröger/Kolbe/Czerwinski/Nagel 2008] vor, das ein einzelnes dreistöckiges Gebäude in einem lokalen Koordinatensystem abbildet. *CityGML* ist ein vom Open Geospatial Consortium (OGC) veröffentlichtes Informationsmodell zur Repräsentation dreidimensionaler, urbaner Objekte. Der Diskursbereich von *CityGML* umfasst die Topologie, Geometrie, Semantik und graphische Darstellung relevanter topographischer Objekte in verschiedenen Generalisierungsstufen.

Das vorliegende Modell ist automatisiert aus extrudierten 2d-Grundrissen entstanden. Es weist zwischen den Stockwerken einen Versatz auf, der nicht der Realität vor Ort entspricht, sondern aufgrund von Mess- oder Zeichnungsfehlern entstanden ist. Dieser Versatz (siehe Abbildung 7.15²) soll bereinigt werden, ohne die lokalen Eigenschaften der Fensterflächen zu verändern.

7.2.1 Transformation CityGML → Popa3d (Schritt 1)

Die *CityGML* Datei wird mit Hilfe einer für das Berliner Stadtmodell entwickelten Software in eine Java-Klassenstruktur [Nagel und Stadler 2008] importiert und das Gebäudemodell nach *Popa3d* überführt. Hierfür müssen die punktparametrisierten Polygone `<gml:LinearRing>` gemäß der Kapitel 2.10 und 2.11 topologisch und geometrisch normalisiert werden. Es werden:

1. die topologischen Primitive Knoten, Kante, Halbkante, Masche und Raum gebildet.
2. die Beziehungen zwischen den topologischen Primitiven ermittelt.

¹Der Anwendungsfall wurde im Rahmen der Masterarbeit [Pasewaldt 2010] entwickelt und untersucht.

²Zur Visualisierung der *CityGML* Datei wurde das Programm LandexplorerTM der Firma Autodesk® verwendet.

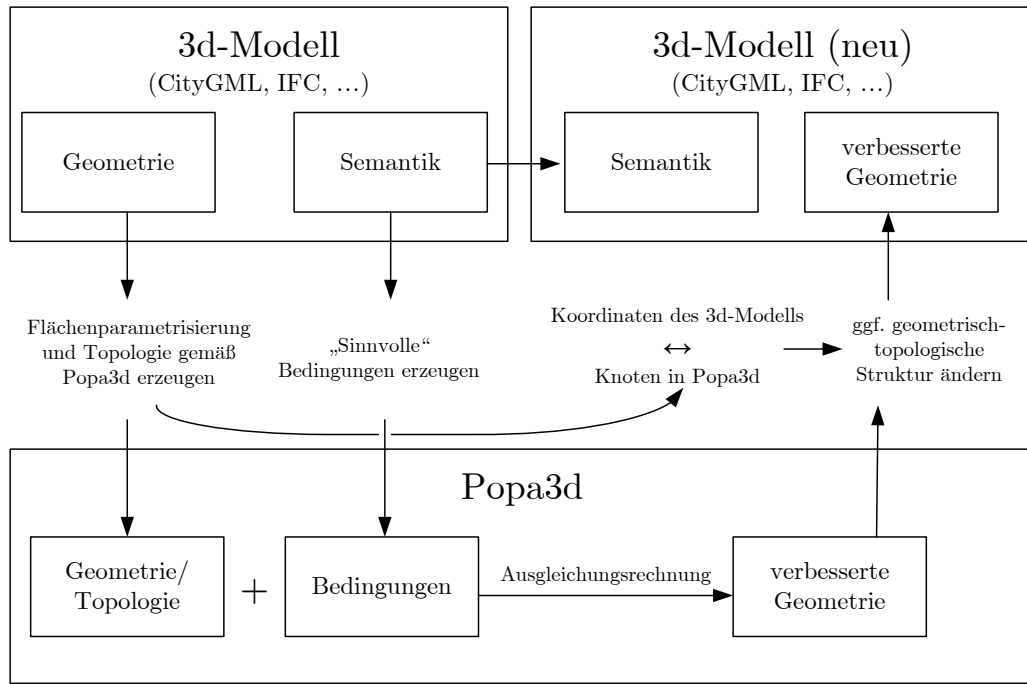


Abbildung 7.14: Informationsfluss Anwendungsfall „Sekundärdatenerfassung“

3. die Referenzen zwischen den topologischen Primitiven innerhalb der *Popa3d* Datenstruktur redundanzfrei gespeichert.
4. zum Zweck der späteren Rücktransformation Referenzen zwischen den Knoten in *Popa3d* und den Koordinaten `<gml:coordinates>` des *CityGML* Dokumentes gespeichert.
5. die geometrischen Elemente Ebene, Normalenvektor und Parameter gebildet.
6. die geometrischen Elemente gemäß Abschnitt 2.11 normalisiert.
7. anhand der semantischen Information geometrische Bedingungen für die Ausgleichungsrechnung erzeugt.
8. ein XML-Dokument gemäß dem *Popa3d* Schema erstellt.

Dabei werden die Bedingungen automatisch so eingeführt, dass zwei parallelen Polygone i und j vom Typ `<citygml:wallsurface>` mit einem „kleinen“ Unterschied $\Delta d_{ij} < \epsilon$ in der Translationskomponente d keinen Abstand haben sollen.

$$l + v = 0 + v = \Delta d_{ij} = d_j - d_i \quad (7.1)$$

Hierfür werden Abstandsbedingungen (bzw. Beobachtungen) vom Typ *Abstand Masche-Masche* gebildet (siehe Kapitel 2.13.2 und 3.2). Neben dieser rein geometrischen wird auch semantische Information ausgewertet: Zur Erhaltung der lokalen Nachbarschaft zwischen Außenwand `<citygml:wallsurface>` und Fenster `<citygml>window>` werden die aus der dem **CityGML**-Dokument abgeleiteten Abstandsmaße $\Delta d_{ij} \neq 0$ als Objekt gebildet und in die Ausgleichungsrechnung eingeführt, auch wenn sie kleiner als der Schwellenwert ϵ sind.

Das Ergebniss von Schritt 1 ist ein XML-Dokument gemäß der *Popa3d* Schemadefinition und eine temporäre Datei. Das *Popa3d* XML Dokument enthält die flächenparametrisierte Gebäudestruktur mit den zusätzlichen Bedingungen. Die temporäre Datei, persistenziert die eindeutige Zuordnung der *Popa3d*-Knoten mit den Koordinaten der *CityGML* Instanz, die für die Rücktransformation benötigt wird. (siehe Abbildung 7.14)

