

Lehrstuhl für Bauinformatik
Fakultät Bauingenieur- und Vermessungswesen
Technische Universität München

Wissensbasierte Unterstützung zur Erstellung von Produktmodellen im Baubestand

Matthias Schleinkofer

Vollständiger Abdruck der von der Fakultät für Bauingenieur- und Vermessungswesen der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr.-Ing. habil. Thomas Wunderlich

Prüfer der Dissertation:

1. Univ.-Prof. Dr. rer. nat. Ernst Rank
2. Univ.-Prof. Dr.-Ing. habil. Dieter K. Fritsch,
Universität Stuttgart

Die Dissertation wurde am 11.07.2007 bei der Technischen Universität München eingereicht und durch die Fakultät für Bauingenieur- und Vermessungswesen am 28.11.2007 angenommen.

Vorwort

Diese Arbeit stellt die Ergebnisse meiner wissenschaftlichen Tätigkeit am Lehrstuhl für Bauinformatik der Technischen Universität München aus dem Zeitraum September 2003 bis Juli 2007 vor. Mein Dank gilt dabei dem Bayerischen Staatsministerium für Umwelt, Gesundheit und Verbraucherschutz für die Förderung des Forschungsprojekts BayFORREST F248 „Vom Laseraufmaß zum Stoffflussmanagement für Altbauten“, wodurch es möglich wurde, dieses interessante, interdisziplinäre Thema zu bearbeiten. Die erfolgreiche Beendigung stellte die Weiterfinanzierung durch den Lehrstuhl für Bauinformatik sicher.

Meinem Doktorvater, Prof. Ernst Rank, danke ich für die Abstimmung des Themas auf meine persönlichen Interessen und ferner für das mir entgegengebrachte Vertrauen in eine Arbeit, mit welcher ich das Forschungsspektrum des Lehrstuhls verbreitern durfte. Prof. Dieter Fritsch von der Universität Stuttgart möchte ich Dank sagen für die Übernahme des Zweitgutachtens und dem damit verbundenen Interesse. Mit der Übernahme des Prüfungsvorsitzes durch Prof. Thomas Wunderlich wurde zudem die gemeinsame, harmonische Arbeit im Forschungsverbund BayFORREST abgerundet.

Für die konstruktive Zusammenarbeit mit meinen Kollegen vom Lehrstuhl für Bauinformatik bedanke ich mich besonders. An erster Stelle steht hier mein Zimmerkollege Martin Egger, von welchem ich zu jeder Zeit fachliche und technische Unterstützung erfahren konnte. Ziad Wassouf hat außer mit fachlichen Diskussionen durch seine konstruktiven Hinweise zur Verbesserung des schriftlichen Teils Wesentliches beigetragen. Die Kompetenzen von Richard Romberg und Dominik Scholz in objektorientierter Programmierung waren der Entwicklungsphase des technischen Teils der Arbeit sehr förderlich. Positiv in Erinnerung habe ich ferner die fachlichen Auseinandersetzungen mit David Franke sowie Martin Schlaffers leidenschaftliche und begeisternde Klärung meiner mathematischen Fragen. Durch ihr großes Engagement bei der Erstellung ihrer Diplomarbeit konnte Almut Bastian das Forschungsprojekt und somit die vorliegende Arbeit voranbringen. Thomas Schäfer vom Lehrstuhl für Geodäsie danke ich für die gute Zusammenarbeit, für seine fachliche Unterstützung im Gebiet Laservermessung sowie die Aufnahme und Zurverfügungstellung von Laserpunktwolken. Bei allen Kollegen des Lehrstuhls für Bauinformatik bedanke ich mich für die gute Atmosphäre, namentlich – weil wesentlich – bei unserer Lehrstuhlsekretärin Hanne Cornils.

Mein herzlichstes Dankeschön geht an meine Eltern Carola und Hans für ihre unermüdliche Ermutigung und Unterstützung, sowie an meine Lebensgefährtin Kathrin für das Verständnis und den Rückhalt, den ich bei ihr fand.

München, im Dezember 2007

Matthias Schleinkofer

Kurzfassung

Modellbasierte Planung gestattet die Durchführung einer Vielzahl von Planungsaufgaben in einer ganzheitlichen Sicht auf das Bauobjekt. Während diese in zunehmendem Maß bei Neubauten eingesetzt wird, findet sie im Bereich der Umbau- und Sanierungsplanung eher geringen Zuspruch. Als wesentlicher Grund für den beschränkten Einsatz im Altbausektor ist das Fehlen eines räumlichen Bauwerksmodells anzusehen, welches bislang nur unter großem manuellem Einsatz erstellt wird.

Die vorliegende Arbeit stellt einen Weg vor, um unter der Prämisse eines hohen Automatisierungsgrades den Baubestand geometrisch zu erfassen und als planungstaugliches Produktmodell aufzubereiten. Der Einsatz terrestrischer Laserscanner erlaubt es, einen wirtschaftlich effizienten Messablauf einzuführen. Berührungslos und in kurzer Zeit werden die Objektoberflächen automatisch und in hoher Genauigkeit abgetastet. Als resultierende Messdaten stehen Punktwolken mit einer sehr großen Anzahl an Einzelpunkten zur Verfügung. Die weitere Bearbeitung kann in örtlicher Distanz zum Objekt erfolgen.

Die Punktwolken werden zu einer Gesamtpunktwolke verschmolzen und ausgedünnt, um Messungenauigkeiten auszugleichen. Im nachfolgenden Schritt findet eine Dreiecksvernetzung der Punkte statt. Benachbarte Dreiecke, die eine analytische Fläche approximieren, bilden Segmente. Aus der Verschneidung aneinandergrenzender Segmente entsteht ein Oberflächenmodell, aus welchem sich Volumenkörper ableiten lassen.

Zu deren weiteren Analyse werden künstliche neuronale Netze verwendet. Mit ihnen ist es möglich, die zunächst unbestimmten Körper als Bauteile zu klassifizieren, ohne konkrete Entscheidungsregeln dafür festlegen zu müssen. Diese leiten sich aus zuvor erlernten Beispielen ab. Gemäß der Klassifizierung werden schließlich, entsprechend der aufgemessenen Geometrie, in einem Produktmodell Objekte erzeugt.

Abstract

Model-based planning enables a plurality of planning tasks to be carried out based on a holistic view of the building project. Although this approach is increasingly employed in new buildings, it does not enjoy such widespread popularity in the field of conversion and refurbishment so far. The lack of a spatial building model, which still involves a great deal of manual work even today, is regarded as one of the fundamental reasons for its restricted employment in the old building sector.

This thesis introduces an approach, under the premise of a high degree of automation, of geometrically capturing the building stock and of processing it as a product-model suitable for planning. The use of terrestrial laser scanners enables an economically efficient measuring procedure to be introduced. The surfaces of the object are automatically scanned without physical contact, within a short space of time and with a high level of precision. The resultant measuring data appear in the form of point clouds with a very large number of individual points. The further processing can be conducted at a local distance from the object.

In the first processing step, the individual point clouds are merged to a joint point cloud. Thinning out the database enables errors of measurement to be reduced. In the following step the three dimensional points are meshed to triangles. Triangles located next to each other and approximating the same analytic surface form segments. Intersecting adjacent segments serves to create a surface model, from which solids can be derived.

Artificial neural networks are used to analyse and classify initially undefined solids as building elements, without the need to stipulate strict decision rules. Working as self-learning systems, artificial neural networks can derive their classification rules from known training patterns. Following neural network-based classification, the solids are finally reconstructed as objects within the product model according to the previously measured geometry of the actual building.

Inhalt

1	Einleitung	7
2	Modellierung im Bauwesen am Computer	12
2.1	Übergang zur 3D-Planung	12
2.2	3D-CAD-Modelle.....	16
2.2.1	Punktwolke	16
2.2.2	Drahtmodell	17
2.2.3	Oberflächenmodell	18
2.2.4	Volumenmodell.....	18
2.2.5	Produktmodell	19
2.3	Strukturierung dreidimensionaler Daten.....	19
2.3.1	Voxel	20
2.3.2	Oktalbaum	21
2.3.3	Ballungen	23
2.3.4	Delaunay-Vernetzung.....	24
2.3.5	Ausgleichsebenen	25
3	Bauaufnahme im Altbau.....	28
3.1	Zweck der Bauaufnahme	28
3.2	Aufmaßverfahren	28
3.2.1	Handvermessung	28
3.2.2	Laserdistanzmessgerät	29
3.2.3	Reflektorlos distanzmessender Tachymeter.....	30
3.2.4	Terrestrische Laserscanner.....	31
3.3	Bestandsaufnahme mittels Laserscanner	32

4	Erstellung eines Oberflächenmodells aus Aufmaßdaten	35
4.1	Stand der Wissenschaft und Technik	35
4.1.1	Approximation durch ausgleichende geometrische Primitive	36
4.1.2	Bereichswachstumsverfahren	36
4.1.3	Vernetzung von Punktwolken aus Einzelaufnahmen	37
4.1.4	Vernetzung von Gesamtpunktwolken	38
4.1.5	Segmentierung in analytische Teilflächen	41
4.1.6	Schnittkanten und -punkte zwischen Segmenten	42
4.2	Überblick über das Vorgehen	43
4.3	Aufbereitung der Messdaten (Ausdünnung)	44
4.4	Dreiecksvernetzung der Messpunkte	46
4.5	Segmentierung	54
4.6	Schnittpunkt-, -kanten- und Regionsbestimmung	59
4.7	Beispielanwendung	60
4.7.1	Oberflächenrekonstruktion als prototypische Implementierung	60
4.7.2	Parameterbestimmung	61
4.7.3	Beispiele	65
5	Virtuelle neuronale Netze	74
5.1	Aufbau	76
5.2	Netztopologie	80
5.3	Netzeingabe	83
5.4	Netzausgabe	85
5.5	Auswahl von Trainingsdaten	88
5.6	Netzprägung	92
5.6.1	Back-Propagation-Algorithmus	92
5.6.2	Initialisierung	94
5.6.3	Festlegung des Lernparameters	94
5.6.4	Einbringung der Lerninformation	96
5.6.5	Abbruch des Lernvorgangs	96

6	Bauteilerkennung unter Zuhilfenahme virtueller neuronaler Netze	100
6.1	Vorgehensweise	100
6.2	Festlegung und Aufbereitung der Eingangsparameter.....	101
6.3	Auswertung der Netzentscheidung	105
6.4	Festlegung der Netztopologie	106
6.5	Aufstellen der Trainingsbasis	107
6.6	Konditionierung des Netzes	109
6.7	Beispielanwendung.....	109
6.7.1	Bauteilerkennung mit einem kommerziellen CAD-System	109
6.7.2	Erhebung der Eingangsparameter	110
6.7.3	Training	111
6.7.4	Klassifizierung	111
6.7.5	Konvertierung in Bauteile	111
6.7.6	Beispiele.....	115
7	Zusammenfassung und Ausblick.....	120

1 Einleitung

Die Planung eines Bauwerkes erfordert die Zusammenarbeit unterschiedlicher Fachplaner. Dabei erfolgt die Ausführung domänenspezifischer Arbeiten räumlich getrennt und zeitlich versetzt. Der notwendige Austausch planungsrelevanter Daten zwischen den Beteiligten findet immer häufiger auf digitalem Wege statt, für welchen die 2D-Zeichnung den Standard darstellt. Dies ist unter anderem darin begründet, dass sich die Planungsleistungen an der Honorarordnung für Architekten und Ingenieure (HOAI) orientieren [1], die eine Vergütung des Mehrwertes einer modellbasierten Planung nicht vorsieht. Der in der Baupraxis übliche Wechsel von Planern führt dazu, dass bislang keine über einzelne Entwurfsschritte hinwegreichende Dienste erbracht werden: Die zusätzlichen Leistungen würden erst im Laufe des Baufortschrittes an späterer Stelle eine Effizienzsteigerung und Fehlervermeidung darstellen oder nach Fertigstellung eine geeignete Weiterverwendung finden.

Die zeichnungs-basierte, zweidimensionale Vorgehensweise besitzt jedoch gravierende Nachteile: Da keine Relation zwischen den verschiedenen Ansichten und Schnitten besteht, können Inkonsistenzen zu Planungsfehlern führen, die oftmals sehr spät entdeckt werden und unter großem Aufwand zu lösen sind. Nachträgliche Änderungen sind dabei schwer einzuarbeiten und erzwingen mitunter eine Neuerstellung der Planung, welche weitreichende Folgen bis hin zur Neukonzeption von Bauabläufen haben kann. Zeitliche Verzögerungen und eine Erhöhung der Kosten gehen damit einher. Aufgrund der zweidimensionalen Projektion ist es nicht möglich, alle geometrischen Aspekte des Bauwerkes zu erkennen. Die Interpretation dieser Pläne bleibt zudem vornehmlich nur wenigen fachkundigen Personen vorbehalten.

Grundsätzlich bietet eine IT-gestützte Planung die Möglichkeit, die digitalen Daten fortzuführen. Die durchgängige Verwendung eines digitalen Planes scheitert jedoch häufig daran, dass die einzelnen Fachplaner unterschiedliche Softwarelösungen mit proprietären Datenformaten einsetzen. Die 1995 gegründete *Industrieallianz für Interoperabilität* (IAI) hat es sich deshalb zum Ziel gesetzt, ein Datenaustauschformat einzuführen, das den verlustfreien Austausch zwischen unterschiedlichen Softwareanwendungen im Bauwesen erlaubt. Dazu sind von ihr die *Industry Foundation Classes* (IFC) entwickelt worden, welche „alle am Bauwerk existierenden Bauteile als

Objekte definiert“ [37], was eine Loslösung vom linienbasierten Datenaustausch erlaubt.

Bauwerksinformationsmodell

Die Entwicklung der IFC zielt darauf ab, eine durch alle Planungsphasen durchgängige, rechnergestützte dreidimensionale Planung zu realisieren. Sie erlaubt durch die Erstellung eines digitalen Bauwerksinformationsmodells, auch als Produktmodell bezeichnet, eine ganzheitliche Sicht auf das Bauwerk, wobei durch jeden Fachplaner das Modell um weitere Details angereichert wird. Neben der sequenziellen Bearbeitung eines solchen Modells sind auch Szenarien für die parallele Entwicklung eines Bauwerks möglich. Das dafür verwendete, zentrale Produktmodell ermöglicht grundsätzlich auch Prüfmechanismen, die Konflikte zwischen den Arbeiten der Beteiligten verhindern: Verändert ein Planer beispielsweise die Geometrie, so können Inkonsistenzen entstehen, wenn gleichzeitig von anderen Personen auf Grundlage des alten Datenstandes geplant wird. Deshalb werden entweder alle Beteiligten über die Änderung informiert, deren Datenbasis aktualisiert oder eine Konsistenzprüfung beim Einpflegen der Modifikationen ausgeführt.

Auf Basis eines dreidimensionalen Modells lassen sich die nach wie vor relevanten, zweidimensionalen Ausführungspläne erstellen, indem vom Softwaresystem Schnittebenen in das virtuelle Gebäude gelegt werden. Da die Pläne auf Grundlage einer einheitlichen Datenquelle generiert werden, schließen sich Widersprüche zwischen den erstellten Planzeichnungen aus.

Mit über die reine Geometrie hinausgehenden, semantischen Informationen angereichert ist ein solches Modell die Basis vielfältiger weiterer Untersuchungen und Anwendungen, welche den gesamten Produktlebenszyklus [2] betrachten: Beispielsweise lassen sich die Heizenergieverluste zum Nachweis nach Energieeinsparverordnung [28] oder der Primärenergiebedarf zur Erstellung des Energiepasses [18] automatisch berechnen [32]. Durch Ergebnisse fachkundiger Untersuchungen des Materialzustandes angereichert steht das Bauwerksmodell für den Planungsprozess einer Gebäudesanierung zur Verfügung [48]. In ein entsprechendes Programm integriert lässt sich das Modell ferner für Facility Management Anwendungen bis hin zur Reststoffanalyse beim Rückbau verwenden.

Planung im Bestand

Während bei Neubaumaßnahmen die erste Entwurfshandlung bereits mit einem dreidimensionalen CAD-Programm (engl.: *computer aided design*) erfolgen kann, so ist die Planung im Bestand auf die Erfassung der vorhandenen Geometrie angewiesen. Vorhandene Papieraufzeichnungen bieten eine mögliche Ausgangsbasis für eine digitale Weiterverarbeitung. Stand der Technik ist die Vektorisierung der gescannten Papierpläne. Das Resultat ist jedoch eine ernüchternde Ansammlung von Linien, die nur von einem mit dem Kontext vertrauten Betrachter, keinesfalls aber von einem Computer interpretierbar sind. Es ist deshalb sinnvoll, die Linien algorithmisch weiter auszuwerten, um so ein für die weitere Planung nutzbares 2D- oder idealerweise sogar 3D-Modell zu erzeugen und nicht die gescannte Papiervorlage lediglich als Hintergrundbild zur Orientierungshilfe am Computer einzublenden [6].

Existieren keine Aufzeichnungen, misstraut man diesen oder möchte man diese verifizieren, so muss die Objektgeometrie vor Ort erhoben werden. Die Forschung befasst sich schon seit einiger Zeit mit dem Thema, möglichst effizient den Bestand zu erfassen und als Computermodell aufbereitet zur Verfügung zu stellen [24][87]. Die technischen Möglichkeiten in der Ingenieurvermessung erlauben heute den Einsatz moderner, berührungslos messender Verfahren, insbesondere die Anwendung terrestrischer Laserscanner findet immer größere Beachtung [16]. Die Instrumente vermessen in relativ kurzer Zeit automatisch ihre Umgebung in hoher Auflösung. Ein angeschlossener Computer steuert den Laserscanner, empfängt von diesem die Messdaten und erstellt daraus sogenannte Punktwolken. Die Zusammenführung von Aufnahmen aus verschiedenen Standorten wird von entsprechender Software zuverlässig erledigt. Allerdings endet an dieser Stelle die automatische Datenverarbeitung. Die Aufbereitung als planungstaugliches Bauteilmodell muss unter großem Einsatz menschlicher Mitwirkung erfolgen [65], weshalb unter wirtschaftlichen Gesichtspunkten „das Verfahren des Laser-Scannings für die planungsbezogene Bestandserfassung heute noch nicht eingesetzt werden kann“ ([94], S. 3). Für den Bereich eines eng begrenzten Spektrums an Messobjekten, beispielsweise im Rohrleitungsbau, ist es dennoch bereits möglich, die vermessenen Gegenstände zu rekonstruieren. Grund dafür ist die geringere Anzahl an Freiheitsgraden, welche zur Modellerstellung bestimmt werden müssen [17].

Daraus begründet sich die Vision, automatisch und ohne Einschränkungen ein volumenbasiertes Geometriemodell der bestehenden Bausubstanz zu erzeugen. Mit dieser Arbeit werden wesentliche Schritte auf einem Weg vorgestellt, aus Laseraufmaßdaten ein objektorientiertes Produktmodell zu erzeugen (Abb. 1).

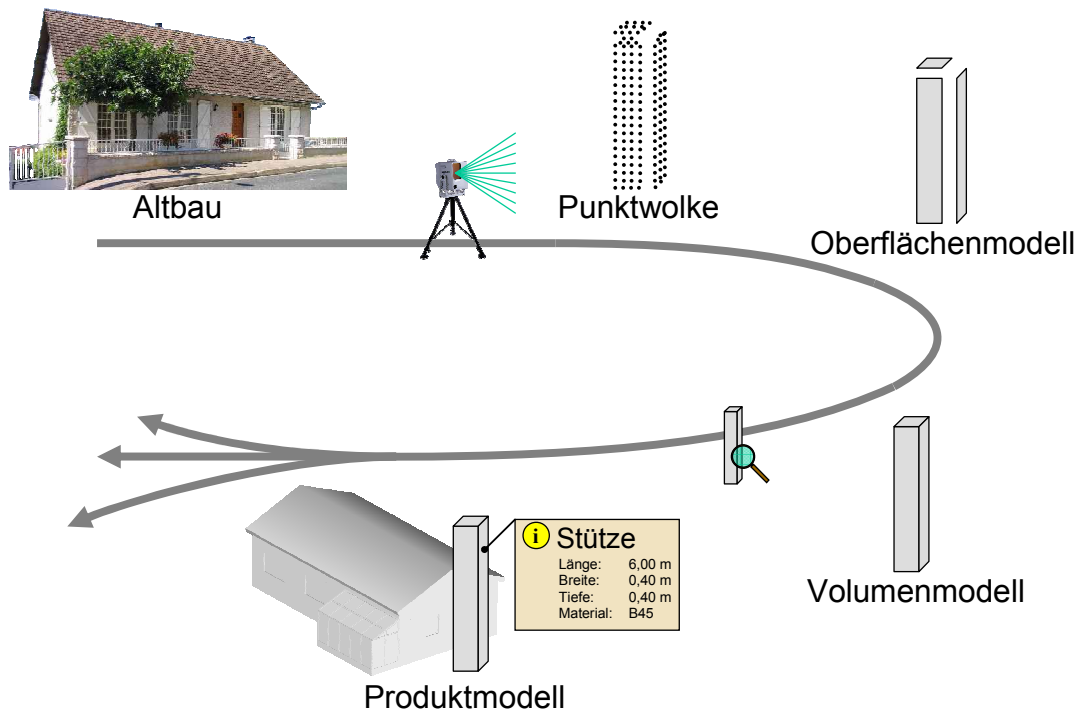


Abb. 1: Der bei der Punktwolke beginnende Arbeitsablauf errechnet ein Oberflächenmodell; stellt dieses in ein konsistentes Volumenmodell dar, lässt es sich in ein Produktmodell, welches als Grundlage für den weiteren Planungsprozess dienen kann, konvertieren

Gliederung der Arbeit

Diese Arbeit motiviert zunächst in Kapitel 2 die IT-basierte dreidimensionale Modellierung. Dazu werden verschiedene geometrische Repräsentierungsformen diskutiert und Möglichkeiten zur Strukturierung dreidimensionaler Daten gegeben.

Die Auseinandersetzung mit der Bauaufnahme im Altbau unter Einsatz lasergestützter Technologien ist das Thema von Kapitel 3. Der Schwerpunkt liegt dabei auf 3D-Laserscannern und ihrem praktischen Einsatz.

Einen durchgängigen Weg, der aus den Messergebnissen eines 3D-Laseraufmaßes über mehrere Zwischenschritte ein Oberflächenmodell erzeugt, stellt Kapitel 4 vor. An praktischen Beispielen kann die Tauglichkeit des Verfahrens gezeigt werden.

Beschreibt das Oberflächenmodell einzelne Körper, so lässt sich dieses als Volumenmodell interpretieren. Diese Volumenkörper sind im Weiteren als Bauteile eines Produktmodells zu klassifizieren. Da nicht jede Entscheidungsregel uneingeschränkt in geschlossener Form ausgedrückt werden kann [103], stellt Kapitel 5 virtuelle neuronale Netze als eine Methode der schwachen künstlichen Intelligenz zur Mustererkennung vor.

Mit deren Hilfe ist es möglich, CAD-Körper in Bauteilklassen einzuordnen. Kapitel 6 diskutiert, welche Parameter der Volumenkörper zur Klassifizierung geeignet sind und wie ein virtuelles neuronales Netz dafür zu konfigurieren ist. Das methodische Vorgehen zur Konvertierung eines Volumenmodells in ein Produktmodell wird exemplarisch für Wände gezeigt. Mehrere Beispiele, welche das Verfahren anwenden, schließen das Kapitel ab.

2 Modellierung im Bauwesen am Computer

Insbesondere bei großen Vorhaben, welche sich durch die Länge ihrer Zeitdauer oder die Größe der aufzubringenden Anstrengungen auszeichnen, bedient sich der Mensch der Planung. Durch diese imaginäre Vorwegnahme erwarteter und auch unerwarteter, erwünschter und auch unerwünschter Ereignisse mit ihren jeweiligen Folgen lassen sich die Erfolgchancen einer Handlung um ein Vielfaches erhöhen.

Arbeiten im Bauwesen zeichnen sich überwiegend durch einen großen Materialeinsatz und einen umfangreichen Koordinationsaufwand der Beteiligten aus und sind zumeist durch ein erhöhtes Risiko widriger äußerer Einwirkungen gekennzeichnet. Somit besteht im Umfeld des Bauens die permanente Notwendigkeit der Planung.

2.1 Übergang zur 3D-Planung

Für Planungsprozesse jeglicher Art werden Modelle verwendet. Durch diese wird die Realität auf eine abstrakte Art und Weise – häufig aus der Perspektive eines speziellen Gesichtspunkts – abgebildet. Die meisten im Bauwesen eingesetzten Modelle basieren auf der Grundlage einer geometrischen Repräsentation. Sieht man vom Modellbau in der Architektur ab, so handelt es sich dabei auch heute noch vorwiegend um papierbasierte 2D-Pläne.

Beim papierbasierten Entwurf geht der Planer den Weg, Bauteile durch Strichzeichnungen zu repräsentieren. Diese Skizze symbolisiert die Objekte, die im Gedankenentwurf das zu Planende formen. In technischen Zeichnungen werden diese Bauobjekte nach allgemein anerkannten Regeln so niedergelegt, dass eine zweidimensionale Darstellung das räumliche Objekt eindeutig beschreibt.

Mit der Einführung von 2D-CAD-Systemen wurden die Arbeitsschritte des technischen Zeichnens adaptiert. Es können dort außer den Grundobjekten Punkt und Linie verschiedene Kurven wie Kreise konstruiert werden. Das Computersystem unterstützt den Anwender dabei mit vielfältigen Werkzeugen: So lassen sich automatisch Lote fällen, Schnittpunkte ermitteln, Tangenten konstruieren und Unzähliges mehr. Durch stete Weiterentwicklung über viele Jahre erreichten die CAD-Programme auf diesem Gebiet einen immer leistungsfähigeren Umfang: Es wurden z. B. abschnittsweise parametrisierte Kurven [5] implementiert, ein- und

ausblendbare Ebenen eingeführt, automatische Flächenfüllungen und Bemaßungen verbessert und Symbolbibliotheken ergänzt.

Ein papierbasiertes Modell ist immer auf seine Zweidimensionalität beschränkt. In Ausnahmefällen finden sich zwar isometrische und perspektivische Ansichten eines Entwurfes, doch bestehen dabei keinerlei Assoziationen zu den technischen Zeichnungen wie Grund- oder Aufriss.

Der Einsatz von Computern erlaubt die Modellierung im dreidimensionalen Raum. Dort lassen sich zusätzlich zu den schon im 2D möglichen Objekten Punkt, Linie und Fläche auch Volumenkörper erzeugen. Diese Art der Darstellung ist den vorangehenden Methoden deutlich überlegen, da es in der Realität im Bauwesen ausschließlich Objekte mit räumlicher Ausdehnung gibt, die nun adäquat repräsentiert werden können. Das Erstellen von 3D-Modellen ist allerdings erheblich aufwändiger, weil übliche CAD-Systeme nur über 2D-Eingabe- und 2D-Visualisierungsmöglichkeiten verfügen. Während die Darstellung in perspektivischer Sicht erfolgen kann, was dem Beteiligten aus anderen Kontexten heraus geläufig ist, so stellt die Konstruktion eines 3D-Modells am Computer einen völlig neuen Vorgang in der Planung dar. Diese nicht direkt aus einem bereits bekannten Arbeitsschritt ableitbare Vorgehensweise führt zu einer sehr verhaltenen Einführung von 3D-Modellen.

Dreidimensionale CAD-Modelle stellen jedoch ein enormes Potential zur Verfügung: Während der Planer vormals seine räumlichen Vorstellungen des Entwurfs mit abstrakten Symbolen auf ein 2D-Medium projiziert darstellen musste, so kann er nun mit räumlichen Objekten modellieren. Es lassen sich dabei Räume entwerfen und frei verschieben, Wände können als Begrenzung der Räume definiert und entsprechend platziert werden. Aktuelle CAD-Software unterstützt den Planer durch automatische Verschneidung der Bauelemente, Tür- und Fensteröffnungen lassen sich jederzeit in einer Wand platzieren.

Werden die Volumenobjekte durch zusätzliche Eigenschaften spezieller definiert, so spricht man von einem Produktmodell. Im Bauwesen führt man dafür Bauteile als eigene Objekte ein. Der Planer verwendet nicht mehr allgemeine Volumenobjekte, sondern er wählt sich Objekte aus einem Bauteilklassen-Katalog aus. Dabei muss es sich nicht nur ausschließlich um greifbare Objekte wie Wände und Decken handeln: Auch abstrakte Objekte wie Räume, die den Entwurf gliedern, oder Öffnungen, die massive Bauteile durchbrechen, stehen zur Verfügung und erleichtern das

Modellieren, hat doch der Planer oft eigentlich diese Objekte in der Entwurfsphase vor Augen.

Die Bauteilklassifizierung ist der Schlüssel zu den Vorteilen des computergestützten Entwerfens. Ein steigender Informationsgehalt erlaubt weitere Analysen am virtuellen Bauwerksmodell. Variantenstudien sind schnell durchführbar, wenn beispielsweise alle Wände durch eine andere Ausführung ersetzt werden. Verwenden intelligente Programme das Bauwerksmodell als Datenbasis, so lassen sich grundsätzlich Analysen zum Heizwärmebedarf, zu den Erstellungs- und Betriebskosten (welche auch aus anderen Blickrichtungen als der rein finanzwirtschaftlichen, beispielsweise aus der ökologischen Betrachtung [26][93] durchgeführt werden können) oder sogar zur Bewertung von Evakuierungssituationen [54] erstellen. Bei der Verwendung eines zentralen Bauwerkmodells für alle am Bauwerk Beteiligten kann ferner sichergestellt werden, dass das Modell in sich konsistent bleibt [27]. Widersprüche in der Planung, die durch gegensätzliche Veränderungen zweier unterschiedlicher Fachplaner entstehen könnten, kann ein zentrales Modell mit entsprechender Intelligenz erkennen.

Die Vorteile eines computergestützten, dreidimensionalen Bauwerkmodells bestehen aber nicht nur für Neubauten. Auch oder gerade Altbestand benötigt unterschiedliche Planungsvarianten, hier für die Sanierung, die in ihren Vor- und Nachteilen gegenübergestellt werden müssen. Wie eben beschrieben, ist die Verwendung eines Computermodells des Bauwerks dabei eine äußerst nützliche Arbeitsgrundlage. Jedoch ist die Erhebung im Bestand mit einer anderen Herangehensweise verknüpft. Während man bei der Neubauplanung die Bauelemente im CAD-System als parametrisierte Bauteilobjekte neu entwirft, so gibt der Altbestand oft als Rahmenbedingungen eine fixe Geometrie vor. Diese einfach und wirtschaftlich zu erheben, wird in Kapitel 3 vorgestellt.

Bei der Erstellung eines dreidimensionalen CAD-Modells muss darauf geachtet werden, dass sich die Konstruktion auch an Arbeitsprozessen orientiert. Das Modell darf deshalb nicht nur unter rein geometrischen Aspekten entstehen. Wände sollten immer geschoßweise erstellt werden, um später ebene Teilplanungen zu erlauben (siehe Abb. 2). Auch ist es entscheidend, korrekte Objekte einzusetzen. Die Frage, ob es sich um eine Wand mit mittigem Durchbruch oder zwei Stützen handelt mag im Plan unerheblich, bei einer Nutzung durch automatische Verfahren wie statische Analysen aber durchaus erheblich sein (siehe Abb. 3). Insbesondere für die

in Kapitel 6 gezeigte Bauteilerkennung ist es deshalb notwendig, sich auf einheitliche Modellierungspraktiken zu einigen.

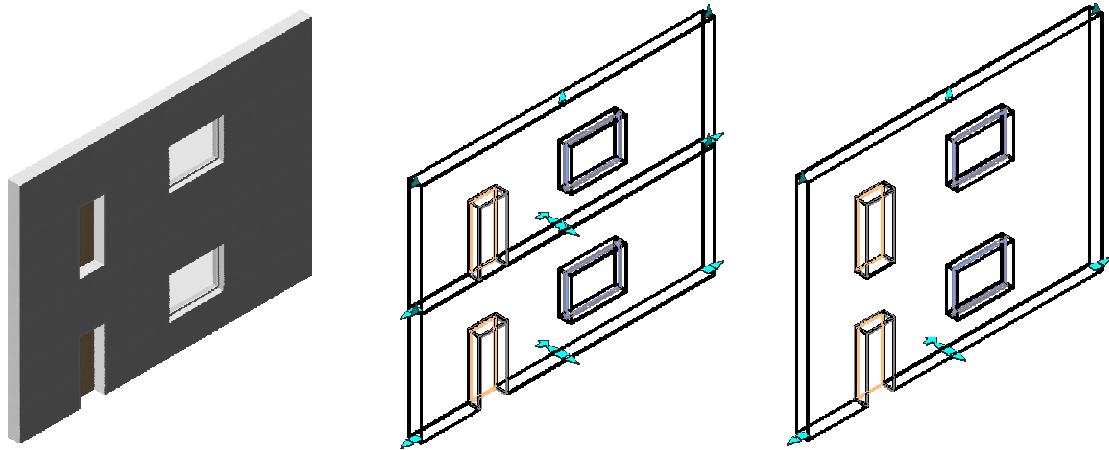


Abb. 2: Durch die geschosswise Modellierung einer Wand (mitte) lassen sich Fenster und Türen mit einheitlicher Brüstungs- bzw. Schwellenhöhe einsetzen

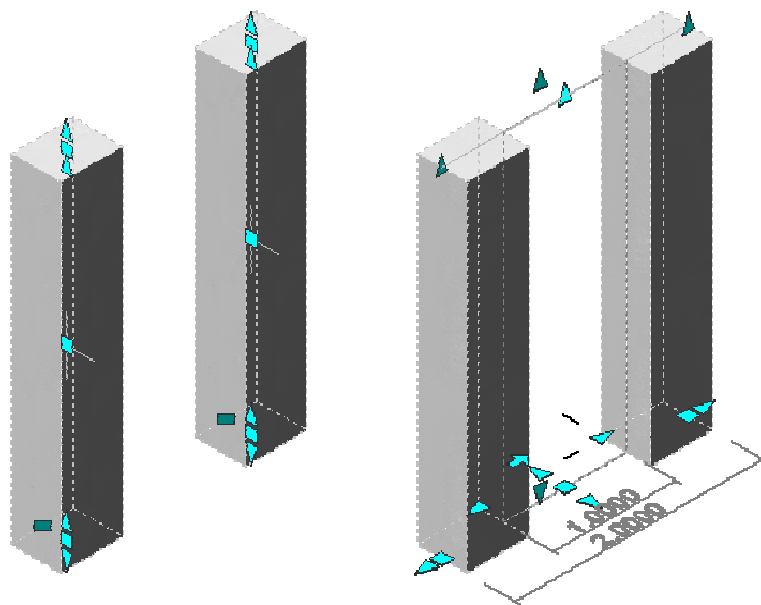


Abb. 3: Die Modellierung eines Durchgangs als beidseitige Stütze (links) oder als durchbrochene Wand (rechts) führt zu gleicher Visualisierung, jedoch unterschiedlicher Interpretation, wodurch z. B. bei automatisierten Verfahren ein anderer Bewehrungsplan erzeugt wird

2.2 3D-CAD-Modelle

Papierbasierte Skizzen, 2D-Computerzeichnungen, 3D-CAD und Produktmodelle haben jeweils ihre Berechtigung. Zum einen muss der Aufwand, dieses Modell zu erstellen, berücksichtigt werden, ebenso die noch praktikable Handhabung der zum Teil immensen Datenvolumina. Auf der anderen Seite ist jedes Modell eine Reduktion der Realität, so dass sich der Anwender der jeweiligen Einschränkungen und Schwachstellen bewusst sein muss. Selbst bei größtem Detaillierungsgrad unter Verwendung moderner Verfahren zur Detailauflösung (level of detail [15], Detailstufenverwaltung [77]) können nie alle Facetten der Wirklichkeit abgebildet werden.

2.2.1 Punktwolke

Im Raum platzierte, einzelne Punkte können ein dreidimensionales Objekt bereits geometrisch repräsentieren. Spezielle Aufmaßverfahren wie z. B. Tachymeter (siehe Abschnitt 3.2.3) sind darauf abgestellt, signifikante Punkte – üblicherweise Eckpunkte – zu erheben. Ohne topologische Information ist es allerdings bei komplexen Körpern nicht mehr möglich, das Objekt zu rekonstruieren, da keine Zugehörigkeit zu Kanten oder Flächen abgeleitet werden kann (siehe Abschnitt 2.2.2).

Nimmt man eine Vielzahl von auf der Objektoberfläche liegenden Punkten auf und visualisiert diese, kann ein menschlicher Betrachter daraus zumeist die ursprüngliche Geometrie ableiten (Abb. 4). Dazu müssen die Punkte nicht zwingend an charakteristischen Orten liegen. Demgegenüber ist es algorithmisch nicht einfach möglich, diese Geometrie zu finden. Dennoch wird die punktweise Aufnahme verwendet, um durch automatisierte Aufmassverfahren wie Laserscanner (siehe Abschnitt 3.2.4) schnell die Oberflächen zu vermessen.

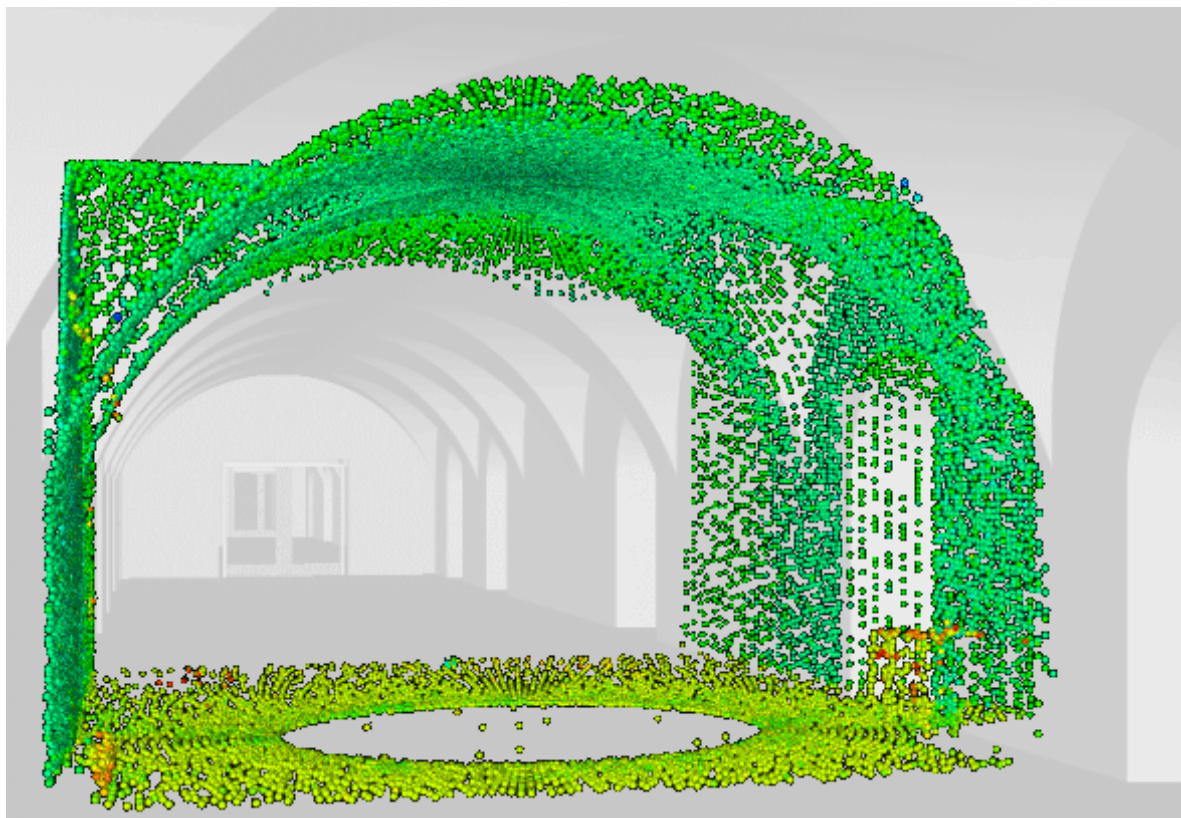


Abb. 4: In einem Gewölbe aufgenommene Punktwolke, wobei der Grad der Reflexion des Laserstrahls durch die Farbe symbolisiert wird

2.2.2 Drahtmodell

Werden die Eckpunkte eines Objektes mit Kanten verbunden, so spricht man von einem Drahtmodell. Dieses besitzt keinerlei Flächen. Wird von einem Drahtmodell auf ein reales Objekt geschlossen, so ist nicht sichergestellt, dass alle angenommenen Flächen korrekt in das Modell interpretiert werden (siehe Abb. 5) [19].

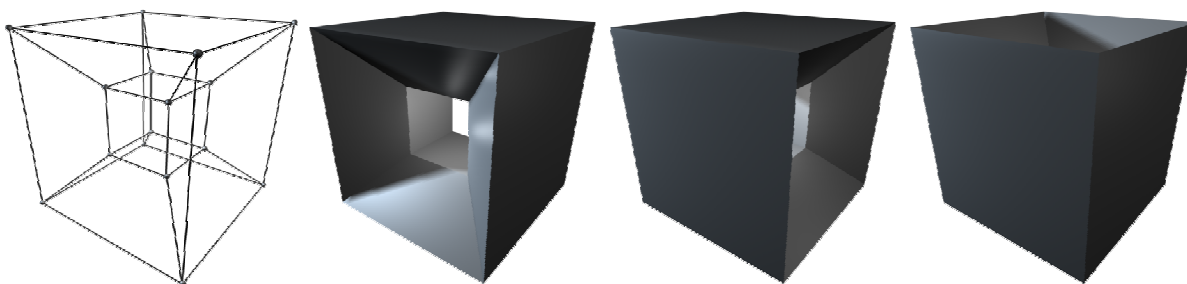


Abb. 5: Als Drahtmodell erstelltes, mehrdeutig interpretierbares Objekt (nach [19])

2.2.3 Oberflächenmodell

Ein Oberflächenmodell schließt die Lücke des Drahtgittermodells und definiert zwischen den Berandungskanten in der Regel polygonale Oberflächen. Dabei sind Dreiecke als einfachstes Flächenprimitiv am weitesten verbreitet. Algorithmen, die mit Polygonnetzen arbeiten, sind effizient formulierbar [8].

Die polygonal definierten Regionen besitzen keine topologische Information bzgl. anderer Oberflächen. So kann nicht gewährleistet sein, dass durch dieses Modell konsistente Körper beschrieben sind. Klaffungen zwischen Flächen können nicht geschlossene Körper ergeben, bei denen keine eindeutige Zuordnung von innen und außen möglich ist. Ferner sind Konstruktionen denkbar, die zu nicht-orientierbaren Flächen führen (siehe Abb. 6).