¹Anfangs und Endpunkt werden doppelt gespeichert

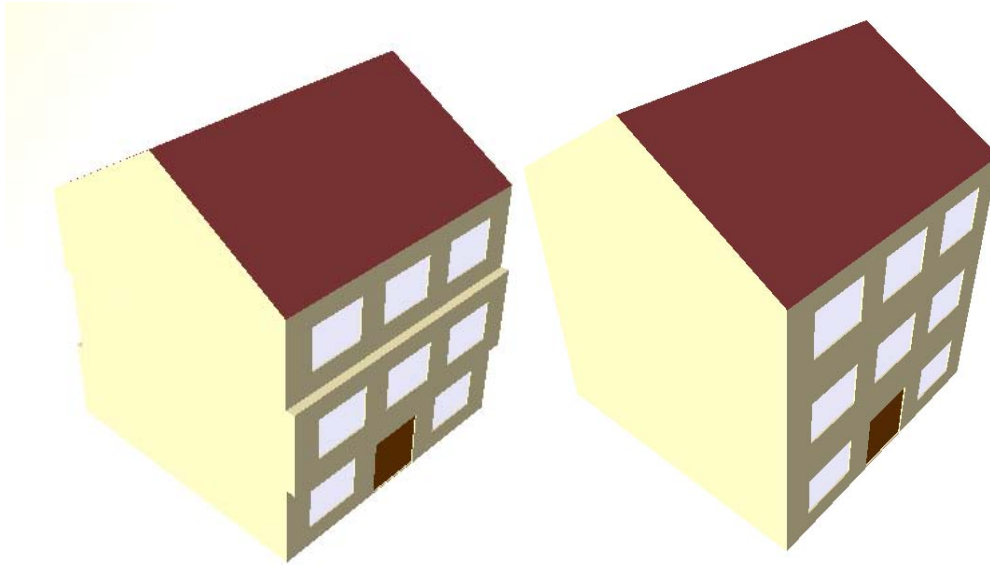


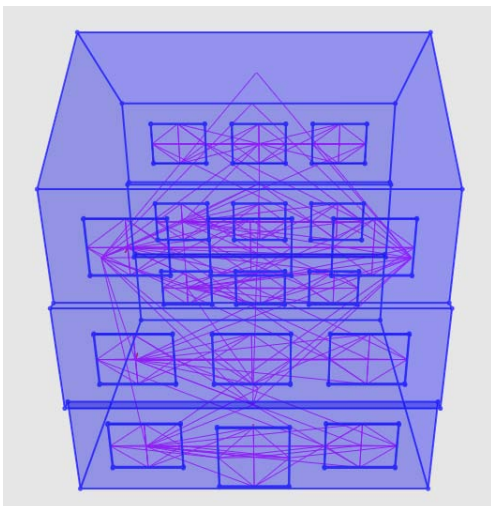
Abbildung 7.15: Das 3d-Modell (CityGML) *vor* und *nach* der Ausgleichung

Typ	CityGML	Typ	Popa3d
Polygone	111	Maschen	105
LinearRings	129	Halbkanten	510
		Kanten	255
Punkte	514 (643 ¹)	Knoten	170
Ebenen	111	Ebene	31
Normalenvektoren	111	Normalenvektor	8
Parameter	333	Parameter	19

Tabelle 7.4: Anzahl der Primitive im Beispieldatensatz

7.2.2 Ausgleichungsrechnung (Schritt 2)

Das XML Dokument kann nun in die *Popa3d* Software eingelesen, visuell inspiziert und gegebenenfalls um weitere Bedingungen erweitert werden.



- 97 Parallelitätsbedingungen
- 78 Koplanaritätsbedingungen
- 252 Orthogonalitätsbedingungen
- 18 Abstandsbeobachtungen (Wand-Fenster)
- 50 Identitätsbeobachtungen
- 8 Betrag des normierten NV = 1

Abbildung 7.16: Bedingungen

Abbildung 7.16 zeigt die auszugleichende Objektgeometrie (blau) und die Bedingungen (rot). Die aus den Gebäudegrundrissen entstandene Geometrie wird in der Ausgleichung schwach gewichtet, während die Abstandsbedingungen einen starken Einfluss auf das Ausgleichungsergebnis haben. Das Ergebnis der Berechnung ist ein geometrisch verändertes Gebäudemodell, dessen Aussenwände, wie gefordert, koplanar sind. Der ursprüngliche Abstand der Aussenfassade beziehungsweise der Innenwand zu den Fenstern wird unter Einhaltung der lokalen Nachbarschaft beibehalten.

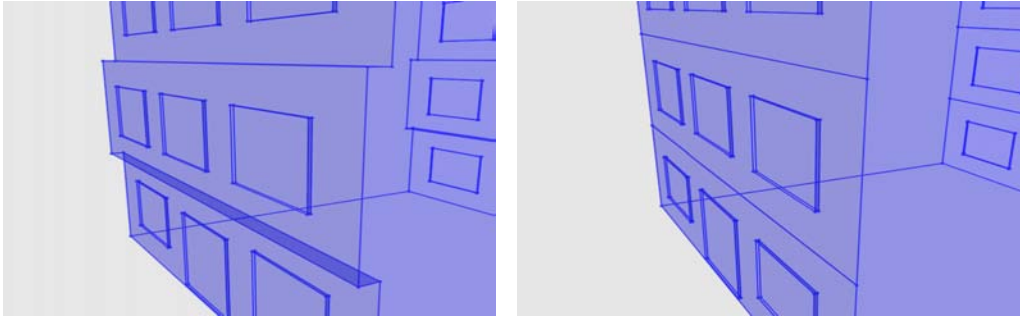


Abbildung 7.17: Das 3d-Modell (*Popa3d*) *vor* und *nach* der Ausgleichung

7.2.3 Transformation *Popa3d* \rightarrow *CityGML* (Schritt 3)

Bei der Überführung der verbesserten Geometrie werden die Koordinaten der *CityGML* Datei bearbeitet. Im vorliegenden Beispiel induziert die geometrische Veränderung darüberhinaus zwei topologische Typen von Änderungen:

- Die Polygone des Versatzes, die im Original waagrecht sind, sind nach der Ausgleichung ohne geometrische Ausdehnung (keine Fläche). Sie werden bei der Konvertierung *Popa3d* \rightarrow *CityGML* aus dem Modell entfernt. Topologische Nachbarschaftsbeziehungen zwischen Maschen, Halbkanten, Kanten und Knoten werden dabei kaskadierend nachgeführt.
- Polygone, die semantisch gleich (mit `<citygml:WallSurface>`) und koplanar sind, werden zu einem Polygon vereinigt und die Referenzen, zum Beispiel der Fenster, nachgeführt.

	Typ	CityGML (original)	CityGML (neu)
	Polygone	111	101
	LinearRings	129	119
	Punkte	514 (643 ³)	170 (597)

Tabelle 7.5: Anzahl der Primitive vor und nach der Fehlerbereinigung

7.2.4 Bewertung des Anwendungsfalls

Das einfache Beispiel zeigt die Möglichkeit, mit den Methoden der Ausgleichungsrechnung und einer flächenparametrisierten Geometriebeschreibung einen existierenden Datensatz geeignet zu vereinfachen und geometrisch zu verbessern. Die Ausgleichungsrechnung ist ein geeignetes Mittel zur Informationsintegration (Bedingungen, Modell) und Validierung. Es wird gezeigt, dass eine gemeinsame und kohärente Spezifikation von Semantik und Geometrie für eine automatisierte Definition von Bedingungen genutzt werden kann. Die Ausgleichungsrechnung auf Grundlage des *Popa3d* Informationsmodells hat hierbei die Funktion der *Informationsintegration* und Bewertung. Sie schätzt das „beste“ Ergebnis gemäß der Bedingungen und erlaubt statistische Aussagen über die Genauigkeit und Zuverlässigkeit der vorgenommenen Integration.

Ein Ergebnis weiterer Forschungsarbeiten könnte eine auf der semantischen Beschreibung einzelner Gebäudeteile beruhende *Bedingungs-Grammatik* sein, die automatisiert sinnvolle Bedingungen erzeugt. Ergänzend zur Geo-

metrie der Originaldatei könnten weiterhin einige wenige zusätzliche Messungen in die Ausgleichungsrechnung integriert und damit das Gebäudemodell zuverlässiger beschrieben bzw. kontrolliert werden.

Der Anwendungsbereich von *Popa3d* bleibt nicht auf die Ersterfassung beschränkt. Die gleichen Prinzipien und Methoden (Flächenparametrisierung, Normalisierung, Ausgleichungsrechnung und empirische Primärdaten) können auch im Kontext der Generalisierung und Sekundärdatenerfassung gewinnbringend eingesetzt werden.

8 Zusammenfassung und Ausblick

Die vorliegende Arbeit verfolgt zwei Ziele. Zum einen wird eine Methode zur Bestandsdokumentation entwickelt, die das Aufmaß von Innen- und Außenraum eines Gebäudes im Sinne einer Ersterfassung effizient gestaltet. Das zweite Ziel sind Methoden zur Validierung bereits vorhandener 3d-Modelle eines Gebäudes, um die geometrisch-topologisch Abbildung der Gebäudestruktur zu verbessern. Die Ziele werden mit den Verfahren der Ausgleichsrechnung erreicht. Herkömmliche Datenmodelle sind für eine stochastische Optimierung allerdings ungeeignet, weil sie die Objektgeometrie redundant speichern und daher numerisch schlecht handhabbar sind. Das in dieser Arbeit entwickelte Datenmodell integriert die Vermessung und Qualitätskontrolle als Bestandteil des Informationsmodells in die Datenhaltung und -analyse. Die ingenieurgeodätischen Methoden der Parameterschätzung und statistischen Tests sind so in die Software integriert, dass sie auch von Laien benutzt werden können.

Hinsichtlich des geometrisch-topologischen Datenmodells geht die vorliegende Arbeit neue Wege: Träger der geometrischen Information ist nicht der Punkt, sondern die Fläche. Durch eine explizite Spezifikation der Objekttopologie und eine für die effiziente Parameterschätzung optimierte Datenstruktur können die herkömmlich als getrennt behandelten Modelle der 3d-CAD-Welt mit denen der Ausgleichsrechnung verbunden werden. Die Neuartigkeit der vorliegenden Arbeit besteht in der konzeptionellen Erweiterung des in [Gielsdorf 2007] entwickelten Ansatzes, der methodischen Umsetzung in einem funktionalen und stochastischen Modell mittels erweiterbarer Software und in der Validierung der Methode an konkreten Anwendungsbeispielen.

Popa3d und die Implementierung der Parameterschätzung bilden eine generische Grundlage zur **Integration weiterer Sensortypen** in das Datenmodell und die Ausgleichsrechnung. Ausgleichsansätze aus dem Anwendungsbereich der Photogrammetrie, des Laserscannings oder 3d-Videos müssen hierfür in der funktionalen Modellierung an die Flächenparametrisierung angepasst und in ihrer datentechnischen Integration an das entwickelte Informationsmodell gebunden werden. Aus bautechnischer Sicht ist es interessant, die Dimension von Bauteilgruppen oder Fertigbauteilen in das funktionale Modell der Parameterschätzung aufzunehmen. Beispielsweise könnten standardisierte Mauerstärken automatisiert detektiert und dann mit einem „hohen“ Gewicht in die Ausgleichsrechnung eingeführt werden.

Im Rahmen dieser Arbeit wurde die Gestaltung der **graphischen Benutzerschnittstelle** nicht genauer untersucht. Eine effiziente Bearbeitung des 3d Modells ist neben einer geeigneten Datenstruktur und kurzen Rechenzeit der Ausgleichsrechnung nur dann gewährleistet, wenn der Benutzer bei der Interaktion mit dem Modell ausreichend graphisch unterstützt wird. Hierfür sind weitere Untersuchungen hinsichtlich der Eingabe und Bearbeitung von Beobachtungen und Bedingungen, der Visualisierung der Residuen und statistischen Kenngrößen und eine geeignete Benutzerführung bei der Sachdateneingabe notwendig. Die in [Nichelmann 2007] entwickelten Algorithmen zur Darstellung und Selektion von Abstandsbeobachtungen in flächenparametrisierten Geometriemodellen werden in der eingesetzten Testsoftware bereits verwendet.

Bei der **Spezifikation der Topologie** durch einen Benutzer kann mit Euler-Poincaré Operatoren das topologische Modell von *Popa3d* stets in einem topologisch konsistenten Zustand gehalten werden [Kruschwitz 1996]. Untersuchungen hierzu wurden in [Krämer 2007a] durchgeführt. In der vorliegenden Arbeit werden die Modelle mit dem dreidimensionalen Entwurfsprogramm *Google SketchUp* erzeugt und mit im Rahmen dieser Arbeit angefertigten Plug-ins die Topologie und Geometrie auf Konsistenz geprüft und Fehler visualisiert. Dann wurden die Daten von der *Popa3d* Software eingelesen. Es muss untersucht werden, ob es sinnvoll ist, auch die Eingabe, Bearbeitung und Ausgleichung von Beobachtungsdaten und geometrischen Bedingungen mit einem zu erweiternden kommerziellen 3d-Modellierungswerkzeug durchzuführen, oder ob - wie in der vorliegenden Arbeit - eine eigene graphische Benutzerschnittstelle notwendig ist. Weiterhin müssen die Methoden zur Spezifikation der dritten topologischen Dimension (Raum, Körper) verbessert werden, weil sowohl der Raum, im Sinne von „Zimmer“, als auch das Bauteil das Gebäudemodell geeignet gliedern.

Die Validierung von Gebäudemodellen, die in **Standardmodellen** gespeichert sind erfordert die Möglichkeit eines verlustfreien Informationsaustausches zwischen den Datenmodellen. Die Verlustfreiheit muss dabei auf die Abbildung der Gebäudestruktur (Geometrie und Topologie) begrenzt bleiben, da es keinen Standard zur Spezifikation der Beobachtungstopologie und Messwerte gibt. Die Transformation kann nach den einzelnen Schritten der topologischen und geometrischen Normalisierung, wie sie in den Abschnitten 2.10 und 2.11 vorgestellt werden, folgen. Eine Transformation von *Popa3d* zu anderen Datenmodellen und zurück wurden in einigen Diplomarbeiten am Fachgebiet für Geodäsie und Geoinformationstechnik untersucht: In [Michel 2006] konnte gezeigt werden, dass ein Export von *Popa3d* in das *IFC* Modell (Industry Foundation Classes) möglich ist, das Importieren der *IFC* Modelle aber einen hohen Konvertierungsaufwand erfordert, weil die IFC eine große Anzahl verschiedener Modellierungskonzepte für Geometrie und Topologie unterstützen. In [Pasewaldt 2010] konnte gezeigt werden, dass eine Transformation der geometrisch-topologischen Struktur zwischen *Popa3d* und *CityGML* grundsätzlich möglich ist und dass die in einem *CityGML*-Dokument angegebene Semantik genutzt werden kann, um geometrische Bedingungen für die Ausgleichsrechnung automatisch zu generieren. Weitere Forschungsarbeiten sind hier nötig, um *Popa3d* in einen möglichst voll automatisiert im Kontext der Sekundärdatenerfassung einzusetzen. In einem vielversprechenden Ansatz wurde in [Tischer 2009] das *Popa3d* Datenmodell relational in einer Oracle Datenbank abgebildet und dann mit den räumlichen Datentypen einer **objektrelationalen Datenbank** erweitert. Zum Zweck der Speicherung von raumbezogenen 3d-Massendaten kann hier eine objektrelationale Datenbank in Verbindung mit einer relationalen Modellierung der Beobachtungstopologie und Bedingungen als eine Mittelschicht zwischen Geographischem Informationssystem und Ausgleichsrechnung genutzt werden. Das DBMS (Datenbank Management System) übernimmt dann die Aufgaben der Datenspeicherung, Sicherung, Versionierung, Indizierung etc. und realisiert standardisierte Schnittstellen zu 3d-GIS/3d-CAD Software.

Popa3d Modelle beschränken sich in der vorliegenden Arbeit auf lokale dreidimensionale Bezugssysteme. Die Transformation von *Popa3d* Modellen in ein geographisches oder **geodätisches Bezugssystem** zur Verarbeitung in einem Geographischen Informationssystem (GIS) wurde in dieser Arbeit nicht untersucht. Ein praktisches und einfaches Vorgehen wäre es, das Datenmodell so zu erweitern, dass jeder Relation zwischen „Gebäude“ und „Koordinatensystem“ eine geometrische Transformationsvorschrift zugeordnet ist.

Die vorliegende Arbeit liefert einen Beitrag zur ingenieurgeodätischen Methodenentwicklung auf dem Gebiet der dreidimensionalen Erfassung und Validierung von Gebäudestrukturen. Der praktische Nutzen der Kombination aus Flächenparametrisierung, Topologie und Ausgleichsrechnung ist sicherlich noch nicht vollständig erkannt und wird in Zukunft in vielfältigen Anwendungsgebieten vertreten sein.