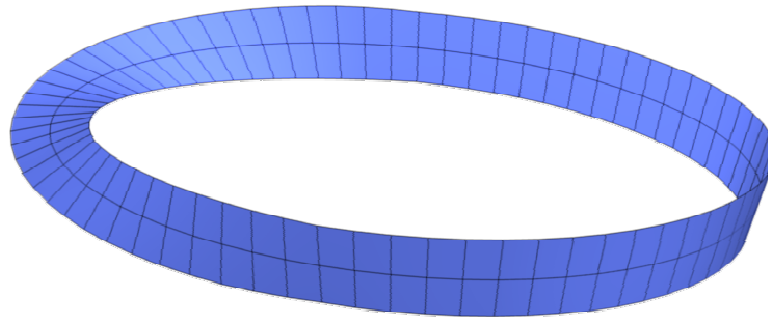


Abb. 6: Das Möbiusband besitzt nur eine Fläche, wodurch ein Innen und Außen nicht unterschieden werden kann

2.2.4 Volumenmodell

Das Volumenmodell beschreibt konsistente (geschlossene) Körper. Aus dem Oberflächenmodell ist dieses unter Zuhilfenahme syntaktischer Informationen generierbar. Man spricht dabei von einem *Boundary representation*-Modell (B-rep), wenn die Beschreibung der Volumina indirekt durch deren umhüllende Oberflächen geschieht. Beim *Constructive Solid Geometry*-Modell (CSG) erfolgt die Volumenbeschreibung hingegen als direkte Repräsentierung durch modellierende Verknüpfungen (oder, und, ohne) von definierten einfachen und parametrisierten Grundkörpern (z. B. Quader, Kugel, ...) [19].

Ein Volumenmodell stellt allerdings nicht eine syntaktisch korrekte Konstruktion sicher. Die Platzierung sowie Dimensionierung der Einzelkörper kann zu einer unsinnigen Modellierung führen (siehe Abb. 7).

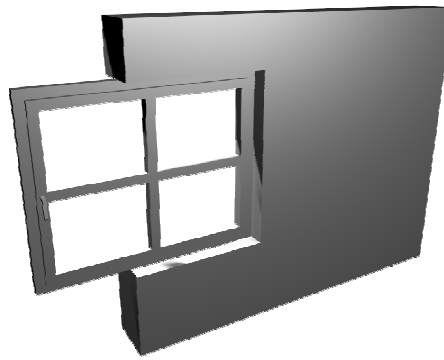


Abb. 7: Fehlerhafte Konstruktion beim Volumenmodell

2.2.5 Produktmodell

Das Produktmodell ergänzt das Volumenmodell um zusätzliche Attribute. Die Einzelkörper werden Objektklassen zugeordnet. Ein Regelwerk zur syntaktisch korrekten Konstruktion stellt die Anordnung der Einzelkörper zueinander sicher. So ist beispielsweise die Platzierung von Fenstern nur innerhalb von Wandobjekten zulässig. Die geometrische Beschreibung erfolgt dabei parametrisiert, wodurch bedingte Abhängigkeiten möglich werden: Wird eine modellierte Wand verschoben, folgen die darin liegenden Fenster und Türen dieser und behalten ihre relative Position, wird die Dicke der Wand verändert, so passen sich die Öffnungsobjekte dieser dynamisch an.

Angereichert durch zusätzliche Informationen kann ein Produktmodell als allgemeine Planungsgrundlage und Basis von Variantenstudien eingesetzt werden.

2.3 Strukturierung dreidimensionaler Daten

Voraussetzung für ein computergestütztes Arbeiten mit geometrischen Modellen ist deren räumliche Gliederung. Im Folgenden sollen Methoden vorgestellt werden, die im Rahmen dieser Arbeit zur Anwendung kommen.

Unter Diskretisierung versteht man dabei die Strukturierung „kontinuierlicher Körper in Form von Gittern oder Netzen“ ([72], S. 18). Diese dient der Beschleunigung von Berechnungsverfahren, da das ursprüngliche, meist komplexe Objekt in einfachere Substrukturen unterteilt wird, welche deutlich schneller verarbeitet werden können. Für lokale Einwirkungen, die sich auf das globale System fortpflanzen, wie beispielsweise Verformungen, ist die Kommunikation mit benachbarten Elementen von Bedeutung. Diese topologische Information ist innerhalb einer räumlich

vernetzten Struktur an definierte Nachbarschaften gekoppelt. Sie muss nicht zwingend im direkten geometrischen Zusammenhang stehen, es können auch Ballungen, die ein übergeordnetes System als eine Einheit betrachtet, und Beziehungen zu entfernteren Einheiten angelegt werden. Die Erzeugung analytischer Flächen, in welche segmentierte Bereiche überführt werden, ist in dieser Arbeit auf Ebenen beschränkt. So wird die Einbeschreibung einer Ausgleichsebene in eine Punktmenge beschrieben.

2.3.1 Voxel

Der Begriff Voxel leitet sich aus der Übertragung zweidimensionaler Computer-Bildpunkte (engl. Pixel) in ein räumliches (engl. volumetric) System ab. Durch Voxel lässt sich der kontinuierliche Raum in diskrete Zellen auflösen. Aus praktischen Gründen verwendet man üblicherweise Würfel. Alle Elemente besitzen dabei die gleiche Größe, weshalb man auch von Normzellen spricht [72]. Die Kantenlänge der Zellen bestimmt dabei den möglichen Auflösungsgrad des Raumes.

Um ein Objekt als Voxelmatrix darzustellen, wird für jede einzelne Zelle entschieden, ob sie sich innerhalb oder außerhalb des Objektes befindet. In diesem Fall kann ein Voxel die Zustände gesetzt oder ungesetzt annehmen. Für Voxel, die vom Objektrand durchschnitten werden, ist zu definieren, welcher Gruppe sie zuzuordnen sind (Abb. 8).

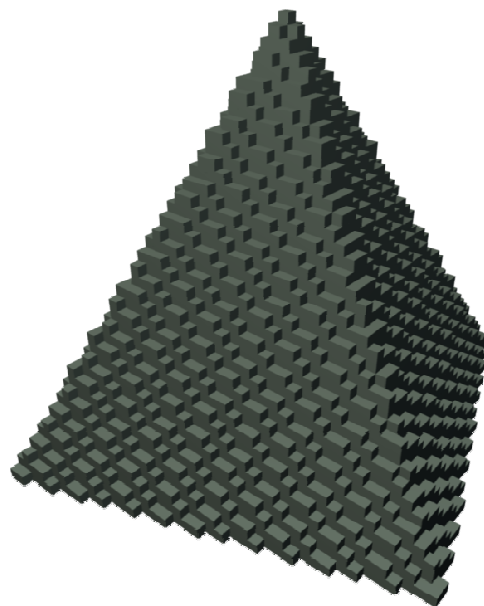


Abb. 8: Als Voxelmmodell dargestellte Pyramide

Der Vorteil in der Anwendung von Voxel ist die für Rechenoperationen geschickte Matrixstruktur. Nachbarschaftsbeziehungen lassen sich damit sehr schnell ermitteln. Insbesondere Algorithmen, die Informationen zwischen Zellen propagieren, wie beispielsweise Lattice-Boltzmann-Strömungssimulationen [88], profitieren von dieser Eigenschaft. Es ist deshalb naheliegend, aus einer – beispielsweise in Form eines Oberflächenmodells repräsentierten – Geometrie automatisch Normzellen zu generieren. In Anlehnung an die erzeugte Struktur spricht man dabei von Voxelierung [96].

Die einzelnen Zellen können neben diskreten Zuständen wie gesetzt oder ungesetzt [72] bzw. Attributen mit größerem Wertebereich wie beispielsweise Farbwerten auch Objekte aufnehmen [54].

2.3.2 Oktalbaum

Ein Oktalbaum [77] löst den Raum ebenfalls in Substrukturen auf. Diese sind im Allgemeinen jedoch nicht uniform sondern verfeinern adaptiv. So können komplexe Geometrien innerhalb des Oktalbaums abgelegt werden, ohne einen Kompromiss zwischen einer hohen Raumaufösung in Gebieten großer Strukturiertheit einerseits und Speicheraufwand in unberührten Bereichen andererseits eingehen zu müssen. Wegen des nicht-uniformen Aufbaus ist die Suche nach benachbarten Zellen jedoch mit größerem Rechenaufwand verbunden.

Ein Oktalbaum besitzt drei verschiedene Arten von Zellen: Die quaderförmige Wurzelzelle umfasst vollständig den zu diskretisierenden Raum. Auch hier wird aus algorithmischen Gründen oft eine einheitliche Seitenlänge gewählt. Jede Zelle kann durch acht Nachfolgezellen, welche die halben Kantenlängen ihres Vorgängers besitzen, verfeinert werden. Diese decken mit ihren Teilräumen das ursprüngliche Gebiet der Elternzelle ab. Zellen, die weitere Nachfolger besitzen, werden als innere Knoten, ansonsten als Blätter bezeichnet. Die Gliederung eines Baumes geschieht in Ebenen, beginnend bei der Wurzelzelle mit Ebene 0. Mit jeder Unterteilung einer Zelle inkrementiert sich die Ebenennummer um eins (Abb. 9). Die Tiefe eines Baumes entspricht der maximalen Ebene.

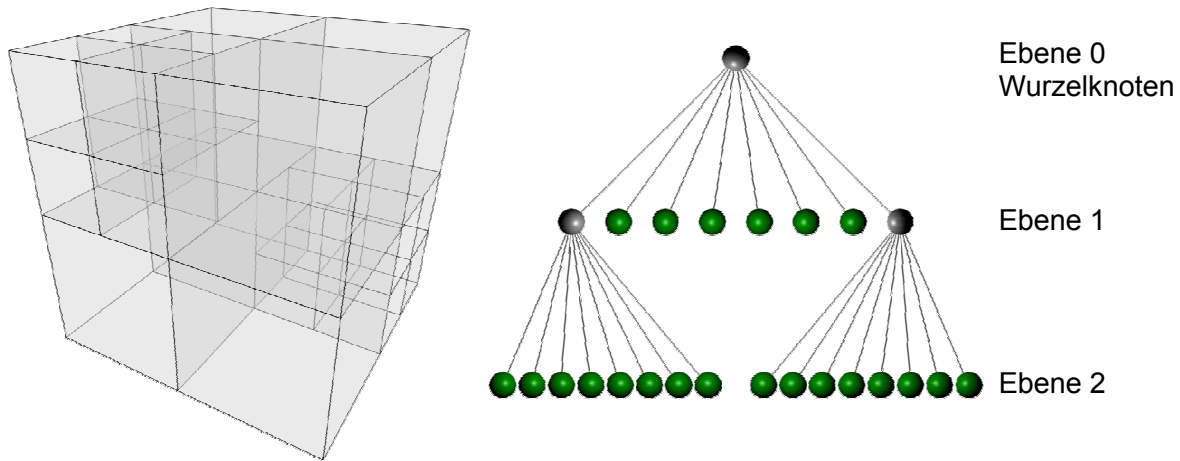


Abb. 9: Geometrische Repräsentation (links) eines Oktaalbaums (rechts) mit lokaler Verfeinerung, innere Knoten grau, Blätter grün dargestellt

Um eine Geometrie in einem Oktaalbaum zu speichern wird diese zunächst in einer neu erzeugten Wurzelzelle abgelegt. Mit einem rekursiven Algorithmus wird diese und jede aus ihr hervorgehende Subzelle dann verfeinert, wenn sie ein vorgegebenes Kriterium erfüllt, beispielsweise die Objektoberfläche schneidet. Andernfalls wird festgehalten, ob sich das Blatt innerhalb oder außerhalb des Objektes befindet [40]. Eine oftmals vorher festgelegte maximale Baumtiefe schreibt die Anzahl an Rekursionen vor.

Mit Oktalbäumen lassen sich sehr vorteilhaft Vorentscheidungen für Kollisionsberechnungen durchführen [12]. So ist eine aufwändige Berechnung von polygonalen Durchdringungen nur dort nötig, wo sich Blätter mit Objekträndern befinden. Da die Tiefe des Oktaalbaums jederzeit veränderbar ist, lässt sich dieser auch auf eine dynamische Formänderung der zu speichernden Geometrie anpassen.

Generell sind Baumdatenstrukturen auch in niedriger- oder höherdimensionalen Räumen anwendbar. Die Anzahl an Subzellen errechnet sich allgemein zu

$$n = 2^D \quad (2.1)$$

mit D als Anzahl an Raumdimensionen

Die Ermittlung benachbarter Blätter innerhalb eines Oktaalbaumes kann auf unterschiedliche Arten erfolgen [7][45]. Eine Möglichkeit zur Bestimmung der Nachbarn des aktuellen Blattes ist, zu dessen Mittelpunkt durch die Mitte der sechs Flächen, der zwölf Kanten und der acht Eckpunkte des Hüllwürfels ein Vektor mit der Länge der Kante des Hüllwürfels zu addieren (Abb. 10). Vom Wurzelknoten aus

startet zunächst eine Suche nach den Zellen, in denen der Ort liegt, auf den die Vektoren zeigen. Die Suche nach den Nachbarzellen erfolgt zunächst bis zur maximal gleichen Ebene des Startblattes. Besitzen diese Zellen noch weitere Kindzellen, so werden jene weiterverfolgt, die in entgegengesetzter Vektorrichtung liegen. Der Algorithmus ist rekursiv, d. h. liegen mehrere Kindzellen in dieser Richtung, werden alle bis zu den entsprechenden Blättern durchlaufen. Äste, die nicht so fein aufgelöst sind, liefern Blätter aus höheren Ebenen.

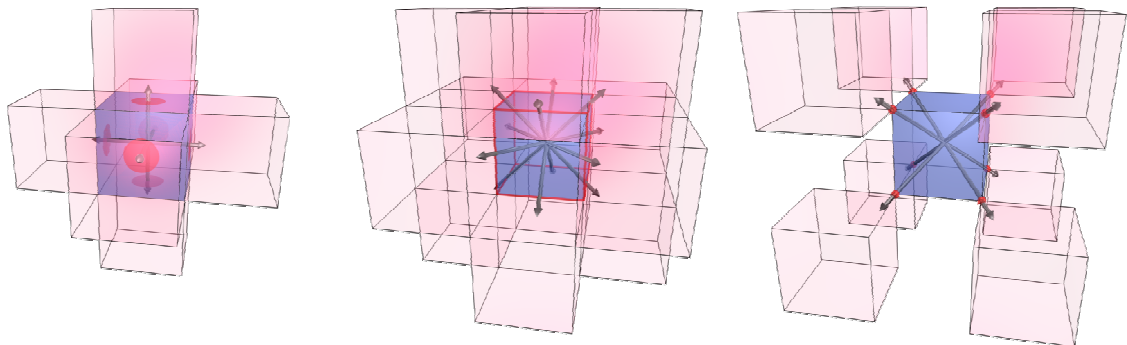


Abb. 10: Markierung der Nachbarzellen durch vektoriell durchschreiten der Zellenflächen (links), -kanten (Mitte) und -ecken (rechts)

2.3.3 Ballungen

Die Ballungsanalyse wird verwendet, um Objekte mit ähnlichen Eigenschaften zu Gruppen zusammenzufassen. Diese Anhäufung wird häufig auch mit ihrem englischen Begriff als Clusterverfahren bezeichnet [75]. In der hierarchischen Ballungsanalyse sind zwei verschiedenen Vorgehensweisen möglich: Der agglomerative (anhäufende) bottom-up-Ansatz, der einzeln betrachtete Objekte immer größer werdenden Gruppen zuteilt, und der diversive (teilende) top-down-Ansatz, welcher die Gesamtmenge der Objekte schrittweise unterteilt. Die Anzahl an Merkmalen, anhand derer die Objekte sortiert werden, kann eine beliebige natürliche Zahl sein.

Um die Objekte zusammenfassen zu können, ist eine Distanzfunktion zu definieren, welche den Abstand eines Elements zu einer Gruppe bestimmt. Eine mögliche Methode zur Berechnung des Abstandes ist die Verwendung der Euklidischen Distanz. Zwischen zwei Vektoren berechnet sich dieser zu

$$d(\underline{x}, \underline{y}) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (2.2)$$

mit $\underline{x} = [x_1 \ \dots \ x_n] \in \mathbb{R}^n$ als Merkmalsvektor eines Objektes

und $\underline{y} = [y_1 \ \dots \ y_n] \in \mathbb{R}^n$ als Merkmalsvektor einer Gruppe

Dabei gilt es festzulegen, wie sich der Vektor \underline{y} aus den Merkmalsvektoren der Mitglieder der bestehenden Gruppe errechnet. Üblicherweise geschieht dies durch Berechnung des Mittelwertes.

$$\underline{y} = \frac{\sum_{i=1}^n \underline{y}_i}{n} \quad (2.3)$$

mit \underline{y}_i als Merkmalsvektor des i-ten Objektes der Gruppe

und n als Anzahl der in der Gruppe enthaltenen Objekte

In einem anderen Anwendungsfall kann auch die Verwendung des Minimal- oder Maximalabstandes, also die Entfernung zum nächsten bzw. weitest entfernten Objekt in der Gruppe sinnvoll sein. Ebenso ist die Berechnung der Distanz über eine andere als die Euklidische Metrik möglich [9]. Die Wahl der Distanzfunktion orientiert sich an den Objekteigenschaften, die im Merkmalsvektor abgelegt sind.

Ein zuvor festgelegter Grenzwert bestimmt, ob ein Element in eine bestehende Menge integriert wird oder eine neue Gruppe anzulegen ist. Unterschreitet der Abstand zu mehreren Ballungen diesen Wert, so fällt die Entscheidung zugunsten der Gruppe mit der geringsten Distanz. Somit wird sichergestellt, dass es zu keinen überlappenden Bereichen kommt.

Durch die Aufnahme eines neuen Mitglieds ändert sich stets der Merkmalsvektor der Gruppe. Die Konsequenz daraus ist, dass bereits enthaltene durch später hinzugekommene, weitere Elemente die Abstandsbedingung verletzen können. Durch überprüfende Iterationsschleifen werden deshalb korrigierende Verschiebungen vorgenommen.

2.3.4 Delaunay-Vernetzung

Eine Methode, um im Raum eine ungeordnete Menge an Punkten zu vernetzen, hat der sowjetische Mathematiker Boris Nikolaevich Delaunay 1934 vorgestellt [23]. Für den zweidimensional Fall gilt: Die Punkte werden so geschickt zu Dreiecken vernetzt,

dass innerhalb jedes Umkreises eines Dreiecks keine anderen Punkte enthalten sind. Ein Delaunay-Netz lässt sich erzeugen, indem man seinen dualen Graphen, das Voronoi-Diagramm [62][92], in der Punktmenge konstruiert. Jeder beliebige Ort innerhalb einer Polygonfläche des Voronoi-Diagramms besitzt die Eigenschaft, dass er zum zentralen Punkt dieses Polygons näher liegt als zum zentralen Punkt eines jeden anderen Polygons. Alle Punkte, die in diesem Diagramm eine gemeinsame Voronoi-Kante trennt, sind direkte Nachbarn und werden deshalb durch Dreieckskanten verbunden [47].

Dieses Verfahren gilt auch in drei Dimensionen, wobei eine Vernetzung mit Tetraederelementen entsteht. Innerhalb deren Umkugeln – vier nicht in einer Ebene liegende Punkte, also die Tetraedereckpunkte, definieren eine Kugel eindeutig – existieren wiederum keine anderen Punkte [44]. Der duale Graph zum Tetraedernetz besteht nun aus Flächen, welche benachbarte Punkte trennen. Um die Ränder eines Volumenkörpers korrekt abzubilden kann es notwendig sein, zusätzliche Punkte einzufügen [76]. Die Literatur gibt vielerlei Verfahren an, um ein Delaunay-Netz zu erzeugen [20].

2.3.5 Ausgleichsebenen

Für eine Weiterverarbeitung einer Menge von Punkten ist es oftmals erwünscht, eine analytische Fläche, welche die Punkte zu repräsentieren versucht, zu definieren. Diese Arbeit beschränkt sich ausschließlich auf ebene Flächen.

Eine Ebene wird durch drei nicht auf einer Geraden liegenden Punkte eindeutig festgelegt. Um eine Ebene zu finden, die durch mehr als drei Punkte beschrieben wird und um diese streuen, ist eine Ausgleichsebene zu berechnen. Über die Minimierung der Summe der Quadrate der orthogonalen Abstände aller Punkte können der Aufpunkt und der Normalenvektor als eindeutig beschreibende Parameter der Ausgleichsebene ermittelt werden [70]. So bestimmt sich der Aufpunkt S aus allen betrachteten Punkten über die Berechnung des Mittelwertes entsprechend Formel (2.3). S wird auch als Schwerpunkt bezeichnet.

Wird die gesuchte Ebene in Form der Hesseschen Normalform

$$\vec{x} \cdot \vec{n}_0 - d = 0 \tag{2.4}$$

mit $\vec{n}_0 = [n_x \ n_y \ n_z]$ als normierter Ebenennormalenvektor
und d als skalarer Abstandswert vom Koordinatenursprung

ausgedrückt, so errechnet sich der Abstand v_i , den ein Punkt $P_i^T = [x_i \ y_i \ z_i]$ von der Ebene besitzt, zu

$$v_i = x_i \cdot n_x + y_i \cdot n_y + z_i \cdot n_z - d \quad (2.5)$$

bzw. für alle m Punkte als Abstandsvektor

$$\underline{v} = \underline{\underline{A}} \cdot \underline{n} - \underline{1} \cdot d \quad (2.6)$$

$$\text{mit } \underline{\underline{A}} = \begin{bmatrix} x_1 & y_1 & z_1 \\ \vdots & \vdots & \vdots \\ x_m & y_m & z_m \end{bmatrix}$$

$$\text{und } \underline{n} = \vec{n}_0 = [n_x \ n_y \ n_z]^T$$

$$\text{und } \underline{1} = [1 \ \dots \ 1]^T \text{ als Einsvektor}$$

Durch Betrachtung der Punktkoordinaten im lokalen Koordinatensystem mit Ursprung um den Schwerpunkt $S = [x_s \ y_s \ z_s]$ lässt sich der Abstandsterm eliminieren [50].

Die Gleichung vereinfacht sich zu

$$\underline{v} = \underline{\underline{A}}^* \cdot \underline{n} \quad (2.7)$$

$$\text{mit } \underline{\underline{A}}^* = \begin{bmatrix} x_1 - x_s & y_1 - y_s & z_1 - z_s \\ \vdots & \vdots & \vdots \\ x_m - x_s & y_m - y_s & z_m - z_s \end{bmatrix}$$

Die Minimierung des Fehlerquadrats ε , welches sich als Summe der Abstandsquadrate errechnet

$$\varepsilon = \sum_{i=1}^m (v_i)^2 = \underline{v}^T \underline{v} = (\underline{\underline{A}}^* \cdot \underline{n})^T (\underline{\underline{A}}^* \cdot \underline{n}) = \underline{n}^T \cdot \underline{\underline{A}}^{*T} \cdot \underline{\underline{A}}^* \cdot \underline{n} \quad (2.8)$$

ist also

$$\min_{\underline{n}} \left(\underline{n}^T \cdot \underline{\underline{A}}^{*T} \cdot \underline{\underline{A}}^* \cdot \underline{n} \right) \quad (2.9)$$

Die Spektralzerlegung der reellwertigen symmetrischen Matrix

$$\underline{\underline{N}} = \underline{\underline{A}}^{*T} \cdot \underline{\underline{A}}^* \quad (2.10)$$

führt zu deren Eigenwerten und orthogonalen Eigenvektoren [86]. Der in Gleichung (2.8) definierte Fehler ist minimal, wenn \underline{n} dem Eigenvektor mit dem kleinsten Eigenwert von \underline{N} entspricht [25]. Die beiden anderen Eigenvektoren spannen die Ausgleichsebene auf.

3 Bauaufnahme im Altbau

3.1 Zweck der Bauaufnahme

Die Motivation für eine Bestandsaufnahme im Bauwesen entspringt unterschiedlichen Bereichen: Im Zusammenhang mit der Fertigung des Bauwerks dient das Aufmaß zur Sicherstellung der planmäßigen Umsetzung. Spätere bauliche Veränderungen setzen eine Sanierungsplanung voraus, für die der Ist-Zustand durch Aufmaßverfahren erhoben werden muss. In der Betriebsphase erfolgt die Überwachung von Ingenieurbauwerken wie Staumauern und Brücken. Insbesondere Veränderungen der Standsicherheit sind aus Abweichungen von der Sollgeometrie abzuleiten [59]. Betriebswirtschaftlichen Überlegungen beim Eigentümer- oder Nutzungswechsel gehen ebenfalls Bestandsaufnahmen voraus. Ein großes Anwendungsgebiet findet die Bauaufnahme schließlich im Bereich des Denkmalschutzes. Für jede dieser Anwendungen ist eine andere Detailtiefe, welche nicht nur die geometrische Genauigkeit, sondern auch die daneben zu erfassenden Informationen meint, notwendig. [66]

Die Grundlage bei der Erfassung von Baubestand ist die Aufnahme der Geometrie. Um diese zu bestimmen, sind im ersten Schritt Längenmaße, unter Umständen auch Winkel zu erheben. Aus den gewonnenen Werten lässt sich dann ein Abbild erstellen, welches häufig in einem kleineren Maßstab und durch eine ebene Projektion geschieht. Die Wahl des Maßstabes legt den Detaillierungsgrad der Darstellung fest. Für die Vermessung werden im Folgenden vier Methoden vorgestellt. Im Anschluss an jede Aufnahme, die auch aus den verschiedenen Verfahren zusammengesetzt sein kann, folgt die Erstellung des Modells. Erst in der Kombination von Aufmaß und Modellierung zeigt sich die Wirtschaftlichkeit der gewählten Methode für die Bauaufnahme.

3.2 Aufmaßverfahren

3.2.1 Handvermessung

Die einfachste Methode eines Aufmaßes ist das direkte Messen von Längen. Bei den Dimensionen im Bauwesen verwendet man dazu ein Maßband oder einen Meterstab (Gliedermaßstab). Es ist damit möglich, gerade Abstände entlang einer Oberfläche oder im Raum zu ermitteln. Bei Längenmessungen im Bereich von mehreren Metern

liegt die Messunsicherheit bei ungefähr ± 5 mm. Sie ist zum einen begründet in der Qualität des Maßbandes, aus Metal gefertigt unterliegt dieses einer geringen, Temperatur abhängigen Längenänderung, der aus Holz gefertigte Meterstab reagiert insbesondere auf Feuchtigkeit. Die Mechanik des Meterstabes stellt eine weitere Quelle für die Unschärfe dar. Fehler in der Anwendung, beispielsweise bei Verwendung eines nicht exakt ausgeklappten Meterstabes oder bei einem durchhängenden Maßband, ungewollt schrägen Messungen und fehlerhaften Anhaltepunkten beziehungsweise Ablesungen beeinträchtigen die Genauigkeit von Anwenderseite. Der erkannte Messfehler von Maßbändern bezüglich Temperatur und Durchhang ist durch Einsatz von Reduktionsformeln [38] korrigierbar.

Ein Sonderfall in der Anwendung ist die Messung von konvexen Oberflächen, welche mit einem Maßband am geeignetsten durchgeführt werden kann.

3.2.2 Laserdistanzmessgerät

Ein Laserdistanzmessgerät bestimmt den Abstand zwischen zwei Punkten durch die Kombination aus Laufzeitmessung und Phasenvergleich eines Laserpulses [98]. Dazu braucht nur mehr ein Messpunkt direkt zugänglich und der zweite vom ersten aus sichtbar sein. Das am zugänglichen Messpunkt positionierte Messgerät ist auf den zweiten Punkt auszurichten (Abb. 11). Der emittierte Laserstrahl wird von der Oberfläche des angepeilten Ziels diffus reflektiert und mittels eines Fotosensors wieder am Gerät erfasst. Aus der halben Zeitdifferenz zwischen Emission und Detektion errechnet das Messgerät unter Kenntnis der Lichtgeschwindigkeit die Entfernung. Beim Phasenvergleichsverfahren werden auf den Laserimpuls weitere langwelligere Frequenzen aufmoduliert. Durch die Phasenverschiebung der aufmodulierten Schwingung vom ausgesendeten zum reflektierten Signal lässt sich ebenfalls die Entfernung bestimmen. Da es sich um eine periodische Aufmodulation handelt, kann das Ergebnis mehrdeutig sein. Diesem Dilemma wird entweder durch mehrere Messungen mit unterschiedlichen Modulationswellenlängen oder der parallelen Laufzeitmessung begegnet. Die Messgenauigkeit des Gerätes liegt bei wenigen Millimetern. Gerätehersteller geben diesen mit einem festen Wert von drei Millimetern zuzüglich einem Messdistanz abhängigen Teil von $3 \cdot 10^{-6}$ an.

Wie bei der Handvermessung ist das Laserdistanzmessgerät im Bereich der Erhebung weniger, gerader, aussagekräftiger Längen, welche zugänglich sein müssen, wirtschaftlich einzusetzen.

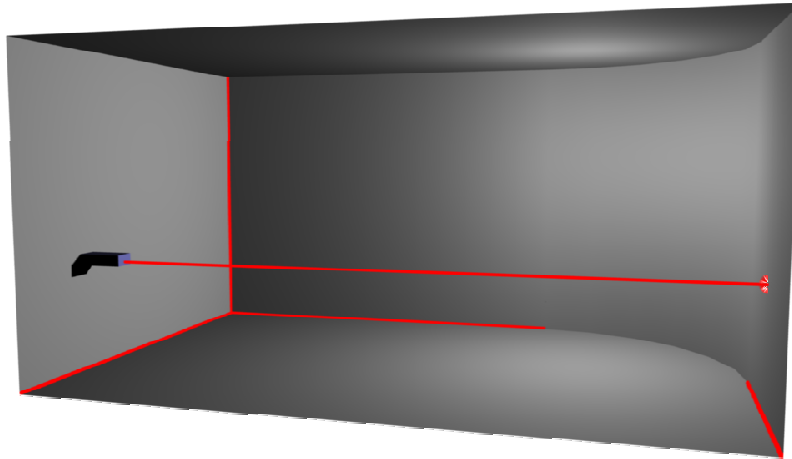


Abb. 11: Streckenmessung mit Laserdistanzmessgerät

3.2.3 Reflektorlos distanzmessender Tachymeter

Ein Tachymeter ist die Kombination aus einem Theodolit und einer elektronischen Distanzmessung. Dazu muss er vom Benutzer auf die Zielpunkte eingestellt werden und misst zu diesen jeweils den Horizontal- und Vertikalwinkel sowie die Schrägdistanz (Abb. 12). Moderne Geräte ermitteln die Stellwinkel elektronisch, bei reflektorlos messenden Geräten erfolgt die Bestimmung der Entfernung wie bei Laserdistanzmessgeräten durch die Laufzeitmessung eines Laserstrahls in Kombination mit dem Phasenvergleichsverfahren. Aus den eingestellten Winkeln und der Entfernung lässt sich der Messpunkt vom Polarkoordinatensystem des Gerätes in ein lokales kartesisches Koordinatensystem übertragen. Die Messgenauigkeit des Systems setzt sich aus einer Unschärfe in der Längenmessung von einem Millimeter sowie einer Ungenauigkeit in der Winkelmessung zusammen und führt zu einer Unsicherheit der Messpunkte von ± 3 mm. Durch ungenaues Anpeilen können zusätzliche Fehler entstehen.

Wirtschaftlich vermessen Tachymeter wenige signifikante Punkte eines Objektes. Diese müssen alle vom Standort des Gerätes direkt sichtbar sein, andernfalls sind zusätzliche Aufstellorte notwendig, die über Referenzpunkte mit der bisherigen Messung in Korrelation gebracht werden. Im Gegensatz zur Handvermessung und zum Laserdistanzmessgerät können beim Einsatz eines Tachymeters alle zu erfassenden Punkte unzugänglich sein. Der Benutzer muss jedoch die Zielpunkte optisch erfassen können, um diese mit dem Gerät anzupeilen. Deshalb ist eine hinreichende Beleuchtung der Szenerie Voraussetzung.

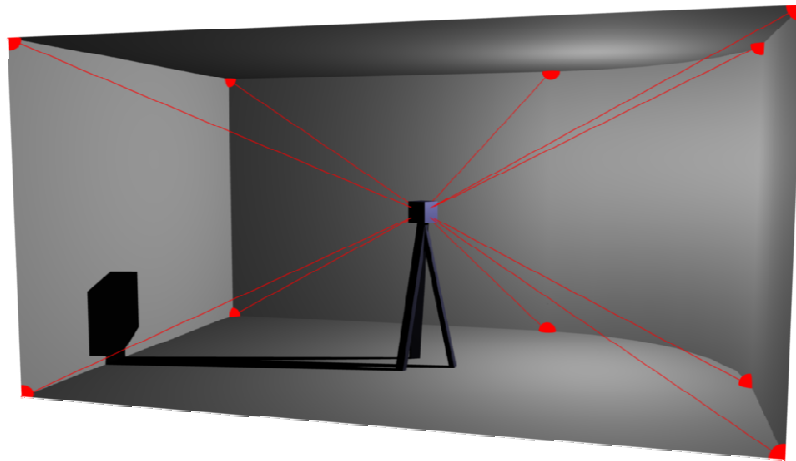


Abb. 12: Aufnahme signifikanter Einzelpunkte mittels Tachymeter

3.2.4 Terrestrische Laserscanner

Laserscanner dienen der vollautomatischen Erfassung von Oberflächengeometrien. Ein Laserstrahl tastet dazu in einer festlegbaren Schrittweite die Umgebung ab (Abb. 13). Wie beim Tachymeter werden dabei die Horizontal- und Vertikalwinkel sowie die Schrägdistanz gemessen. Als zusätzlichen Parameter liefert der Scanner die Intensität des reflektierten Lasersignals [83]. Terrestrische Laserscanner sind als Geräte mit einem horizontalen Sichtbereich von 360° verfügbar und erfassen so innerhalb weniger Minuten mit Ausnahme eines Kegels direkt unter ihnen ihr Umfeld mit bis zu mehreren hunderttausend Punkten pro Sekunde. Die Punkte eines Messvorgangs werden im Anschluss daran vom Polarkoordinatensystem in ein lokales kartesisches Koordinatensystem umgerechnet. Für die spätere Visualisierung steht als Farbinformation der Punkte die Reflexionsintensität zur Verfügung (siehe Abb. 4). Es ist auch möglich, die Messpunkte entsprechend den Farbwerten einer parallel durchgeführten Digitalkameraaufnahme zu colorieren [99].

Bei einem Messfehler von etwa 6 Millimetern [78] scheint die Genauigkeit hinter der des Tachymeters zu liegen. Jedoch kompensiert sich dieser durch die hohe Anzahl an Messpunkten und der darin hineingelegten Modellierung, beispielsweise von Ausgleichsebenen, zu deren Bestimmung mehrere hundert Punkte herangezogen werden.

Durch Laserscanner ist es möglich, in kurzer Zeit eine komplexe Geometrie zu erfassen. Störungen, die durch das Aufmessen resultieren, können auf ein Minimum reduziert werden. Dies ist insbesondere bei Messungen im Industriebereich oder im Straßenverkehrsraum von Vorteil. Ein weiterer Vorzug des Verfahrens liegt darin,

dass das Gerät nicht mehr auf spezielle Punkte ausgerichtet werden muss, weshalb es keiner zusätzlichen Beleuchtung des Messobjektes bedarf. Bei Aufmaßen von Örtlichkeiten, die schwer ausleuchtbar sind wie beispielsweise ein historischer Dachstuhl, reduziert dies den Aufwand beträchtlich. Sind Aufnahmen von mehreren Standorten aus notwendig, werden die einzelnen Punktwolken miteinander verknüpft. Für die Erfassung sehr komplexer Oberflächenstrukturen, wie es zum Beispiel bei historischen Fassaden im Bereich Denkmalschutz notwendig ist, ist der Laserscanner ein äußerst wirtschaftliches Mittel.

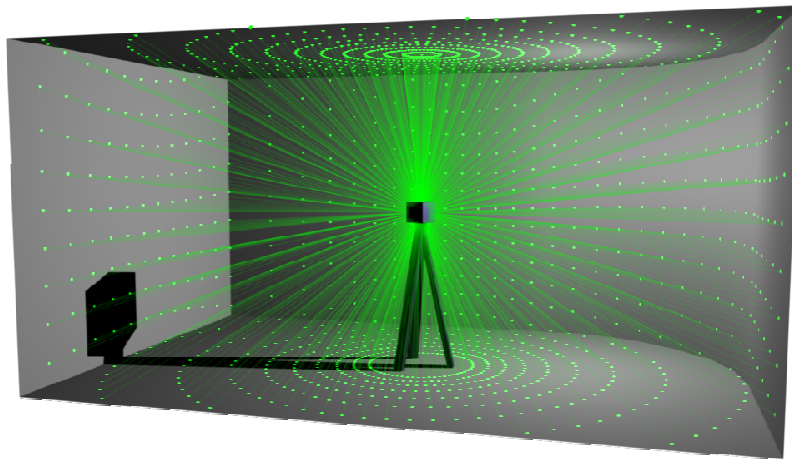


Abb. 13: Aufnahme einer Punktwolke mit einem 3D-Laserscanner

3.3 Bestandsaufnahme mittels Laserscanner

Die Möglichkeiten eines Laserscanners lassen ein automatisches Erfassen und Modellieren erwarten. Wenngleich der eigentliche Messvorgang vom Gerät bereits völlig autark durchgeführt wird, so ist es zum momentanen Stand der Technik noch nicht möglich, aus den erhobenen Messpunkten automatisch ein CAD-Modell abzuleiten. Für diesen fehlenden Bereich möchte diese Arbeit in den Kapiteln 4 und 6 einen Beitrag leisten. Im Folgenden wird näher erläutert, welche Schritte zur Erstellung des Aufmaßes mit einem Laserscanner notwendig sind.

Ziel der Aufnahme ist es, ein umfassendes Bild der Umgebung aus der Punktwolke abzuleiten. Eine konstante Abtastrate der Oberfläche fördert diesen Prozess. Dazu muss vom Anwender die gewünschte Auflösung festgelegt werden. Diese wiederum lässt sich durch die Winkelschrittweiten des Laserscanners ausdrücken, steht aber noch in Abhängigkeit vom Aufstellort des Gerätes. Die sorgsame Planung der einzelnen Standpunkte ist Voraussetzung für eine ausgewogene Messung, bei der ein festgelegter Bereich für die Punktdichte nicht unter- oder überschritten wird [11].

Ein günstiger Standort erfasst mit seinem Sichtbereich ein großes Gebiet des Messobjektes. Hierfür sind insbesondere Verdeckungen zu beachten.

Erfordert der Sichtbereich auf das Aufmaßobjekt ein Umsetzen des Laserscanners, so sind Referenzmarken zu setzen, die von beiden Standorten eingesehen werden können [60]. Die Anzahl der notwendigen Passpunkte, aus denen die relativen Aufstellpositionen ermittelt werden, ist von der Gerätebauart abhängig. Scanner, die selbständig ihre Stehachsschiefe kompensieren und damit die Vertikalachse ermitteln können, benötigen mindestens zwei Punkte, ansonsten sind drei zu verwenden. Als Referenzmarken können zum einen auf Oberflächen aufgeklebte Scheiben oder im Raum aufgestellte Kugeln dienen (Abb. 14). Aufgrund der Reflexionseigenschaften und einer äußerst feinen Abtastauflösung in diesem Bereich errechnet die Scannersoftware den Mittelpunkt der Scheibe. Ähnlich wird im Falle der Referenzkugel verfahren: Aus dem in der Punktwolke vom Benutzer ausgewählten Bereich der Kugeloberfläche ermittelt die Software den Mittelpunkt.

Bei der anschließenden Registration werden die einzelnen Punktwolken, deren Aufnahme von den verschiedenen Standorten aus erfolgt ist, miteinander verbunden. Mittels der angebrachten Referenzmarken, die nun aus den einzelnen Aufnahmen einander zugeordnet werden, lassen sich Transformationsmatrizen errechnen, die die einzelnen Punktwolken in ein gemeinsames Koordinatensystem übertragen. Das dafür übliche Verfahren ist die Helmert-Transformation [55]. Dieser Vorgang geschieht vollautomatisch innerhalb weniger Sekunden bis einiger Minuten, abhängig von der Menge an Punkten.



Abb. 14: Als Referenzmarke an einer Wand angebrachte Scheibe (links) und auf einem Dreibein befestigte Kugel (rechts)

4 Erstellung eines Oberflächenmodells aus Aufmaßdaten

Dieses Kapitel stellt ein im Rahmen dieser Arbeit entwickeltes Verfahren zur Rekonstruktion eines Oberflächenmodells vor. Basis ist die mit Hilfe von Methoden der 3D-Laserscanning erzeugte, topologisch unstrukturierte Punktwolke. Zunächst gibt das Kapitel eine Übersicht über die in diesem Zusammenhang in der Praxis eingesetzten Verfahren und stellt den Stand der Forschung dar.

Für einen menschlichen Beobachter erscheint es einfach, die bei der Betrachtung einer Punktwolke fehlenden Informationen zu ergänzen. Auf dem Erfahrungsschatz und dem erkannten Kontext basierend wird das Punktemodell imaginär in ein Flächen- oder Volumenmodell umgeformt. Die menschliche Interpretationsfähigkeit lässt sich durch eine entsprechende Visualisierungsmethode noch weiter unterstützen: Scheiben, welche eine Ausrichtung entlang einer vermuteten Oberfläche erfahren, substituieren die Punkte und erzeugen einen flächenhaften Eindruck [11].

Im Gegensatz zu den weitreichenden kognitiven Fähigkeiten des menschlichen Gehirns im Bereich der Mustererkennung ist eine Maschine nicht in der Lage, eine solch komplexe Auswertung direkt vorzunehmen. Die Extraktion und Interpretation der Information, welche in einer Punktwolke indirekt vorliegt, stellt eine anspruchsvolle Aufgabe dar.

4.1 Stand der Wissenschaft und Technik

Für die Weiterverarbeitung von Punktwolken werden in der Literatur verschiedene computerorientierte Algorithmen angeführt. Diese Verfahren liefern als Ergebnis entweder ein flächen- oder volumenbasiertes Geometriemodell oder stellen Teillösungen auf dem Weg dorthin dar.

Während die meisten Verfahren eine Gesamtpunktwolke zugrunde legen, bei der die Standorte der Laserscanner nicht mehr zur Verfügung stehen, so gibt es auch Strategien, welche die Punkte jedes einzelnen Aufnahmeortes unter Berücksichtigung der Positionierung des Aufnahmegerätes verarbeiten. Ein übliches Vorgehen ist die stufenweise Entwicklung eines Oberflächenmodells, wie es in Abb. 15 dargestellt ist.