Literaturverzeichnis

- [Bendtsen und Stauing 1996] BENDTSEN, C.; STAUNING, O.: *FADBAD, a Flexible C++ Package for Automatic Differentiation*. Technical Report IMM-REP-1996-17 Department of Mathematical Modelling. Lyngby, Denmark: Technical University of Denmark, 1996
- [Böck 1961] BÖCK, R.: Allgemeinste Formulierung der Ausgleichungsrechnung nach der Methode der kleinsten Quadratsumme. In: *Zeitschrift für Vermessungswesen*. 86. Jahrgang (1961), S. 98-106.
- [Bradley und Paul 2009] BRADLEY, P.E.; PAUL, N.: *Architektonische Komlexe*. ww1.mathematik.uni-karlsruhe.de/~bradley/Arch/Archkomp/index.html (04/2009)
- [Cooors 2003] COORS, V.: 3D GIS in Networking Environments In: *Computer, Environment and Urban Systems*, Band 27(4), S. 345-357.
- [Foley u.a. 1995] FOLEY, J.; VANDAM, A.; FEINER, S.: *Computer Graphics: Principles and Practice in C*. 2.Ausgabe, Amsterdam, Niederlande: Addison-Wesely Longman, 1995
- [GML 2007] PORTELE, C (Hrsg.): *OpenGIS Geography Markup Language (GML) Encoding Standard*, Version 3.2.1, OGC Document 07-036, Open Geospatial Consortium, 2007
- [Gielsdorf 2007] GIELSDORF, F.: *Ausgleichungsrechnung und raumbezogene Informationssysteme* Deutsche Geodätische Kommission DGK bei der Bayerischen Akademie der Wissenschaft, Reihe C, Heft 593, München, 2007
- [Gröger 2000] GRÖGER, G.: *Modellierung raumbezogener Objekte und Datenintegrität in GIS*. Heidelberg: Wichmann Verlag, 2000
- [Gröger/Kolbe/Czerwinski/Nagel 2008] GRÖGER, G.; KOLBE, T.; CZERWINSKI, A.; NAGEL, C.: *OpenGIS City Geography Markup Language (CityGML) Encoding Standard* Version 1.0.0, International OGC Standard Open Geospatial Consortium, Doc. No. 08-007r1 2008
- [Gründig 2003] GRÜNDIG, L.: *Grundlagen der Ausgleichungsrechnung*, Vorlesungsskript. Technische Universität Berlin, 2003 (unveröffentlicht)
- [Gründig und Gielsdorf 2002] GRÜNDIG, G.; GIELSDORF, F.: Geometrical Modelling for Facility Management Systems Applying Surface Parameter. In: International Federation of Surveyors (FIG): *XXII FIG International Congress*, Washington, USA: International Federation of Surveyors (FIG), 2002
- [Hau 2000] HAU, T.: *Generische Strukturen zur Modellierung der Parameterschätzung*. Dissertation. Rheinische Friedrich-Wilhelms-Universität zu Bonn, 2000
- [Heunecke u.a. 2004] HEUNECKE, O.; NIEMEIER, W.; RETSCHER, G.: Paradigmenwechsel bei der Auswertung ingenieurgeodätischer Messungen. In: INGENSAND, H.: *emphIngenieurvermessung 2004*, Zürich: ETH Zürich, 2004
- [Huhnt und Gielsdorf 2006] HUHT, W.; GIELSDORF, F.: Topological Information as Leading Information in Building Product Models. In: K. GÜRLEBECK u. KÖNKE C. (Hrsg.): *17th International Conference on the Application of Computer Science and Mathematics in Architecture and Civil Engineering*, Weimar: Bauhausuniversität Weimar, 2006
- [IEEE Std 754-2008] IEEE COMPUTER SOCIETY: *IEEE Std 754 - Standard for Floating-Point Arithmetic*. New York, USA, 2008
- [ISO19107:2003] ISO - INTERNATIONAL STANDARD: *ISO 19107 - Geographic Information - Spatial Schema*. Genf, Schweiz, 2003
- [Jählich 2005] JÄHNICH, K.: *Topologie*. Berlin, Heidelberg, New York: Springer-Verlag, 2005
- [Kazakos u.a. 2002] KAZAKOS, W.; SCHMIDT, A.; TOMCZYK, P.: *Datenbanken und XML*. Berlin, Heidelberg, New York: Springer-Verlag, 2002
- [Kersten 2008] KERSTEN, J.: *Anwendung der Differentiations-Arithmetik für Aufgaben der Ausgleichungsrechnung*. Diplomarbeit. Technische Universität Berlin, 2008 (unveröffentlicht)
- [Krämer 2007a] KRÄMER, T.: *Relativmaß als Grundlage für die Konstruktion von Bauteilen*. Diplomarbeit. Technische Universität Berlin, 2007 (unveröffentlicht)
- [Krämer 2007b] KRÄMER, T.: Von der Topologie zur CAD-Konstruktion. In: MERKEL, A.P. AND SCHÜTZ, R. AND WIESSFLECKER, T.: *Forum Bauinformatik 2007*, Graz, 2007
- [Kruschwitz 1996] KRUSCHWITZ, E.: *Euler-Modellierung dreidimensionaler Körper*. Dissertation an der Technischen Universität Berlin. Aachen: Shaker-Verlag, 1996
- [Kuipers 1999] KUIPERS, J.: *Quaternions and Rotation Sequences*. Princeton, New Jersey, USA: Princeton University Press, 1999

- [Leser und Naumann 2007] LESER, U.; NAUMANN, F.: *Informationsintegration*. Heidelberg: dpunkt Verlag, 2007
- [Opić 2006] OPIĆ, M.; KOCH, S.; MAY, M.; REINECKE, W.: Datenerfassung und Datenaustausch für CAFM. In MAY, M. (Hrsg.): *IT im Facility Management erfolgreich einsetzen. Das CAFM Kanbuch*. 2. Auflage. Berlin, Heidelberg, New York: Springer-Verlag, 2006, S. 169-198.
- [Oracle 2009] MURRAY, C.: *Oracle Spatial Developer's Guide, 11g Release 2 (11.2)*, E11830-05, Oracle, 2009
- [Mäntylä 1976] MÄNTYLÄ, M.: *An Introduction to Solid Modeling*. Rockville, Maryland, USA: Computer Science Press, 1976
- [Michel 2006] MICHEL, H.: *Datenaustausch mit den Industry Foundation Classes (IFC)*. Diplomarbeit. Technische Universität Berlin, 2006 (unveröffentlicht)
- [Mikhail und Ackerman 1976] MIKHAIL, E.; ACKERMAN, F.: *Observations and Least Squares*. New York, USA: University Press of America, 1976
- [Molenaar 1990] MOLENAAR, M.: A Formal Data Structure for Three Dimensional Vector Maps. In: BRASSEL, K. u.a. (Hrsg.): *Proceedings of the 4th International Symposium on Spatial Data Handling*, Zürich, Schweiz: 1990, S. 830-843.
- [Nagel und Stadler 2008] NAGEL, C.; STADLER, A.: Die Oracle-Schnittstelle des Berliner 3D-Stadtmodells. In: CLEMEN, C. (Hrsg.): *Entwicklerforum Geoinformationstechnik 2008* Aachen: Shaker-Verlag, 2008, S. 197-221.
- [Neitzel und Petrovic 2008] NEITZEL, F.; PETROVIC, S.: Total Least Squares (TLS) im Kontext der Ausgleichung nach kleinsten Quadraten am Beispiel der ausgleichenden Gerade. *Zeitschrift für Vermessungswesen*, 134. Jahrgang (2008), S. 141-148.
- [Nichelmann 2007] NICHELMANN, K.: *Implementierung einer graphisch-interaktiven 3D Bestandsdatenerfassung von Gebäudegeometrien mit OpenGL*. Diplomarbeit. Technische Universität Berlin, 2008 (unveröffentlicht)
- [Niemeier 2002] NIEMEIER, W.: *Ausgleichungsrechnung*. Berlin, New York: Walter de Gruyter, 2002
- [Pahl und Damrath 2000] PAHL, P.J.; DAMRATH, R.: *Mathematische Grundlagen der Ingenieurinformatik*. Berlin, Heidelberg, New York: Springer-Verlag, 2000
- [Pasewaldt 2010] PASEWALDT, A.: *Geometrical and Topological Improvement of CityGML Building Models by Means of Geometric Constraints*. Master Thesis. Technische Universität Berlin, 2010 (unveröffentlicht)
- [Pilouk 1996] PILOUK, M.: *Integrated Modelling for 3D GIS*. Dissertation. International Institute for Geo-Information Science and Earth Observation ITC, Enschede, Niederlande, 1996
- [Rall 1986] RALL, L.B.: The arithmetic of differentiation. *Mathematics Magazine*, Volume 59, S. 275-282.
- [Shi u.a.2003] SHI, W.; YANG, B.; LI, Q.: An Object-oriented Data Model for Complex Objects in Three-dimensional Geographic Information Systems. *International Journal of Geographic Information Science*, Volume 17, Issue 5, S. 411-430.
- [Stadler 2007] STADLER, A.: Kohärenz von Geometrie und Semantik in der Modellierung von 3D Stadtmodellen CLEMEN, C. (Hrsg.): *Entwicklerforum Geoinformationstechnik 2007* Aachen: Shaker-Verlag, 2007, S. 167-181.
- [Thiemann 2001] THIEMANN, F.: *Modellierung, geometrische Parametrisierung und Ausgleichung von Gebäudedaten*. Diplomarbeit. Technische Universität Berlin, 2001 (unveröffentlicht)
- [Thompson 2007] THOMPSON, R.J.: *Towards a Rigorous Logic for Spatial Data Representation*. Dissertation. Technische Universität Delft, Niederlande, 2007
- [Thurrow 2004] THUROW, T.: Digitaler Architekturbestand - Untersuchungen zur computergestützten, schrittweisen Erfassung und Abbildung der Geometrie im Kontext der planungsrelevanten Bauaufnahme. Dissertation an der Fakultät Architektur der Bauhaus-Universität Weimar, <http://e-pub.uni-weimar.de/volltexte/2004/138/> (aufgerufen 04/2009)
- [Tischer 2009] TISCHER, J.: *Modellierung und Ausgleichung flächenparametrisierter 3d-Gebäudemodelle mit der objektrelationalen Datenbank Oracle Spatial*. Diplomarbeit. Technische Universität Berlin, 2009 (unveröffentlicht)
- [Tse u.a. 2008] TSE, R.O.; KOLD, C.; KIDNER, D.: 3D City Modelling from LIDAR Data. In: OOSTEROM P.v. & ZLATANOVA S.: *Advances in 3D Geoinformation Systems* Berlin, Heidelberg, New York: Springer-Verlag, S. 161-170.
- [Weferling 2002] WEFERLING, U.: *Bauaufnahme als Modellierungsaufgabe*. (Bd. 561 Reihe C). Deutsche Geodätische Kommission DGK bei der Bayerischen Akademie der Wissenschaft, Reihe C, Heft 561, München, 2002
- [Zlatanova 2000] ZLATANOVA, S.: *3D GIS for Urban Development*. Dissertation. ITC Dissertation, Series 69, 2000
- [Zlatanova 2004] ZLATANOVA, S.; RAHMAN, A. A.; SHI, W.: Topological Models and Frameworks for 3D Spatial Objects. In: *Computer & Geoscience*, Volume 40, issue 4, S. 419-428.

Abbildungsverzeichnis

1.1	Typische Fehler bei der Bestandsdokumentation	8
1.2	Inhalt der Arbeit	9
2.1	Entity Relationship Diagramm	10
2.2	UML (Unified Modeling Language) Beispiel Diagramm mit Erläuterung	11
2.3	XML-Schema Schichtenmodell. Nach [Kazakos u.a. 2002]	12
2.4	Objektgeometrie als Ordnungsrahmen [Weferling 2002]	13
2.5	Beobachtungen als Primärdaten - Die Geodätische Sicht	16
2.6	Drei-Schichten Modellierung nach [Mäntylä 1976, S. 31]	17
2.7	Constructive Solid Geometry (CSG) Modell	18
2.8	Randrepräsentation nach [Bradley und Paul 2009])	18
2.9	Homöomorphie und 2-Mannigfaltigkeit nach [Foley u.a. 1995, S. 566]	20
2.10	Topologie ist invariant gegenüber stetigen Abbildungen	21
2.11	Topologiefreies 3D Modell	21
2.12	vier Ebenen der Topologischen Redundanz	22
2.13	Vermeidung einer Mehrfachspeicherung von Punktkoordinaten	22
2.14	Vermeidung von Kantenredundanz	22
2.15	Das Konzept der Halbkante dient der Beseitigung der Kantenredundanz	22
2.16	Eine Halbkante ist ein Verweis auf eine Kante und ein Vorzeichen zur Orientierung	23
2.17	Das Konzept des Loop	23
2.18	Das topologisch normalisierte Modell mit Loops	23
2.19	Topologisch Normalisiertes 3D Modell	24
2.20	vier Ebenen der Geometrischen Redundanz	25
2.21	a) Punkt-Repräsentation b) Flächen-Repräsentation c) Punkt als Ebenenschnitt	25
2.22	Erste geometrische Normalisierung: Ebenen-parametrisiertes 3d-Modell	26
2.23	Zweite Geometrische Normalisierung: Ebene wird referenziert	26
2.24	Dritte geometrische Normalisierung: Normalenvektor wird referenziert	27
2.25	Parallelität und Orthogonalität	27
2.26	Vierte Normalisierung: Der Parameter wird Referenziert	27
2.27	Normalenvektor #1	29
2.28	Minimierung der Parameterzahl um 91% am Beispiel eines Wohnhauses	29
2.29	Abstandsbeobachtungen zwischen topologischen Primitiven a)Face-Face b)Face-Edge c)Face-Node d)Edge-Edge e)Edge-Node f)Node-Node	31
2.30	a)PointOnFace b)PointOnEdge c)PointOnNode	32
2.31	A priori Relativdistanzen	33
3.1	Geometrische Interpretation eines Quaternions	35
3.2	a) Abstand Masche-Masche b) Abstand Masche-Kante	37
3.3	a)Abstand Masche-Knoten b) Abstand Kante-Kante	38
3.4	a)Abstand Kante-Knoten b)Abstand Knoten-Knoten	38
3.5	Inkrementelle Suche nach $\Sigma v^2 = \min$ (Inkrement = 1 gon)	40
3.6	Startwert a) und Ergebnis b) der inkrementellen Suche	41
3.7	a) Nährungswerte b) Datum mit Identitätsbeobachtungen c) Datum mit Teilspurminimierung d) Datum mit Festhalten	43

4.1	Berechnungsgraph Funktion	49
4.2	Berechnungsgraph Forward Methode	50
4.3	Berechnungsgraph Backward-Methode	51
5.1	Parameternetz	57
5.2	Parameternetz am Beispiel des Gleichungsobjekts „Abstand Kante-Kante“	57
6.1	Schichtenmodell für die Konsistenzprüfung	59
6.2	Auswirkung einer falschen Kodierung von Fließkommazahlen	60
6.3	Verletzung der topologischen Integrität beim Anfertigen der 3d Skizze	64
6.4	3D-Körper mit Löchern a)2-Mannigfaltigkeit b)Zellkomplex c)Popa3d	64
6.5	Ein Knoten wird indirekt von 3 linear abhängigen Ebenen referenziert	65
6.6	Die Integritätsbedingung GA02	66
7.1	Anfertigen der Skizze mit Hilfe von Konstruktionslinien	68
7.2	Selektieren, Kopieren und Verschieben von Blöcken	68
7.3	Visualisieren von Zeichnungsfehlern (a)	68
7.4	Visualisieren von Zeichnungsfehlern (b)	69
7.5	Visualisieren von Zeichnungsfehlern (c)	69
7.6	Visualisieren von Zeichnungsfehlern (d)	70
7.7	Modell vor und nach der Vereinigung von gleichen Normalenvektoren	70
7.8	Modell vor und nach der Vereinigung von gleichen Ebenen	71
7.9	Dreidimensionale Benutzereingabe bei der Messung mit einem Entfernungsmessgerät	72
7.10	Visualisierung der Objektgeometrie vor und nach der Ausgleichsrechnung	72
7.11	Analyse der Objektgeometrie während der Vermessung	73
7.12	Veränderung der Objektgeometrie nach einer ersten Ausgleichung	73
7.13	Visualisierung noch nicht gemessener Elemente	73
7.14	Informationsfluss Anwendungsfall „Sekundärdatenerfassung“	75
7.15	Das 3d-Modell (CityGML) vor und nach der Ausgleichung	76
7.16	Bedingungen	76
7.17	Das 3d-Modell (Popa3d) vor und nach der Ausgleichung	77
A.1	Abstandbeobachtungen zwischen topologischen Primitiven	87
A.2	UML: Polare Beobachtungen auf topologischen Primitiven	88
A.3	Winkelbedingungen zwischen topologischen Primitiven	89
A.4	A priori Differenzen zwischen Elementen der Objekt-Geometrie	90