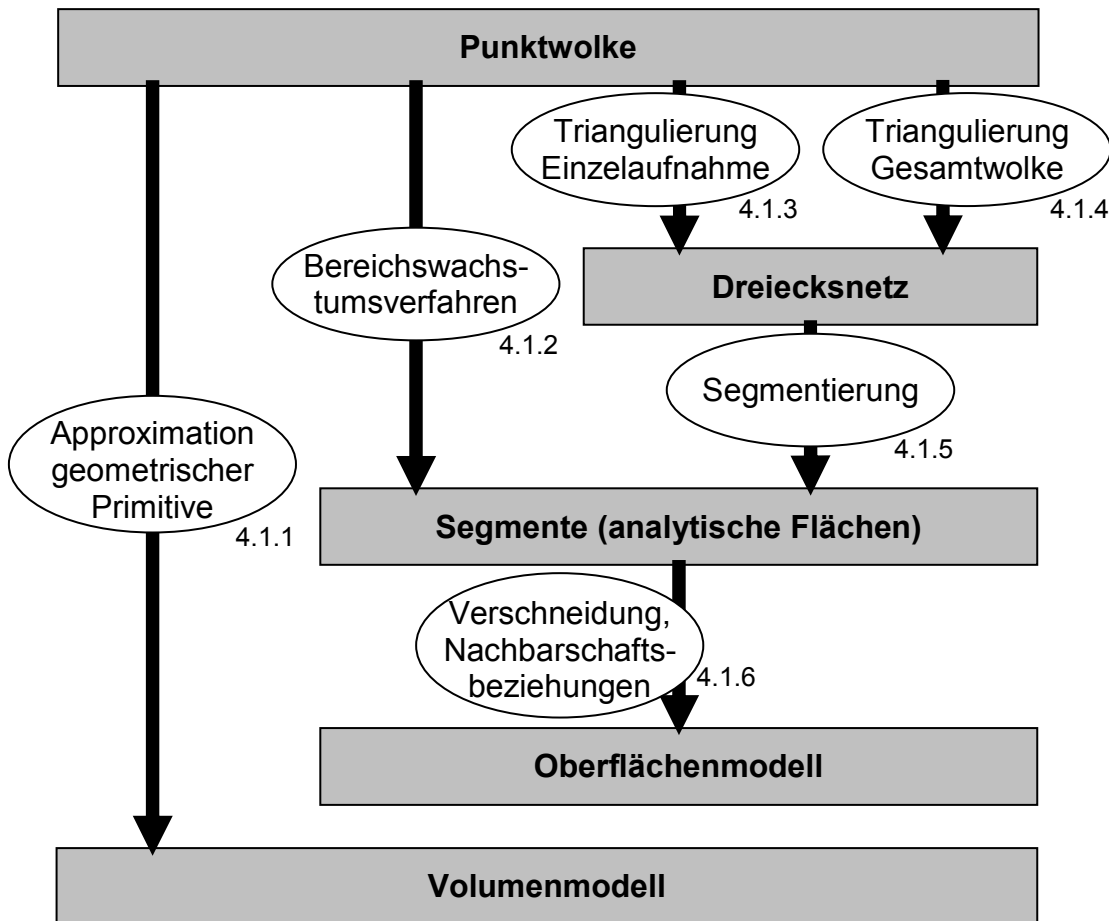


Abb. 15: Strategien zur Erzeugung eines Oberflächenmodells aus Punktwolken mit Angabe des Kapitels, in dem diese vorgestellt werden

4.1.1 Approximation durch ausgleichende geometrische Primitive

Ein halbautomatischer Weg ist die Selektion eines Bereiches der Punktwolke und die anschließende Zuordnung eines Objektes aus einem Katalog von Grundkörpern. Beide Schritte sind dabei durch einen Benutzer am Computer manuell durchzuführen.

Durch Lageänderung und Skalierung können Objekte in den spezifizierten Teil der Punktwolke eingepasst werden [79]. Am Markt angebotene Softwaresysteme, wie beispielsweise *Cyclone* von *Leica Geosystems* [49], verfolgen diesen Ansatz. Für die Approximation der Punktwolke durch standardisierte Körper, beispielsweise aus dem Stahl- oder Rohrleitungsbau, ist dieses Vorgehen wirtschaftlich praktikabel.

4.1.2 Bereichswachstumsverfahren

Um aus der Punktwolke direkt Oberflächen abzuleiten eignet sich das Bereichswachstumsverfahren [14]. Bei diesem Verfahren ist ein Saatpunkt zu

bestimmen, der zusammen mit seinen unmittelbaren Nachbarn eine Ebene aufspannt. Diese Nachbarschaft wird iterativ erweitert und aus der Menge der Punkte eine Ausgleichsebene errechnet. Dabei werden so lange Punkte aufgenommen, bis im Randbereich der Fläche keine Punkte mehr gefunden werden, die dieser Ebene unter Vernachlässigung eines Fehlers zugeteilt werden können. Ein günstiger Saatpunkt besitzt in seiner weiteren Nachbarschaft nur Punkte, welche mit einer geringen Abweichung annähernd in einer Ebene liegen. Liegt ein solcher vor, lässt sich mit hoher Wahrscheinlichkeit die ebene Fläche bis zu ihren Kanten bestimmen. Raue Oberflächen oder stark zergliederte Teilflächen behindern das Verfahren. Nachteile der Methode sind der hohe Rechenaufwand, der sich aus der wiederholten Berechnung der Ausgleichsebene ergibt, so wie die Notwendigkeit der Festlegung des Saatpunktes je Ebene. Letztere erfolgt in derzeit auf dem Markt erhältlichen Softwaresystemen durch den Benutzer [10].

4.1.3 Vernetzung von Punktwolken aus Einzelaufnahmen

Über die Erzeugung eines Dreiecksnetzes lassen sich topologische Zusammenhänge in der Punktwolke festlegen, welche eine gute Basis für die Weiterverarbeitung sind. Die Reduzierung des Problems auf Punkte, welche von einem einzelnen Standort aus ermittelt wurden, erleichtert die Aufgabe der Vernetzung. Ist der Aufnahmestandort selbst nicht bekannt, lässt sich aus der Sichtbarkeitsbedingung ein Bereich, in dem das Gerät positioniert war, ermitteln.

Über die Kenntnis des Aufnahmestandortes kann im gerätebezogenen Koordinatensystem jeder Messpunkt P_i mit seinen Polarkoordinaten, nämlich dem Horizontalwinkel u_i , dem Vertikalwinkel v_i und dem Abstand d_i im lokalen Koordinatensystem des Laserscanners dargestellt werden. So ist eine Projektion in den \mathbb{R}^2 als Plattkarte [82] unter Vernachlässigung des Abstands d_i zweidimensional möglich.

Die nun in lokalen zweidimensional Koordinaten dargestellten Punkte werden mit der Delaunay-Triangulierung vernetzt [41]. Handelt es sich um eine einzelne Aufnahme mit konstanten Winkelschrittweiten in Horizontal- und Vertikalrichtung, so ist eine Vernetzung auch mit einfachen Mitteln auf Basis der Matrixstruktur durchführbar.

Dreiecke mit annähernd parallelem Normalenvektor lassen sich daraufhin zu größeren ebenen Segmenten zusammenfassen. Kann von einem Aufnahmestandort aus das Aufnahmeobjekt vollständig erfasst werden, so lässt die Verschneidung der Segmente schließlich die Ermittlung von Eckpunkten und Kanten zu [41]. Andernfalls

sind weitere Verfahren notwendig, welche die Dreiecksnetze der verschiedenen Standorte vereinigen.

4.1.4 Vernetzung von Gesamtpunktwolken

Die Vernetzungen von Laserscandaten findet überwiegend mithilfe der Delaunay-Vernetzung (siehe Abschnitt 2.3.4), insbesondere als zweidimensionale Triangulierung [80], statt. Das erwünschte Ziel ist die möglichst präzise Approximation der ursprünglichen Oberfläche. Da eine Vernetzung nur topologische Verbindungen zwischen existierenden Punkten schafft und die Punkte statistisch gesehen ausnahmslos in den Flächen, nicht aber in Kanten oder Ecken erhoben wurden, ist es evident, dass durch die Triangulierung die Flächenbegrenzungen nicht korrekt rekonstruiert werden. Die Ermittlung derselben bleibt einem nachfolgenden Schritt vorbehalten.

Ein verbreitetes Vorgehen zur räumlichen Vernetzung ist, lokale Tangentialebenen an den Punkten zu erzeugen. Diese werden als Ausgleichsebene (Abschnitt 2.3.5) unter Einbeziehung einer zu definierenden Menge an Nachbarpunkten ermittelt. In dieser Hilfsebene findet eine lokale Vernetzung, üblicherweise eine Triangulierung, statt. Im Folgenden werden zwei Verfahren vorgestellt:

Triangulierungsverfahren nach Hoppe

Hoppe [33] verwendet zur Berechnung der Normalenrichtung der Tangentialebene alle Punkte, deren Abstand zum aktuell betrachteten einen maximalen Abstand nicht überschreiten. Dieser Punktabstand steht in Abhängigkeit zur Dichte und zum Rauschen der Messpunkte in der Gesamtpunktwolke. Diese beiden Parameter sind vom Benutzer anzugeben. Ein aufwändiger Schritt ist die nachfolgende Anpassung der Normalenorientierungen.

Auf Basis einer räumlichen Auflösung des Objektes in Voxel (siehe Abschnitt 2.3.1) findet die eigentliche Vernetzung mit einem Umrissverfolgungsverfahren (ähnlich der *marching cubes* -Methode [51]) statt. Über die Verschneidung der Tangentialflächen mit den Normzellen lassen sich Polygonzüge erstellen. Mit der Vereinigung der erzeugten Schnittflächen benachbarter Würfel entsteht ein Netz über die gesamte Oberfläche. Weil die Polygone nicht zwingend eben sind, wird zuvor eine Triangulierung vorgenommen [100].

Dieses Verfahren ist im Softwarepaket *The visualization toolkit* von *Kitware* implementiert [42]. Für eine effiziente Anwendung dieser Methode muss die Punktwolke eine passende Verteilung der Punkte besitzen. Allerdings wird die benötigte Rechenzeit für eine Vernetzung nach diesem Verfahren als hoch bewertet [91], was der folgende Algorithmus zu kompensieren versucht.

Triangulierungsverfahren nach Kós

Die von Kós [43] verwendete Methode zur Festlegung der Tangentialebene berücksichtigt den Abstand der Nachbarpunkte zum aktuellen Punkt. Im Gegensatz zum Verfahren der kleinsten Fehlerquadrate trägt man so stark gekrümmten Oberflächen in Randbereichen Rechnung. Als Nachbarpunkte wird eine in ihrer Anzahl $n \in \{25; 50; 100\}$ festgelegte Menge an nächstliegenden Punkten verwendet. Neben der Tangentialebene erhält jeder Punkt einen Nachbarschaftsgraph, welcher Punkte bis zu einem festgelegten Abstand, aus Effizienzgründen jedoch maximal 25, enthält.

Die Grundlage der an eine Delaunay-Triangulierung (siehe Abschnitt 2.3.4) angelehnten Vernetzung ist die Unterteilung eines imaginären Viereckelements: Für die Entscheidung, welche Diagonalen innerhalb eines Vierecks ABCD für die Erzeugung von zwei Dreiecken zu verbinden sind, werden die Innenwinkel ausgewertet: Die Punkte B und D werden verbunden, wenn $(\angle ABC + \angle CDA) > (\angle BCD + \angle DAB)$ gilt, sonst A mit C (Abb. 16).

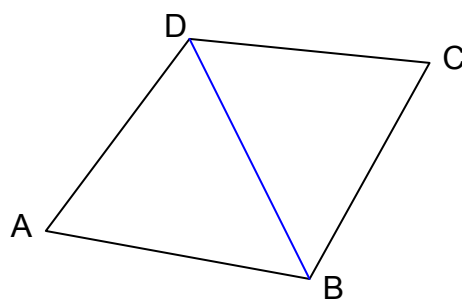


Abb. 16: Aus der größeren Winkelsumme $\angle ABC + \angle CDA$ gegenüber $\angle BCD + \angle DAB$ folgert die Erzeugung der Kante BD

Die Triangulierung arbeitet mit zwei Listen, einer Kandidatenliste und einer Ergebnisliste. Ein beliebiger Punkt dient als Startpunkt und wird mit A bezeichnet. Die Elemente der Ergebnisliste werden mit festgelegtem Umlaufsinn um die Ebenennormale von A mit B_1 bis B_k indiziert. Diese Liste ist zunächst leer, die

Kandidatenliste beinhaltet die in die Tangentialebene an A projizierten Punkte des Nachbarschaftsgraphen von A.

Der Algorithmus beginnt mit der Wahl des Punktes C aus der Kandidatenliste, welcher zu A am nächsten liegt. Für den Sektor B_iAB_{i+1} , in dem C liegt, folgt eine Überprüfung des Vierecks AB_iCB_{i+1} im vorgenannten Sinne (Abb. 17). Sind A und C zu verbinden, so wird C in die Ergebnisliste zwischen B_i und B_{i+1} eingefügt, andernfalls C für die weitere Betrachtung bezüglich A ignoriert.

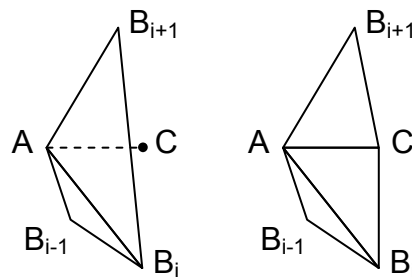


Abb. 17: Im Sektor B_iAB_{i+1} gelegener Punkt C erfüllt die Winkelbedingung und erzeugt die Dreiecke AB_iC und ACB_{i+1} (nach [43])

Sofern C nun Element der Ergebnisliste ist, werden dessen Nachbarn in die Kandidatenliste von A aufgenommen. Ferner folgt eine Überprüfung aller Punkte der Ergebnisliste: Ein Punkt B_j ist zu löschen, wenn dieser in einem der beiden Vierecke $AB_{j-1}B_jC$ oder ACB_jB_{j+1} nicht mit A aufgrund der Innenwinkel gemäß obiger Regel verbunden würde (Abb. 18).

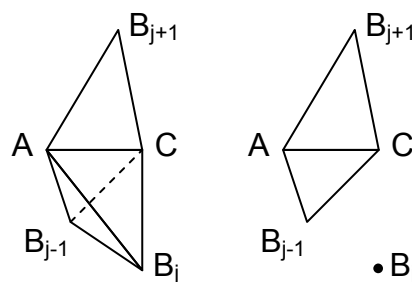


Abb. 18: B_j muss entfernt werden, da im Viereck $AB_{j-1}B_jC$ die Kante AB_j zu $B_{j-1}C$ umgeklappt wird (nach [43])

Dieses Vorgehen wiederholt sich, bis die Kandidatenliste leer ist. Jedes Dreieck, das aus der Ergebnisliste mit AB_iB_{i+1} erstellt werden kann, überträgt die Topologie auf die ursprünglichen Messpunkte. Ein solches Dreieck wird als Kandidatendreieck bezeichnet und erhält eine positive Bewertung vom erzeugenden Punkt A.

Nachdem um alle Punkte Kandidatendreiecke bestimmt wurden, können daraus die des endgültigen Netzes ausgewählt werden. Die Kandidaten werden entsprechend der ihnen zugefallenen positive Bewertungen absteigend, sowie bei Gleichstand anhand des Winkelunterschieds ihres Normalenvektors zum Mittelwert der Eckpunkt-Normalen aufsteigend sortiert. In dieser Reihenfolge werden sie dem endgültigen Netz zugeordnet.

4.1.5 Segmentierung in analytische Teilflächen

Die im vorhergehenden Abschnitt vorgestellten Verfahren erstellen ein Dreiecksnetz. Für die Weiterverarbeitung ist eine Zusammenfassung passender Dreiecke zu größeren Einheiten, den Segmenten sinnvoll. Segmente sollen aus zusammenhängenden Dreiecken hervorgehen und analytische Flächen repräsentieren.

Einfache analytische Oberflächen wie Ebenen, Zylinder- und Kugelflächen können im Oberflächennetz durch Tests erkannt und darin einbeschrieben werden. Ein Segment entsteht dadurch, dass passende Dreiecke, also solche, die eine zulässige Abweichung von der gewählten Form nicht überschreiten, zusammengefasst werden.

Prinzipiell ist es auch möglich passende Freiformflächen zu finden. Soll das Objekt jedoch nicht durch eine einzige, globale Freiformfläche dargestellt werden, welche alle vernetzten Punkte mit einbezieht, ist eine Festlegung für Segmentgrenzen zu treffen. Da die Messpunkte einem Messrauschen unterliegen, Freiformflächen sich jedoch beliebig genau anpassen, entsprechen die automatisch einbeschriebenen Flächen oft nicht dem gewünschten Ergebnis. Es erfordert daher der Unterstützung durch den Anwender zur Erzeugung von Oberflächen höherer Qualität [91].

Das Verschmelzen zu größeren Einheiten kann durch das Bereichswachstumsverfahren geschehen [101]. Ein Segment wird solange durch weitere Dreiecke vergrößert, wie diese einen zulässigen Abstandsfehler zur Ausgleichsfläche des Segments nicht überschreiten. Der Erfolg des Verfahrens hängt von der Auswahl des Startelements, der sogenannten Saatregion, ab. Bei der zufälligen Auswahl läuft man Gefahr, als Orientierung des Segments eine unglückliche Wahl zu treffen. In kommerziellen Anwendungen, die mit dem Bereichswachstumsverfahren arbeiten, muss deshalb der Saatpunkt für jedes Segment durch den Benutzer festgelegt werden.

4.1.6 Schnittkanten und -punkte zwischen Segmenten

Durch die Vermessung mit einem 3D-Laserscanner werden willkürliche Punkte auf der Objekt Oberfläche erhoben. Eine direkte Vermessung von Kanten oder gar Ecken entsteht daher nur rein zufällig. Jedoch dienen genau diese herausgestellten Punkt- und Kantenmerkmale der gängigen Beschreibung von Körpern oder allgemeiner Oberflächenstrukturen. Um nun diese als exakte Grenzen zwischen den einzelnen Segmenten zu ermitteln, müssen zwischen den Segmenten Nachbarschaftsbeziehungen ermittelt werden.

Zwischen Segmenten zu tatsächlich existierenden Oberflächen sind oft Segmente enthalten, die keinen realen Flächen entsprechen. Typischerweise werden diese Phantomsegmente durch eine im Verhältnis zu den erwünschten Segmenten deutlich geringere Anzahl an Dreiecken gebildet. Sie entstehen z. B. durch die Ausrundung an den Kanten und Ecken, bedingt durch die dort fehlende Information (Abb. 19). Diese ausrundenden Segmente müssen eliminiert werden, wobei die angrenzenden, größeren Segmente im Nachbarschaftsgraph direkt zu verbinden sind.

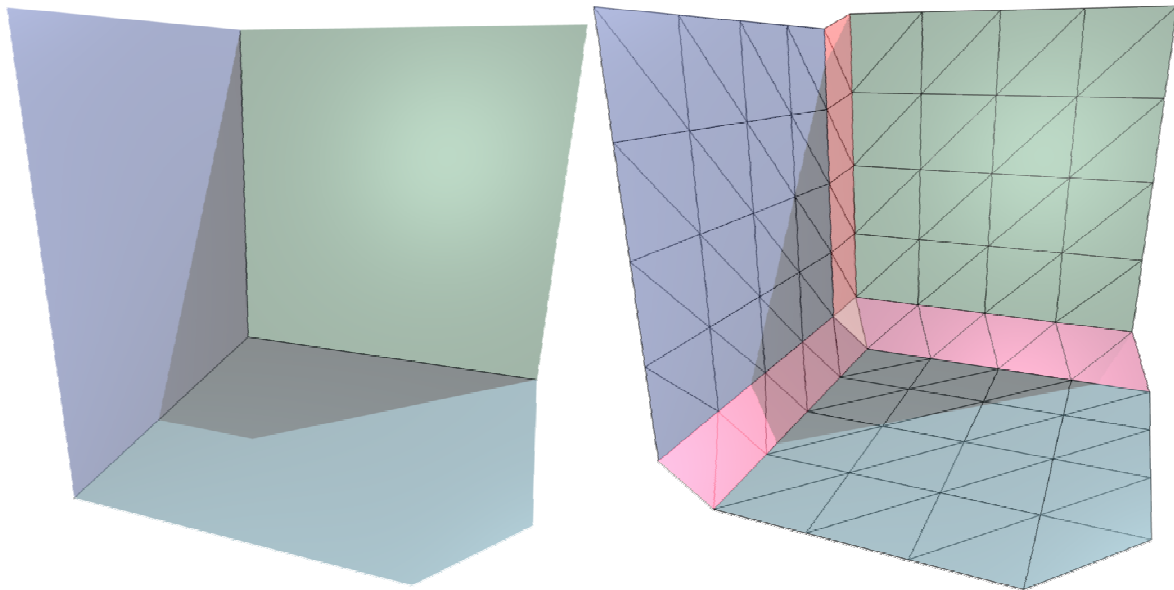


Abb. 19: Gewünschte Geometrie (links) und ausrundende Segmente in den Kanten und Ecken (rechts)

Durch die im Nachbarschaftsgraph hinterlegten Beziehungen lassen sich dann Zwei- und Dreifachschnitte erzeugen, die zu den gesuchten Kanten und Eckpunkten der realen Geometrie führen. Um durch Mess- und Interpretationsfehler ungültige Dreifachschnitte zu detektieren, kann die Determinante aus den drei Ebenennormalenvektoren ausgewertet werden: Liegt der Wert der Determinante

unter einem Grenzwert, beispielsweise wenn zwei der Ebenen annähernd parallel verlaufen, so wird der Schnitt für ungültig erklärt [41]. Die Überprüfung des Zwischenwinkels zweier Ebenennormalen erfüllt Gleiches bei einem Zweifachschnitt.

4.2 Überblick über das Vorgehen

Wenngleich es diverse Ansätze zur Rekonstruktion eines Oberflächenmodells aus Laseraufmaßdaten gibt, so existiert bislang kein Weg, der diese Aufgabe uneingeschränkt löst. Die vorliegende Arbeit greift das Verfahren der schrittweisen Rekonstruktion des Oberflächenmodells auf (Abb. 20) und konzentriert sich dabei auf die im Bauwesen weit verbreiteten ebenen Flächen.

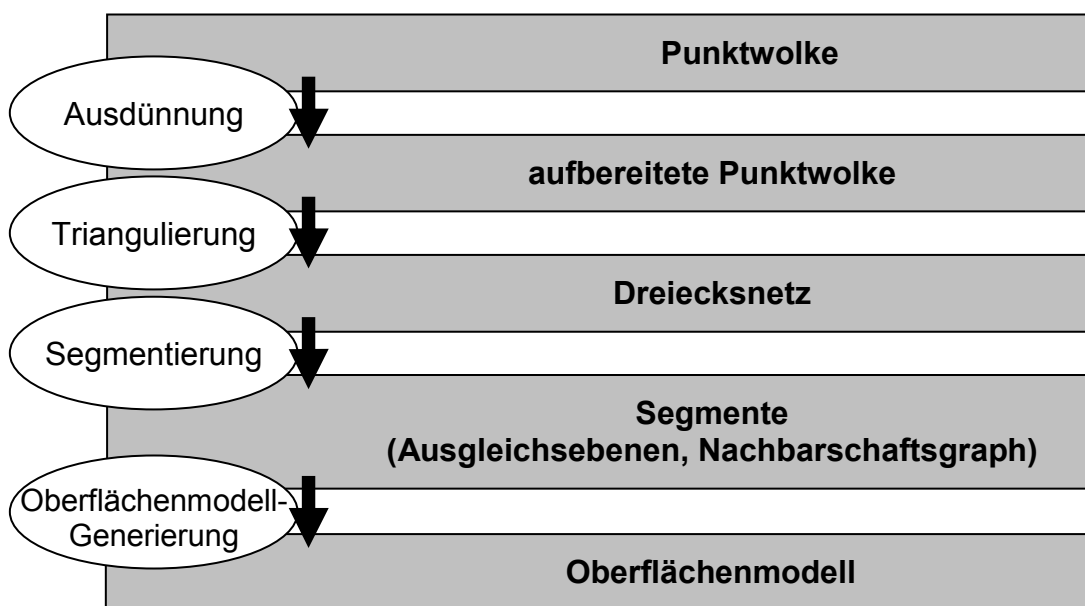


Abb. 20: Die vier Arbeitsschritte mit den zugehörigen Ein- und Ausgabedaten bzw. Zwischenergebnissen

Zunächst werden die Aufmaßdaten durch Ausdünnung der Punktmenge aufbereitet. Anschließend sind zwischen den Messpunkten Relationen zu erkennen, um ein Facettenmodell zu erzeugen. Dies erfolgt in Form einer Triangulierung und stellt die Basis für die folgende Segmentierung dar. Nebeneinander liegende Dreiecke, die eine ähnliche Ausrichtung ihrer Flächennormalen besitzen, werden in einem Segment zusammengefasst. Über Verschneidungen dieser Segmente lässt sich im abschließenden Schritt die gesuchte Oberfläche durch Schnittpunkte, -kanten und Flächen extrahieren [14].

4.3 Aufbereitung der Messdaten (Ausdünnung)

Für eine effektive Vernetzung ist es notwendig, eine gut konditionierte Punktwolke vorliegend zu haben. Wie in Abschnitt 3.3 erwähnt, kann durch die Planung geeigneter Aufstellorte des Messgerätes die Voraussetzung für eine passende Verteilung geschaffen werden. Dennoch führt das Abtasten von Flächen, welche in einem unterschiedlichen Winkel zum Laserstrahl stehen oder die verschieden weit vom Messursprung entfernt sind, zu einer inhomogenen Verteilung der Punkte. Werden die Aufmaßobjekte von unterschiedlichen Standorten aus aufgenommen, entstehen nach der Vereinigung der Einzelpunktwolken im Überlappungsbereich des Weiteren Gebiete einer ungewollt höheren Punktdichte.

Zu nahe nebeneinander liegende Punkte ergeben bei der Vernetzung in eigentlich ebenen Regionen eine übermäßige Oberflächenrauheit, weil die Punkte einem Messrauschen unterliegen. Je dichter die Punkte liegen, das heißt je feiner aufgelöst wird, desto mehr rückt die Messungenauigkeit des 3D-Laserscanners (siehe Abschnitt 3.2.4) in den Vordergrund [38]. Die redundante Information führt damit zu Widersprüchen. Als rau versteht man in diesem Zusammenhang, wenn in unmittelbarer Nachbarschaft liegende Punkte eine lokale Unebenheit bezüglich einer übergeordneten analytischen Fläche besitzen. Dies führt im Späteren zu einem Dreiecksnetz, in dem benachbarte Flächennormalen einen großen Richtungsunterschied aufweisen (Abb. 21). Eine zu dichte Punktwolke erfordert daneben einen großen Rechenaufwand im Vernetzungsschritt.

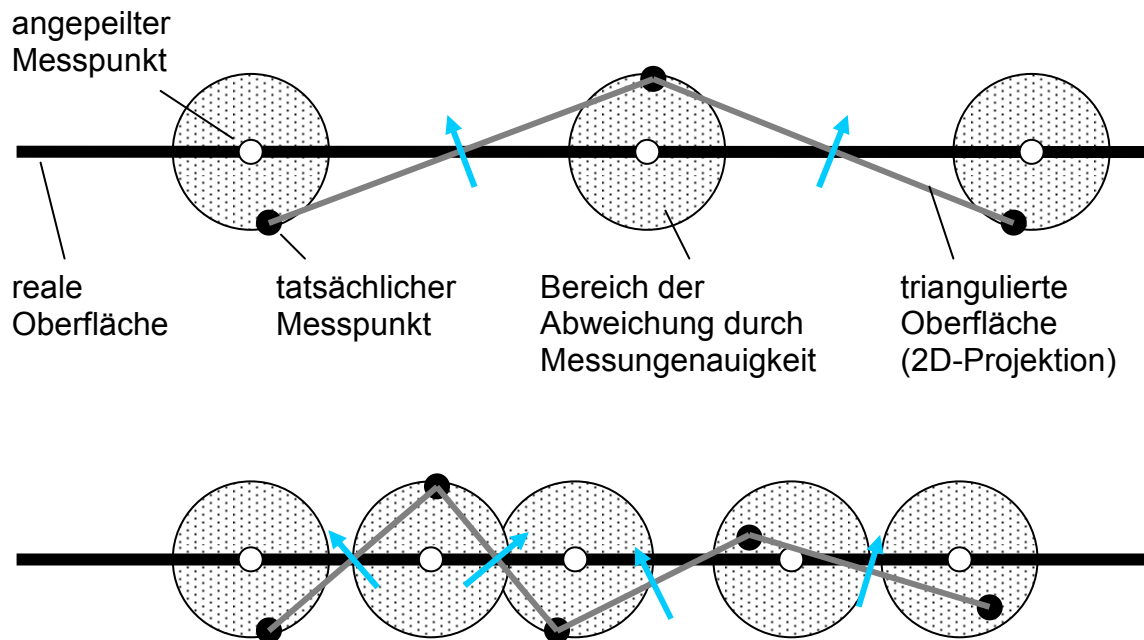


Abb. 21: Durch Messungenauigkeit verursachte Rauheit der triangulierten Oberfläche (oben, hier als Kanten projiziert, symbolisiert durch die Winkeländerung benachbarter Normalen), welche durch eine Verdichtung der Punktmenge verstärkt wird (unten)

Die Triangulierungsmethoden basieren darauf, dass räumlich nächste Punkte vernetzt werden. Sind auf einer Oberfläche erhobene Messpunkte anisotrop verteilt, so können Fehler in der korrekten Vernetzung entstehen. Eine Verdichtung oder Lichtung der Punktmenge muss deshalb entlang der Oberfläche isotrop sein und darf nicht sprunghaft erfolgen. Die Punkte, welche als Nachbarn bestimmt wurden, müssen auf der gleichen realen Oberfläche liegen und dort ebenfalls benachbart sein. An Kanten ist diese Bedingung nicht immer einzuhalten. Wie später noch gezeigt wird, können Falschvernetzungen an diesen Stellen jedoch geheilt werden.

Ein Einfügen von Punkten in Bereiche zu geringer Dichte ist nicht möglich, da keinerlei Information über die Topologie vorliegt. Deshalb muss beim Scannen ein Punktabstand gewählt werden, der jeden Bereich hinreichend genau auflöst. Anschließend ist die zur Verfügung stehende Gesamtpunktmenge soweit auszudünnen, dass die repräsentierte Oberfläche durch Punkte in einem einheitlichen Rasterabstand dargestellt wird.

Zur Ausdünnung wird die Methode der Ballungsanalyse (clustering) eingesetzt (siehe Abschnitt 2.3.3). Über den festzulegenden Grenzwert des Abstandes lässt sich bestimmen, wie detailliert die resultierende Punktmenge die Ursprungsgeometrie beschreibt. Sind alle Punkte anhand ihrer Koordinaten in dreidimensionale Cluster

eingeteilt, so lässt sich die Punktwolke konsequent ausdünnen. Für nachfolgende Schritte wird jedes Cluster durch einen Punkt repräsentiert. Um keine zusätzlichen Fehler zu erzeugen [91], wird dabei nicht der Mittelwert der Punkte einer Ballung weiterverwendet, sondern der Messpunkt, der diesem am nächsten liegt (Abb. 22).

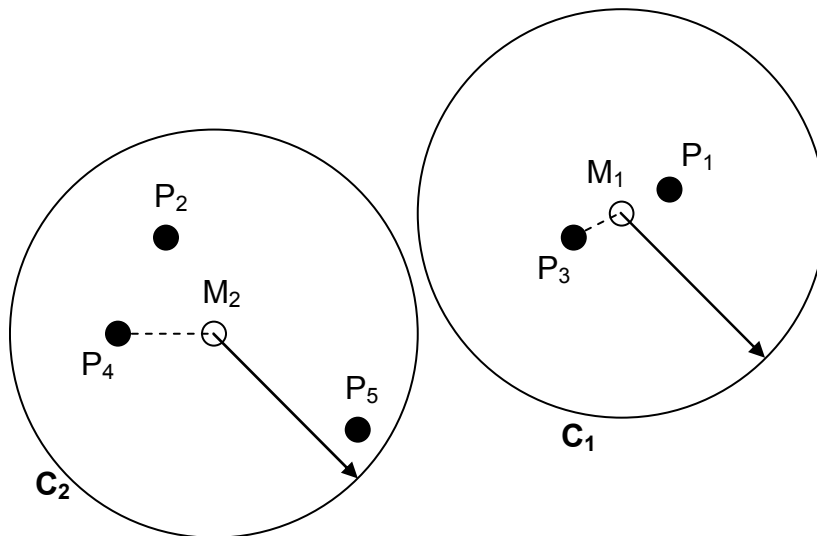


Abb. 22: Einteilung der Punkte P_1 bis P_5 in zwei Cluster C_1 und C_2 mit jeweiligen Mittelwert (Mittelpunkt) M_1 bzw. M_2 , anschließende Ausdünnung auf die Punkte P_3 und P_4 , welche zum jeweiligen Ballungszentrum den geringsten Abstand besitzen

Alternativ lässt sich die Punktwolke auch anhand einer Voxelmatrix (siehe Abschnitt 2.3.1) verkleinern. Die Kantenlänge der einzelnen Würfel definiert dabei die Ausdünnungsrate. Liegt in einer Zelle mehr als ein Punkt, kann der mit dem geringsten Abstand zum Würfelmittelpunkt beibehalten und alle anderen verworfen werden.

Die Methode der Ballungsanalyse bedarf einer größeren Rechenzeit, da iterative Optimierungen notwendig sind. Die Ausdünnung geschieht dabei jedoch flexibler und damit geeigneter als mit starren Gittern.

4.4 Dreiecksvernetzung der Messpunkte

Im \mathbb{R}^3 erzeugt der Delaunay-Algorithmus ein Netz aus Tetraeder-Elementen. Allerdings lassen sich lediglich konvexe Körper mit einem solchen Netz repräsentieren: Werden alle erzeugten Tetraeder zu einem Körper verschmolzen, so approximiert dessen Oberfläche die des vermessenen. Ist das Ziel die Suche nach einem konvexen Körper (bzw. der konvexen Hülle eines Körpers), so lässt sich diese ohne boolesche Vereinigung auch direkt [63] erzeugen.

Für nicht rein-konvexe Körper sowie bei komplexeren Szenen mit mehreren Objekten innerhalb einer Aufnahme ist aus der Tetraeder-Vernetzung keine Beschreibung der vermessenen Körper ableitbar – es lässt sich kein Innen und Außen, auch keine Oberfläche ableiten (Abb. 23).

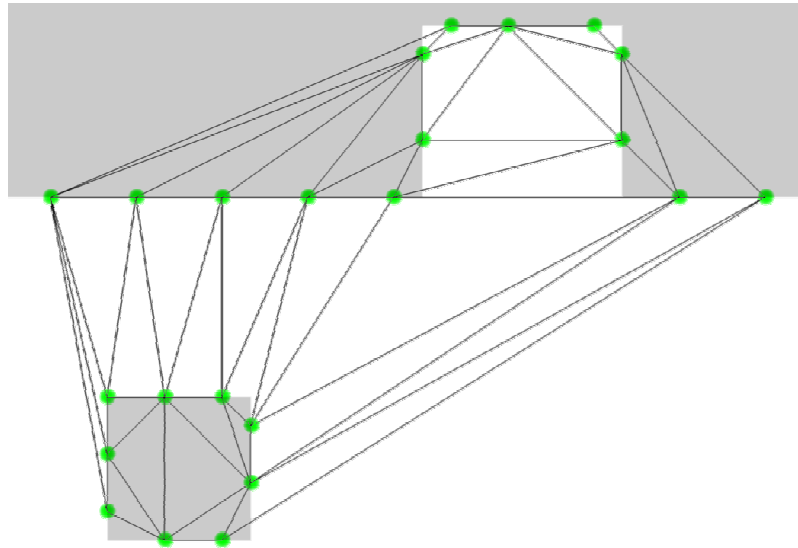


Abb. 23: Die Vernetzung (schwarz) der auf der Oberfläche von Wand und Stütze (grau) liegenden Messpunkte (grün), hier als Horizontalschnitt, veranschaulicht das unzureichende Ergebnis, aus dem sich aus den Tetraederflächen (hier als Kanten projiziert) nicht die Lage der ursprünglichen Oberflächen ableiten lässt

Da sich die vorliegende Problemstellung nicht auf konvexe Objekte beschränkt, ist es notwendig, Abwandlungen des Verfahrens einzuführen, die statt mit Volumen- mit Dreieckselementen arbeiten.

Unter dem Aspekt, dass sich im Bauwesen die Bauteile häufig entlang der Vertikalachse und der Horizontalebene orientieren, ist es in einem ersten semi-automatischen Ansatz denkbar, die Objekte vom Benutzer in der Grundrissansicht mit Linien vorskizzieren zu lassen. Dies zeigt Abb. 24 und folgende am Beispiel einer Wand. In der vertikalen Projektion der Punktwolke ist die Konstruktion dieser Linien gut realisierbar (Abb. 25, Abb. 26). Die in einer definierten Umgebung zu den Skizzenlinien liegende Punkte werden auf die Ebene, welche die Liniensegmente mit der Vertikalachse aufspannen (Abb. 27), projiziert. In diesen lässt sich der Delaunay-Algorithmus im Zweidimensionalen anwenden. Die ursprünglichen Punkte, welche als Quelle der 2D-Punkte über diese zu dereferenzieren sind, übernehmen die erstellte Topologie (Abb. 28). Die erwünschten Kanten und Eckpunkte lassen sich durch die Verschneidung der im nachfolgenden Schritt generierten

Ausgleichsebenen (Abb. 29) berechnen (Abb. 30). Gegenüber der vollständig manuellen Rekonstruktion muss ein Benutzer nur die grobe Ausrichtung der Flächen angeben.

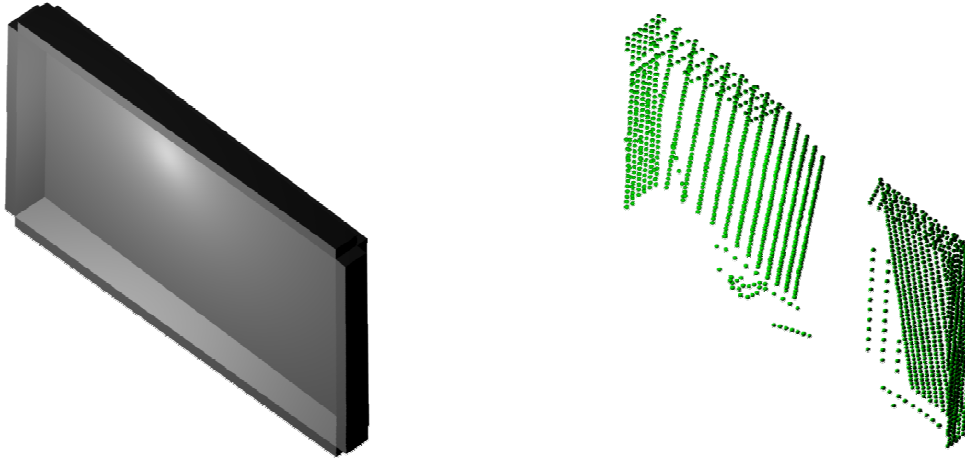


Abb. 24: Die aus dem gescannten Wandbereich erzeugte Punktwolke

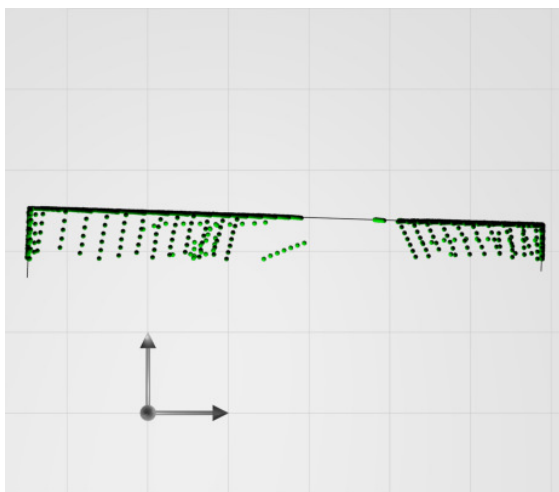


Abb. 25: Projektion der Punktwolke in die Grundrissebene

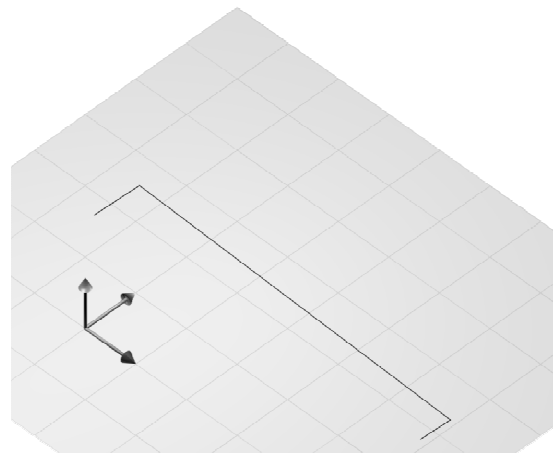


Abb. 26: Skizzenlinie in der Grundrissebene

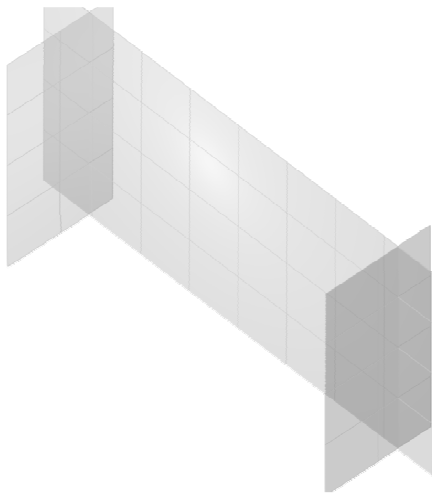


Abb. 27: Hilfsebenen zur lokalen
Triangulierung

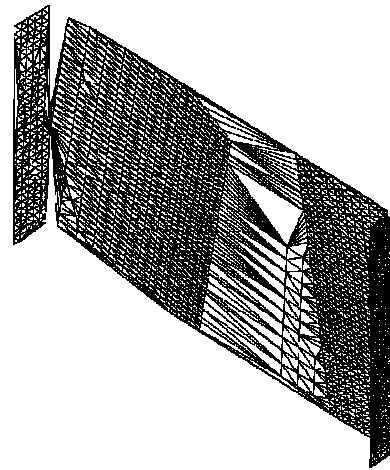


Abb. 28: Ergebnis der automatischen
Vernetzung

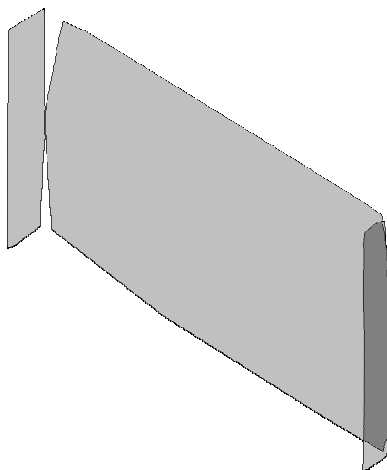


Abb. 29: Ausgleichsebenen und erkannte
Berandung

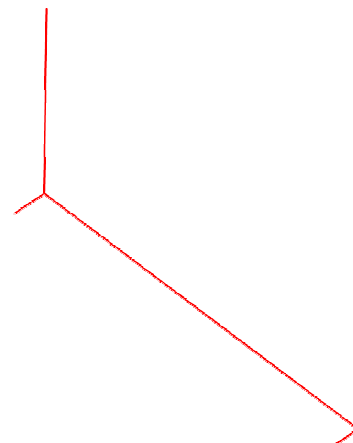


Abb. 30: Durch Verschneidung erzeugte
Kanten und Eckpunkte

Dreidimensionale Vernetzung

Die im Rahmen dieser Arbeit eingesetzte Lösung des räumlichen Vernetzungsproblems orientiert sich stark am ursprünglichen Delaunay-Ansatz, welcher im \mathbb{R}^3 als Vernetzungskriterium „leere“ Kugeln (frz. „sphère vide“ [23]) verwendet. Wie zuvor erläutert, ist die tatsächlich dreidimensionale Vernetzung in Tetraederelementen nicht zielführend. Als Ergebnis sollen deshalb Dreieckselemente entstehen (vgl. dazu Abschnitt 4.1.4). Die ursprüngliche Regel, die aus vier Punkten ein Tetraederelement erzeugt, wird hier auf drei Punkte und ein Dreieck reduziert.

Die Bedingung für die als Hilfsobjekt verwendete Kugel sei, dass – analog zur Tetraedervernetzung – die drei Punkte auf ihrer Oberfläche liegen. Der zur eindeutigen Definition der Kugel nun fehlende vierte Punkt wird durch die Vorschrift, der Kugelradius sei minimal, ersetzt. Durch diese Prämisse ist eine Kugel exakt konstruierbar, vorausgesetzt die drei Punkte fallen nicht in einem Ort oder auf einer Geraden zusammen: Die Punkte müssen auf einem Kreis liegen, dessen Mittelpunkt und Radius mit dem der Kugel identisch sind. Es kann also weiterhin mit Kugeln als Abstandskriterium gearbeitet werden, wobei deren Mittelpunkt und Radius aus den Dreieckspunkten zu bestimmen ist.

Der Mittelpunkt eines Kreises im \mathbb{R}^3 bestimmt sich aus drei auf dessen Kreislinie liegenden Punkte A, B und C wie folgt [31]: Es sind drei Ebenen zu erstellen, eine (E_1), welche die drei Punkte selbst aufspannen, und zwei weitere (E_2 und E_3) als Mittelebenen zwischen je zwei Punkten (Abb. 31). Diese Ebenen scheiden sich in genau einem Punkt, dem gesuchten Kreismittelpunkt M. Der Kreisradius r ist der Abstand von M zu einem der Punkte. Die Orientierung des Kreises gibt sein Normalenvektor, welcher zugleich auch Normalenvektor der Ebene E_1 ist, an. Dieser wird in diesem Zusammenhang aber nicht weiter benötigt.

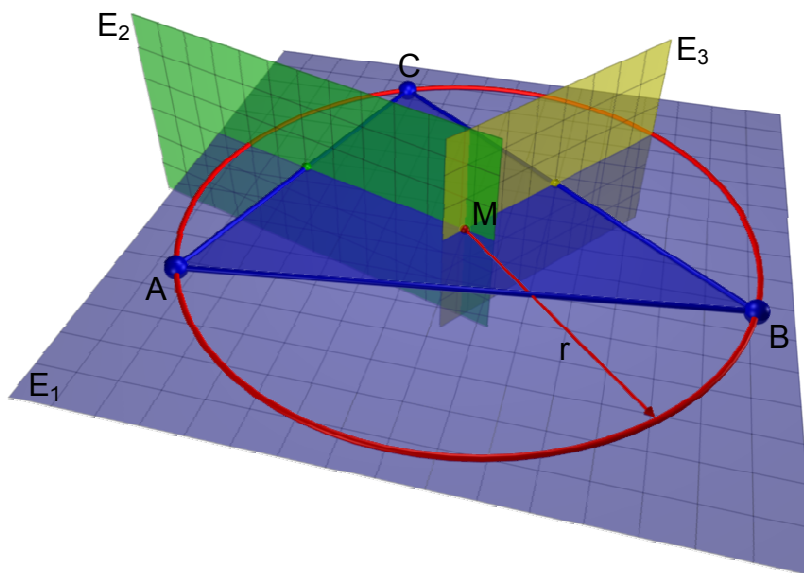


Abb. 31: Konstruktionsskizze Kreis in 3D

Ein Dreieck wird nun innerhalb der Punktwolke erzeugt, falls drei Punkte auf einer durch sie beschriebenen Kugel mit minimalem Radius liegen und sich keine weiteren Punkte innerhalb dieser Kugel befinden (siehe Abb. 47). Der Algorithmus kann in Anlehnung an das ursprüngliche Vorgehen als 2½-D-Delaunay-Vernetzung bezeichnet werden (Abb. 32).


```

Durchlaufe alle Oktaalbaumzellen
Suche alle Punkte der Zelle und der benachbarten Zellen
Für jedes Punkte-3-Tupel
  Erstelle Umkugel
  Wenn kein anderer Punkt innerhalb der Kugel liegt
    Erstelle Dreieck aus 3-Tupel

```

Abb. 32: Vernetzungsalgorithmus als Pseudo-Code

Beschleunigung des Verfahrens durch Diskretisierung des Teilraums

Aus der Erstellung aller möglichen 3-Tupel aus einer Menge von n Punkten und dem anschließenden Test dieser 3-Tupel gegen jeden anderen Punkt ergibt sich ein Berechnungsaufwand von $O(n^4)$. Es ist deshalb erforderlich, die Anzahl n an potentiellen Nachbarpunkten zu begrenzen. Dies ermöglicht eine Diskretisierung des Teilraums, innerhalb dessen sich die Punktwolke befindet. Die Diskretisierung wird über einen Oktaalbaum vorgenommen, welcher eine kubische Wurzelzelle besitzt und lokal soweit verfeinert wird,

- bis die Kantenlänge eines jeden Blattes einen vorgegebenen Wert unterschreitet (dies legt indirekt die Tiefe des Baumes fest),
- solange für eine Unterteilung eines Knotens gilt, dass eine festgelegte Anzahl an Kindzellen eine Mindestanzahl an Punkten besitzt, sowie
- bis ein Blatt aufgrund seiner Anzahl an Punkten nicht weiter unterteilt werden braucht.

Die drei Verfeinerungsregeln werden in dieser Reihenfolge abgearbeitet. Die darin enthaltenen Parameter sind vom Benutzer in Abhängigkeit der gegebenen Punktwolke anzugeben. Dabei muss die maximale Kantenlänge an den Skalierungsmaßstab der Punktwolke sowie die Punktdichte angepasst werden. Die zuvor durchgeführte Ausdünnung mit dem Ziel einer Homogenisierung der Punktdichte dient dem Zweck, die Oktaalbaumverfeinerung auf einer passenden Datengrundlage durchzuführen.

Findet eine zu starke Verfeinerung statt, stehen die Blätter isoliert im Raum und besitzen keine mit Punkten besetzten Nachbarn. Der Triangulierungs-Algorithmus ist dann nicht in der Lage, potentielle Kandidaten für Dreieckspunkte zu finden. Die Festlegung auf eine zu geringe Baumtiefe führt hingegen zu einer deutlichen Verlängerung der Rechenzeit. Ziel ist es, dünn besetzte Blätter entlang der zu

erwarteten Oberfläche aneinander grenzen zu lassen. Zwischen den Flächen sind leere Blätter auch aus algorithmischer Sicht sinnvoll, da dies eine irrtümliche Vernetzung dieser Flächen verhindert.

Die Umkugelbedingung wird so für jedes Punkte-3-Tupel geprüft, das sich aus den Punkten eines Oktalbaumblattes und dessen Nachbarblättern, auch diagonal, bilden lässt. Wegen der adaptiven Verfeinerung entsteht ein ungeglätteter Oktalbaum [21], das heißt zwei benachbarte Blätter können sich unabhängig voneinander auf verschiedenen Ebenen befinden. Folglich ist die Anzahl an Nachbarblättern beliebig. Das Verfahren der Identifizierung von Nachbarblättern ist aufwändig, der benötigte Rechenaufwand jedoch im Verhältnis zur Berechnung des Umkugeltests vernachlässigbar.

Optimierung der Vernetzung

Durch eine Nachlaufrechnung lässt sich das Ergebnis der Dreiecksvernetzung weiter verbessern: So können Dreiecke verworfen werden, wenn der maximale Innenwinkel einen Grenzwert überschreitet. Solch verzerrte Dreiecke treten auf, falls der Punktabstand in eine Richtung deutlich größer ist als in eine andere. Grenzen mehrere besetzte Oktalbaumblätter in unterschiedlichen Richtungen aneinander, entsteht eine solche Konstellation von Punkten. Im Bereich zwischen realen Flächen sind deshalb leere Blätter erwünscht, manchmal aber durch den Winkel der Flächen zum koordinatenachsenparallelen Hüllquader nicht realisierbar. Durch die Verwerfung verzerrter Dreiecke kann eine unerwünschte Vernetzung an diesen Stellen behoben werden.

Wird zur Modellierung des Dreiecksnetzes ein Knoten-Kanten-Flächen-Graph (vef-Graph) verwendet, lassen sich alle Kanten nach ihrer Zugehörigkeit zu Dreiecken untersuchen. Dadurch können Dreiecke identifiziert werden, die keine Entsprechungen in der Realität besitzen. Weist ein Dreieck eine nur einmal verwendete Kante auf, wird es näher untersucht:

- Wenn an keine seiner Kanten ein weiteres Dreieck stößt, handelt es sich um ein alleine stehendes und wird entfernt (Abb. 34 – A).
- Besitzt das Dreieck nur an einer Seite einen Nachbarn, entscheidet der Winkel zwischen den auf der gemeinsamen Kante senkrecht stehenden Vektoren, welche jeweils auf den dritten Punkt der Dreiecke zeigen (Abb. 33), ob es sich um ein herausstehendes Ende einer Fläche oder ein unerwünscht vernetztes Gebiet

handelt (Abb. 34 – B). Durch die Wahl dieser Vektoren kann die Orientierung der Dreiecke und somit der Normalenvektoren unbeachtet bleiben: Der Zwischenwinkel der beschriebenen Vektoren ist um 90° größer als der Winkelunterschied der Flächenausrichtung.

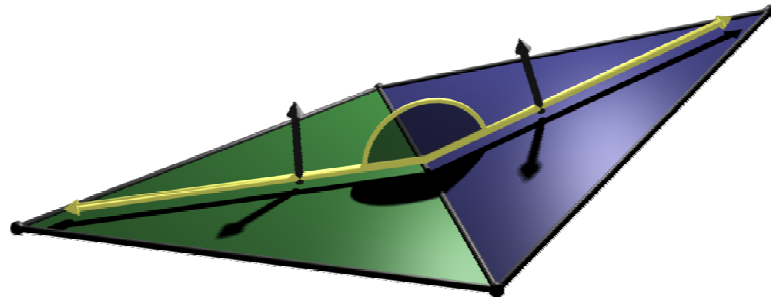


Abb. 33: Bestimmung des Zwischenwinkels zweier Dreiecke durch die von der gemeinsamen Kante wegzeigenden, aufspannenden Vektoren (gelb)

- Für Kanten, welche mehr als zweimal referenziert sind, werden die Richtungsvektoren zu den dritten Punkten der angrenzenden Dreiecke wie zuvor bestimmt. Aufgrund der Annahme, dass die gescannten Punkte von Flächen geringer Krümmung abstammen, werden die beiden Dreiecke beibehalten, die miteinander den größten Winkel aufspannen. Alle anderen Dreiecke sind zu verwerfen (Abb. 34 – C).

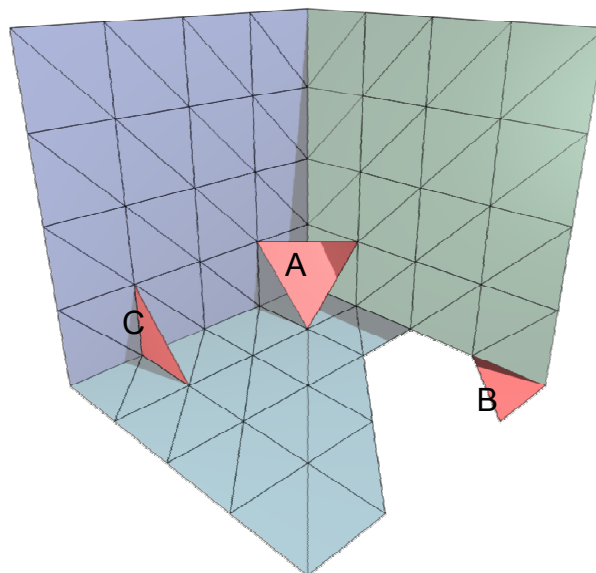


Abb. 34: Beispiele (rot) für in der Nachlaufrechnung identifizierte Vernetzungsfehler

Wegen der Änderung der Topologie, die sich durch das Löschen von Elementen ergibt, werden vorgenannte Betrachtungen solange iterativ durchgeführt, bis keine weiteren Veränderungen mehr auftreten (Abb. 35).

```

Wiederhole, bis keine Änderungen mehr auftreten
  Durchlaufe im vef-Graph alle Dreiecke
  Wenn mindestens eine Kante des Dreiecks nur zu einem
  Dreieck gehört
  Wenn Dreieck keine Nachbarn besitzt
  Oder nur ein Nachbardreieck existiert, zu welchem ein zu
  großer Normalenwinkelunterschied besteht
  Lösche Dreieck
  Wenn eine Dreiecksseite zu mehr als zwei Dreiecken
  gehört
  Lösche alle von der Kante referenzierten Dreiecke bis
  auf jene zwei, die zueinander den kleinsten
  Normalenwinkelunterschied besitzen

```

Abb. 35: Netzoptimierungsalgorithmus als Pseudo-Code

4.5 Segmentierung

Das erzeugte Dreiecksnetz repräsentiert zwar in einer ersten Näherung die Ursprungsgeometrie, ist jedoch als geometrische Planungsgrundlage nicht ausreichend. Ziel ist es nun, die Eckpunkte der vermessenen ebenen Bauteilflächen zu ermitteln. Da diese nur indirekt über die Verscheidung der rekonstruierten Ebenen zu erhalten sind, müssen die einzelnen Dreiecke zuerst zu größeren zusammenhängenden Bereichen vereinigt werden. Dieser Schritt wird als Segmentierung bezeichnet. Einem Segment ist dabei eine analytische Fläche zuzuordnen. Die Dreiecke werden so den Segmenten zugeteilt, dass diese mit einer vorgeschriebenen Genauigkeit der analytischen Fläche folgen. Beim Ersetzen einer facettierten Oberfläche durch größere Einheiten ist zu beachten, dass dies immer eine Veränderung, meist eine Vergrößerung der Geometrierepräsentation darstellt.

Es sei zunächst nochmals das Bereichswachstumsverfahren betrachtet. Im vorliegenden Fall handelt es sich um Facetten, die annähernd ebenen Flächen entsprungen sind. Eine andere Möglichkeit als die Verwendung der aufwändig zu berechnenden Ausgleichsebene ist, die Dreiecksnormalen zu untersuchen. Nachbarn, deren Normalen zu der des Startdreiecks eine nach oben hin beschränkte Winkelabweichung besitzen, sind zu einem Segment zu vereinigen. Dadurch bleibt, wie in Abschnitt 4.1.5 genannt, die Wahl des Startelements ausschlaggebend. Die

optimale Ausgangslage bietet ein Element mit der Flächennormale der ursprünglichen Oberfläche. Durch die Messungenauigkeit, der die Messpunkte unterliegen, kann auf die ursprüngliche Fläche nicht zurückgegriffen werden. Eine Alternative stellt die Normalenrichtung der Ausgleichsebene eines Segments dar. Bedauerlicherweise ist diese nicht bekannt, soll vielmehr hiermit erst rekonstruiert werden.

Diese Überlegungen führen zu dem Ergebnis, die schon in Kapitel 4.3 verwendete Methode der Ballungsanalyse einzusetzen. Aus den erstellten Clustern kann nicht nur die mittlere Normalenrichtung – sie repräsentiert jene der ursprünglichen Oberfläche – abgelesen werden, die Mitglieder einer Ballung stellen bereits ein Segment dar.

Für die ebenenweisen Ballungen sind nun die charakterisierenden Parameter festzulegen. Als Kriterium für eine Ebenenzugehörigkeit lassen sich der Normalenvektor jedes Dreiecks sowie der Abstandswert der Ebene, die das Dreieck beschreibt, zum Ursprung einsetzen. Zuletzt wird überprüft, ob die Segmente aus zusammenhängenden Dreiecksstrukturen hervorgehen, im gegenteiligen Fall findet eine Unterteilung in kleinere Segmente statt.

Im ersten Schritt werden die Dreiecke anhand ihrer Normalenrichtungskoordinaten gruppiert. Die Normalen sind dazu als Einheitsvektoren zu verwenden. In einer Gesamtpunktwolke steht keine Information zur Richtung des scannenden Laserstrahls mehr zur Verfügung, wodurch keine Orientierungsrichtung der Dreiecke festgelegt ist. Benachbarte Dreiecke können so entgegengesetzt gerichtete Normalenvektoren besitzen, obwohl beide Dreiecke in der gleichen Ebene liegen. Unter Verwendung der Ballungsanalyse würden diese Dreiecke nicht in einem Cluster zusammengezogen, da die Länge des Differenzvektors zwischen den normierten Dreiecksnormalenvektoren die in der Einheitskugel größtmögliche Distanz $d = \|\vec{n}_1 - \vec{n}_2\| = 2$ besitzt (Abb. 36). Ein Ausrichten der Vektoren in spezielle Oktanten ist nicht zielführend, da es an den Oktantengrenzen zu Sprüngen kommt. Die Orientierung muss demzufolge der tatsächlichen Oberflächenbeschreibung, beispielsweise vom Objekt weg in den Raum, folgen. Für einen isoliert stehenden Körper, dessen Oberflächen durch ein Dreiecksnetz beschrieben wird, ist in [34] ein entsprechendes Vorgehen beschrieben. Da es sich allerdings beim vorliegenden Netz nicht um eine gültige Hülle eines Körpers handelt, da Lücken und Klaffungen vorhanden, zum Teil auch nicht real existierende Flächen modelliert sind (siehe Abb. 34), ist eine Ausrichtung der Normalen unter diesem Gesichtspunkt hier nicht

möglich. Eine wie in [90] gezeigte Auftrennung der Objekte ist in diesem Stadium der Analyse mangels Kenntnis über die zu repräsentierende Topologie ebenfalls nicht realisierbar.

Ein Ausweg aus dem Dilemma der nicht gleich orientierten Vektoren ist, alle Dreiecksnormalen in ihrer ursprünglichen Richtung sowie als am Ursprung punktgespiegeltes Duplikat zu speichern (Abb. 37). Die normierten Dreiecksnormalen werden anhand ihrer Richtungskordinaten in Ballungen eingeteilt. Der Grenzwert für die Ballungszugehörigkeit wird aus dem nicht zu überschreitenden Zwischenwinkel zweier Vektoren als Länge des Differenzvektors in der Einheitskugel berechnet.

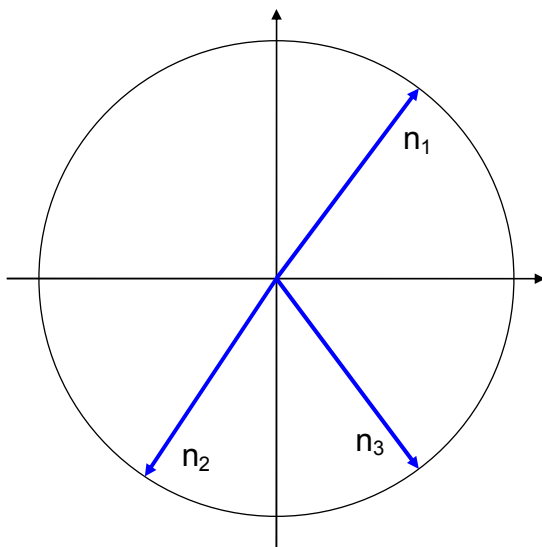


Abb. 36: In der Einheitskugel angetragenen Flächennormalen

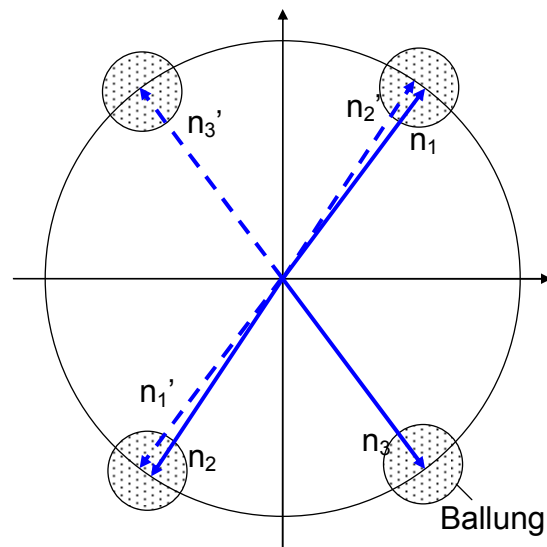


Abb. 37: Gespiegelt duplizierte Flächennormalen werden geballt

In der Menge der Ballungen lassen sich Paare bilden, die aus Elementen mit jeweils gegengleichen mittleren Normalenvektoren bestehen. Aus diesen Paaren wird jeweils eine Ballung entfernt. Innerhalb einer Ballung sind die Normalen nun homogenisiert, d. h. die durch sie referenzierten Dreiecke werden gegebenenfalls umgedreht (Abb. 38).

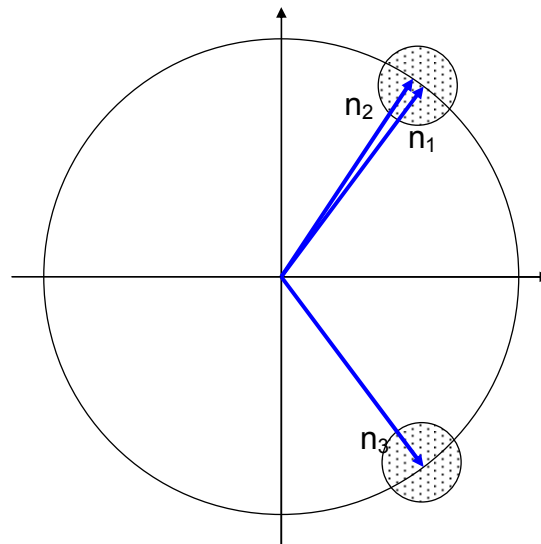


Abb. 38: Löschung jeweils einer Ballung aus Paaren mit gegengleichen mittleren Normalenvektoren aus Abb. 37, Anpassung aller ursprünglichen Dreiecksnormalen (hier: n_2 , vgl. dazu Abb. 36)

In den vorliegenden Ballungen sind die Dreiecke anhand ihrer Ausrichtung gruppiert. Innerhalb einer Ballung liegen die Dreiecke jedoch noch keinesfalls in einer Ebene. Diese notwendige Untergruppierung geschieht durch erneute Ballung. Diesmal ist der charakteristische Wert der Abstand, den jede Dreiecksebene zum Ursprung besitzt. Ein festgelegter Wert definiert den zulässigen Unterschied, den der Abstand eines Dreiecks zum Mittelwert der Gruppe besitzen darf, um nicht abgegrenzt zu werden. Um im konkreten Anwendungsfall dem Rauschen in der Orientierung der einzelnen Dreiecke, welches sich auf die Ebenengleichung und damit den Abstandswert auswirkt, Rechnung zu tragen, wird nicht der Abstand des Ursprungs zur Dreiecksebene bestimmt (Abb. 39). Es hat sich als praktikabler herausgestellt, den Abstand des Dreiecksschwerpunkts zur Ursprungsebene mit Orientierung gemäß dem Mittelwert der zu prüfenden Dreiecksnormalen zu prüfen (Abb. 40).

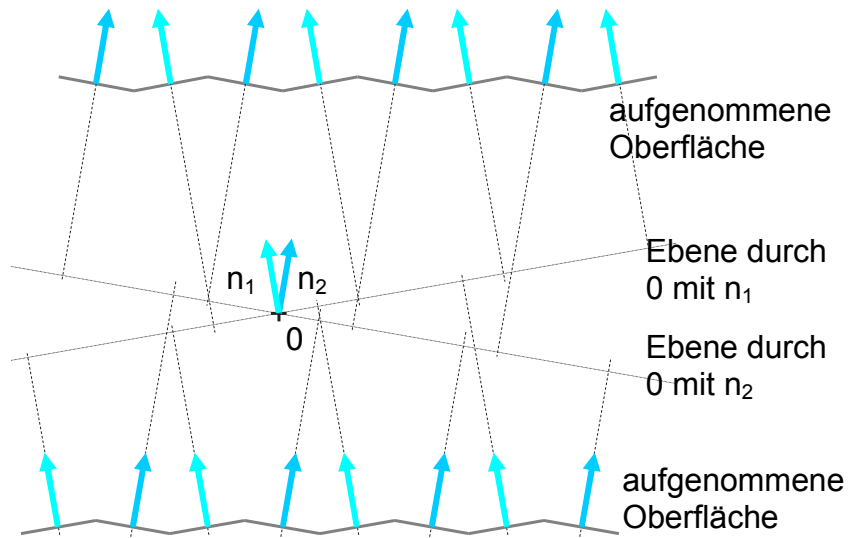


Abb. 39: Die Distanz der Facetten einer rauen Fläche zur Ursprungsebene ändert sich, wenn die Normalenrichtung der Ursprungsebene jener der Facetten entspricht

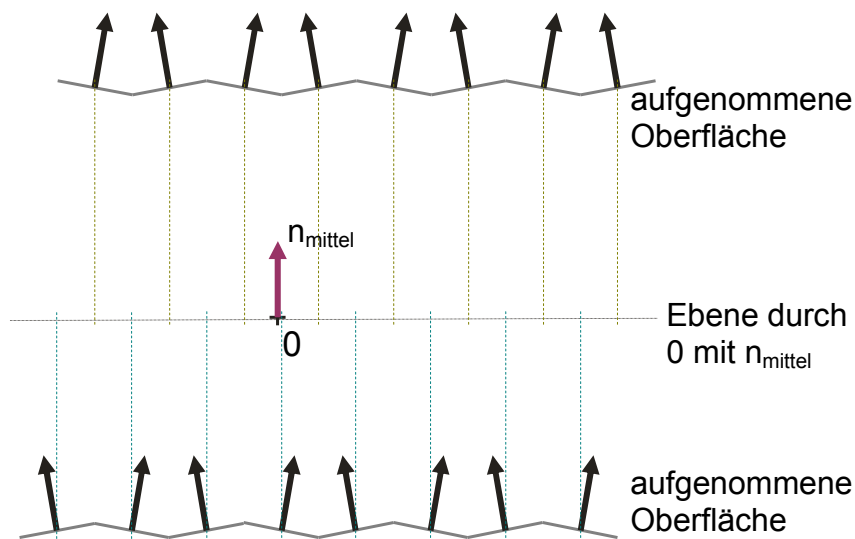


Abb. 40: Die Distanz der Facetten einer rauen Fläche zur Ursprungsebene ist annähernd gleich, wenn die Normalenrichtung der Ursprungsebene der mittleren Normalenrichtung der Facetten entspricht: Verschiedene Flächen können anhand der Distanz getrennt werden

Die Dreiecke jeder Ballungen liegen nun annähernd in einer Ebene. Als Nächstes findet eine Unterteilung der Ballungen in zusammenhängende Bereiche statt. Als Nachbarschaftsbedingung gilt hier, dass Dreiecke über Eckpunkte miteinander in Verbindung stehen müssen. Von einem beliebigen Dreieck einer Ballung ausgehend durchschreitet ein rekursiver Algorithmus alle unmarkierten Nachbardreiecke. Jedes erreichte Dreieck erhält eine Markierung, um ein erneutes Verwenden zu vermeiden. Sobald keine unmarkierten Dreieckselemente über gemeinsame Eckpunkte mehr

erreichbar sind, werden alle markierten Dreiecke zu einer Gruppe zusammengefasst und mit den verbliebenen das Verfahren wiederholt. Dies geschieht so lange, bis die Ballung keine unmarkierten Dreiecke mehr enthält.

Aus den gruppierten Dreiecken lassen sich im folgenden Schritt die Segmente ableiten. Segmente besitzen in diesem Kontext eine Ebene sowie Informationen über ihre Nachbarschaften. Als Ebene wird die in Abschnitt 2.3.5 vorgestellte Ausgleichsebene errechnet. Der Nachbarschaftsgraph der Segmente ergibt sich, indem sämtliche Dreiecksnachbarschaften, welche sich aus gemeinsamen Kanten ergeben, analysiert werden: Liegen zwei benachbarte Dreiecke in unterschiedlichen Segmenten, so stehen diese Segmente in nachbarschaftlicher Beziehung.

Optimierung der Segmentierung

Für nachfolgende Schritte ist es notwendig, die Segmentierung zu verbessern. Ausrundende Segmente (siehe Abb. 19) werden, wie in Abschnitt 4.1.6 beschrieben, aufgrund der Anzahl ihrer Dreiecke erkannt und eliminiert. Da die Anzahl an Elementen, die ein Segment besitzt, von der Punktdichte nach der Ausdünnung abhängig ist, ist die Mindestanzahl an notwendigen Dreiecken, also der Grenzwert für ausrundende Segmente, vom Anwender zu festzulegen.

Als weiterer Optimierungsschritt werden Nachbarsegmente, die ähnliche Parameter der Ausgleichsebenen besitzen, zu einem gemeinsamen Segment vereinigt. Somit vermeidet man die spätere Verschneidung von dicht nebeneinander liegenden, annähernd parallelen Ebenen unter einem zu flachen Winkel.

4.6 Schnittpunkt-, -kanten- und Regionsbestimmung

Die in den vorherigen Schritten bestimmten Segmente besitzen neben ihrer geometrischen Information Angaben zu den topologischen Beziehungen. Daraus lassen sich Verscheidungsverschriften zwischen den Segmenten folgern. Die ursprünglichen Eckpunkte und Kanten der vermessenen Oberflächen werden mit dieser Operation extrapoliert. Das Ergebnis ist ein Oberflächenmodell.

In diesem Oberflächenmodell ist die Geometrie ausschließlich durch die Koordinaten von Punkten festgelegt. Die Kanten sind topologische Elemente und referenzieren auf jeweils zwei Endpunkte. Die hier nur ebenen Oberflächen führen zu geraden Kanten. Die Flächen selbst sind durch Polygone vertreten, welche wiederum von den Kanten begrenzt werden. Die Punkte eines Polygons liegen jeweils in einer Ebene.

Daraus leitet sich das Vorgehen zur Erzeugung des Oberflächenmodells ab: Zunächst werden aus drei benachbarten Segmenten 3-Tupel und aus benachbarten Segmentpaaren 2-Tupel gebildet. Die 3-Tupel bestimmen die Schnittpunkte. Dabei kann gemäß 4.1.6 über die Determinante aus den Normalen der Ausgleichsflächen verhindert werden, dass sich die Ebenen in einem zu spitzen Winkel treffen. Zur Schnittgeraden, die aus den 2-Tupeln hervorgehen, werden passende Schnittpunkte gesucht. Können zwei Punkte gefunden werden, verweist die Kante auf diese als ihre Endpunkte. Sofern zwei benachbarte Segmente in mehr als zwei 3-Tupeln enthalten sind, existieren mehrere Schnittpunkte auf der Schnittgeraden. Die Schnittkante setzt sich dann aus mehreren Teilstrecken zusammen (Abb. 41).

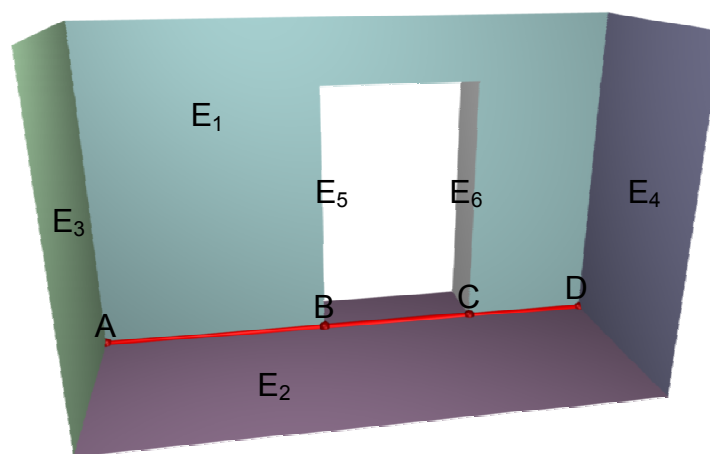


Abb. 41: Die Schnittkante zwischen den Ebenen E_1 und E_2 setzt sich aus den Teilstrecken AB, BC und CD zusammen, da außer den Ebenen E_3 und E_4 , welche die Kante begrenzen, auch die Ebenen E_5 (im Bild verdeckt) und E_6 mit den Ebenen E_1 und E_2 ein 3-Tupel bilden

Eine Fläche definiert sich in diesem Modell durch einen geschlossenen Polygonzug. Deshalb werden die Kanten eines jeden Segments untersucht. Lässt sich ein geschlossener Kantenzug erzeugen, so stellt die umschlossene Region eine Teilfläche des Oberflächenmodells dar.

4.7 Beispielanwendung

4.7.1 Oberflächenrekonstruktion als prototypische Implementierung

Die Implementierung des vorgestellten Algorithmus wurde als eigenständiges Programm vollzogen. Der Import der Laserpunktwolke erfolgt als Text-Datei mit den xyz-Koordinaten der Punkte. Die Scanner-Software, welche die Lasermessung steuert und innerhalb derer die Einzelpunktwolken miteinander verknüpft werden,

bietet dieses Dateiausgabeformat an. Das erzeugte Oberflächenmodell wird als Knoten-Kanten-Flächen-Graph (vef-Graph) gespeichert und ist beispielsweise mit dem kommerziellen CAD-Programm *Autodesk AutoCAD 2005* weiterverwendbar. Somit konnte ein lückenloser Informationsfluss vom Laseraufmaß zum CAD-Modell geschaffen werden.

Das Programm ist mit einer grafischen Benutzeroberfläche ausgestattet, worin die geometrische Visualisierung der dreidimensionalen Daten realisiert wurde. Dadurch ist es möglich, die Erstellung des Oberflächenmodells am Bildschirm in Einzelschritten mitzuverfolgen. Um dem Anwender möglichst viele Informationen an die Hand zu geben, können auch Hilfsgrößen eingeblendet werden. Der Anwender kann sich folgende Daten darstellen lassen:

- die ursprüngliche Punktwolke (Abb. 43)
- die ausgedünnte Punktwolke (Abb. 44)
- die mit Punkten der ausgedünnten Wolke besetzten Blätter des raumdiskretisierenden Oktaalbaums (Abb. 45)
- das Dreiecksnetz (Abb. 48) und die zu dessen Erstellung verwendeten Umkugeln (Abb. 47) und Kreise (Abb. 46), sowie die Dreiecksnormalen
- die unterschiedlich kolorierten Segmente (Abb. 49), die daraus erzeugten Ausgleichsebenen sowie der Nachbarschaftsgraph (Abb. 50)
- die Schnittpunkte, -kanten und Regionen des Oberflächenmodells (Abb. 51)

Die einzustellenden Parameter, welche über die Menüleiste konfiguriert werden, lassen sich auf diese Weise heuristisch anpassen. Es ist möglich, die triangulierte Punktwolke als Zwischenschritt abzuspeichern.

4.7.2 Parameterbestimmung

Für jeden Einzelschritt des in Abb. 20 vorgestellten Arbeitsprogramms können spezifische Parameter eingestellt werden. Damit lässt sich der Algorithmus auf die gegebenen Aufmaßdaten abstimmen und dadurch die Qualität des zu erzeugenden Oberflächennetzes deutlich verbessern. Im Folgenden werden praktikable Werte für die Parameter der Teilschritte vorgeschlagen.

Ausdünnung

- Ballungsradius

Die Ausdünnung des Datensatzes erfolgt durch die Ballungsanalyse. Der Ballungsradius gibt an, wie groß der zulässige Abstand von Messpunkten zu den Mittelwerten der bereits in Ballungen zusammengefassten Punkte sein darf. Aus jeder dieser Teilmengen wird ein Punkt beibehalten, alle anderen verworfen. Die Angabe der Distanz gibt somit an, wie die Vergrößerung des Messrasters ausfällt. Da die gescannten Bauteile mit einer Dichte von 10 cm noch hinreichend aufgelöst werden, eignet sich dieser Wert als Ballungsabstand.

Oktalbaum

- maximale Kantenlänge
- maximale Punkteanzahl
- Anzahl Kindzellen mit minimaler Punkteanzahl
- minimale Punkteanzahl für Kindzellen

Die Tiefe des Oktalbaumes legt den Suchbereich für potentielle Dreieckspunkte fest. Zu kleine Blätter isolieren die enthaltenen Punkte von Nachbarn, zu große führen neben einer Erhöhung der Berechnungszeit zu eventuellen Fehlvernetzungen, da Nachbarschaftsbeziehungen unterstellt werden.

Die maximale Kantenlänge legt indirekt fest, bis zu welcher Entfernung Punkte als Kandidaten für die Dreiecksvernetzung in Frage kommen. Dieser Wert muss größer sein wie der Ballungsradius für die Ausdünnung. Um nicht in Verbindung stehende Flächen voneinander zu trennen, sind leere Blätter zwischen ihnen erwünscht. Für sehr dünne oder schlanke Bauteile ist dies im Hinblick auf die vorgenannte Prämisse jedoch nicht immer realisierbar. Als Wert für die maximale Kantenlänge haben sich 25 cm als geeignet erwiesen.

Die Blätter des Oktalbaumes dürfen dünn besetzt sein, wenn ebenfalls besetzte Nachbarzellen in ausreichender Zahl vorhanden sind. Sind nicht mehr als ein gegebener Grenzwert an Punkten in einem Blatt gespeichert, wird diese nicht mehr weiter unterteilt. Ein passender Wert für die maximale Punkteanzahl sind drei Punkte.

Eine weitere Unterteilung findet wie in Abschnitt 4.4 dargestellt statt, wenn für potentielle Subzellen eine Mindestbesetzung durch Punkte vorhanden ist. Aufgrund geometrischer Überlegungen sollten mindestens vier Kindzellen mit Punkten besetzt sein. So ist gewährleistet, dass eine Verfeinerung in Ausdehnungsrichtung der Flächen keine Inselbildung zur Folge hat. Eine stärkere Bedingung als vier besetzte Kindzellen kann bei Wolken mit speziellen Punktconstellationen zu besseren Ergebnissen führen. Als minimale Punkteanzahl der neuen Blätter ist in den meisten Fällen ein Punkt ausreichend. Die Erhöhung dieser beiden Parameter führt zu einer geringeren Auflösung des Oktalbaumes und damit zu einer längeren Vernetzungszeit.

Triangulierung

- maximaler Innenwinkel
- maximale Normalenänderung für Randdreieck

Wie in Abb. 34 gezeigt können in einer Nachlaufrechnung falsch vernetzte Dreiecke identifiziert und entfernt werden. Stark verzerrte Dreiecke vernetzen häufig Punkte, welche nicht auf der gleichen Oberfläche erfasst wurden. Die Festlegung eines maximalen Dreiecksinnenwinkels von beispielsweise 135° deckt solche unpassenden Dreiecke auf. Als Randdreiecke werden solche bezeichnet, die nur an einer Kante mit anderen zusammenstoßen. Ist die Orientierung dieser beiden Flächen stark voneinander abweichend, liegt die Vermutung nahe, dass an einer Flächenkante ein nicht zugehöriger Punkt mitvernetzt wurde. Als Grenzwinkel zwischen den Normalenvektoren, ab dem diese Dreiecke zu entfernen sind, können 20° angenommen werden.

Segmentierung

- Dreieck aufnehmen bei maximaler Winkelabweichung
- Dreieck aufnehmen bei maximalem Abstandsunterschied
- Mindestanzahl an Dreiecken im Segment
- Segmente vereinigen bei maximaler Winkelabweichung
- Segmente vereinigen bei maximalem Abstandsunterschied

Die Segmentierung ist über die Ebenenparameter beeinflussbar. Die Werte maximale Winkelabweichung des Normalenvektors sowie maximaler Unterschied zum Ebenenabstandsparameter legen fest, wie groß die Abweichung der Orientierung eines neuen Dreiecks zum Mittelwert der im Segment vorhandenen Dreiecke sein darf. Wie sich in der Erprobung gezeigt hat, dürfen diese Werte nicht zu klein angenommen werden. Grund dafür ist die durchaus signifikante Oberflächenrauheit der vermessenen Flächen. So sind Winkelabweichungen bis 30° und Unterschiede bis 10 cm durchaus gewöhnlich.

Ein geeignetes Kriterium, um ausrundende Segmente zu identifizieren, ist die in Relation zu anderen geringe Anzahl an zugehörigen Dreiecken. Deshalb können Segmente verworfen werden, wenn sie eine Mindestanzahl an enthaltenen Dreiecken unterschreiten. Ein Grenzwert von 25 Dreiecken je Segment ist praktikabel, kann aber abhängig von der Objektgröße und zum Teil auch von der Strukturiertheit des ursprünglichen Objektes variieren.

Die Verschmelzung benachbarter Segmente zu einem bedient sich wieder der Ebenen beschreibenden Charakteristika Abweichungen vom Normalenwinkel und Abstandsparameter. Hierbei sollte der Winkel jedoch kleiner als zuvor, beispielsweise 5° , gewählt werden.

Verschneidung zur Schnittpunkt-, -kanten- und Regionsbestimmung

- minimale Determinante für Schnittpunktsuche

Der Nachbarschaftsgraph, der die Verschneidungs-Tupel bestimmt, wird aus den Nachbarschaften der Dreiecke an den Segmentgrenzen erstellt. Durch die vorgenannten Einstellmöglichkeiten repräsentiert die erzeugte Topologie weitgehend die gewünschte. Um jedoch die in Ausnahmefällen auftretende Verschneidung von Ebenen in einem zu flachem Winkel auszuschließen, was bei parallel verlaufenden Ebenen zu Singularitäten führen kann, wird über die Determinante aus den Normalenvektoren der Verschneidungswinkel abgeschätzt. Es hat sich gezeigt, dass es ausreichend ist, über den minimalen Wert der Determinante lediglich die Unlösbarkeit der Schnittpunktbestimmung bei parallelen Ebenen zu verhindern. So ist ein Wert von 0,01 genügend.

4.7.3 Beispiele

Nachstehend wird die praktische Anwendung des Algorithmus anhand von Beispielen demonstriert. Als Ausgangsbasis dienen Punktwolken aus realen Laseraufmaßen.

Stütze

Das erste Beispiel ist eine innerhalb eines größeren Raumes vermessene Stütze (Abb. 42). Das Objekt wurde aus drei unterschiedlichen Perspektiven abgetastet und die Einzelaufnahmen zu einer Gesamtpunktwolke verschmolzen. 5157 Aufmaßpunkte repräsentieren die Stütze (Abb. 43). Die Ausdünnung reduziert diese auf 510 Punkte mit homogener Verteilung (Abb. 44).