Tabellenverzeichnis

4.1	Berernungssequenz der Funktion	49
4.2	Berernungssequenz der Forward-Methode	50
4.3	Berernungssequenz der Backward-Methode	51
6.1	Axiome der referentiellen Integrität auf Modellebene	62
6.2	Axiome der Topologischen Integrität	63
6.3	Vergleich der unterschiedlichen Euler-Poincaré Formeln	63
6.4	Axiome der geometrischen Integrität	65
6.5	Axiome der Integrität für die Ausgleichungsrechnung	65
7.1	Parameterreduktion durch Normalisierung	71
7.2	Beispiel: Ausgleichung mit 36 Distanzmessungen und Skizzengeometrie	71
7.3	Analyse der Ausgleichung mit 36 Distanzmessungen und Skizzengeometrie	74
7.4	Anzahl der Primitive im Beispieldatensatz	76
7.5	Anzahl der Primitive vor und nach der Fehlerbereinigung	77
B.1	Gleichungsobjekte in Popa3d	91

Anhang

A UML-Klassendiagramme

A.1 Abstände zwischen topologischen Primitiven

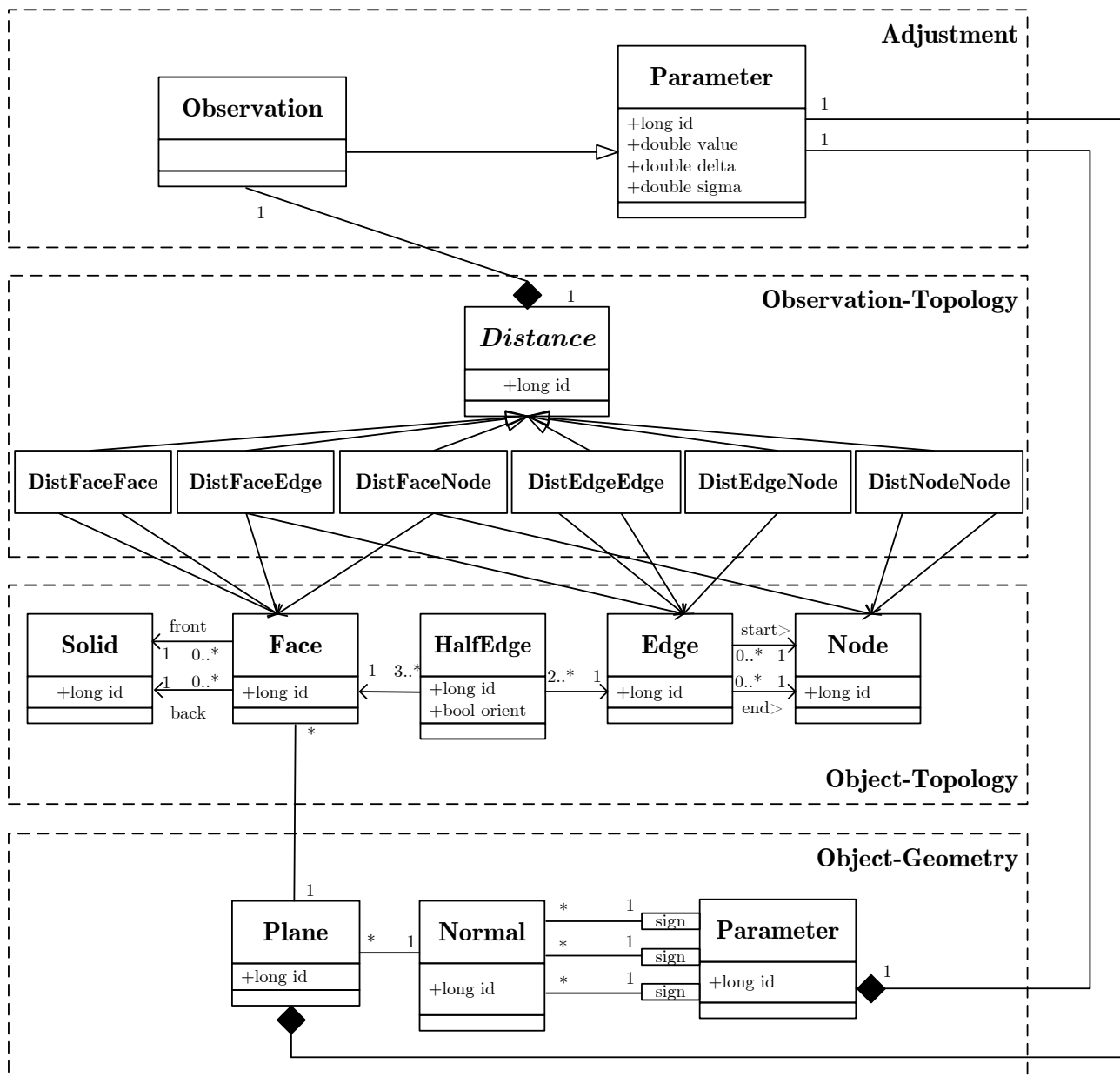


Abbildung A.1: Abstandbeobachtungen zwischen topologischen Primitiven

Beschreibung des konzeptionellen Modells: Kapitel 2.13.2 Seite 30

Beschreibung des funktionalen Modells: Kapitel 3.2 Seite 36

A.2 Lokale Polarkoordinaten auf topologischen Primitiven

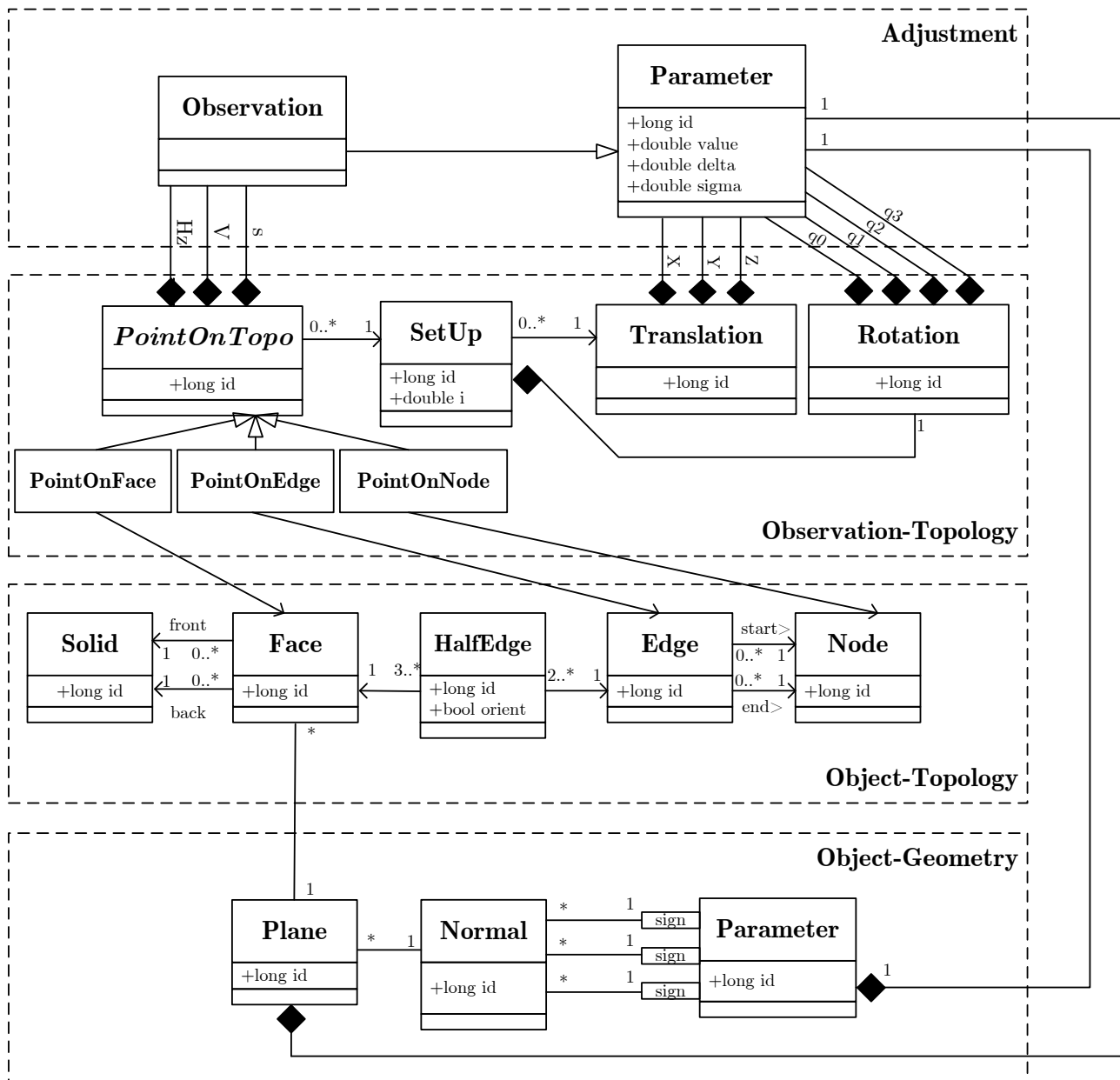


Abbildung A.2: UML: Polare Beobachtungen auf topologischen Primitiven

Beschreibung des konzeptionellen Modells: Kapitel 2.13.3 Seite 31

Beschreibung des funktionalen Modells: Kapitel 3.3 Seite 39

A.3 Winkelbedingungen zwischen topologischen Primitiven

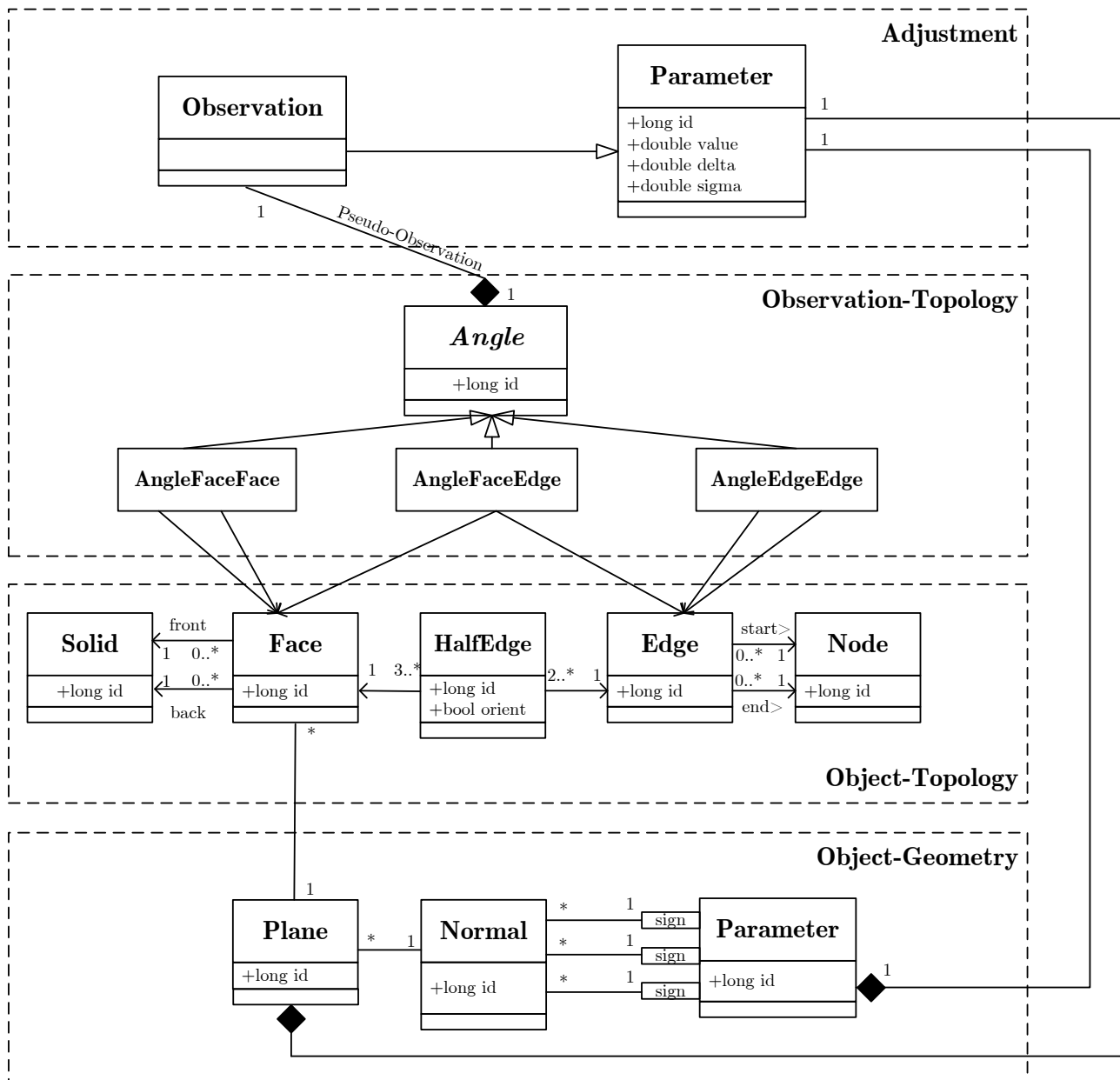


Abbildung A.3: Winkelbedingungen zwischen topologischen Primitiven

Beschreibung des konzeptionellen Modells: Kapitel 2.13.4 Seite 32

Beschreibung des funktionalen Modells: Kapitel 3.4 Seite 41

A.4 A priori Differenzen zwischen Abständen und Normalenvektoren