Abb. 42: Fotoaufnahme der vermessenen Stütze

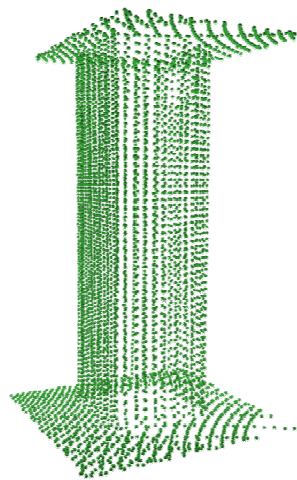


Abb. 43: Ungefilterte Punktwolke

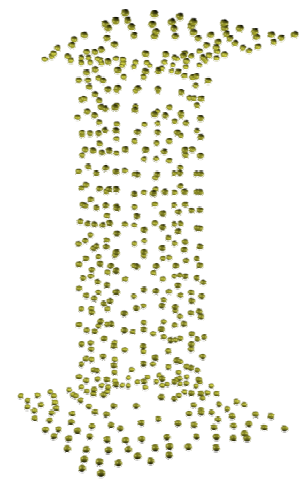


Abb. 44: Ausgedünnte Punktwolke

Die Grundlage für die Triangulierung stellt der Oktaibaum dar (Abb. 45), in welchem die ausgedünnte Punktwolke gespeichert ist. Die Triangulierung selbst beachtet die Umkugelbedingung, welche sich am Delaunay-Algorithmus anlehnt. Die Kugeln werden auf Grundlage von Kreisen erzeugt, die wiederum durch je drei Punkte gelegt werden (Abb. 46). Ein Dreieck ist gültig, wenn seine Eckpunkte auf solch einer Kugeloberfläche (Abb. 47) liegen und die Kugel keine weiteren Punkte beinhaltet.

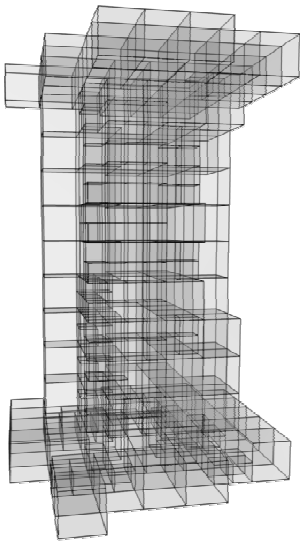


Abb. 45: Oktalbaum als
Basis der Suche
benachbarter Punkte

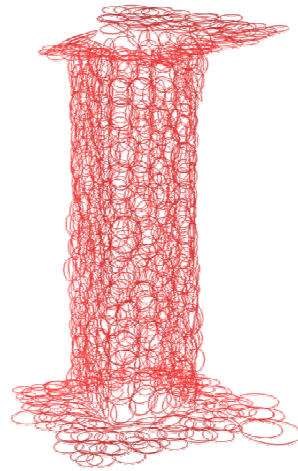


Abb. 46: Hilfskreise zur
Konstruktion der Umkugeln
(Abb. 47)

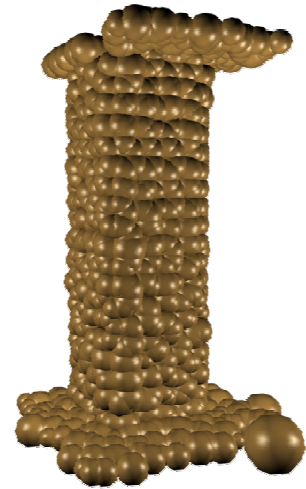


Abb. 47: Umkugeln zur
Prüfung auf gültige
Dreiecke

Unter dieser Bedingung entstand ein Dreiecksnetz mit 913 Elementen (Abb. 48), innerhalb dessen der Algorithmus sechs Segmente identifizierte (Abb. 49). Dabei sind die vier Seiten der Stütze, sowie die Boden- und Deckenebene durch je ein Segment vertreten. In der Explosionszeichnung deutlich zu erkennen sind die freien, weil nicht vermessbaren Bereiche im Boden- und Deckenanschluss. Die Verbindungen zwischen den einzelnen Segmenten, welche schließlich durch ausgleichende Ebenen ersetzt werden, stellt der Nachbarschaftsgraph dar (Abb. 50).

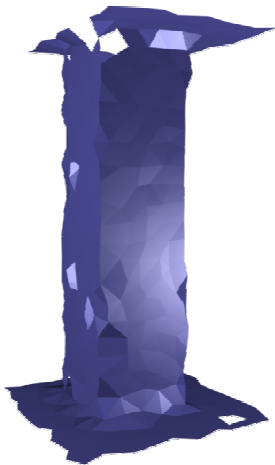


Abb. 48: Mit Dreiecken vernetzte Punktwolke

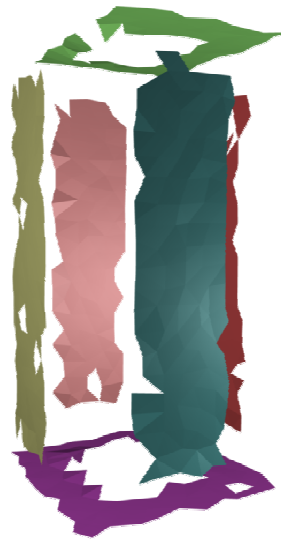


Abb. 49: Im Dreiecksnetz gefundene Segmente, hier farblich gekennzeichnet und als Explosionszeichnung visualisiert

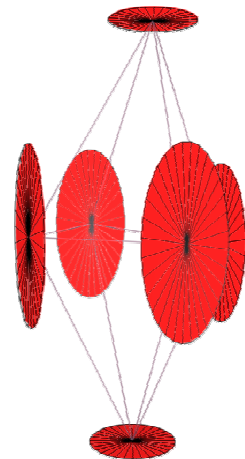


Abb. 50: Ausgleichsebenen und Nachbarschaftsbeziehungen der Segmente

In diesem bestehen Relationen zwischen den Seiten und der Boden- bzw. Deckenebene, wodurch sich die Eckpunkte und Schnittkanten korrekt ergeben. Neben den Seitenflächen, welche unten und oben korrekt an den jeweiligen Horizontalflächen enden, entstehen auch Flächen zwischen Stütze und Boden bzw. Decke (Abb. 51). Der Algorithmus, welcher Regionen über in einer Ausgleichsfläche liegende, geschlossene Kantenzüge erzeugt, prüft nicht, ob die umschlossenen Bereiche mit Messpunkten besetzt sind, also dort eine reale Fläche existiert. Die positive Konsequenz daraus ist, dass ein geschlossener Volumenkörper entsteht. Für alle Arbeitsschritte zusammen beträgt die Berechnungszeit etwa eine Sekunde (PC, Intel Pentium 4, 2,4 GHz).



Abb. 51: Das erzeugte Oberflächenmodell ergibt einen geschlossenen Körper, da die Decken- und Bodenplatte im Bereich der Stütze als abschließende Flächen erkannt werden

Wand

Als zweites Beispiel dient eine von beiden Seiten aufgenommene Wand (Abb. 52). Die Punktwolke, welche im Bereich der Türe feiner aufgenommen wurde, besteht aus 45703 Punkten (Abb. 56). Das Ziel der detaillierteren Messung war, die Türleibung zu erfassen. In der Punktwolke kann neben dieser auch das Türblatt identifiziert werden. Ein Ballungsabstand von 1 cm ist notwendig, um die Türleibung in der ausgedünnten Punktwolke zu erhalten. Unglücklicherweise führt diese Punktdichte bei Flächen, die in einem stumpfen Winkel ($> 45^\circ$) vermessen wurden, zu einem sehr rauen Dreiecksnetz. Wie in Abb. 21 gezeigt, wirkt sich bei sehr nahe beieinander liegenden Punkten die Messungenauigkeit signifikant aus, infolge derer eine raue Oberfläche modelliert wird. Die Segmentierung des rauen Netzes ergibt eine Zersplitterung in mehrere Hundert Elemente (Abb. 53), deren Verschneidung wiederum erwartungsgemäß zu keinem brauchbaren Oberflächenmodell führt, wenngleich ein Benutzer in der Ergebnismenge der Verschneidungskanten die erwünschte Struktur zu erkennen glaubt (Abb. 55).

Dieses Beispiel führt deutlich vor Augen, dass eine höhere Punktdichte nicht zwangsläufig zu einer höheren Genauigkeit, sondern im Gegenteil, zu schlechteren Aussagen führt. Um die Messungenauigkeit auszugleichen, wird an dieser Stelle vorgeschlagen, für die Segmentierung eine größere Toleranz zwischen den Dreiecksebenengleichungen zuzulassen. Damit sich dies nicht negativ auf Bereiche

geringerer Dichte auswirkt, ist die Verwendung eines größeren Toleranzbereichs räumlich zu beschränken. Ferner darf, durch die in Abschnitt 4.3 aufgeführte Problematik einer anisotropen Punkteverteilung begründet, eine Änderung der Dichte nicht sprunghaft erfolgen, sondern muss einen stetigen Übergang aufweisen. Diese Verbesserungen werden bei künftigen Aktivitäten zu integrieren sein.

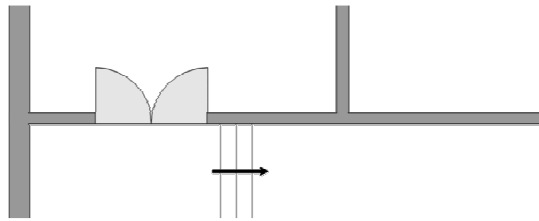


Abb. 52: Grundriss der gescannten Wand

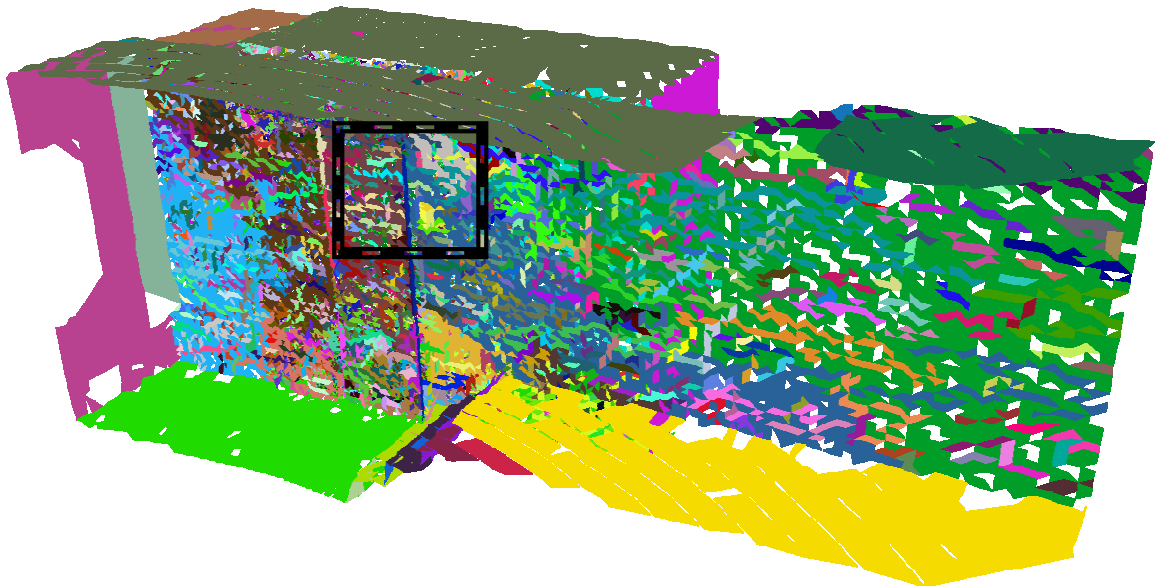


Abb. 53: Die mit hoher Punktdichte aufgenommenen Wandflächen führen zu einer großen Segmentanzahl



Abb. 54: Im vergrößerten Ausschnitt aus Abb. 53 ist zu erkennen, in welche kleine Segmente das Dreiecksnetz aufgeteilt wird: 99% der Segmente beinhalten weniger als 10 Dreieckselemente

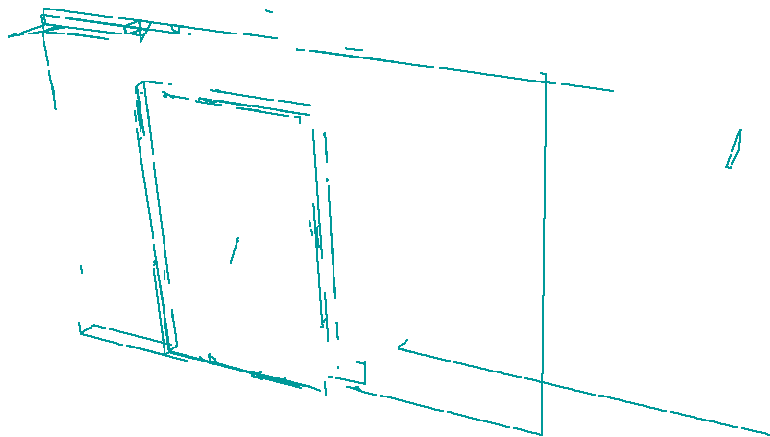


Abb. 55: Aus den vielen kleinen Segmenten ist kein für CAD-Zwecke taugliches Oberflächenmodell mehr generierbar, wenngleich von einem Menschen aus den gefundenen Kanten auf die aufgemessene Geometrie geschlossen werden könnte

Eine Ausdünnung auf das übliche Rastermaß von 10 cm liefert 2530 Punkte (Abb. 57). Diese lassen sich zu 4651 Dreiecken vermaschen (Abb. 58) und in 8 Segmente untergliedern (Abb. 59). Das resultierende Oberflächenmodell besteht aus den zwei Wandflächen sowie einer Stirnseite (Abb. 61). Der Grund für diese Modellierung findet sich im Nachbarschaftsgraph (Abb. 60): Da die Außenwand links, der Boden und die Decke für jeden Raum jeweils als getrennte Segmente erkannt wurden, kann keine Region an diesen Stirnseiten der Wand modelliert werden. Eine fälschliche Beziehung der senkrecht anschließenden Innenwand mit der abgewandten Fläche der vermessenen Wand lässt nicht nur diese Fläche an der gemeinsamen

Schnittkante enden sondern auch eine Stirnfläche dort entstehen. Den größten Anteil an der Rechenzeit (PC, Intel Pentium 4, 2,4 GHz) beansprucht in diesem Beispiel die Ausdünnung mit 58 Sekunden. Dies begründet sich in der iterativen Ausgleichung zwischen den 2530 Ballungen. Die Triangulierung erfolgt in weiteren sechs Sekunden, die Segmentierung in einer. Sowohl die Oktalbaumgenerierung als auch die Verscheidung der Segmente geschieht in Sekundenbruchteilen.

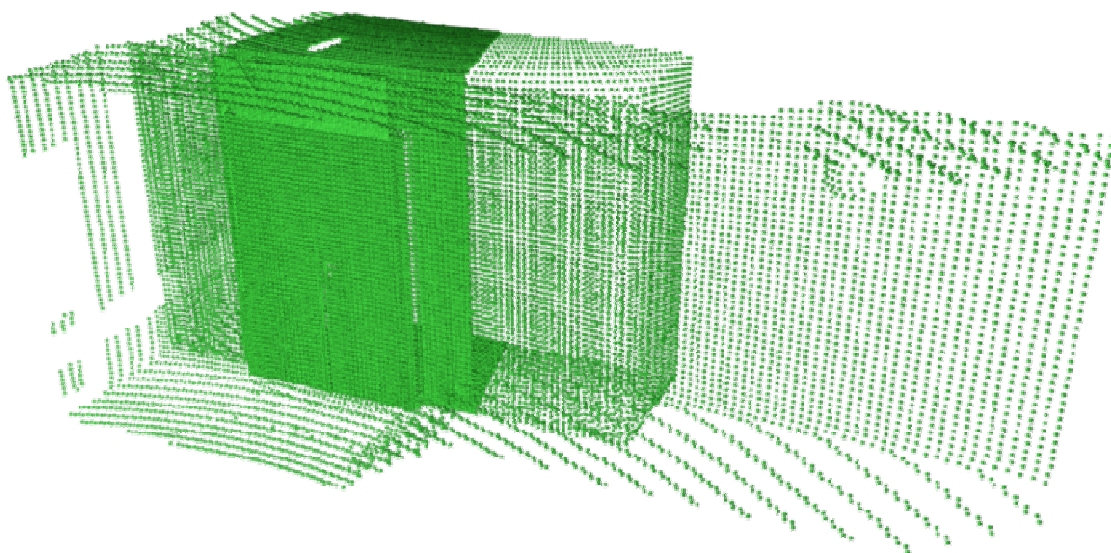


Abb. 56: Original Punktwolke



Abb. 57: Ausgedünnte Punktwolke

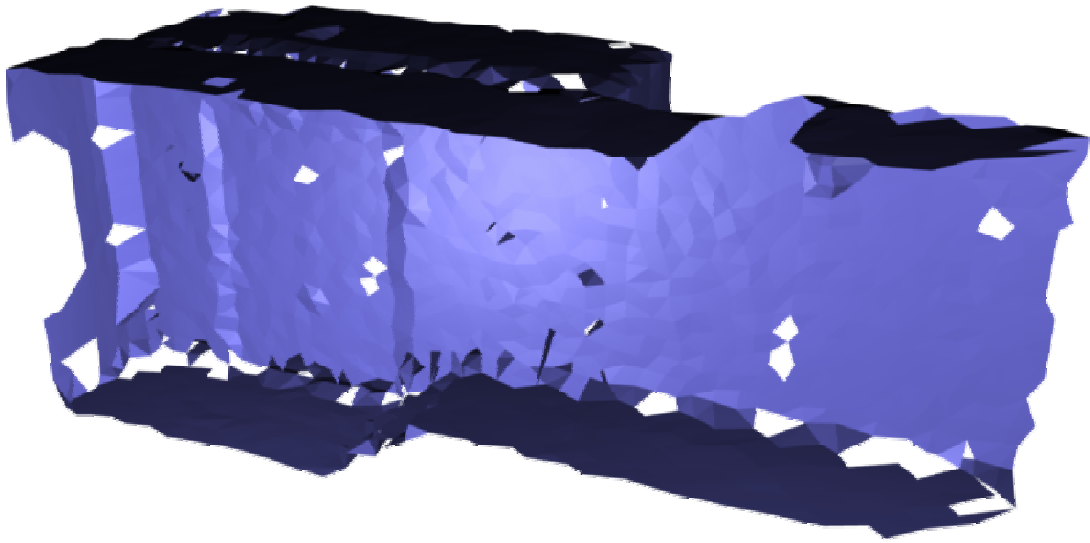


Abb. 58: Dreiecksvernetzte Punktwolke

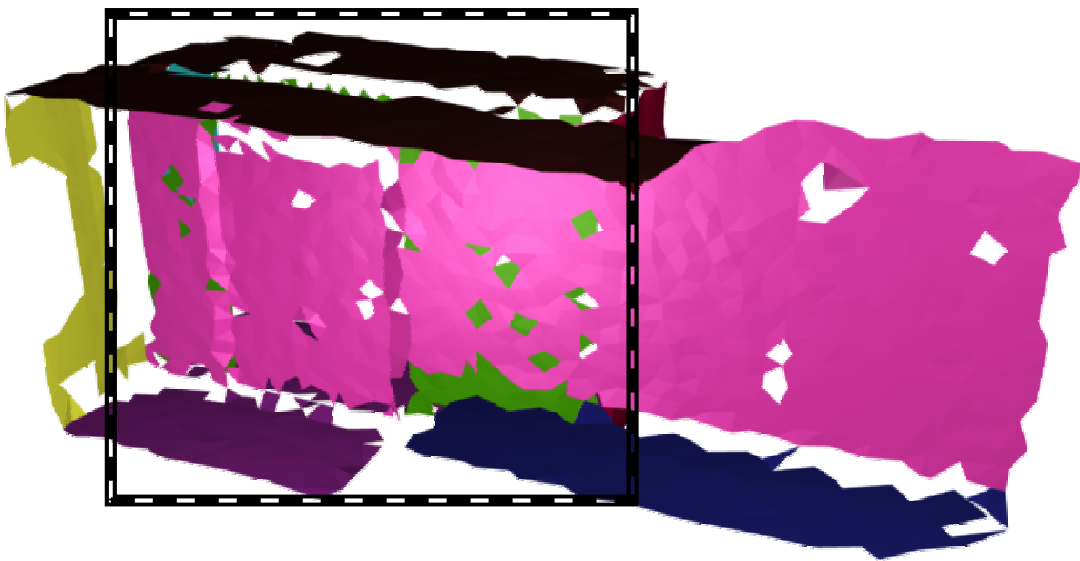


Abb. 59: Segmentierte Dreiecke mit markiertem Bereich des späteren Oberflächenmodells (Abb. 61)

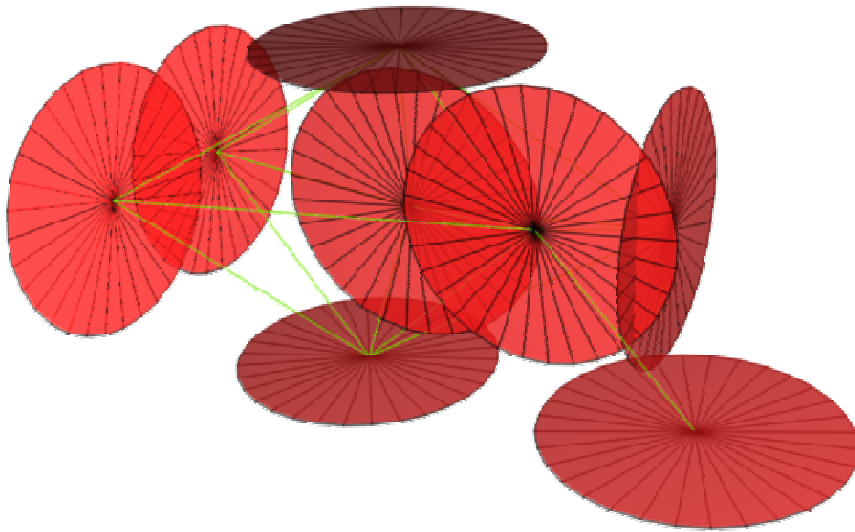


Abb. 60: Nachbarschaftsgraph

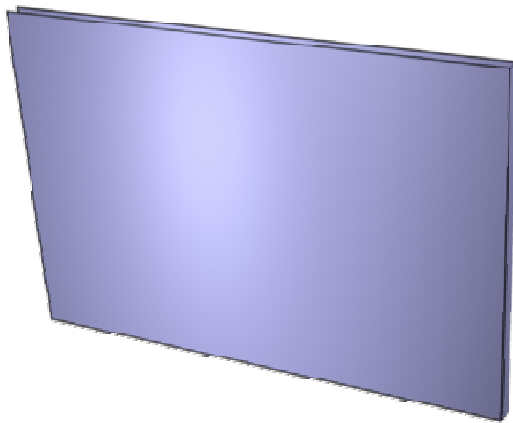


Abb. 61: Oberflächenmodell, bestehend aus der vorderen und hinteren Wandfläche sowie der rechten Stirnfläche

5 Virtuelle neuronale Netze

Die Fähigkeit, Daten in enormem Umfang zu speichern, ist eine herausragende Eigenschaft von Computern. Das Durchsuchen dieses Datenspeichers ist für explizit gegebene Kriterien durch die Maschine lösbar, Verfahren zur Beschleunigung der Suche wie die Indizierung von Einträgen optimieren den Zugriff. Soll der Computer aber nicht nur Daten, sondern Wissen speichern, so zeigt sich, dass dessen Formulierung und Abbildung innerhalb des Computersystems eine anspruchsvolle Aufgabe ist. Grund dafür ist, dass die Von-Neumann-Architektur [29], nach deren Prinzip heute gebräuchliche Computer arbeiten, ein selbstständiges Entwickeln von Wissen oder das Reproduzieren menschlicher Entscheidungsvorgänge nicht direkt unterstützt. Die deshalb notwendige Aufbereitung von Wissen in computergerechter Form ist eine diffizile Angelegenheit, da menschliche Entscheidungen nicht ausschließlich festen Regeln folgen, sondern sich auch auf Heuristiken und subjektive Empfindungen stützen. [84]

Die knappe Vorstellung dreier Formen der computergestützten Wissensrepräsentierung zeigt, welche unterschiedlichen Konzepte in einem Rechner zur Auswertung zur Anwendung kommen können: Expertensysteme, welche auf Basis von Regeln exakte Entscheidungen treffen, die Fuzzy-Logic, welche Zustände zwischen wahr und falsch kennt, und schließlich virtuelle neuronale Netze, die sich selbst Entscheidungsregeln von bekannten Mustern ableiten.

Expertensysteme sind ein Ansatz, Eingabeparameter maschinell auszuwerten und daraus eine Handlungsanweisung zu folgern. Es wird dabei vereinfacht davon ausgegangen, dass menschliche Experten aus einem Repertoire an schlichten Entscheidungsregeln schöpfen, um diese zu einem Gesamtkonzept passend zu verknüpfen und so komplexe Aufgaben zu lösen. Ein Computersystem bildet dies nun nach: Das Wissen eines Experten wird formalisiert als Regelsatz abgelegt. Dieser kann als das Wissen des Systems angesehen werden, welchem ein Mechanismus zur Anwendung dieser Regeln, die Problemlösungsstrategie, gegenübersteht. Bei dieser regelbasierten Methode bestimmt der Regelinterpretierer die Auswahl und sinnvolle Reihenfolge der Einzelanweisungen. Expertensysteme trennen somit Wissensbasis und Interpretierer. Eine sinnvolle Umsetzung erfolgt objektbasiert, wobei die Objekte Zustände annehmen, Informationen weitergeben und Anweisungen entgegennehmen können. Die Pflege eines so aufgebauten Expertensystems ist komfortabel und überschaubar. Dies steht im Gegensatz zu

einem anweisungsbasierten System, welches eine chronologisch fest vorgegebene Befehlsfolge besitzt. Die starre Struktur eines Entscheidungsbaumes ist ein Beispiel dafür. Da dort der sequenzielle Einsatz der einzelnen Entscheidungsregeln exakt festgelegt ist, stellt dies eine Verwebung von Wissen und Lösungsstrategie dar. Eine Erweiterung des anweisungsbasierten Algorithmus wird durch dessen verwobene Struktur gehemmt. [67]

Die Methode der Fuzzy-Logic führt eine neue Eigenschaft für Entscheidungen ein: Sie bedient sich unscharfer Aussagen [102]. Dieses Vorgehen motiviert sich aus der Beobachtung der Realität, dass Kenngrößen oft nicht exakt bestimmbar sind, weil sie zum einen einer Messungenauigkeit unterliegen oder andererseits über subjektive Empfindungen Auskunft geben. Die zweiwertige Logik, welche zwischen den Aussagen wahr und falsch unterscheidet, wird deshalb erweitert. Der in Analogie zu einem Interpolationsverfahren eingeführte, stetige Übergangsbereich zwischen den Extremwerten erlaubt vor allem die bessere Darstellung menschlicher Wahrnehmung. So bietet sich die Möglichkeit, komplexe Situationen, die in ihrem Umfang nicht ganzheitlich durch Parameter abgebildet werden können, mit einer gewissen Unsicherheit behaftet auszudrücken. Viele Faustregeln, die von Menschen aufgestellt werden, lassen sich nur durch ungenaue Beschreibungen in der menschlichen Sprache formulieren. Die Fuzzy-Logic ist hier in der Lage, diese vagen Zustände zu verarbeiten. Typischerweise stellt auch die Ausgabe der Fuzzy-Logic keine scharfen Größen dar. Es ist damit möglich, auf die Ausgangssituation angemessen zu reagieren, was sich in der Praxis insbesondere in der Maschinensteuerung bewährt hat. Um mit unscharfen Mengen operieren zu können, sind zunächst die Rechenoperationen zu definieren, welche auf diese Mengen angewendet werden. Deren konkrete Festlegung wird dem jeweiligen Anwendungsfall angepasst. [53]

Während bei den beiden vorgenannten Verfahren das Entscheidungsverhalten durch einen Regelsatz festzulegen ist, handelt es sich bei der Methoden der virtuellen neuronalen Netze um ein lernfähiges System. Insbesondere für Anwendungen, bei welchen der Zusammenhang zwischen Eingangsparametern und Ergebnis nicht hinreichend bekannt ist, lassen sich so Entscheidungsregeln trainieren, ohne dass diese explizit formuliert sind. Ein solches System lernt, indem es Eingabedaten aus einer vorhandenen Trainingsdatenmenge verarbeitet und die Berechnungsergebnisse den Sollausgaben gegenüberstellt. Dieses Lernprinzip wird überwachtes Lernen genannt, da das virtuelle neuronale Netz die eigene Entscheidung mit der vom Anwender erwünschten überprüfen kann und seine internen Entscheidungs-

regeln darauf anpasst. Aus diesen intern angelegten Berechnungsvorschriften kann ein Mensch allerdings keine Entscheidungsregeln ableiten, das erlernte Wissen bleibt im nach dem black-box-Prinzip arbeitenden System verborgen [69]. Ziel ist es, ein generalisierendes Verhalten zu erreichen, wodurch während der Nutzungsphase eine adäquate Auswertung neuer, dem Netz unbekannter Eingabedaten möglich ist.

Virtuelle neuronale Netze werden auch künstliche neuronale Netze genannt. Für Klassifizierungsaufgaben, insbesondere in der Mustererkennung, werden sie bereits seit langem erfolgreich eingesetzt [104]. Dieses Kapitel stellt künstliche neuronale Netze in ihrer Arbeitsweise vor. Durch zwei einfache Beispiele, die jeweils einen anderen Gesichtspunkt herausstellen, wird diese näher erläutert. Das erste Beispiel behandelt die Klassifizierung dreier unterschiedlicher geometrischer Formen: Kreise, gleichseitige Dreiecke und Quadrate. Die einzelnen Objekte sind in verschiedenen Größen ausgeprägt und sollen anhand ihres Typs in Klassen, sogenannte Objektklassen, eingeteilt werden. Im zweiten Beispiel sind verschiedene Farben zu analysieren und auf ihre Zugehörigkeit zu gelb zu prüfen. Die Eigenschaft gelb wird dabei durch Muster vorgegeben.

5.1 Aufbau

Virtuelle neuronale Netze bestehen aus einzelnen „Zellen“, die miteinander vernetzt sind. Ihr Vorbild finden sie in der Biologie (Abb. 62): Die Arbeitsweise des Nervensystems besteht grundsätzlich darin, dass einzelne Nervenzellen ihre elektrischen Eingangsreize über die Dendriten im Zellkörper auswerten (Abb. 63). Diese Eingangsreize werden dabei verschieden gewichtet und im Zellkörper aufsummiert. Überschreitet die Summe der Signale einen Schwellwert, so gibt die Zelle über ihren Ausgang, das Axon, einen elektrischen Impuls weiter. [56]



Abb. 62: Mikroskopaufnahme einer Nervenzelle des Rückenmarks (mit freundlicher Genehmigung von [61])

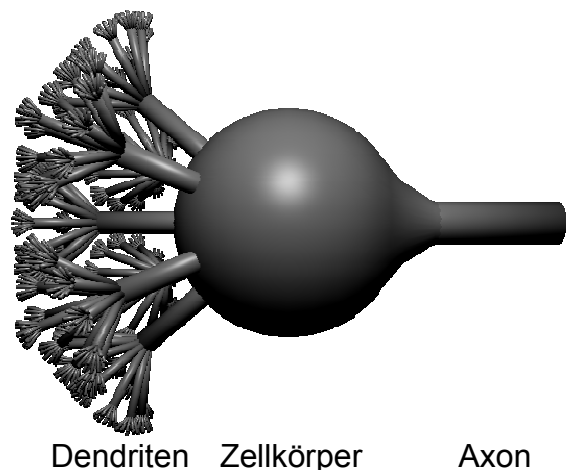


Abb. 63: Schemazeichnung einer Nervenzelle

Entsprechend diesem Vorbild werden auch künstliche neuronale Netze aus einfachen, gleichartigen Elementen konstruiert. Das in der Informatik verwendete Analogon zur Nervenzelle ist das künstliche Neuron (Abb. 64). Seine Eingangssignale (x_1 bis x_n) können durch Gewichte (w_1 bis w_n) verstärkt oder abgeschwächt werden. Eine definierte Entscheidungsfunktion, zum Beispiel die Schwellwertfunktion (5.1) bildet diese Eingangsgrößen unter Berücksichtigung eines Schwellwertes auf eine Ausgabe ab.

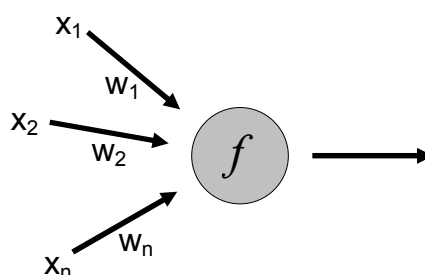


Abb. 64: Schemazeichnung eines künstlichen Neurons

Den ersten Ansatz für künstliche Neuronen repräsentieren die 1958 vom amerikanischen Psychologen Frank Rosenblatt vorgestellten Perzeptronen [73]. Rosenblatt führte die Gewichtung der Zelleneingangssignale ein. Von Marvin Minsky und Seymour Papert wurden die Perzeptronen weiter erforscht [57]. Sie definierten deren Verhalten so, dass die Zelleneingangswerte mit ihrer jeweilig zugeordneten Gewichtung multipliziert und diese Produkte dann aufsummiert werden.

Überschreitet die Summe den Schwellwert S der Zelle, so gibt die Zelle den Wert Eins weiter, sonst Null. [71]

$$f(s_E) = \begin{cases} 0 & \text{für } s_E < S \\ 1 & \text{für } s_E \geq S \end{cases} \quad (5.1)$$

mit $s_E = \sum_{i=1}^n x_i \cdot w_i$ als Summe der gewichteten Eingangssignale

und $S \in \mathbb{R}$ als Schwellwert

Durch Einführung eines fiktiven Neurons mit der konstanten Aktivierung -1 lässt sich der Schwellwert als zusätzliches Kantengewicht w_{n+1} ausdrücken. Die Summe der am Neuron eingehenden, verstärkten Signale aus (5.1) wird mit dem Schwellwert als zusätzlichen Summanden erweitert

$$s_E = \sum_{i=1}^{n+1} x_i \cdot w_i \quad (5.2)$$

mit $x_{n+1} = -1$

und $w_{n+1} = S$

wodurch die Entscheidungsfunktion selbst keine Variable mehr enthält:

$$f(s_E) = \begin{cases} 0 & \text{für } s_E < 0 \\ 1 & \text{für } s_E \geq 0 \end{cases} \quad (5.3)$$

Die Vernetzung von einzelnen biologischen Nervenzellen (Abb. 65) lässt ein komplexes Entscheidungsorgan entstehen. Dabei wird ein Axon mit Dendriten anderer Neuronen verbunden, zwischengeschaltete, nur einseitig passierbare Übergänge (Synapsen) verhindern Fehlleitungen. Die komplexeste und höchstentwickelte Form eines solchen Nervennetzes stellt das Gehirn dar.

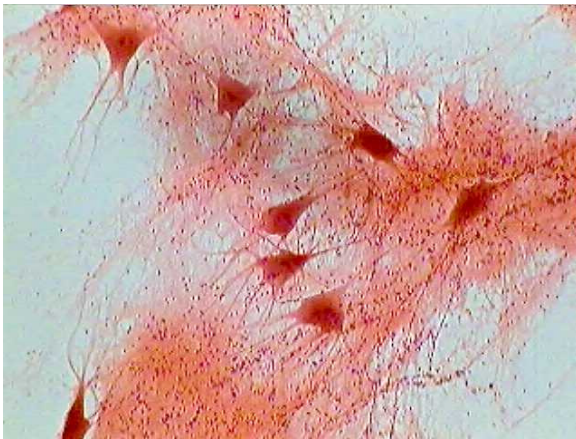


Abb. 65: Mikroskopaufnahme vernetzter Nervenzellen des Rückenmarks (mit freundlicher Genehmigung von [61])

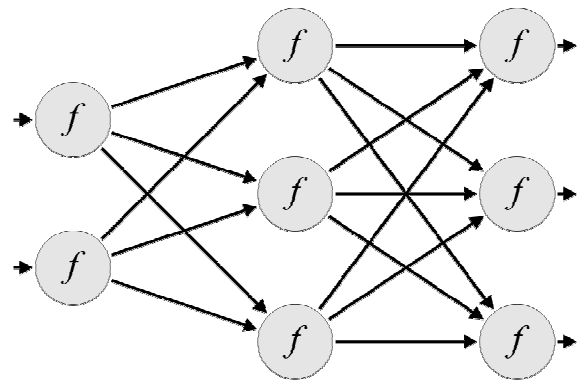


Abb. 66: Schemazeichnung eines künstlichen neuronalen Netzes, vorwärtsgerichtet hierarchisch

Beim Perzeptronen-Netzwerk werden die künstlichen Neuronen strukturiert in Schichten angeordnet (Abb. 66), durch welche hindurch ein n-dimensionaler Eingabevektor propagiert und auf einen m-dimensionalen Ergebnisvektor abgebildet wird [95]:

$$\underline{y} = g(\underline{x}) \quad (5.4)$$

mit $\underline{x} = [x_1 \ \dots \ x_n] \in \mathbb{R}^n$ als Eingabevektor

und $\underline{y} = [y_1 \ \dots \ y_m] \in \mathbb{R}^m$ als Ergebnisvektor

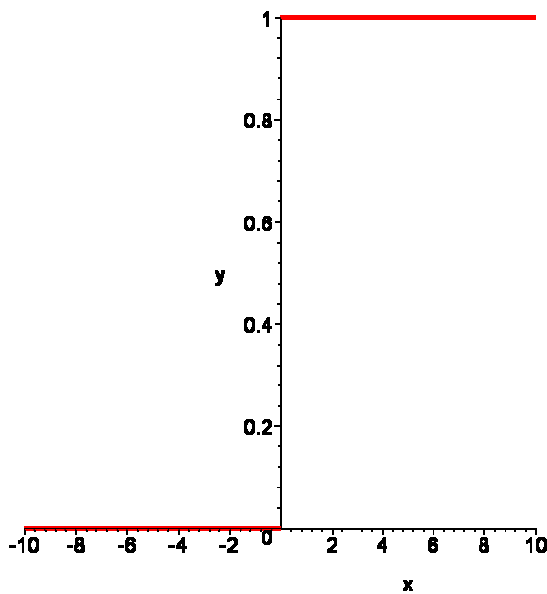
Das Training eines Perzeptronen-Netzwerks geschieht durch ein speziell dafür entwickeltes Gradientenabstiegsverfahren. Um die Richtung des Abstiegs berechnen zu können, muss die in Abb. 67 dargestellte Entscheidungsfunktion (5.3) durch eine differenzierbare Funktion substituiert werden. Die Sigmoidfunktion

$$f(x) = \frac{1}{1 + e^{-x}} \quad (5.5)$$

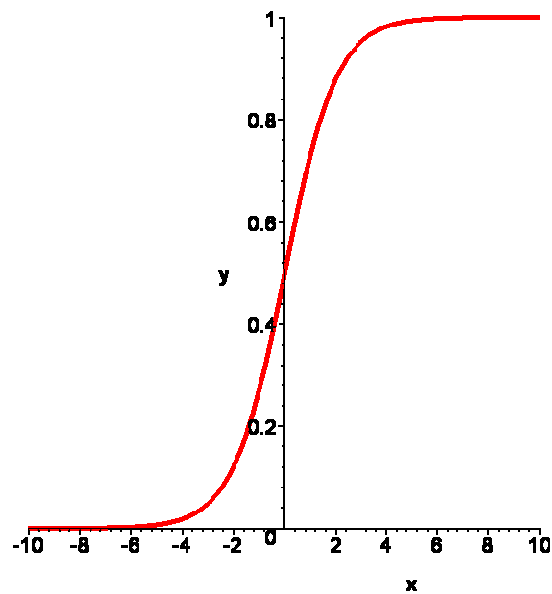
mit $x = s_E$ nach (5.2)

welche eine Ähnlichkeit zur Schwellwertfunktion hat (Abb. 68), erfüllt diese Bedingung. Zudem besitzt sie in ihrer Ableitung eine vorteilhafte Form, für welche lediglich der Funktionswert ausgewertet werden muss.

$$f'(x) = \left(\frac{1}{1+e^{-x}} \right)' = \frac{e^{-x}}{(1+e^{-x})^2} = f(x) \cdot (1-f(x)) \quad (5.6)$$



$$f(x) = \begin{cases} 0 & \text{für } x < 0 \\ 1 & \text{für } x \geq 0 \end{cases}$$



$$f(x) = \frac{1}{1+e^{-x}}$$

Abb. 67: Graph der Schwellwertfunktion Abb. 68: Graph der Sigmoiden Funktion

Die Abweichung, die die Abbildung des Eingabevektors auf den Ausgabevektor zum gewünschten Ausgabevektor besitzt, drückt der mittlere Netzfehler E_{mittel} aus. Er berechnet sich für bekannte Muster durch das mittlere euklidische Abstandsquadrat zwischen deren Soll- und Ist-Ausgabe-Vektoren:

$$E_{\text{mittel}} = \frac{\sum_{j=1}^N \sum_{i=1}^m (o_i^{\text{soll},j} - o_i^j)^2}{N \cdot m} \quad (5.7)$$

mit $o_i^{\text{soll},j}$ als gewünschte Aktivierung

bzw. o_i^j als tatsächliche Aktivierung

für das i -te Ausgabeneuron von m Neuronen in der Ausgangsschicht
und das j -te Validierungsmuster von insgesamt N

5.2 Netztopologie

Die Schichten eines Netzes kann man in drei Gruppen unterteilen: In Gruppe 1 (Eingangsschicht) befinden sich die Eingangsneuronen. Sie dienen ausschließlich

der Übernahme der auszuwertenden Informationen. Gruppe 2 ist die Ausgangsbeschicht, welche die berechnete Entscheidung des Netzes wieder an den Benutzer zurückgibt. Dazwischen liegen beliebig viele innere Schichten. Sie verbinden Eingangs- und Ausgangsbeschicht und sind von außen nicht ansprechbar, weshalb sie auch verborgene Schichten genannt werden. Im Zusammenhang mehrschichtiger Netze vernachlässigt man bei der Zählung der Schichten im Allgemeinen die Eingangsbeschicht. Grund dafür ist, dass deren Neuronen keine veränderbaren Kantengewichte besitzen, sondern lediglich die Netzeingangsparameter an die nachfolgende Schicht übergeben. Gezählt werden infolgedessen nur trainierbare Schichten. Das sind alle verborgenen Schichten sowie die Ausgangsbeschicht.

Ist die Kommunikation der Schichten immer nur auf nachfolgende Schichten gerichtet, so spricht man von einem vorwärtsgerichteten, hierarchischen Netz (Abb. 69 links). Dies ist die einfachste Form und aufgrund fehlender rekursiver Schleifen gut handhabbar.

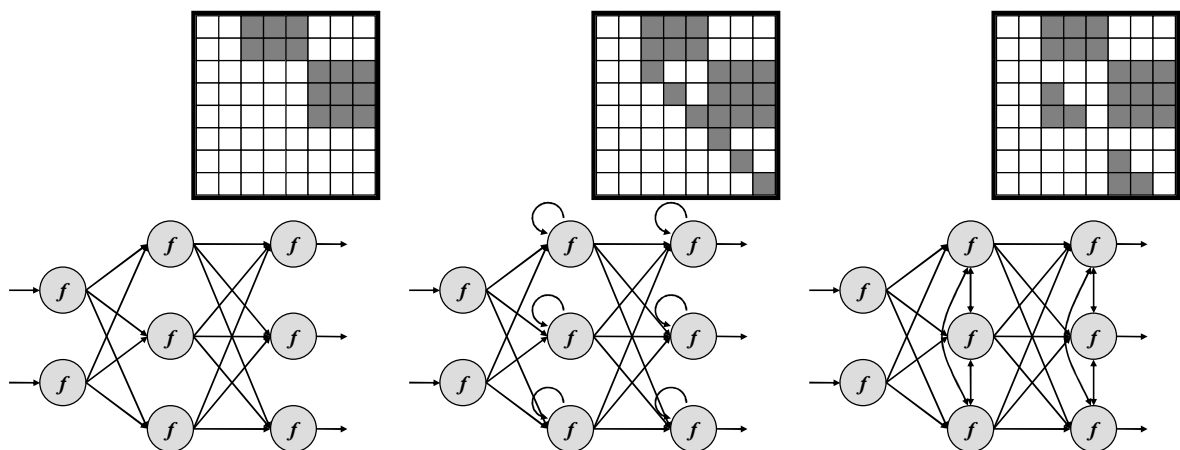


Abb. 69: Beispiele von Verbindungsnetzwerk-Topologien mit zugehörigen Verbindungsmatrizen: vorwärtsgerichtet, ebenenweise verbunden (links), direkte Rückkopplung (mitte) und laterale Rückkopplung (rechts) (nach [104])

Obwohl vorwärtsgerichtete, hierarchische Netze nur sehr abstrakt ihr biologisches Vorbild imitieren, eignen sie sich bereits, um komplexe Entscheidungsvorgänge nachzuahmen. Sie können dazu nichtlineare Funktionsabbildungen erlernen. Deshalb gilt es zu untersuchen, in welchem Maße die Topologie eines Netzes diese Fähigkeit beeinflusst. Wie in [35] bewiesen, kann bereits mit einem zweischichtigen Netz eine beliebige Funktion angenähert werden. Ein veranschaulichendes Beispiel ist die Approximation der Sinusfunktion mittels eines künstlichen neuronalen Netzes.

Dazu wird der Wertebereich der Sinusfunktion in den annähernd linearen Abbildungsbereich der Sigmoiden Funktion transformiert. Dies geschieht z. B. durch folgende Skalierung und Translation:

$$y = \frac{\sin(x)}{9} + 0,5 \quad (5.8)$$

als Transformation in den Wertebereich $\left[\frac{7}{18}; \frac{11}{18}\right]$

mit $\sin(x)$ als ursprüngliche Sinusfunktion mit Wertebereich $[-1; 1]$

Als Trainingsdaten stehen 630 Zahlenpaare der Abbildungsvorschrift $y = \sin(x)$ zur Verfügung. Die x-Werte liegen homogen in der Definitionsmenge $[0; 2\pi]$ verteilt und decken so eine Periodenlänge ab. Ein zweischichtiges Netz mit fünf verdeckten Neuronen ist in der Lage, die Funktionsabbildung zu erlernen. Abb. 70 veranschaulicht dies unter Verwendung der Rücktransformation der Netzausgabe auf den Wertebereich der Sinusfunktion.

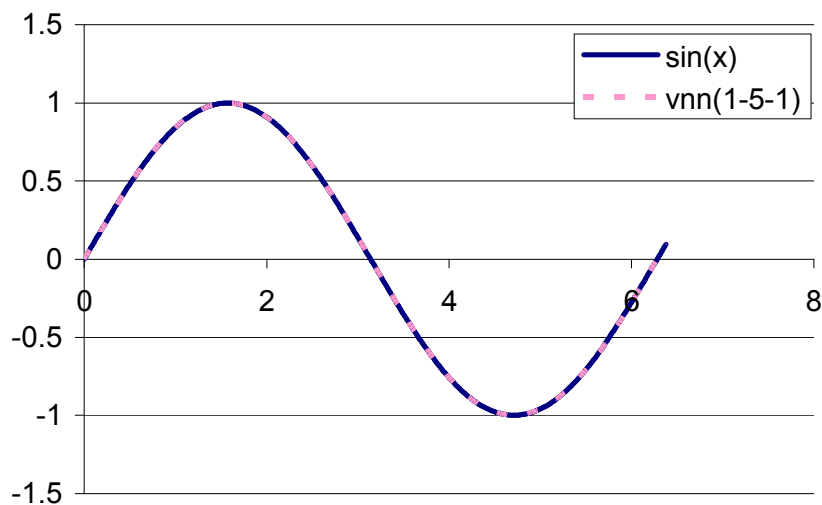


Abb. 70: Ein zweischichtiges Netz mit fünf verborgenen Neuronen approximiert die Sinusfunktion erfolgreich mit einer Genauigkeit von 10^{-4}

Dennoch kann es abhängig von der Problemstellung sinnvoll sein, auch mehrere verborgene Schichten einzusetzen [13]. Ebenso wie die Zahl der Schichten ist auch die Anzahl der darin vorhandenen Neuronen festzulegen. Die Freiheitsgrade des Netzes, also die Kantengewichte und Schwellenwerte, bestimmen die Prägbarkeit des Netzes auf die Aufgabe. Zu große Netze sind schwer zu trainieren, da zur Bestimmung der großen Zahl an Variablen eine umfassende Menge an heterogen verteilten Trainingsdaten notwendig ist. Ferner tendieren unnötig große Netze dazu,

sich auf die Trainingsmuster zu spezialisieren (overfitting) und dadurch ihre Generalisierungsfähigkeit zu verlieren. Wird das Netz auf der anderen Seite zu klein gewählt, ist es auf die Problemstellung nicht mehr anwendbar [97].

In der Fachliteratur herrscht Übereinstimmung, dass das Festlegen der Netztopologie, insbesondere der inneren Schichten, einem „entwerferischen Akt“ gleichkomme, da es zwar Vorschläge und Empfehlungen, jedoch keine festen Regeln für die Anordnung der Neuronen in den verborgenen Schichten gäbe [13][30][95]. Ein sinnvoller methodischer Ansatz zur Entwicklung der Netztopologie ist der Start mit einem kleinen Netz, welches sukzessiv vergrößert und in der Leistung verbessert wird [36]. Dieses Initialnetz kann ein zweischichtiges Netz sein, dessen verborgene Schicht die Mächtigkeit der kleineren der beiden äußeren besitzt.

5.3 Netzeingabe

Der Datenaustausch mit dem virtuellen neuronalen Netz geschieht über die Eingangs- und die Ausgabeschicht. Die Eingabeschicht erwartet einen n -dimensionalen Eingabevektor, welcher mit Fließkommazahlen zu besetzen ist. Die Dimension n entspricht dabei der Anzahl an Neuronen, die die Eingabeschicht besitzt.

Die vom Netz auszuwertenden Parameter müssen als Eingabewerte entsprechend aufbereitet werden. Maßzahlen und Kenngrößen können dabei direkt im Eingabevektor abgelegt werden. Boolesche Variablen, welche die Werte wahr oder falsch annehmen, sind auf reelle Zahlen abzubilden. Dazu gilt zu beachten, dass nicht die Zahl 0,0, wie zu nächst naheliegend, einen der beiden Zustände repräsentiert. Der Grund dafür ist, dass im Lernschritt die Kantengewichte, welche als Verstärkungsfaktoren der Signale anzusehen sind, angepasst werden müssen, ein Null-Signal aber vom Verstärkungsfaktor unabhängig ist. Eine Substitution der beiden booleschen Zustände durch die Zahlen -1,0 und +1,0 umgeht das Problem [85]. Gleiches gilt für Maßzahlen, die den Wert 0,0 annehmen können. Damit diese Werte ebenfalls zum Lernvorgang beitragen, findet eine Translation des Wertebereichs dahingehend statt, dass 0 kein Element dieser Menge mehr ist.

Wie sich in Studien zu geeigneten Eingangswerten gezeigt hat [3], kann die Aufbereitung der Trainingsdaten von absoluten Größen zu deren Verhältniswerten zueinander eine Verbesserung darstellen. Dies hat folgenden Grund: In Kapitel 5.1 wurde erläutert, dass die Neuronen Signale auswerten, indem sie diese zunächst

gewichten und anschließend aufsummiert mit einem Referenzwert, dem Schwellwert, verglichen. Produkte oder Quotienten zwischen den Eingabewerten können durch dieses Vorgehen nur bedingt abgebildet werden, weshalb diese sinnvollerweise durch eine Aufbereitung der Datenbasis in einer Vorlaufrechnung gebildet werden.

Am ersten Eingangsbeispiel lässt sich dies praktisch veranschaulichen: 30 Muster der Geometrieklassen Kreis, gleichseitiges Dreieck und Quadrat sollten ein zweischichtiges Netz mit drei Neuronen in der verborgenen Schicht so trainieren, dass dieses beliebige andere Ausprägungen dieser Klassen richtig klassifizieren kann. Bei Verwendung der vier geometrischen Parameter Flächenzahl, Umfang sowie der Ausdehnungen eines kleinstmöglichen Hüllquaders für die Objekte der Geometrieklassen lässt sich das Netz nicht so trainieren, dass es die Identifizierungsvorschrift abbildet und so mit brauchbaren Ergebnissen klassifiziert. Abb. 71 zeigt das nichtkonvergierende Lernverhalten. Bereits nach 40 Trainingszyklen stagniert der mittlere Netzfehler bei etwa 1,1 und nimmt im weiteren Lernverlauf sogar wieder etwas zu.

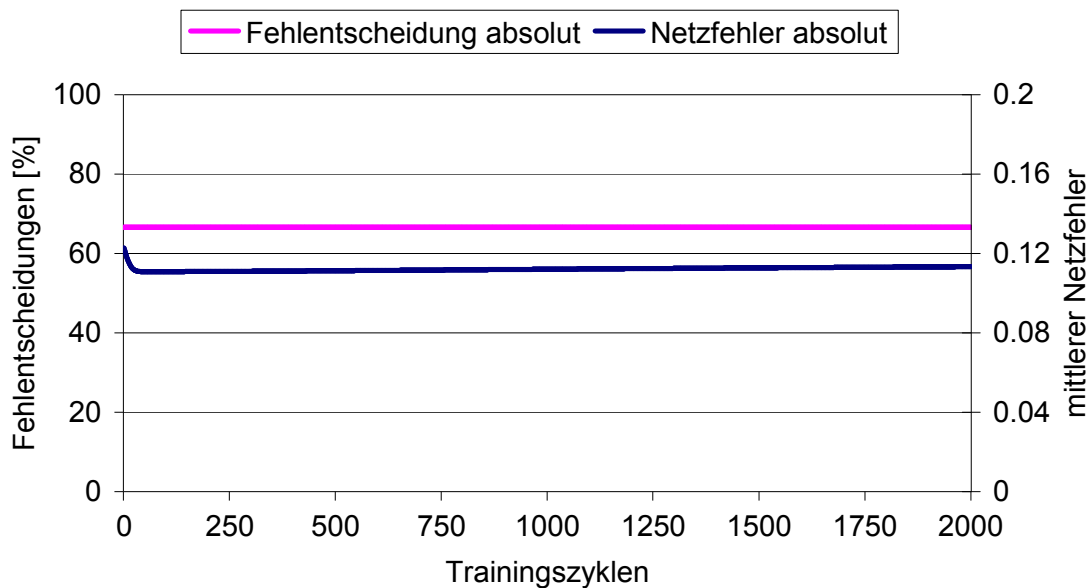


Abb. 71: Nichtkonvergierendes Lernverhalten bei den unverknüpften Parametern

Verwendet man jedoch statt dieser vier Eingangswerte nur zwei, welche sich als Verhältniswerte aus den ursprünglichen vier durch Multiplikation und Division berechnen lassen, so kann ein virtuelles neuronales Netz mit gleicher Topologie in der verborgenen Schicht die Objekte klassifizieren. Der mittlere Netzfehler wird mit fortschreitenden Trainingszyklen stetig kleiner, wodurch eine Verringerung der Fehlentscheidungen einhergeht (Abb. 72). Aus jedem der als Eingabewerte

verwendeten Quotienten lässt sich eindeutig auf die zugrunde liegende Objektklasse im Sinne von Abschnitt 5.0 schließen (Abb. 73). Obwohl die Ausgangsbasis in beiden Fällen die vier geometrischen Parameter sind, kann das verwendete künstliche neuronale Netz diese offensichtlich nicht selbstständig derart in Verbindung bringen, dass dies den beiden Quotienten entspricht.

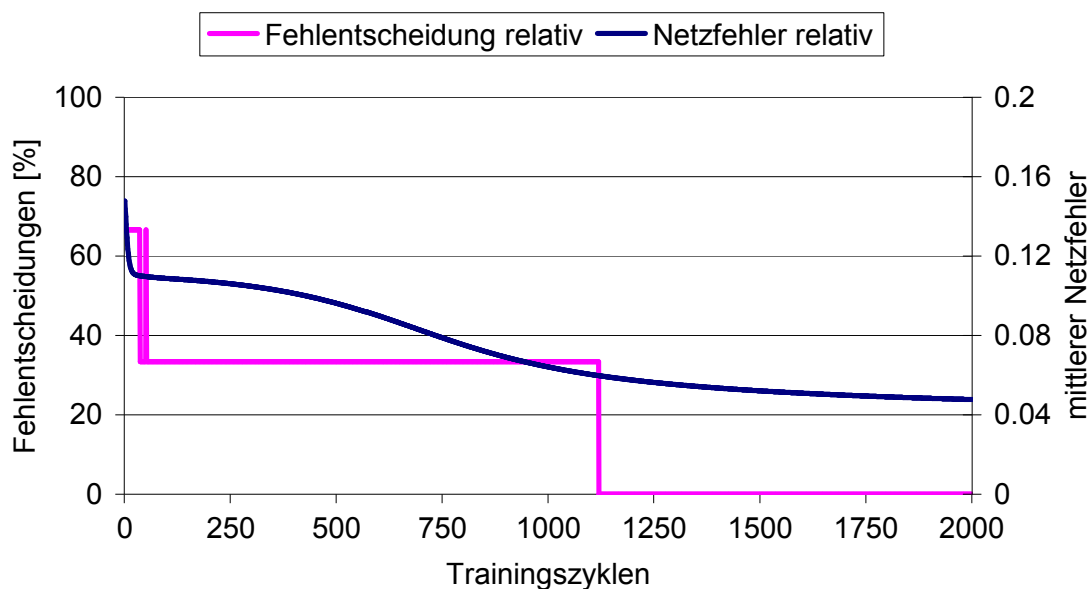


Abb. 72: Konvergenz bei Verwendung von Quotienten als Trainingsparameter

	$\frac{\text{Umfang}}{\text{Ausdehnung}_1}$	$\frac{\text{Fläche}}{\text{Ausdehnung}_1 \cdot \text{Ausdehnung}_2}$
Kreis	π	$\frac{\pi}{4}$
gleichseitiges Dreieck	3	$\frac{1}{2}$
Quadrat	4	1

Abb. 73: Die Quotienten der geometrischen Parameter erlauben eine eindeutige Zuordnung

5.4 Netzausgabe

Analog zur Netzeingabe findet die Netzausgabe statt: Die m Neuronen der Ausgabeschicht speichern ihre Aktivierung in einem m -dimensionalen Vektor. Jeder

Eintrag dieses Vektors ist eine reelle Zahl im Bereich der Wertemenge der Aktivierungsfunktion eines Ausgabeneurons. Wird dazu die Sigmoidfunktion zugrunde gelegt, so ist dies das Intervall $]0; 1[$. Andere Ansätze verwenden speziell für die Ausgabeschicht eine andere Aktivierungsfunktion oder geben die Summe der gewichteten Neuroneneingangssignale direkt aus, so dass dabei der Wertebereich die gesamte Menge der reellen Zahlen abdeckt.

Die Eingangssignale werden durch das Netz auf eine Aktivierung der Ausgabeneuronen abgebildet. Es ist deshalb notwendig, sich zunächst Gedanken über die gewünschte Information auf der Ausgabeseite zu machen.

Wird ein neuronales Netz dazu verwendet, eine Funktion zu approximieren, so muss der Wertebereich des Ausgabeneurons in den Wertebereich der gesuchten Funktion transformiert werden, wie dies in Abschnitt 5.2 für die Sinusfunktion bereits gezeigt wurde. Die Beschränkung auf den annähernd linearen Bereich dient dazu, dass Abweichungen innerhalb dieses Intervalls kongruent bewertbar sind. Durch die Abbildungsvorschrift der Sigmoiden Funktion wird der Unterschied von Soll- zu Ist-Werten außerhalb dieses Bereichs sonst nichtlinear skaliert wiedergegeben.

Eine andere Art der Funktionsabbildung stellt der Einsatz eines virtuellen neuronalen Netzes für Klassifizierungsaufgaben dar. Betrachtet man für die Menge aller Trainingsdaten die korrespondierende Menge an Ausgabevektoren, so müssen sich diese innerhalb des Wertebereichs so ballen, dass Objekte gleicher Klassen in denselben Ballungen zu liegen kommen. Sind die zu unterscheidenden Klassen Komplementärmenge, so ist eine eindeutige Zuordnung gefordert. Idealerweise bilden Eingangssignale gleicher Klassen immer auf die gleiche Aktivierung der Ausgabeneuronen ab.

Als naheliegende Möglichkeit wird zunächst der eindimensionale Ausgabevektor diskutiert. Der durch die Sigmoidfunktion bestimmte Wertebereich $]0; 1[$ des Ausgabeneurons könnte in Abschnitte für die verschiedenen Objektklassen unterteilt werden. Prinzipiell ist dieses Vorgehen eine Möglichkeit, verschiedene Klassen zu unterscheiden [46]. Bei größer werdender Zahl an Klassen führt die Nichtlinearität der Sigmoiden Abbildungsfunktion (5.5) bei dieser Methode jedoch zu Problemen. Eine Lösungsmöglichkeit stellt die Beschränkung auf den annähernd linearen Bereich $[0,3; 0,6]$ oder die Verwendung einer linearen Abbildungsfunktion für die Neuronen der Ausgabeschicht dar. Dies wurde im Rahmen dieser Arbeit jedoch nicht weiter verfolgt. Ein weiteres Problem ergibt sich während der Lernphase: Die

internen Entscheidungsregeln passen sich über mehrere Iterationen schrittweise den gewünschten Ausgabesignalen an. Eine Korrektur in Richtung der geforderten Aktivierung verstärkt deshalb nicht zwingend die Entscheidung für die gewünschte Klasse: Die Änderung verschiebt möglicherweise die Aktivierung des Ausgabeneurons zugunsten einer dritten Klasse, deren Wertebereich zufällig zwischen dem der zuerst votierten und der angestrebten liegt. Dies kann zu wiederholten Falscheingrupperungen in unterschiedliche Klassen führen, was den Lernfortschritt im günstigsten Fall nur verlangsamt. Auf den Benutzer wirkt dieses Lernverhalten nicht nachvollziehbar und deshalb chaotisch.

Die Verwendung eines Ausgabevektors höherer Dimension bietet dagegen die Möglichkeit, von den beschriebenen Unzulänglichkeiten eines in Abschnitte für die einzelnen Klassen eingeteilten Wertebereichs der Sigmoiden Funktion unabhängig zu sein. Allerdings ist dann festzulegen, durch welche Aktivierung der einzelnen Ausgabeneuronen auf eine Klassenzugehörigkeit geschlossen werden soll. Prinzipiell wäre eine binäre Kodierung in der Zuordnung möglich, bei der die Binärverschlüsselung einer Klasse zugeordnet wird. Die Einträge des Ausgabevektors sind dabei in Binärziffern zu konvertieren: Ein Wert größer gleich 0,5 setzt 1, andernfalls steht 0.

Es ist jedoch zielführender, jeder Klasse ein Ausgabeneuron zuzuordnen und zu erwarten, dass bei einer Entscheidung für eine Klasse genau dieses Neuron die maximal mögliche und alle anderen die minimale Aktivierung besitzen (Abb. 74) [30]. Durch diesen Ansatz kann geprüft werden, wie sicher – also eindeutig – die Klassifizierung ist. Besitzen beispielsweise mehrere Ausgabeneuronen eine ähnlich große Aktivierung, könnte auf eine zweifelhafte Eindeutigkeit geschlossen werden.

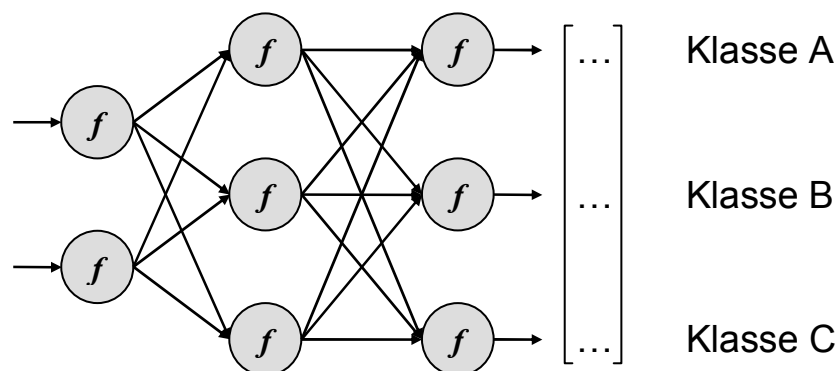


Abb. 74: Direkte Zuordnung der Ausgabeneuronen zu Klassen

In Bezug auf das Training ist die Wahl von einem Ausgabeneuron je Klasse besonders geeignet. Der erwünschten Klasse wird ein Neuron mit der Aktivierung 1 zugeschrieben, allen anderen 0. Der errechnete Ausgabevektor lässt sich so leicht mit dem Zielvektor vergleichen. Die Anpassung geschieht für jede Klasse direkt, das heißt eine Erhöhung einer Neuronenaktivierung führt ausschließlich zu einer Stärkung der Entscheidung für die mit ihr in Korrelation stehende Klasse, eine Reduzierung schwächt die Wahl.

5.5 Auswahl von Trainingsdaten

Virtuelle neuronale Netze erlernen ihre Aufgabe anhand gegebener Beispiele, welche die auszuwertenden Informationen und die Zielwerte enthalten. Ein einzelnes Beispiel besteht aus zwei Vektoren, dem Eingabe- und dem Ausgabevektor. Man bezeichnet solch ein Paar als Trainingsmuster. Die Menge aller Trainingsmuster ergibt die Trainingsbasis.

Im Zusammenhang mit der Klassifizierung durch ein virtuelles neuronales Netz wird für die Merkmale, die ein zu untersuchendes Objekt übergibt, gefordert, dass diese für die „Zugehörigkeit zu einer Klasse charakteristisch sind“ ([58], S. 19). Die ausgewählten Parameter müssen eine eindeutige Zuordnung zu einer Klasse erlauben. Ist dies nicht der Fall, bedingt dies die Hinzunahme anderer, besser bezeichnender Merkmale, welche in Kombination die Klassifizierung ermöglichen.

Sind die Eingangsparameter festgelegt, müssen geeignete Beispiele gefunden werden, da die Anzahl und die Qualität der Trainingsmuster die Generalisierungsfähigkeit des trainierten Netzes maßgeblich bestimmen [30]. Deshalb ist es notwendig, die Datensätze der Trainingsbasis gezielt auszuwählen. Dabei gilt ähnliches wie für die Festlegung der Topologie: Es gibt einige Ansätze und Regeln zur Bestimmung der Größe der Trainingsbasis [4] und zur Auswahl von Trainingsmustern, im konkreten Fall muss deren Eignung aber empirisch geprüft werden. Die Literatur gibt zur Ermittlung der Trainingsbasisgröße folgende Faustformel an [30]:

$$N > \frac{W}{\varepsilon} \tag{5.9}$$

mit N als Anzahl an Trainingsmustern
und W als Anzahl der Freiheitsgrade im Netz
und ε als angestrebter Fehler

Um die Qualität der Trainingsdaten abschätzen zu können, bedarf es Überlegungen zu deren Verteilung. Die Eingabeparameter beschreiben die vom Netz auszuwertende Information, weshalb der Eingabevektor auch als Merkmalsvektor bezeichnet wird. Diese Merkmalsvektoren liegen im Merkmalsraum. Für das Beispiel der Farbklassifizierung stellt Abb. 75 diesen Raum dar, welchen die Intensitäten für gelb (als additive Farbmischung von rot und grün) und blau aufspannen.

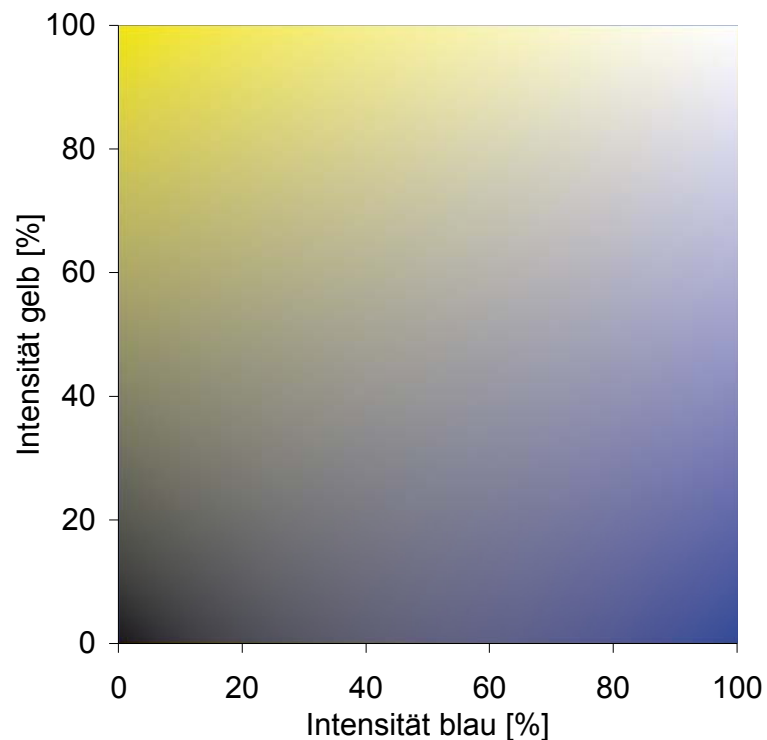


Abb. 75: Von den Farbintensitäten blau und gelb aufgespannter Merkmalsraum

Der Merkmalsraum lässt sich durch diese Parameter in Bereiche trennen [52][58]. Im Kontext dieser Aufgabenstellung wurde festgelegt, dass ein Muster als gelb zu betrachten ist, wenn $Intensität_{gelb} - Intensität_{blau} > 0,6$ gilt. Die zum Lernen verwendeten Beispiele müssen nun so gewählt werden, dass sie diese Bereiche im Merkmalsraum geeignet besetzen. Erst durch die Auswahl passender Trainingsmuster kann das Netz die durch sie repräsentierten, dahinter stehenden Klassifizierungsregeln erlernen. Das zu trainierende Netz besitzt zwei Eingabeneuronen, welche die beiden Farbintensitäten entgegennehmen, eine verborgene Schicht mit ebenfalls zwei Neuronen, sowie ein Ausgabeneuron, dessen Aktivierung den Wert 1 annehmen soll, wenn gelb erkannt wurde, andernfalls 0.

Eine optimale Trainingsbasis besitzt im Merkmalsraum homogen verteilte Merkmalsvektoren. Insbesondere Objekte, die durch ihre Parameter nicht im Zentrum

des Wertebereichs der Klasse sondern auch abseits Richtung Bereichsgrenzen zu liegen kommen, sind erwünscht. Nur so lassen sich die Abgrenzungen zwischen den zu unterscheidenden Klassen passend trainieren. Jedem Beispiel wird die gleiche Gewichtung während der Trainingsphase zugeschrieben. Es sind deshalb redundante Datensätze zu vermeiden, da diesen sonst eine übermäßige Bedeutung zugemessen wird.

Zufällige Muster, wie am Beispiel der Farbklassifizierung in Abb. 76 gezeigt, gewährleisten von sich aus nicht zwingend eine gleichmäßige Besetzung des Merkmalsraums. Eine quantitative Angleichung, wodurch die Beispielmengen beider Klassen eine gleichgroße Mächtigkeit erhalten, löst dieses Problem nicht, wie Abb. 77 veranschaulicht: Die ungleiche Verteilung der Trainingsmuster im Merkmalsraum verschärft sich, unbesetzte Bereiche können entstehen. Eine optimierte Trainingsbasis lässt sich jedoch gewinnen, indem die Muster mit Raumdiskretisierungsverfahren untersucht werden. Mit der Ballungsanalyse lassen sich dicht bei- oder gar aufeinander liegende Merkmalsvektoren identifizieren und entfernen (Abb. 78). Die Diskretisierung durch Voxelierung [96] deckt des Weiteren auf, ob die Trainingsmuster homogen im Gültigkeitsbereich des Merkmalsraumes verteilt sind. Bereiche mit zu geringer Dichte können erkannt und gemeldet werden.

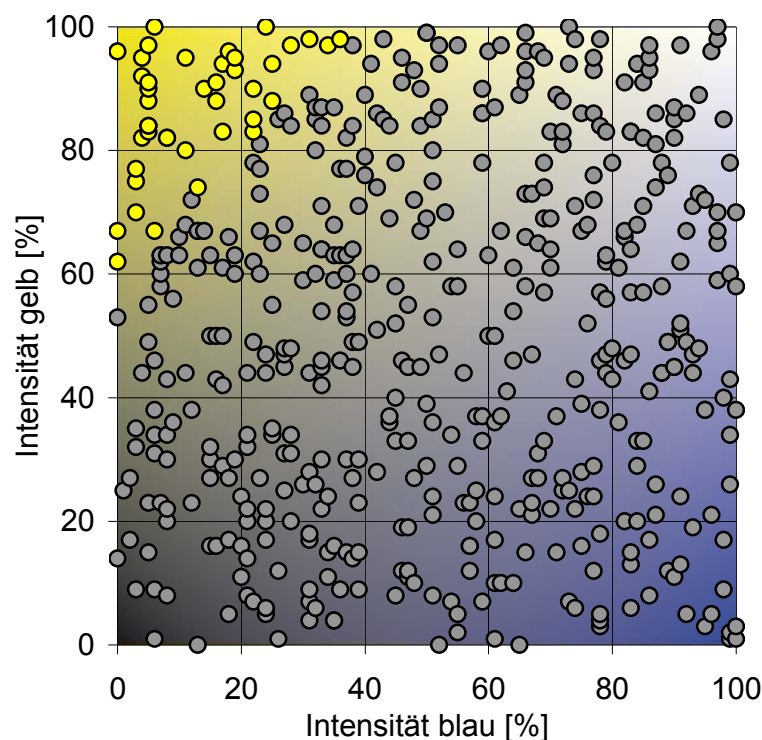


Abb. 76: Willkürliche Besetzung des Merkmalsraums durch zufällig erzeugte Muster

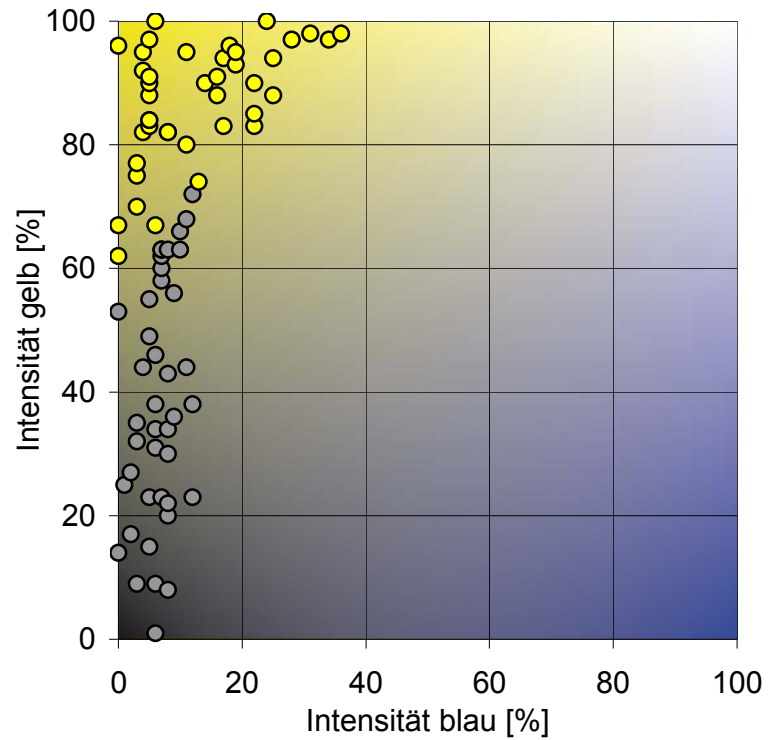


Abb. 77: Die quantitative Angleichung der beiden zu unterscheidenden Klassen verschärft das Problem einer ungleichen Verteilung im Merkmalsraum

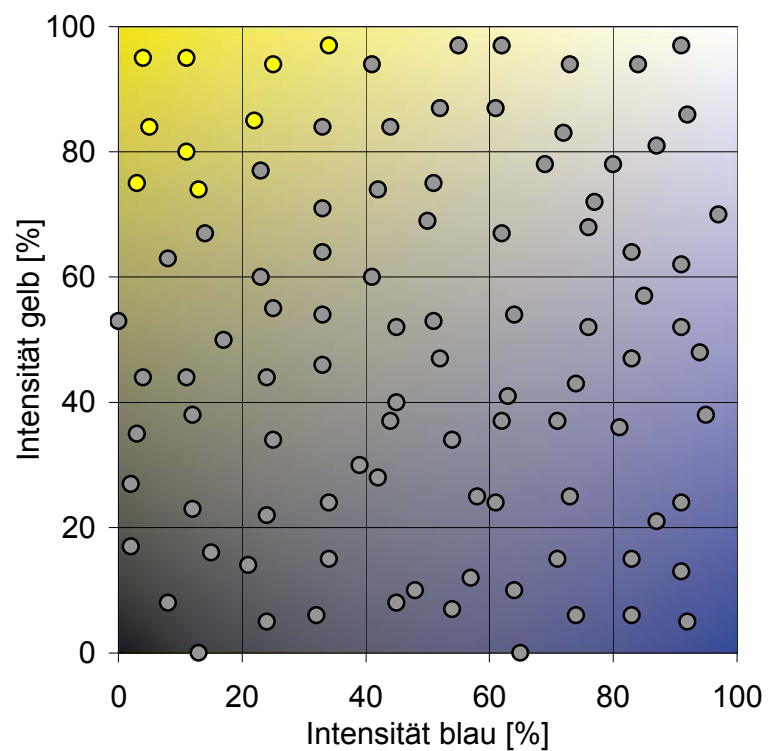


Abb. 78: Mögliche automatische Optimierung der Trainingsbasis durch Ballungsanalyse

5.6 Netzprägung

5.6.1 Back-Propagation-Algorithmus

Die Fähigkeit zur Entscheidung, das heißt das Wissen eines neuronalen Netzes, steckt in dessen internen, anpassbaren Gewichten. Diese sind im Lernschritt so anzupassen, dass das Netz eine Menge gegebener Eingabevektoren auf die zugehörigen Ausgabevektoren unter Vernachlässigung eines tolerierten Fehlers abbildet.

Die beschriebenen mehrschichtigen Perzeptronennetzwerke können mit dem dafür entwickelten Back-Propagation-Algorithmus lernen [74]. Mit diesem Verfahren werden alle Freiheitsgrade des Netzes angepasst.

Zur Herleitung des Lernalgorithmus sei zunächst die Vorwärtsrechnung betrachtet: Wie in Abb. 66 dargestellt, findet die Informationsübertragung innerhalb des hierarchischen Netzes schichtweise statt. Die Aktivierung der Eingabeneuronen entspricht den Werten des Eingangsvektors. Beginnend bei der ersten verborgenen Schicht (bei einem einschichtigen Netz bereits die Ausgabeschicht) akkumulieren die Neuronen die Aktivierung ihrer Vorgänger, gewichten diese und errechnen mit der Aktivierungsfunktion ihre eigene Aktivierung. Jede folgende Schicht verfährt analog, bis schließlich die Neuronen der Ausgabeschicht ihre Aktivierung bestimmt haben.

In der Lernphase, welche sämtliche Variablen des Netzes, zusammenfassbar im Vektor

$$\underline{W} = (w_{ij})_{i \rightarrow j} \quad (5.10)$$

mit $i \rightarrow j$ als existierende Kantengewichtung zwischen den Neuronen i und j [22] bzw. als Schwellwert des Neurons j

bestimmt, wird nun der errechnete Ausgabevektor mit dem Soll-Ausgabevektor verglichen. Die Abweichung der Soll- von den Ist-Werten drückt die von den Freiheitsgraden des Netzes abhängige Fehlerfunktion $E(\underline{W})$ aus (der Faktor $\frac{1}{2}$ dient der Vereinfachung beim Differenzieren):

$$E(\underline{W}) = \frac{1}{2} \sum_{i=1}^m (o_i^{soll} - o_i)^2 \quad (5.11)$$

mit o_i^{soll} als gewünschte Aktivierung des i-ten Ausgabeneurons
und o_i als tatsächliche Aktivierung des i-ten Ausgabeneurons
für den zum Training verwendeten Merkmalsvektor

Mit einem Gradienten-Abstieg minimiert das Back-Propagation-Verfahren diesen Fehler. Wie in [104] näher beschrieben, müssen alle Gewichtungen \underline{W} des Netzes um eine Schrittweite η in Richtung des negativen Gradienten der Fehlerfunktion $E(\underline{W})$ verändert werden. Der Vektor

$$\underline{\Delta W} = -\eta \cdot \nabla E(\underline{W}) \quad (5.12)$$

hält die Änderungen fest. Für die Änderung eines einzelnen Kantengewichts gilt:

$$\Delta w_{ij} = -\eta \cdot \frac{\partial}{\partial w_{ij}} E(\underline{W}) \quad (5.13)$$

mit w_{ij} als Kantengewichtung zwischen den Neuronen i und j

Unter Verwendung der Sigmoidfunktion als Aktivierungsfunktion und der Fehlerfunktion $E(\underline{W})$ lassen sich die einzelnen Kantengewichte w_{ij} anpassen. Nach partieller Ableitung der Fehlerfunktion unter Verwendung der Kettenregel [104] und der Ableitung (5.6) der Sigmoidfunktion ergibt sich bei Betrachtung des Neurons j für die Änderung des Kantengewichtes

$$\Delta w_{ij} = \eta \cdot o_i \cdot \delta_j \quad (5.14)$$

mit o_i als bestehende Aktivierung des Vorgängerneurons i
und δ_j als Fehlersignal der Neurons j, abhängig von der
Schichtzugehörigkeit

mit

$$\delta_j = f'(s_E) \cdot (o_j^{soll} - o_j) = f(s_E)(1 - f(s_E)) \cdot (o_j^{soll} - o_j) = o_j(1 - o_j) \cdot (o_j^{soll} - o_j) \quad (5.15)$$

für Neuron j als Element der Ausgabeschicht

mit $o_j = f(s_E)$ als Ergebnis der Entscheidungsfunktion nach (5.5)

und

$$\delta_j = f'(s_E) \cdot \sum_k (\delta_k \cdot w_{jk}) = f(s_E)(1 - f(s_E)) \cdot \sum_k (\delta_k \cdot w_{jk}) = o_j(1 - o_j) \cdot \sum_k (\delta_k \cdot w_{jk}) \quad (5.16)$$

für Neuron j als Element einer der inneren Schichten

mit \sum_k als Summe über alle dem Neuron j nachfolgenden Neuronen

k in der nächsten Schicht

und $o_j = f(s_E)$ als Ergebnis der Entscheidungsfunktion nach (5.5)

Zu beachten ist, dass zunächst alle Werte δ_j berechnet werden müssen. Da die δ_j der inneren Schichten in Abhängigkeit von ihren Nachfolgern bestimmt werden, sind diese als erstes in der Ausgangsschicht zu ermitteln. Die jeweilige Vorgängerschicht (daher der Begriff Back-Propagation) berechnet dann ihre δ -Werte, beginnend bei der letzten bis zur ersten inneren Schicht. Da die Eingabeschicht keine modifizierbaren Kantengewichte besitzt, wird sie übersprungen. Sind nun alle δ festgelegt, können die Änderungen aller Kantengewichte gemäß (5.14) errechnet werden.

5.6.2 Initialisierung

Ein untrainiertes Netz ist vor dem ersten Trainingsschritt zu initialisieren. Dabei dürfen die Gewichte nicht auf einen einheitlichen Wert gesetzt werden. Andernfalls würden identische Kantengewichte bei der Vorwärtsrechnung zunächst zu gleich großer Aktivierung aller Ausgabeneuronen und nachfolgend gemäß (5.16) zu symmetrischen Änderungen führen, das heißt die Kanten zu den Vorgängerneuronen erfahren stets die gleichen Anpassungen. Die Vermeidung dieses Problems wird als symmetry breaking bezeichnet [104]. Bereits ein nur geringer Unterschied in den Initialisierungswerten der Kantengewichte umgeht dieses Phänomen. Eine sinnvolle Vorbelegung der Startgewichte erfolgt beispielsweise mit Zufallswerten aus dem Intervall $]-1; +1[\setminus \{0\}$.

5.6.3 Festlegung des Lernparameters

Das Problem eines Gradientenabstiegsverfahrens ist, dass die Möglichkeit des Nichterreichens des globalen Minimums besteht. So kann es zum Hängenbleiben in einem lokalen Minimum kommen (Abb. 79 links). Zu diesem typischen Problem wurde bei der Einführung des Back-Propagation-Lernalgorithmus festgestellt, dass es dennoch selten vorkäme, dass der Algorithmus in einem lokalen Minimum ende, welches signifikant schlechter sei als das globale. In solch einem Fall könne eine

geringfügige Vergrößerung des Netzes Verbesserung bringen [74]. Ein anderer Fehler ist das Überspringen des globalen Minimums (Abb. 79 rechts).