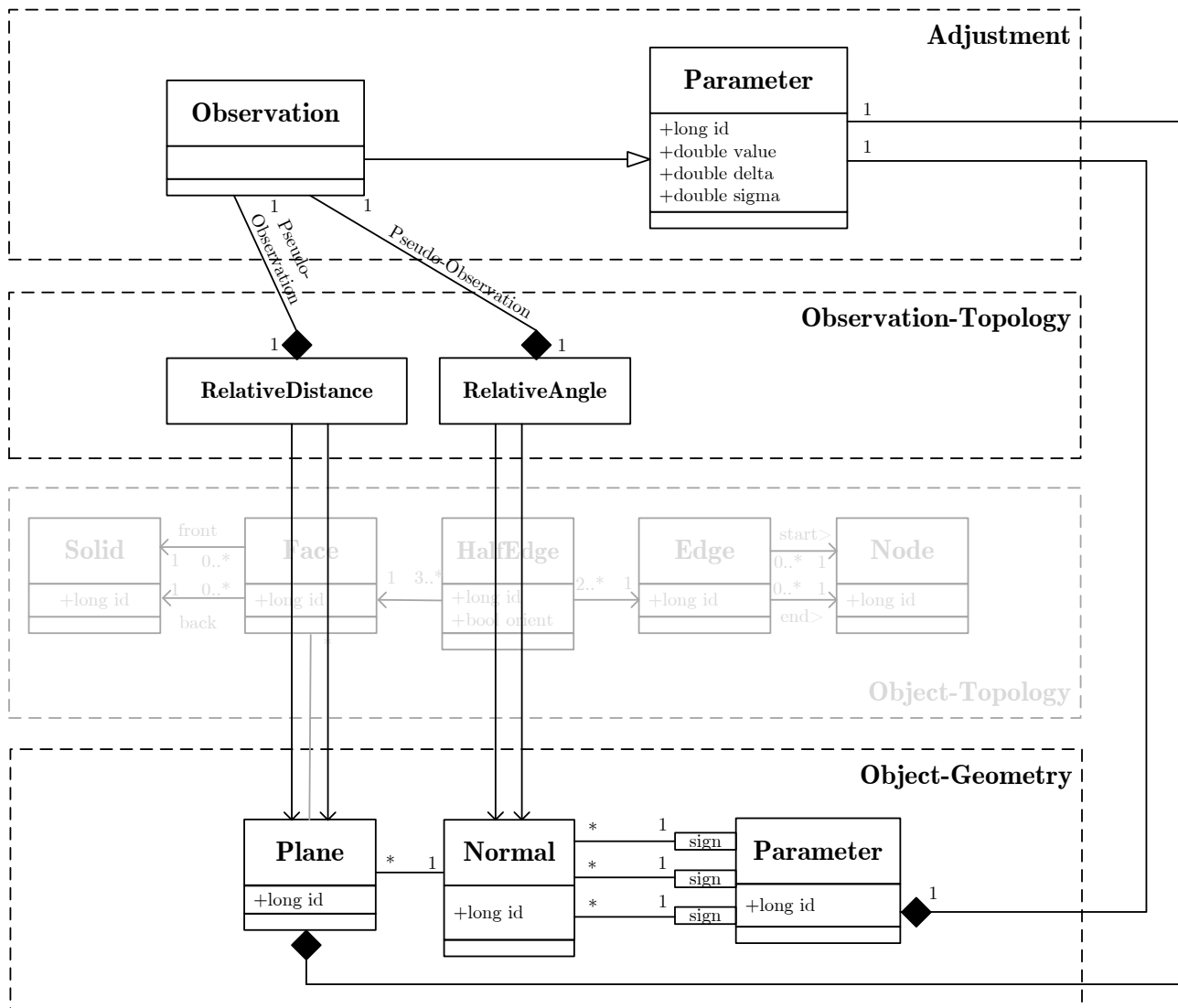


Abbildung A.4: A priori Differenzen zwischen Elementen der Objekt-Geometrie

Beschreibung des konzeptionellen Modells: Kapitel 2.13.5 Seite 32

Beschreibung des funktionalen Modells: Kapitel 3.5 Seite 42

B Liste aller Gleichungsobjekte

Tabelle B.1: Gleichungsobjekte in *Popa3d*

Name	Popa3d Objekt	d	nv	Trans.	Rot.	f ^a	e ^b	v ^c	n' ^d	u' ^e	linear	Konzept	Gleichung	UML
Abstand Masche-Masche	DistanceFaceFace	2	0	0	0	2	0	0	1	2	ja	S. 30	(3.15)	A.1
Abstand Masche-Kante	DistanceFaceEdge	3	9	0	0	1	1	0	1	12	nein	S. 30	(3.17)	A.1
Abstand Masche-Knoten	DistanceFaceNode	4	12	0	0	1	0	1	1	16	nein	S. 30	(3.20)	A.1
Abstand Kante-Kante	DistanceEdgeEdge	4	12	0	0	0	2	0	1	16	nein	S. 30	(3.22)	A.1
Abstand Kante-Knoten	DistanceEdgeNode	5	15	0	0	0	1	1	1	20	nein	S. 30	(3.26)	A.1
Anstand Knoten-Knoten	DistanceNodeNode	6	18	0	0	0	0	2	1	24	nein	S. 30	(3.29)	A.1
Punkt auf Masche	PointOnFace	1	3	3	4	1	0	0	3	11	nein	S. 31	(3.42)	A.2
Punkt auf Kante	PointOnEdge	2	6	3	4	0	1	0	3	15	nein	S. 31	(3.44)	A.2
Punkt auf Knoten	PointOnNode	3	12	3	4	0	0	1	3	22	nein	S. 31	(3.46)	A.2
Winkel Masche-Masche	ConstraintFaceFace	0	6	0	0	2	0	0	1	6	nein	S. 32	(3.48)	A.3
Winkel Masche-Kante	ConstraintFaceEdge	3	9	0	0	1	1	0	1	12	nein	S. 32	(3.49)	A.3
Winkel Kante-Kante	ConstraintEdgeEdge	4	12	0	0	0	2	0	1	16	nein	S. 32	(3.50)	A.3
Ebenendifferenz	RelativeDistancePlanePlane	2	0	0	0	0	0	0	1	2	ja	S. 32	(3.52)	A.4
Relativwinkel	RelativeAngleNormalNormal	0	6	0	0	0	0	0	1	6	nein	S. 32	(3.54)	A.4
Betrag des NV ist 1	NormNV	0	3	0	0	0	0	0	1	3	nein		(3.5)	
Betrag Quaternion ist 1	NormQ	0	0	0	4	0	0	0	1	4	nein	S. 31	(3.14)	
Festhalten	ParameterFix	0/1	0/1	0/1	0/1	0	0	0	0	1	ja		(3.64)	
Identität	ParameterObs	0/1	0/1	0/1	0/1	0	0	0	1	1	ja		(3.63)	