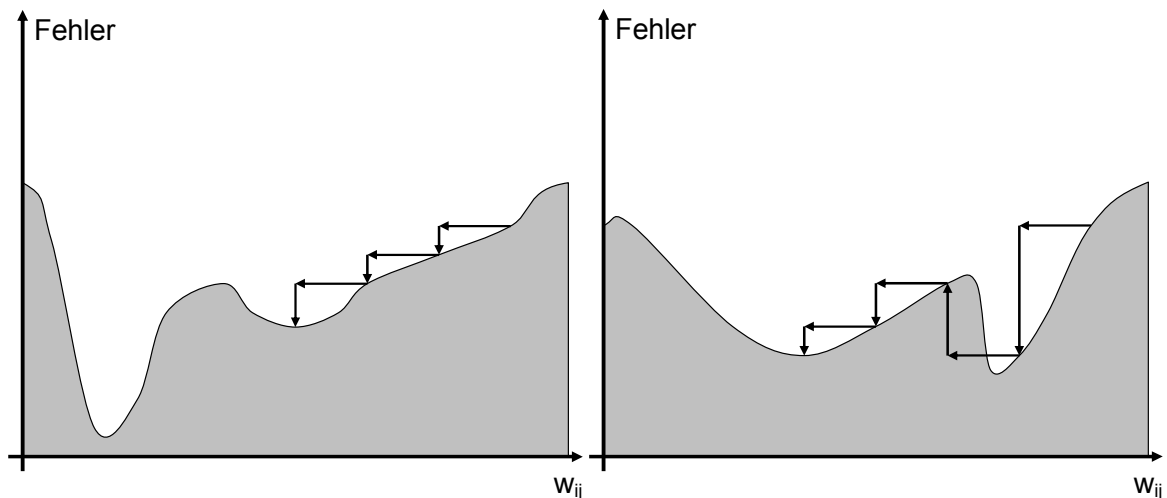


Abb. 79: Stagnation im lokalen Minimum und Herausspringen aus dem globalen Minimum als Beispiele von Problemen beim Gradientenabstieg, veranschaulicht durch Veränderung der Position auf der Fehlerfläche (nach [104], S. 113)

Das Konvergenzverhalten lässt sich durch den Lernparameter η (Schrittweite des Gradientenverfahrens) beeinflussen. Wird dieser zu klein gewählt, neigt das System dazu lokale Minima nicht zu durchschreiten, bei zu großem η tendiert das Netz zum Verlassen des globalen Zielwertes. Die geeignete Wahl des Lernparameters muss empirisch ermittelt werden. Zumeist liegt dieser im Intervall $]0; 1[$ [30].

Eine Vermeidung des Stagnationsproblems ergibt sich durch Einführung eines Momentum-Terms [30][104]. Dabei wird ein Summand eingeführt, der abhängig vom vorhergehenden Lernschritt die Schrittweite des Gradientenabstiegs beeinflusst. Man erreicht damit ein optimiertes Lernverhalten, da auf flacheren Teilen der Fehlerfläche größere Schrittweiten angewendet werden als auf zerklüfteten [104].

$$\Delta w_{ij}^t = \eta \cdot o_i \cdot \delta_j + \alpha \cdot \Delta w_{ij}^{t-1} \quad (5.17)$$

mit $\alpha \in [0; 1]$ als Momentum-Konstante

und t als aktuellen Iterationsschritt

und $t - 1$ als vorhergehenden Iterationsschritt

Unterstellt man, dass mit zunehmendem Iterationsschritt der Fehler näher am globalen Minimum liegt, so ließe sich dessen Überspringen durch Reduzierung der Schrittweite bei jedem Durchlauf beikommen. Durch einen Vorfaktor, welcher sich

indirekt proportional zur Iterationszahl verhält, lässt sich ein kleiner werdender Lernschritt in Gleichung (5.14) integrieren:

$$\Delta w_{ij} = \frac{1}{t_0 + t_1 \cdot t} \cdot \eta \cdot o_i \cdot \delta_j \quad (5.18)$$

mit $t \in \mathbb{N}$ als aktuellen Iterationsschritt

und $t_0, t_1 \in \mathbb{R}$ als Degressionsparameter

5.6.4 Einbringung der Lerninformation

Die Prägung eines virtuellen neuronalen Netzes geschieht durch Anpassung der internen Netzfreiheitsgrade anhand von Beispielen. Für jedes Muster wird mittels des Back-Propagation-Algorithmus eine notwendige Veränderung für alle Netzkantengewichte errechnet. Die eigentliche Aktualisierung der Kantengewichte kann auf zwei unterschiedliche Arten erfolgen: Sie können entweder nach jedem einzelnen Trainingsmuster oder akkumuliert nach Durchlaufen der gesamten Trainingsbasis auf das Netz übertragen werden.

Im online-Verfahren [13], in der Literatur auch als incremental oder pattern mode [30] bezeichnet, werden schrittweise die Trainingsmuster durchlaufen und die Änderung der Kantengewichte sofort in das Netz eingebracht. Da sich die Reihenfolge der vorgezeigten Muster auf das Lernverhalten auswirkt, müssen diese dem Netz unsortiert präsentiert werden. Eine Blockbildung einzelner Klassen, wie sie bei der Zusammenstellung der Trainingsbasis unter Umständen entsteht, ist auf jeden Fall zu vermeiden. Nach jedem vollständigen Durchgang der Trainingsbasis ist diese erneut zu durchmischen.

Das offline- [13] oder auch batch-Verfahren [30] aktualisiert die Kantengewichte erst nach der Analyse der gesamten Trainingsbasis. Die Änderungen, die sich nach jedem Muster errechnen, werden dazu aufsummiert und nach Behandlung aller Beispiele in einem späteren Modifikationsschritt auf die Kantengewichte im Netz übertragen. Bei diesem Vorgehen ist die Reihenfolge, in der die Muster erlernt werden, unerheblich.

5.6.5 Abbruch des Lernvorgangs

Der Lernvorgang soll wiederholt werden, bis das Netz die Entscheidungsregel hinreichend abbilden kann. Jede vollständige Betrachtung der Trainingsbasis wird als Trainingszyklus bezeichnet. Um die Lernschleifen zu beenden ist es notwendig, eine

Definition für den Abbruch festzulegen. Verschiedene Kriterien können herangezogen werden, um den Lernfortschritt zu bewerten.

Grundsätzlich erscheint es sinnvoll, die Bewertung des Netzes nicht von der Qualität der Abbildung der zu lernenden Muster auf die Zielwerte abzuleiten, sondern dafür eigene Testbeispiele bereitzustellen. Dadurch kann gewährleistet werden, dass sich das Netz nicht gezielt auf die Trainingsdaten spezialisiert, man spricht in diesem Fall von overfitting, sondern die Fähigkeit zur Generalisierung behält. Bei einem übertrainierten Netz sinkt die Brauchbarkeit der Berechnungsergebnisse für die Testmuster ab einem gewissen Iterationsschritt wieder ab. Abb. 80 verdeutlicht dies anhand des ansteigenden Prozentsatzes an Fehlklassifizierungen bei Daten, die nicht zum Training beigetragen haben.

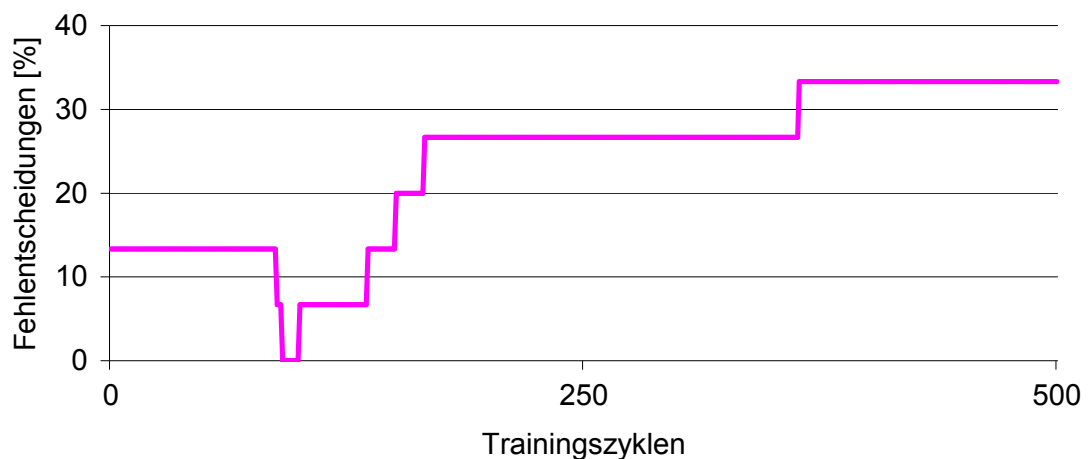


Abb. 80: Anhand von Testmustern wird eine Verschlechterung der Netzqualität durch übermäßige Prägung auf die Trainingsmuster festgestellt

Deshalb ist die Trainingsbasis in zwei Bereiche, die eigentlichen Lernmuster und die Validierungsdatensätze, zu unterteilen. In jedem dieser Teile sollen Daten in geeigneter Qualität und Quantität zur Verfügung stehen.

- Ein Parameter, der den Lernvorgang beschreibt, ist der mittlere Netzfehler E_{mittel} nach (5.7). Unterschreitet dieser einen festgelegten Wert, so kann das Training abgebrochen werden (Abb. 81).

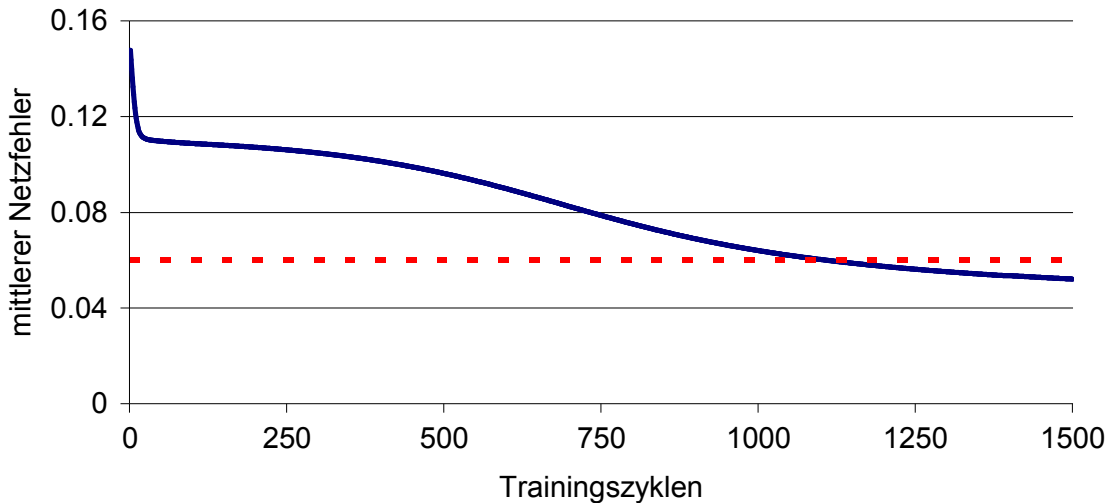


Abb. 81: Abbruch bei Unterschreitung eines Höchstwertes (rot gestrichelt) des mittleren Netzfehlers (blau)

- Für den Benutzer aussagekräftiger ist die Anzahl an Fehlentscheidungen. Der prozentuale Wert gibt Aufschluss über die tatsächliche Leistungsfähigkeit des Netzes, welche anhand der Validierungsmuster überprüft wird. Die Fehlentscheidung ändert sich sprunghaft, da einzelne Muster nur als richtig oder falsch klassifiziert werden. Eine feinere Auflösung erreicht man durch eine größere Anzahl an Testmustern. Der Abbruch erfolgt hier nach Unterschreitung des vorgegebenen Prozentsatzes (Abb. 82).

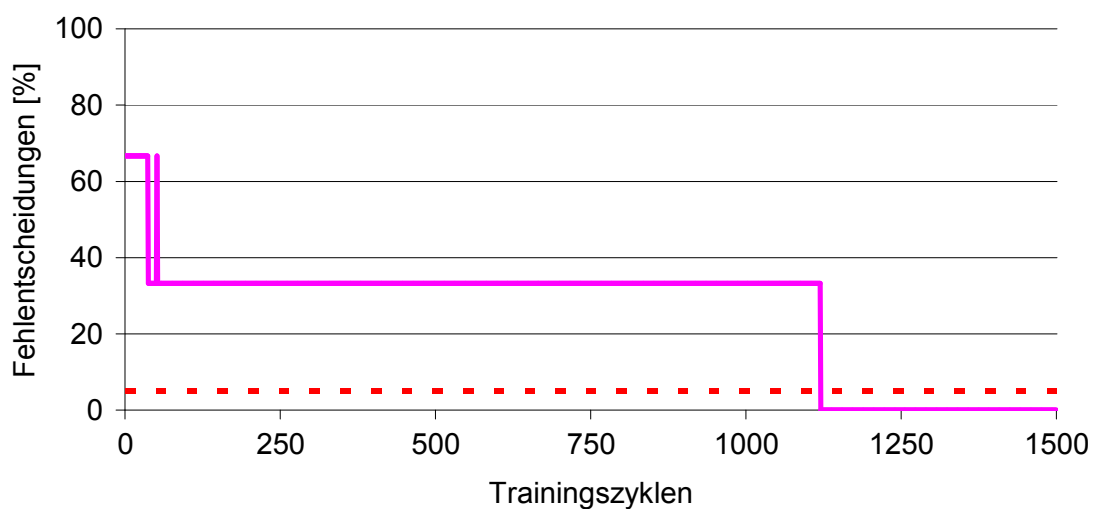


Abb. 82: Abbruch bei Unterschreitung eines Höchstwertes (rot gestrichelt) in den Fehlklassifizierungen (rosa)

- Der Lernfortschritt stagniert, wenn der Änderungsvektor $\Delta \underline{W}$ die Länge 0 besitzt. Das Unterschreiten einer vorgegebenen Länge kann hier als Abbruchkriterium dienen [30]. Infolgedessen gibt die Entwicklung der Kantengewichte Auskunft über den Lernstatus des Netzes. Abb. 83 zeigt die grafische Entwicklung der Kantengewichte während des Lernfortschrittes und veranschaulicht durch die Steigung der einzelnen Kurven das Änderungspotential.

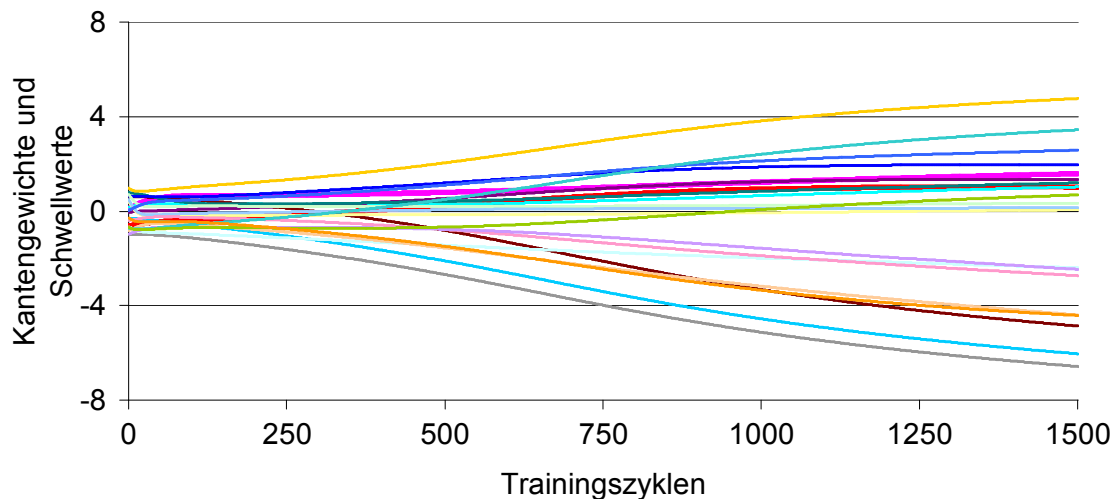


Abb. 83: Entwicklung der Kantengewichte im Netz, Abbruch bei Unterschreitung eines Änderungsdeltas

- Ein immer verwendbarer Abbruchwert ist die Anzahl an Iterationsdurchläufen. Bei Verwendung dieses Wertes zur Steuerung des Lernverhaltens kann allerdings kein Rückschluss auf die Güte des Trainings gezogen werden. Es ist jedoch sinnvoll, die Zahl an Iterationen als zusätzliches Abbruchkriterium zu verwenden. Durch Definition der Gesamtzahl an Trainingszyklen oder mittels Festlegung einer Höchstzahl an Schleifen mit zu geringer Verbesserung sind Endlosschleifen abwendbar. Diese können entstehen, da zum einen nicht per se ein konditionierbares Netz vorliegt – eine unpassende Topologie oder eine ungenügende Trainingsbasis sind mögliche Fehlerquellen – zum anderen ist es nicht immer möglich, das globale Minimum zu erreichen.

6 Bauteilerkennung unter Zuhilfenahme virtueller neuronaler Netze

Wie in Abschnitt 2.1 beschrieben, findet im Bauwesen eine zeitgemäße Planung auf Basis eines dreidimensionalen CAD-Modells statt. Reale Bauteile werden in diesem Ansatz durch Volumenkörper dargestellt, was eine adäquate Repräsentierung ihrer geometrischen Erscheinungsform zulässt. Zur weiteren Verwendung im Sinne eines Produktmodells ist es aber notwendig, die Objekte mit Attributen zu versehen. Die Voraussetzung dafür ist die Kategorisierung der Volumenkörper in Bauteilklassen.

Im Folgenden wird davon ausgegangen, dass die in Kapitel 4 aus Laseraufmaßdaten erzeugten Oberflächenmodelle gültige Beschreibungen von 3D-Körpern darstellen. Die vorliegenden, zunächst noch unspezifizierten Körper werden weiter analysiert und automatisch Bauteilklassen zugeordnet.

Diese Klassifizierungsaufgabe wird mit dem in Kapitel 5 vorgestellten Verfahren der künstlichen neuronalen Netze gelöst. Sie sind in der Lage, nicht explizit formulierte Klassifizierungsregeln durch Beispiele zu erlernen und diese generalisiert auf neue Objekte anzuwenden. Dazu ist die Bereitstellung einer hinreichenden Menge an Eingabemustern notwendig. Virtuelle neuronale Netze eröffnen die Möglichkeit, auch während der Laufzeit weiterzulernen und vom Benutzer korrigierte Entscheidungen in den Erfahrungsschatz, die Trainingsbasis, zu integrieren.

6.1 Vorgehensweise

Die Ausgangsbasis für die Bauteilklassifizierung sind einzelne 3D-Volumenkörper. Diese können prinzipiell beliebig geformt sein, so dass es notwendig ist, sich über die Erhebung aussagekräftiger Parameter für diese Volumenkörper Gedanken zu machen. Sind die Eigenschaften ausgewählt, werden sie in eine dem künstlichen neuronalen Netz zugängliche Form gebracht. Das Netz errechnet mittels der während des Trainings festgelegten Kantengewichte eine Ausgabe für die Eingabeparameter. Dieser Ausgabe ist eine Entscheidung für eine Bauteilkategorie zuzuordnen.

Entsprechend der Klassifizierung wird der Volumenkörper in ein Bauteil überführt. Da in CAD-Systemen Bauteilobjekte mit einem Produktmodell abgebildet werden und diese ihre Geometrie wie in Abschnitt 2.2.5 beschrieben parametrisiert speichern, ist

eine vom eingesetzten CAD-System abhängige Konvertierung von der ursprünglichen Geometriedarstellung in die parameterbasierte Beschreibung notwendig.

6.2 Festlegung und Aufbereitung der Eingangsparameter

Die Basis für eine erfolgreiche Klassifizierung durch ein neuronales Netz sind aussagekräftige Parameter. Diese müssen für alle zu untersuchenden Objekte zur Verfügung stehen. Größen, die nur bei bestimmten Bauteilklassen auftreten, sind bei anderen Typen auf einen neutralen Wert zu setzen, um die Gewichtung dieses Parameters nicht zu verändern. Ferner wird versucht, die Objekte weitgehend unabhängig von ihrer Lage im Gesamtbild auszuwerten. Obwohl aus menschlicher Sicht der Kontext, in dem sich ein Objekt befindet, eine entscheidende Rolle spielt, werden Nachbarschaftsbeziehungen zu anderen Körpern hier nicht ausgewertet. Grund dafür ist, dass solche Objektrelationen nicht geeignet als Attribute für künstliche neuronale Netze aufbereitbar sind. Folgende geometrische Eigenschaften der Volumenkörper wurden als Merkmale untersucht:

- Als vornehmlichste geometrische Eigenschaft gilt die räumliche Ausdehnung eines Körpers. Um auch komplexe Strukturen erfassen zu können, umschließt ein vertikal ausgerichteter Hüllquader den Körper, dessen Kantenlängen als Parameter dienen. Zur Ermittlung der beiden horizontalen Ausdehnungen wird das untersuchte Objekt in die Horizontalebene projiziert. Die Abmessungen des kleinsten umschließenden Rechtecks [64] dieser Projektionspunkte definiert die horizontalen Objektabmessungen x und y . Die Höhe z des Objektes ergibt sich aus den Unterschieden der z -Koordinate des höchsten und des tiefsten Objektpunktes (Abb. 84).

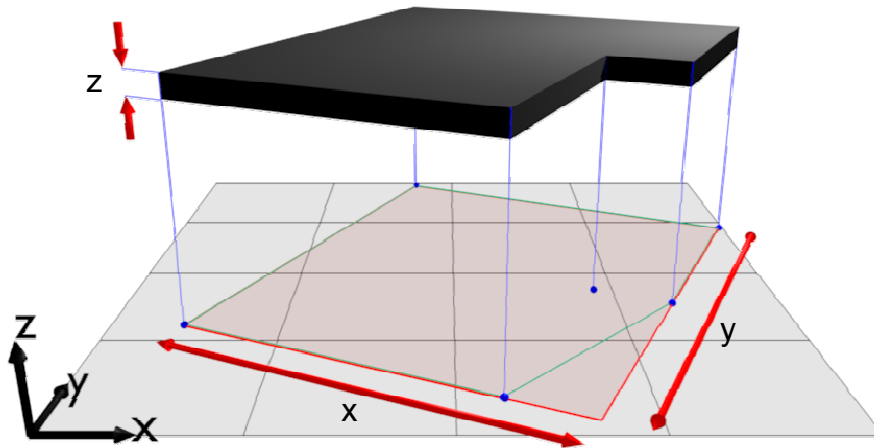


Abb. 84: Ermittlung der Objektausdehnung

- Als weitere Parameter finden die Körperoberfläche A und das umschlossene Volumen V Betrachtung. Aus diesen lässt sich in Zusammenhang mit den räumlichen Abmessungen auf die Gedrungenheit des Körpers schließen.
- Die Prüfung von Verschneidungen des Körpers mit anderen dient der Interpretation der Lage des Objektes innerhalb des Modells. Mit booleschen Operationen kann ein entsprechender Schnittkörper erzeugt werden. In diesem Zusammenhang ist anzumerken, dass nur massebehaftete Elemente als Volumenkörper vorliegen dürfen, nicht aber topologische Bereiche wie der Raum selbst. Eine Einschränkung dieser Forderung ist, dass alle Körper ohne deren herausgeschnittene Öffnungen dargestellt werden, weshalb die (eigentlich masselosen) Öffnungen selbst also notwendigerweise ebenfalls als Volumenkörper repräsentiert sind (Abb. 85).

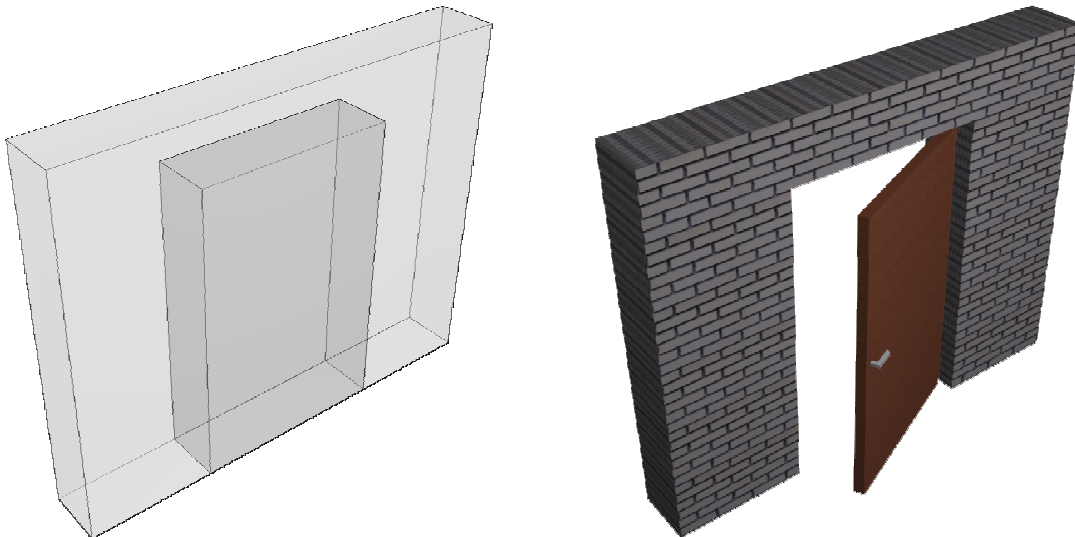


Abb. 85: Eine Wand und eine darin befindliche Türe werden im Volumenmodell als ineinander liegende Volumenkörper repräsentiert

- Eine weitere Deutung der Lage des Objektes erlauben die Objektkoordinaten. Für die betrachteten Volumenkörper wird eine homogene Dichteverteilung angenommen. Unter dieser Prämisse lässt sich über die Objektränder der Massenschwerpunkt ermitteln. Dessen Koordinaten repräsentieren die Position des Objektes und werden ebenfalls als Merkmale herangezogen.

Die vorgestellten Parameter müssen als Fließkommazahlen für den Eingabevektor aufbereitet werden. Das Ergebnis einer Verschneidungsoperation wird wie folgt ausgewertet: Stellt das Volumen des Verschneidungskörpers zu 95% die mathematische Teilmenge des Körpers selbst dar, so wird der Körper unter Vernachlässigung einer praktischen Konstruktionsungenauigkeit als umschlossen oder im anderen liegend angesehen. Für jedes der beiden Objekte wird ein entsprechender Zähler – „liegt innerhalb“ und „beinhaltet“ – inkrementiert. Positive Werte werden als Fließkommazahl in den Eingabevektor übertragen, der Wert 0 durch $-1,0$ ersetzt (siehe Abschnitt 5.3). Längen-, Flächen- oder Volumenzahlen können direkt übernommen werden, ebenso die drei Koordinaten der Position des Objektes.

Als für die gegebene Problemstellung geeignete Parameter konnten folgende vier Werte bestimmt werden:

- x / y : Verhältnis der größeren zur kleineren horizontalen Ausdehnung

- x / z : Verhältnis der größeren horizontalen zur vertikalen Ausdehnung
- "liegt innerhalb von": Anzahl an Objekten, innerhalb deren der aktuelle Körper liegt (im stimmigen Modell 0 oder 1, wobei wie zuvor beschrieben 0 durch $-1,0$ ersetzt wird)
- z-Abstand: Abstand der minimalen z-Werte des aktuellen Objektes und dessen umschließenden (sofern vorhanden transformiert, sonst zu 0 gesetzt). Der z-Abstand wird aus den Objektschwerpunkten sowie den -ausdehnungen in z-Richtung berechnet.

Entgegen der Substitution von 0 durch $-1,0$ bei „keiner Verschneidung“ wird der z-Abstand zu 0 gesetzt, wenn keine Objektverschneidung vorliegt. In diesem Falle ist der Wert nicht definiert. Wie zuvor dargelegt, führt im Lernschritt ein Null-Signal nicht zu einer Veränderung der zugeordneten Kantengewichte. Somit verbleibt die Gewichtung in diesem Fall planmäßig unverändert. Bei vorhandenen Verschneidungen werden die Werte um $1,0$ transformiert, da der Abstand 0 beispielsweise bei einer Türe auftreten kann (Abb. 86) und dort zum Lernprozess beitragen soll.

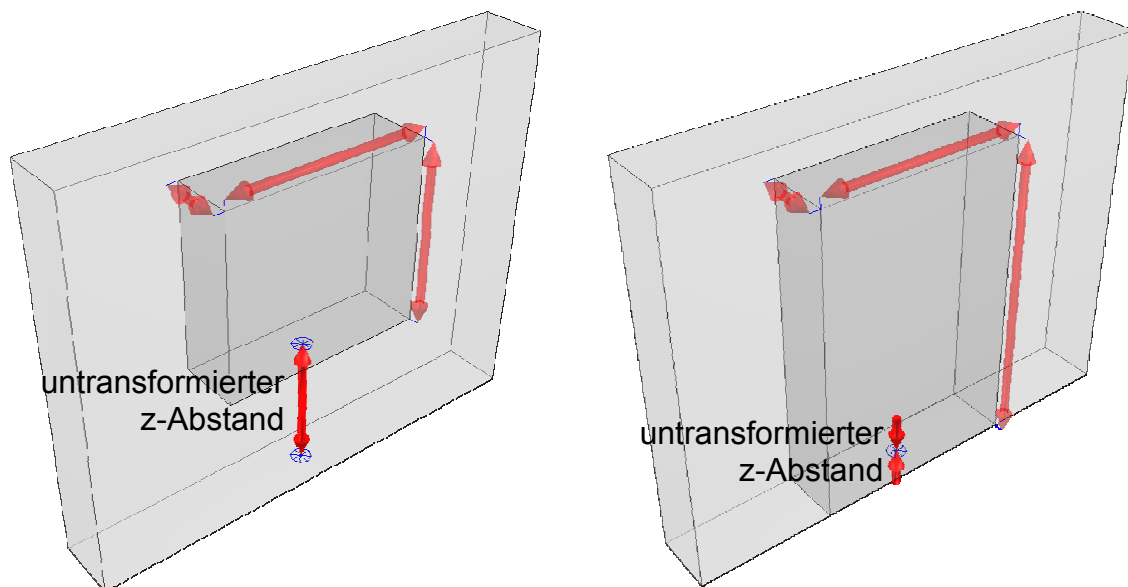


Abb. 86: z-Abstand muss als Eingabewert transformiert werden, um neben Fenstern (links) auch Ungleichheit mit Null bei Türen (rechts) zu erreichen

6.3 Auswertung der Netzentscheidung

Im Rahmen dieser Arbeit wurden sieben verschiedene Bauteilklassen unterschieden (Abb. 87): Die dünnwandigen Klassen sind die vertikal orientierten Wände, Türen und Fenster, sowie die horizontal orientierten Decken und mit deren zugehörigen Durchbrüchen und schließlich die schlanken Objekte Stützen und Balken. Durch diese Festlegung lassen sich Bauwerke schon so weit beschreiben, dass darauf Berechnungen und Variantenstudien wie die des Energieverbrauchs (siehe Kapitel 2.1) möglich sind. Wesentliche Einschränkungen bestehen lediglich bezüglich der Modellierung von Dächern.

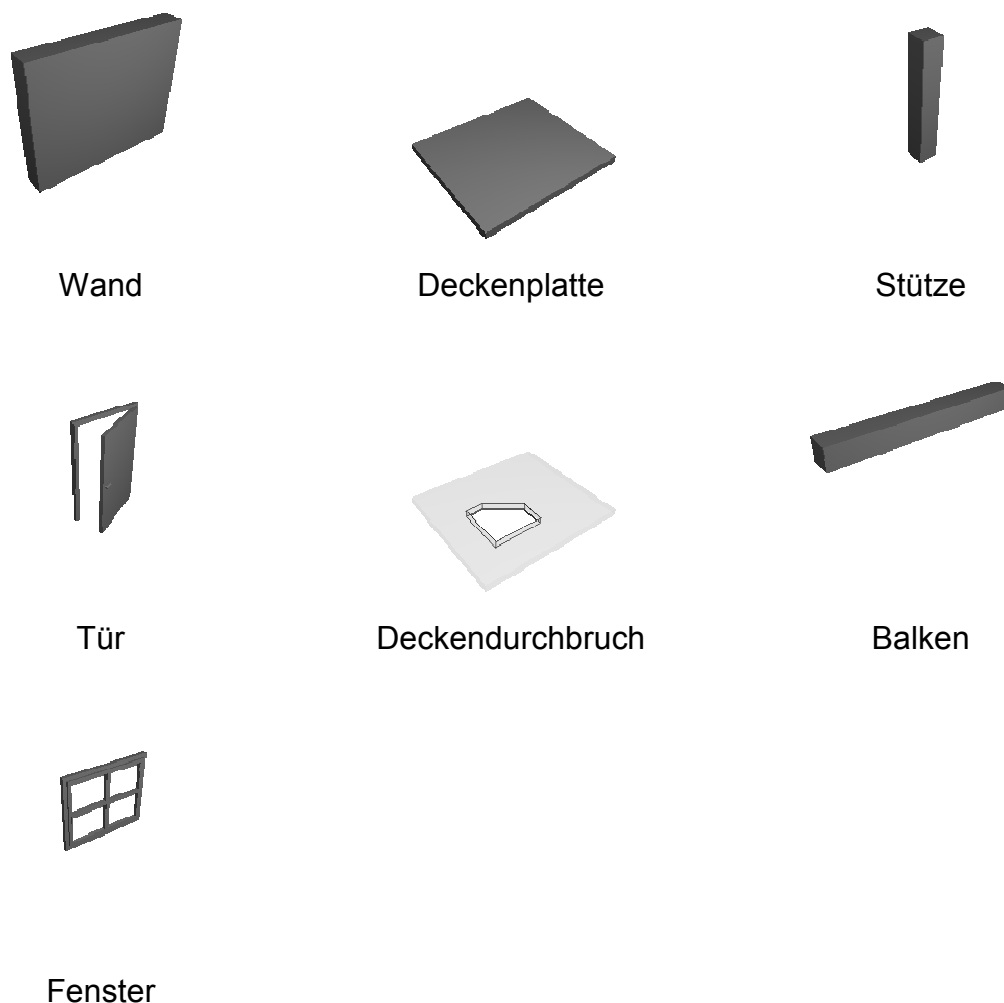


Abb. 87: Erkennbare Bauteilklassen

Aufgrund der unmittelbaren Zuordnung der einzelnen Klassen zu den Neuronen der Ausgabeschicht bestimmt sich deren Anzahl zu sieben (Abb. 88). Die Bauteilkategorie, deren zugeordnetes Neuron den größten Aktivierungszustand aufweist, wird als Netzentscheidung gewertet.

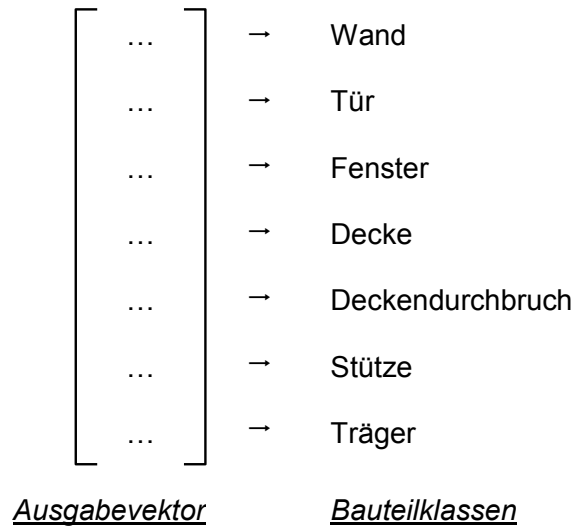


Abb. 88: Den Einträgen des Ausgabevektors zugeordnete Bauteilklassen

6.4 Festlegung der Netztopologie

Durch Studien unterschiedlicher Netzstrukturen [3] konnte ein für die vorliegende Problemstellung passendes Netz entwickelt werden: Zwischen den in Kapitel 6.2 festgelegten vier Eingabeneuronen und den in Kapitel 6.3 beschriebenen sieben Ausgabeneuronen wurde eine Schicht mit sieben verdeckten Neuronen angeordnet (Abb. 89). Diese Topologie erwies sich als am effektivsten.

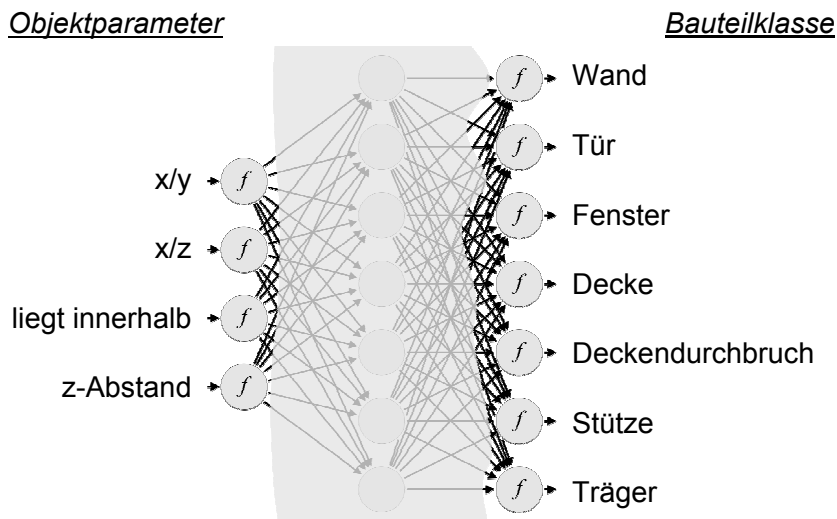


Abb. 89: Verwendetes zweischichtiges Netz der Struktur 4-7-7

Einschichtige Netze scheinen die Komplexität des Problems nicht hinreichend zu erfassen. Der Prozentsatz an Fehlentscheidungen stagnierte dabei bei 9%. Aus der Menge der zweischichtigen Netze konvergierte das gewählte Netz mit sieben verborgenen Neuronen am besten. Wird mehr als ein Neuron der verborgenen

Schicht hinzugefügt oder aus ihr entfernt, so steigt auch der Prozentsatz an Fehlentscheidungen signifikant an. Dreischichtige Netze zeigten eine Verschlechterung im Konvergenzverhalten. Die erste verborgene Schicht erwartet bei der dreischichtigen Topologie ebenfalls eine Mächtigkeit von sechs oder sieben. [3] folgert daraus eine Vorkonditionierung der Eingabedaten durch diese Schicht, wodurch die zweite verborgene Schicht mit weniger Neuronen, beispielsweise drei, auskommen kann.

6.5 Aufstellen der Trainingsbasis

Trainingsdaten können einerseits aus bestehenden CAD-Plänen erhoben oder synthetisch generiert werden. Die geforderte homogene Verteilung der Beispiele wird durch Verwendung synthetischer Muster gewährleistet, entsprechende Direktiven für die automatische Generierung stellen dies sicher. Allerdings muss für die automatische Erstellung von Trainingsdaten die Abbildungsvorschrift der Eingabeparameter auf die gewünschte Netzausgabe formuliert werden. Damit ist der große Vorzug des virtuellen neuronalen Netzes, Entscheidungsregeln nicht explizit formulieren zu müssen, vertan. Deshalb bleiben synthetische Daten akademischen Testumgebungen mit bekanntem Entscheidungsalgorithmus vorbehalten. Da ein solcher für die betrachteten Bauteile nicht vorliegt, scheiden synthetische Muster hier aus.

Als geeignete Beispiele eignen sich bestehende Produktmodelle. Für die in dem Modell enthaltenen Bauteile lassen sich die für das neuronale Netz charakteristischen Merkmale auslesen und so entsprechende Trainingsmuster erzeugen. Dabei müssen aber redundante oder stark ähnliche Bauteile, welche durch ihren Merkmalsvektor ausgedrückt im Merkmalsraum relativ dicht beieinander liegen, vermieden werden. Als identisch einzustufen und somit auszumustern sind Objekte, wenn

- der Unterschied zwischen den Ausdehnungs-Verhältnissen kleiner 0,01 ist,
- beide Objekte den gleichen Wert für „ist enthalten in“ besitzen und
- der Unterschied zwischen den z-Abständen unter einem Millimeter liegt. [3]

Durch die Eliminierung solch redundanter Informationen verringert sich zwangsläufig die Gesamtzahl an Trainingsmustern. Bei einem modernen Hochhaus kann solch eine Filterung die Reduktion des Gebäudes auf die Bauteile eines einzigen

Stockwerkes und noch weniger zur Folge haben. Abb. 90 zeigt dies am Beispiel eines Modells von sechs modellierten Geschößen des Uniqa-Hochhauses in Wien [89]. Nach dem Filtern von Objekten mit ähnlichen Parametern wurden zum Teil mehr als $\frac{5}{6}$ von ihnen eliminiert. Stützen wiederholen sich beispielsweise nicht nur auf unterschiedlichen Stockwerken, sie sind sich großteils auch auf einer Etage ähnlich.

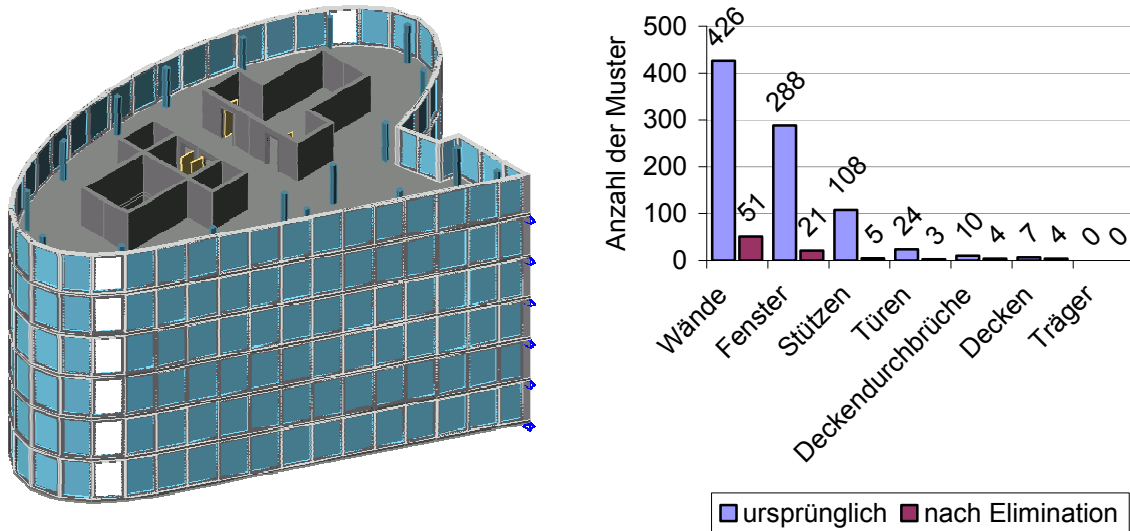


Abb. 90: Ähnliche Bauteile reduzieren die Anzahl der tatsächlichen Trainingsmuster (am Beispiel des Uniqa-Hochhauses in Wien)

Um der geringen Varianz in den Bauteilen zu begegnen, ist es nahe liegend, die Trainingsmuster aus unterschiedlichen CAD-Modellen zu erheben. Werden dabei auch unterschiedliche Gebäudearten berücksichtigt (Abb. 92), so können im Definitionsbereich des Merkmalsraumes große Teile abgedeckt werden.

Aus mehreren unterschiedlichen CAD-Modellen wurden so 1055 Bauteilmuster aus den sieben Bauteilklassen zusammengetragen, die nach Eliminierung als redundant bewerteter Datensätze eine Trainingsbasis mit 329 Mustern ergeben (Abb. 91).