Eigenvektor	(kein Objekt)	alle	alle	0	0	0	0	0	0	alle	nein		(3.62)	

^aAnzahl der Maschen^bAnzahl der Kanten^cAnzahl der Knoten^dAnzahl der Beobachtungen in der Gleichung^eAnzahl der Unbekannten in der Gleichung

Lebenslauf

Dipl.-Ing. Christian Clemen geboren am 11. Juni 1976

Familienstand: verheiratet, zwei Kinder

Ausbildung

09/2009	Promotion zum <i>Doktor - Ingenieur</i> (Dr.-Ing), Dissertation „Ein geometrisch-topologisches Informationsmodell für die Erfassung und Validierung von flächenparametrisierten 3d-Gebäudemodellen“, Gesamturteil: sehr gut mit Auszeichnung
10/1998 – 04/2004	Studium Vermessungswesen Technische Universität Berlin, Diplomarbeit „Datenmodellierung und Ausgleichung hybrider Beobachtungsdaten für die Gebäudeaufnahme“, Abschluss <i>Diplom - Ingenieur</i> , sehr gut
04/1998 – 10/1998	Studium Musikwissenschaften und Philosophie, Technische Universität Berlin
08/1995 – 02/1998	Berufliche Ausbildung zum <i>Vermessungstechniker</i> , bei Dipl.-Ing. A. Nebeling, Öffentlich bestellter Vermessungsingenieur, Mainz
1986 – 1995	Staatliches Gymnasium Mainz-Gonsenheim, Abschluss <i>Abitur</i> , gut

Beruflicher Werdegang

seit 04/2004	Wissenschaftlicher Mitarbeiter am Institut für Geodäsie und Geoinformationstechnik, Fachgebiet für Geodäsie und Ausgleichungsrechnung, Technische Universität Berlin
04/2002 – 04/2004	Studentischer Mitarbeiter für Lehraufgaben und Softwareentwicklung am Institut für Geodäsie und Geoinformationstechnik, Fachgebiet für Geodäsie und Ausgleichungsrechnung, Technische Universität Berlin
2001 – 2002	Vermessungstechniker bei Arc-Greenlab GmbH, Berlin
1998 – 2000	Vermessungstechniker bei Dipl.-Ing. A. Nebeling, Öffentlich bestellter Vermessungsingenieur, Mainz

Berlin, 9. März 2010