Wände	Türen	Fenster	Decken	Decken- durch- brüche	Stützen	Träger
167	49	80	16	9	5	3

Abb. 91: Zusammensetzung der Trainingsbasis

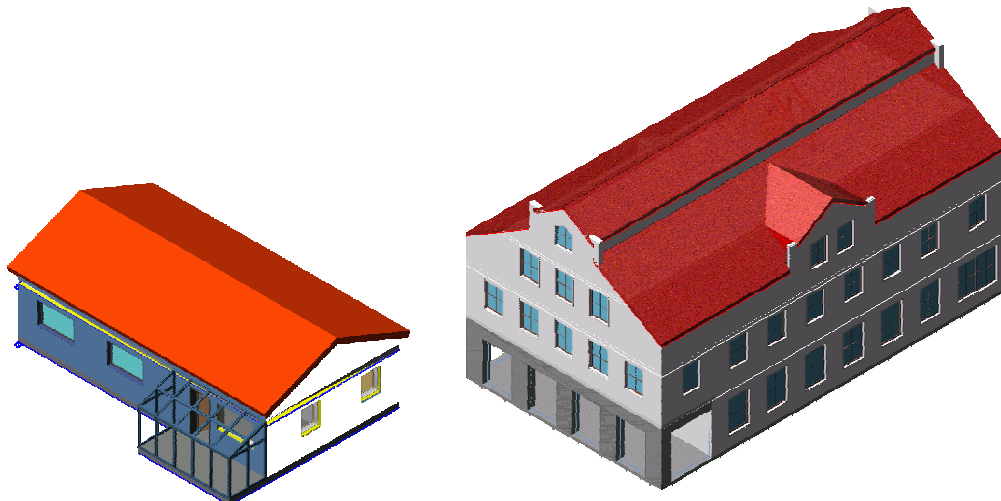


Abb. 92: Ergänzung der Trainingsbasis um Objekte anderer Bauwerkstypen, beispielsweise Einfamilienhaus und Altbau, deren Bauteile deutlich anders ausgeprägt sind

6.6 Konditionierung des Netzes

Das in der Topologie festgelegte Netz erlernt die Klassifizierungsregeln anhand der zusammengestellten Trainingsbasis. Als Lernparameter η wurde ein konstanter Wert von 0,5 gewählt. Für die gegebene Aufgabenstellung war es nicht notwendig, den Lernparameter durch Hilfsternme über die Trainingszyklen (siehe Abschnitt 5.6.3) zu variieren. Das im offline-Verfahren trainierte Netz unterschritt den geforderten mittleren Netzfehler von 0,02 in wenigen Hundert Iterationsschritten.

6.7 Beispielanwendung

6.7.1 Bauteilerkennung mit einem kommerziellen CAD-System

Um das vorgestellte Verfahren der Bauteilerkennung praxisnah zu testen, wurde es in ein CAD-System integriert [68]. Das CAD-System *AutoCAD* von *Autodesk* bietet zur Steuerung und Erweiterung seiner Funktionalität eine Programmier-Schnittstelle, *AutoCAD runtime extension programming environment*, kurz *ObjectARX* genannt. Der Programmierer kann diese Schnittstelle nutzen, um in der Programmiersprache C++ den Funktionsumfang von *AutoCAD* zu erweitern. *Autodesk* nutzt diese Schnittstelle, um dem CAD-System weitere Programmaufsätze hinzuzufügen. So erweitert der *Architectural Desktop (ADT)* Aufsatz das CAD-System um die Möglichkeit, typische Objekte aus dem Bauwesen als Produktmodell abzubilden (Abb. 93). Auch zur Erweiterung des *ADT* wird eine Programmier-Schnittstelle

angeboten: Das *Autodesk AEC Object Modeling Framework* (kurz: *OMF*) ergänzt die Möglichkeiten des *ObjectARX* um die neuen Klassen und Methoden aus dem *ADT*.

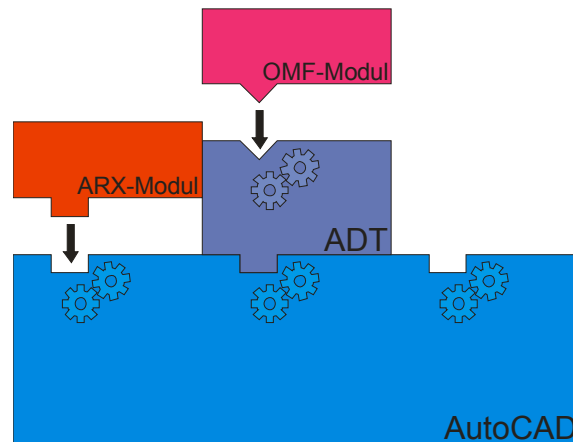


Abb. 93: *Autodesk* bietet mit den Programmierschnittstellen *ObjectARX* und *OMF* die Möglichkeit, die CAD-Software *AutoCAD* bzw. dessen Aufsatz *Architectural Desktop* zu erweitern

Über diese Schnittstellen ist es möglich, die Einzelobjekte innerhalb eines CAD-Modells detailliert zu analysieren. Ferner kann mit ihnen in die Funktionalität des Programms eingegriffen werden. Es lassen sich eigene Abläufe integrieren und aus der Benutzeroberfläche heraus starten. Die entsprechenden Erweiterungsmodule können während der Laufzeit hinzugeladen werden. Damit ist es möglich, innerhalb der CAD-Umgebung bestehende Produktmodelle zu untersuchen und als Trainingsobjekte zu verwenden bzw. die Klassifizierung von nicht näher spezifizierten Volumenobjekten und deren Konvertierung in Bauteile durchzuführen.

Das in vorhergehenden Abschnitten vorgestellte Verfahren zur Bauteilklassifizierung wurde dementsprechend als *OMF-Modul* entwickelt. Über die Kommandozeile des CAD-Systems lässt sich das geladene Modul zur Bauteilidentifizierung konfigurieren. Dabei können die Speicherorte der Trainingsbasis und des Netzes und die Einstellungen für die Lernphase angegeben, sowie das Training angestoßen werden. Die Bauteilerkennung wird ebenso gestartet.

6.7.2 Erhebung der Eingangsparameter

Sofern eine Trainingsbasis nicht bereits vorhanden ist, muss diese zunächst erstellt werden, um das für die Bauteilerkennung angelegte, virtuelle neuronale Netz zu trainieren. Dafür werden bestehende Bauwerksmodelle in die CAD-Umgebung geladen und durch das Modul analysiert. Über vorhandene Funktionen des CAD-

Geometriekerns bestimmt dieses die Abmessungen jedes Einzelobjektes, dessen Schwerpunkt, die Oberfläche der Hülle sowie das Volumen. Des Weiteren wird die Positionierung des Objektes bezüglich der Lage innerhalb anderer Körper durch eine Verschneidungsoperation geprüft. Zusammen mit dem Bauteiltyp werden diese Parameter als Datensatz in einer Textdatei gespeichert. Eine Erweiterung dieser Datenbasis um beliebig viele weitere CAD-Modelle ist jederzeit möglich.

6.7.3 Training

Für die Konditionierung des Netzes werden die Bauteilmuster aus der Trainingsbasis geladen und identische Trainingsmuster anhand der in Kapitel 6.5 angegebenen Ähnlichkeitsgrenzen eliminiert.

Für den Trainingsvorgang müssen die Art und der Wert des Abbruchkriteriums, beispielsweise als Obergrenze des mittleren Netzfehlers, festgelegt werden.

Stehen nicht genügend Trainingsmuster zur Verfügung um die Validierung des Netzes durch getrennte Muster zu prüfen, so ist es möglich, die um ähnliche Trainingsmuster notwendigerweise reduzierte Trainingsbasis sowohl für das Training als auch für die Bewertung anzuwenden.

6.7.4 Klassifizierung

Das trainierte Netz ist in der Lage, Volumenobjekte als Bauteile zu kategorisieren. Dazu muss ein aus Volumenkörpern bestehendes Modell in den Arbeitsbereich des CAD-Programms geladen werden. Das Softwaremodul durchläuft alle im Modellbereich befindlichen Volumenobjekte und übergibt deren charakteristische Merkmale in aufbereiteter Form dem neuronalen Netz. Dieses wertet die Merkmale aus. Die Entscheidung fällt zugunsten der Bauteilklasse, deren zugeordnetes Neuron gemäß Abb. 88 die größte Aktivierung besitzt.

6.7.5 Konvertierung in Bauteile

Wie in Kapiteln 2.2.4 und 2.2.5 erläutert, erfolgt die Beschreibung eines Volumenkörpers durch dessen Oberfläche, speziell durch dessen Eckpunkte, Kanten und Facettenflächen, während Objekte im Produktmodell durch ihre spezifischen Parameter definiert werden. Soll nun ein klassifizierter Volumenkörper als Produktmodell dargestellt werden, so ist dessen Geometrie im eingesetzten CAD-System *Architectural Desktop* parametrisiert zu modellieren. Eine automatische Konvertierung durch das CAD-System ist in der vorliegenden Version nicht möglich.

Deshalb ist die Erstellung der produktmodellbasierten Repräsentierung der CAD-Körper durch eigene Softwareimplementierung durchzuführen.

Grundsätzlich ist für diese Modellierung festzulegen, welche Maße des Volumenmodells in welche Parametergrößen des Produktmodells übertragen werden. Wie gut dieses Vorhaben gelingt, hängt nicht nur von der richtigen Interpretation der Ursprungsgeometrie, sondern auch von den Möglichkeiten des Geometrikerns im CAD-System ab. Anhand des Beispiels einer Wand sei dies im Folgenden näher beschrieben.

Die beiden Stirnflächen (A) der hier modellierten Wand (Abb. 94) sind leicht abgeschragt, sowohl die Unter- (B) wie die Oberkante (C) liegen nicht in der Horizontalebene. Die Oberkante enthält zudem einen Durchbruch (D). Zwei Öffnungen durchbrechen die Wand in einem leicht schrägen Winkel. Eine der Öffnungen (E) ist rechteckig und parallel zur Vertikal- und Horizontalachse, die andere (F) ein gedrehtes Rechteck mit einem zusätzlichen Ausschnitt. Schließlich durchdringt noch eine zur Wandoberfläche nicht parallele Nische (G) das Objekt in einer unteren Ecke etwa bis zu Mittelebene.

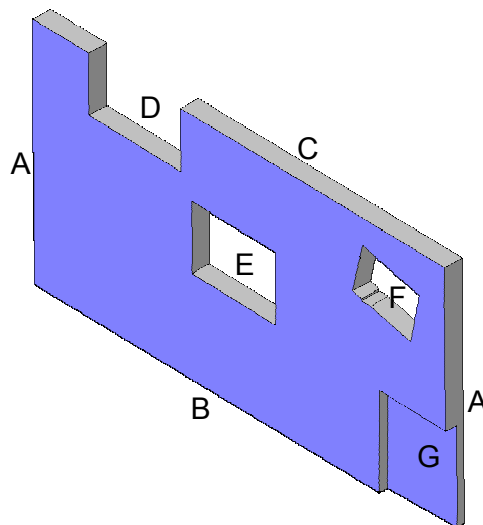


Abb. 94: Testobjekt eines Volumenkörpers zur Konvertierung in ein parametrisiert modelliertes Produktmodell

Nach der Klassifizierung des Volumenobjekts als Wand werden in das Objekt Hauptachsen gelegt. Alle im Rahmen dieser Arbeit klassifizierten Bauteilklassen (siehe Abb. 87) haben gemein, dass sie üblicherweise eine geometrische Ausrichtung entlang der Hauptachsenrichtungen besitzen. Bauteilparameter im Produktmodell entsprechen diesen Ausdehnungen. Den hier verwendeten Wänden

wird eine vertikale Ausrichtung unterstellt, weshalb das Objekt so zu rotieren ist, dass die Hauptachse mit der kleinsten Abweichung zur vertikalen Systemachse auf dieser zu liegen kommt. Die beiden anderen Hauptachsen, welche nun in horizontale Richtungen zeigen, werden verwendet, um die Objektabmessungen Länge und Dicke zu ermitteln. Diese Werte leiten sich aus den Punktkoordinaten des Volumenobjektes ab, welche auf die beiden Hauptachsenvektoren projiziert werden. Um die Höhe des Objektes zu bestimmen, sucht man die minimale und maximale z-Koordinate der Punkte. Somit kann die Wand in Parameterform mit Länge, welche die Basislinie der Wand festlegt, Dicke und Höhe konstruiert werden (Abb. 95).

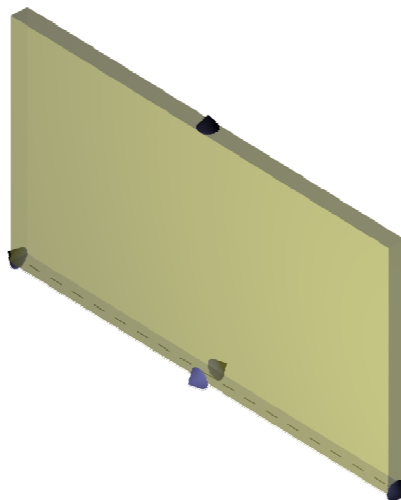


Abb. 95: Festlegung der Basislinie (gestrichelt), der Dicke und Höhe der Wand

Im nächsten Schritt werden die Ober- und Unterkante festgelegt. Eine vertikale Schnittebene (S) entlang der Basislinie erstellt den Wandumriss sowie die Konturen der Öffnungen (Abb. 96): Die größte Schnittregion legt die Silhouette der Wand fest. Zunächst wird diese beidseitig von den Stirnseiten der Wand her um 2,5 cm gestutzt. Dieser heuristisch festgelegte Wert dient bei nicht vertikal stehenden Stirnflächen dazu, dass Ober- und Unterkante eindeutig trennbar sind. Die beiden Enden der Kanten werden bis zur Länge der Basislinie horizontal fortgeführt (Abb. 97). Damit entsteht zwangsläufig ein senkrechter Wandabschluss an den Stirnseiten.

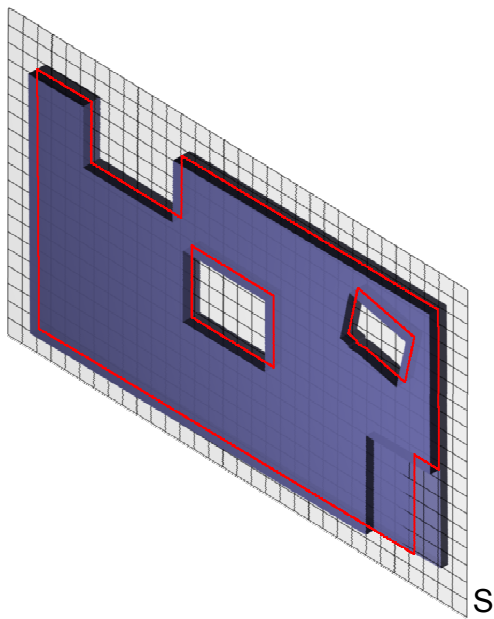


Abb. 96: Mit Hilfe der Schnittebene S erzeugte Konturlinie

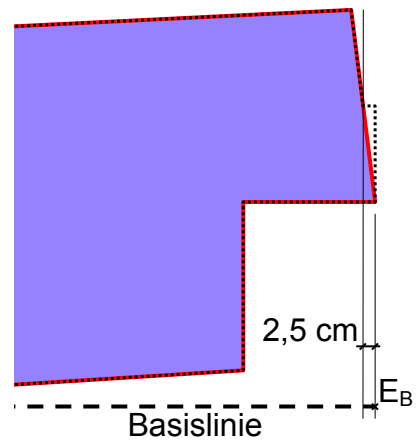


Abb. 97: Die Konturlinie (rot) wird um 2,5 cm gestutzt und wieder bis zum Basislinienendpunkt E_B extrudiert (gestrichelte Linie)

Mit diesem Vorgehen wird der Durchbruch (D) in der Oberkante korrekt dargestellt, die Nische (G) hingegen erscheint nur zum Teil, da sie wegen ihrer partiellen Durchdringung lediglich in diesem Bereich von der Schnittebene (S) erfasst wurde (Abb. 98).

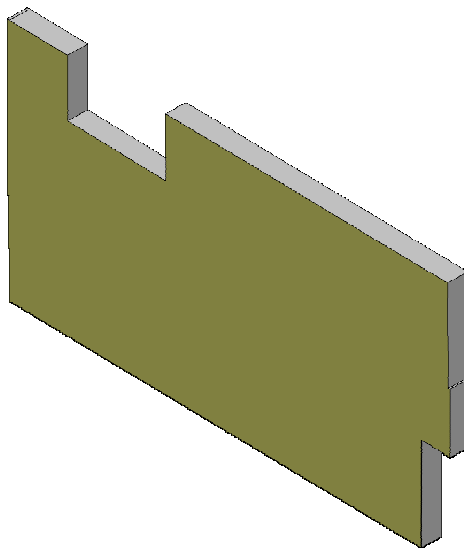


Abb. 98: Modellierung der Ober- und Unterkante der Wand

Die beiden Öffnungen (E, F) ergeben im Schnitt der Ebene (S) mit dem Volumenkörper geschlossene Regionen. An ihrer Stelle werden Öffnungen in der Wand platziert. Das eingesetzte CAD-System bietet nur Öffnungen mit achsenparallelem Rechteckquerschnitt an, weshalb für die gedrehte Öffnung (F) ein vertikal ausgerichteter Hüllquader die Abmessungen für die neue Öffnung liefert.

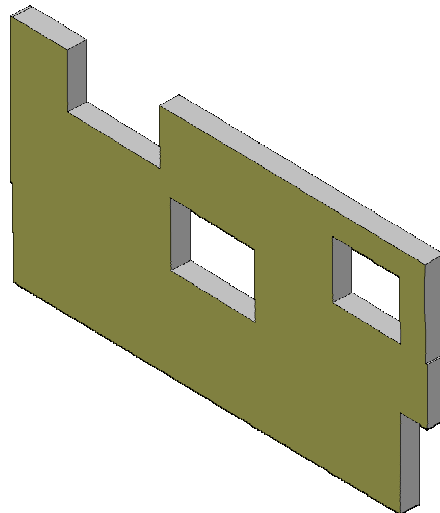


Abb. 99: Erkannte Wanddurchbrüche werden mit Öffnungen modelliert

Das CAD-System bietet zudem sogenannte Modifikatoren für Wände an, mit denen die hier noch einheitliche Dicke der Wand verändert werden könnte. Sie dienen speziell dazu, Sockel zu modellieren. Eine Dickenänderung über die Länge oder Höhe ist mit ihnen nicht praktikabel umsetzbar. Die bis hierher vorgestellte Modellierung ergibt jedoch bereits eine gute Annäherung an die Ursprungsgeometrie.

Wie gezeigt werden konnte, ist zur korrekten Übertragung der Geometrie vom Volumenkörper auf ein parametrisiertes Produktmodell ein aufwändiger Modellierungsprozess notwendig, der zum Teil durch die Fähigkeiten der im CAD-System verfügbaren Objekte beschränkt wird. Im Rahmen dieser Forschungsarbeit wurde dies exemplarisch für eine ausgewählte Anzahl an Bauteilklassen demonstriert.

6.7.6 Beispiele

Das Berandungsmodell der in Abschnitt 4.7.3 vorgestellten Stütze (Abb. 42) konnte nach der Modellierung als Volumenkörper erfolgreich als Bauteil Stütze identifiziert werden (Abb. 100). Die Systemachse des gemäß der Geometrie des Berandungsmodells rekonstruierten Bauteils ist korrekt in Vertikalrichtung festgelegt. Als

parameterbasiertes Objekt besitzt die Stütze Objekt-Griffe (Abb. 100, markiert), über welche der Benutzer die Geometrie interaktiv verändern kann. Die Objektoberfläche passt sich den festgelegten Endpunkten und dem Profil dynamisch an (Assoziativität).

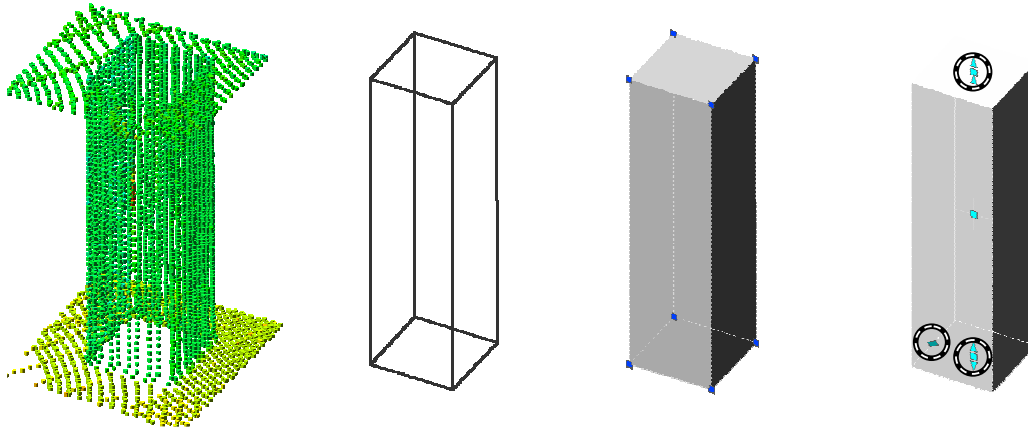


Abb. 100: Die aus der Punktwolke extrahierte Oberflächengeometrie wird im CAD-System als Volumenobjekt erzeugt und nach der Bauteilerkennung in eine Stütze überführt

Als zweites Beispiel steht die ebenfalls in Abschnitt 4.7.3 automatisch erzeugte Geometrie einer Wand zur Verfügung (Abb. 61). Durch manuelle Nacharbeit wurde gemäß der Punktwolke die Tür als zusätzliches Objekt angelegt (Abb. 101). Das neuronale Netz identifizierte die beiden Objekte korrekt, die anschließende parameterbasierte Modellierung als Bauteilmodell liefert die formgerechte Geometrie im Produktmodell. Da Türen nicht detaillierter unterschieden werden, wurde die ursprünglich zweiflügelige Türe durch eine einfache Türe im Modell ersetzt (Abb. 102). Das entstandene Produktmodell bietet den Vorteil, die parametrisierten Objekte unkompliziert zu modifizieren.

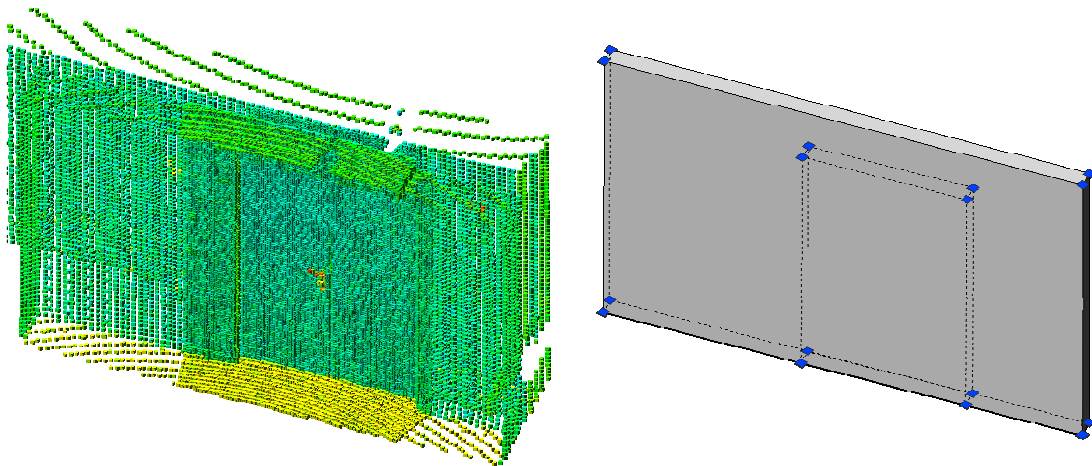


Abb. 101: Die auf Basis der Punktwolke rekonstruierten Volumenkörper Wand und Türe sind die Ausgangsbasis für die Bauteilidentifizierung

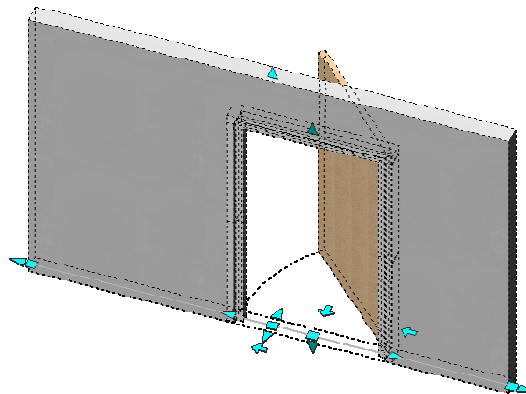


Abb. 102: Passende Überführung in die zugehörigen Produktmodell-Objekte der erfolgreich identifizierten Wand und Türe

Beim letzten Beispiel handelt es sich um ein umfangreiches Volumenmodell eines Altbaus über zwei Geschoße. Im Rahmen einer Forschungsarbeit [68] wurde dieser Gebäudebereich innerhalb eines Altbaus vermessen. Die als Ausgangsbasis für die Bauteilerkennung dienenden Volumenkörper wurden wegen der großen Komplexität nicht durch die in Kapitel 4 beschriebenen Methoden, sondern interaktiv im CAD-System erzeugt (Abb. 103). Diese etwa einhundert Objekte waren anschließend in ein Produktmodell zu konvertieren: Die in Kapitel 5 vorgestellte Methode identifizierte erfolgreich alle Bauteile und erstellte diese parametrisiert im CAD-System (Abb. 104). Die Analyse durch das neuronale Netz und die Überführung in ein Produktmodell benötigte eine Rechenzeit von etwa einer Minute (PC, Intel Pentium 4, 2,4 GHz). Die interne Datenhaltung der einzelnen Objekte innerhalb des CAD-Systems erfordert

dabei komplexe Zugriffsoperationen, insbesondere für schreibende Schritte während der Modellierung. Deshalb beansprucht die Teilaufgabe der parameterbasierten Neumodellierung im Verhältnis zur Klassifizierung den signifikant größeren Zeitaufwand.

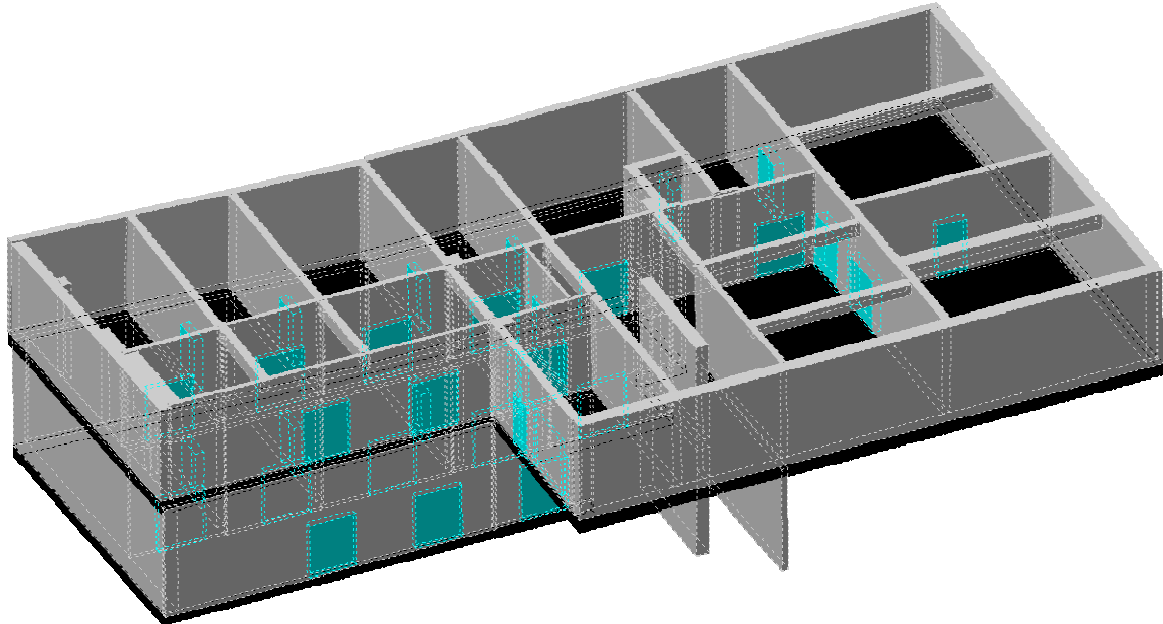


Abb. 103: Das im Rahmen des Forschungsprojektes BayFORREST F248 [68] analysierte Demonstrationsbauwerk, der Lorenzistock des Alten Hofes in München, wurde auf zwei Geschoßen per Laseraufmaß erfasst und von Hand als CAD-Modell aufbereitet

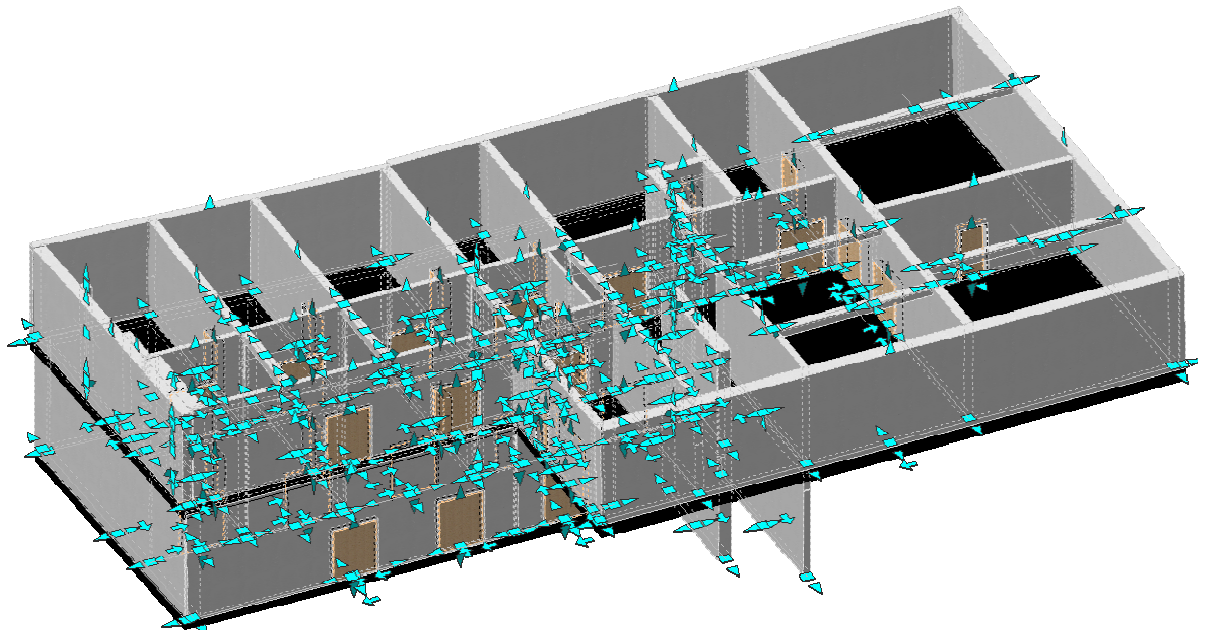


Abb. 104: Ergebnis der erfolgreichen Konvertierung in ein Produktmodell

7 Zusammenfassung und Ausblick

Die vorliegende Arbeit stellte einen nahtlosen Weg vom Aufmaß im Baubestand zum planungstauglichen Produktmodell vor. Schwerpunkt lag dabei auf einem hohen Automatisierungsgrad.

Zunächst wurde in Kapitel 2 diskutiert, welchen Mehrwert eine computergestützte, dreidimensionale Planung auf Basis eines Produktmodells bietet: Neben einem signifikanten Zugewinn an Nutzen und Effektivität durch Verwendung eines digitalen Bauwerksmodells ermöglicht dieses viele aus Inkonsistenzen resultierende Planungsfehler zu vermeiden. Es wurde gezeigt, welche unterschiedlichen Möglichkeiten es zur Repräsentierung von Objekten im dreidimensionalen Raum gibt. Dabei kann festgehalten werden, dass ein objektorientiertes Produktmodell die Realität gut approximiert. Die Grenzen der Rechenleistung und des Arbeitsspeichers verlangen die Optimierung der Verarbeitung dreidimensionaler Daten. Zu diesem Zweck wurden Methoden vorgestellt, die den Datenzugriff beschleunigen und eine Reduktion der Daten unter der Prämisse eines gering zuhaltenden Informationsverlustes erlauben.

Mit verschiedenen Aufmaßverfahren setzte sich Kapitel 3 auseinander. Dabei wurde festgestellt, für welche Anwendungsbereiche eine Effizienzsteigerung mit modernen, berührungslos messenden Geräten möglich ist. Der Einsatz terrestrischer Laserscanner lässt eine Datenerfassung ohne menschliches Eingreifen zu und stellt so den Baustein auf Seite der automatischen Datenerhebung. Die Prozesse Geometrierhebung und Aufbereitung als Gesamtpunktwolke arbeiten wegen ihres hohen Automatisierungsgrades effizient.

Das Kapitel 4 beschäftigte sich mit der Auswertung der gewonnenen Aufmaßdaten. Zunächst fand eine Auseinandersetzung mit dem Stand der Technik und dem Stand der Forschung zu diesem Thema statt. Die direkte Konvertierung von Punktwolken in Körper ist bis heute nur semiautomatisch möglich. Deshalb befasst sich die Literatur schwerpunktmäßig mit der schrittweisen Aufbereitung. Die Dreiecksvernetzung von Punktwolken im Raum ist ein vieldiskutiertes Thema, für das effiziente Verfahren zur Verfügung stehen. Es kommen dazu Abwandlungen des Delaunay-Algorithmus zur Anwendung. Mit einer Segmentierung der Dreiecke lassen sich dahinterliegende Geometrien erkennen und die gruppierten Dreiecke durch analytische Flächenbeschreibungen substituieren. Über die Berechnung der Verschneidung der

analytischen Flächen kann schließlich auf ein Oberflächenmodell geschlossen werden.

Der dargestellte, eigene Ansatz zur Erzeugung eines Oberflächenmodells erfolgt ebenfalls schrittweise. Zunächst musste eine Aufbereitung der gemessenen Punkte erfolgen. Die Punktwolke wurde dabei an Orten lokaler Verdichtungen ausgedünnt. Als nächster Teilschritt erfolgte eine Dreiecksvernetzung in Anlehnung an die räumliche Delaunay-Triangulierung. Zur Beschleunigung des Verfahrens wurde der betrachtete Teilraum durch einen Oktaibaum diskretisiert. Die Dreiecke konnten anschließend segmentiert und die Ausgleichsebenen der Segmente zur Bestimmung der Eckpunkte, Kanten und Teilflächen herangezogen werden. Diese bestimmen das Oberflächenmodell.

Anhand von Beispielen wurde der Arbeitsablauf praxisnah demonstriert. Wie sich dabei bestätigte, besitzt die Beschaffenheit der Punktwolke großen Einfluss auf das Ergebnis. Deshalb ist der Aufbereitung der Messpunkte verstärkt Aufmerksamkeit zu widmen. Während im vorgestellten Schritt für die Punkte eine globale Ausdünnung der Dichte vorgenommen wurde, kann über eine Erweiterung des Algorithmus eine stetige Änderung der Punktdichte erreicht werden. Die Segmentierung, welche im vorliegenden Fall nur ebene Flächen behandelt, wäre in diesem Fall bezüglich der Zugehörigkeitsregel dynamisch anpassbar zu entwickeln, das heißt in Bereichen höherer Dichte muss eine größere Toleranz für die Rauigkeit der Oberfläche zulässig sein. So kann auf die gerätetechnisch bedingten Messungenauigkeiten reagiert werden. Künftige Aktivitäten werden sich ferner auch mit der Segmentierung nicht-ebener Flächen zu befassen haben.

Mit virtuellen neuronalen Netzen setzte sich Kapitel 5 auseinander. Mit ihnen ist es möglich, Datensätze zu klassifizieren, ohne das dahinterliegende Regelwerk zu kennen. Wie gezeigt wurde, erlernt ein virtuelles neuronales Netz zunächst die Klassifizierung bekannter Beispiele und kann anschließend durch seine Generalisierungsfähigkeit unbekannte Muster erkennen. Unter Verwendung des Back-Propagation-Algorithmus werden in der Lernphase die Freiheitsgrade eines neuronalen Netzes an die Kategorisierung der Trainingsmuster angepasst.

Kapitel 6 nutzte die Methode der virtuellen neuronalen Netze und stimmte diese auf die Randbedingungen einer Bauteilerkennung ab. Als geeignete charakteristische Größen, anhand derer die Objekte zu unterscheidenden sind, wurden die Abmessungen eines Hüllquaders, die Volumenzahl des Körpers, die Oberflächenzahl

der Objekthülle, Überschneidung der Körper sowie die globale Position untersucht. Diese Werte sind als Eingabeparameter geeignet aufzubereiten. So werden dem Netz zwei Quotienten der Abmessungen, die Anzahl der das Objekt umschließenden anderen Körper sowie gegebenenfalls der vertikale Abstand der Unterkanten des Objektes zum größten umschließenden übergeben. Für die Klassifizierung von sieben Bauteilklassen erwies sich ein zweischichtiges Netz mit sieben verborgenen Neuronen als am günstigsten. Jeder der Bauteilklassen wurde zur Auswertung der Entscheidung des Netzes ein Ausgabeneuron zugeordnet. Um das Netz zu trainieren, waren passende Objekte zusammenzustellen, wobei ein Teil der Muster zur Überprüfung des Lernverhaltens diente. Eine Umsetzung des Verfahrens fand durch die Erweiterung eines CAD-Systems statt. So standen die zu trainierenden Bauteile wie auch die unbekanntes, zu klassifizierenden Objekte als CAD-Modelle direkt zur Verfügung. An mehreren Beispielen konnte erfolgreich die praktische Tauglichkeit der Methode gezeigt werden.

In dieser Arbeit werden Objekte jener Klasse zugeordnet, die das größte Votum durch die Ausgabeneuronen erfahren hat. Eine mögliche Verbesserung wäre die Abfrage einer Benutzerunterstützung, sobald das Gewinnerneuron nicht mit eindeutigem Abstand Sieger ist. Die Information aus dieser Benutzerinteraktion oder bei Korrekturen eventueller Fehlentscheidungen dient anschließend der Verbesserung der Trainingsbasis. Für den praktischen Einsatz müssen weitere Bauteilklassen erkannt werden können. Insbesondere fehlen Elemente aus der Dachkonstruktion.

Das gezeigte Vorgehen ermöglicht für ebene Geometrien die vollautomatische Erfassung als Punktwolke und deren Überführung in ein Produktmodell. So konnte mit dieser Arbeit ein wirtschaftlicher Weg zur Erzeugung eines Bauteilmodells vorgestellt werden.

Literaturverzeichnis

- [1] AIHONO (idF v. 10.11.2001)
- [2] ARNOLD, Volker ; DETTMERING, Hendrik ; ENGEL, Torsten ; KARCHER, Andreas: *Product Lifecycle Management beherrschen : Ein Anwenderhandbuch für den Mittelstand*. Berlin : Springer, 2005
- [3] BASTIAN, Almut: *Bauteilklassifizierung mit Hilfe virtueller neuronaler Netze*. München, Technische Universität, Lehrstuhl für Bauinformatik, Dipl.-Arb., 2005
- [4] BAUM, Eric B. ; HAUSSLER, David: What Size Net Gives Valid Generalization? In: *Neural computation* 1 (1989), S. 151-160
- [5] BERKHAHN, Volker: *Geometrische Modellierung in der Bauinformatik*. Aachen : Shaker, 2005
- [6] BERKHAHN, Volker ; TILLEKE, Sandra ; SCHLEINKOFER, Matthias ; RANK, Ernst: Re-Engineering im Konstruktiven Ingenieurbau : Von der Bauzeichnung zum Produktmodell. In: *Bauingenieur* (2005), Nr. 11, S. 509-516
- [7] BERNREUTHER Martin: *Geometrische Modellierung mit Simplizialkomplexen : Vom CAD-Modell zur numerischen Analyse*. Stuttgart, Universität, Fachgebiet Informationsverarbeitung im Konstruktiven Ingenieurbau, Diss., 2002
- [8] BEYER, Anderas ; LOHOFF, Markus (Hrsg.): *Bild und Erkenntnis : Formen und Funktionen des Bildes in Wissenschaft und Technik*. München : Deutscher Kunstverlag, 2005
- [9] BEZDEK, James C.: Pattern analysis. In: RUSPINI, Enrique H. (Hrsg.) ; BONISSONE, Piero P. (Hrsg.) ; PEDRYCZ, Witold (Hrsg.): *Handbook of fuzzy computation*. Bristol : Institute of Physics Publishing, 1998
- [10] BÖHM, Jan ; FRITSCH, Dieter: Automated Interpretation of Dense Range Data. In: *Proceedings of SPIE Videometrics VIII*. Santa Clara (USA) : SPIE, 2003 (SPIE 5019), S. 77-86
- [11] BÖHM, Jan ; PATERAKI, Maria: From Point Sample to Surfaces – On Meshing and Alternatives. In: *International Archives on Photogrammetry and Remote Sensing*. Bd. 36, Tl. 5. Dresden : ISPRS Commission V Symposium, 2006, S. 50-55

- [12] BORRMANN, André: *Computerunterstützung verteilt-kooperativer Bauplanung durch Integration interaktiver Simulationen und räumlicher Datenbanken*. München, Technische Universität, Lehrstuhl für Bauinformatik, Diss., 2007
- [13] BRAUSE, Rüdiger: *Neuronale Netze : Eine Einführung in die Neuroinformatik*. Stuttgart : Teubner, 1995
- [14] BRENNER, Claus: *Dreidimensionale Gebäuderekonstruktion aus digitalen Oberflächenmodellen und Grundrissen*. Stuttgart, Universität, Institut für Photogrammetrie, Diss. 2000
- [15] BRENNER, Claus ; KOLBE, Thomas H.: Neue Perspektiven : Wie 3D-Stadtmodelle erstellt und verwendet werden. In: *c't* (2005), Nr. 15, S. 106-111
- [16] BREYER, Benjamin: Die 3. Dimension : 3D-Laserscanning: Messverfahren für Denkmalpflege und Dokumentation. In: *Bautenschutz und Bausanierung* (2006), Nr. 2, S. 40-42
- [17] BRINGMANN, Oliver: Die Punktwolke aus Sicht der Mustererkennung. Laserscanning im Architekturaufmaß. In: *Vermessungsingenieur* (2002), Nr. 2, S. 120-122
- [18] BUNDESMINISTERIUM FÜR VERKEHR, BAU UND STADTENTWICKLUNG (Hrsg.): *Tiefensee: Energieausweis bringt Kosten-Transparenz und hilft beim Klimaschutz : Bundesrat stimmt Energieausweis zu : Pressemitteilung vom 8. Juni 2007, Nr. 153/2007*. URL <http://www.bmvbs.de/-,302.998479/Tiefensee-Energieausweis-bring.htm>. – Aktualisierungsdatum: 08.06.2007 – mailto: buergerinfo@bmvbs.bund.de. – Bundesministerium für Verkehr, Bau und Stadtentwicklung
- [19] BUNGARTZ, Hans-Joachim ; GRIEBEL, Michael ; ZENGER, Christoph: *Einführung in die Computergraphik : Grundlagen, Geometrische Modellierung, Algorithmen*. Braunschweig : Vieweg, 1996
- [20] BURGHARDT, Michael: *Parallele Netzgenerierung für ebene und räumliche Problemstellungen aus dem Bauwesen*. Darmstadt, Technische Universität, Institut für Numerische Methoden und Informatik im Bauwesen, Diss., 2001

-
- [21] CROUSE, Bernd: *Lattice-Boltzmann Strömungssimulationen auf Baumdatenstrukturen*. München, Technische Universität, Lehrstuhl für Bauinformatik, Diss., 2003
- [22] DALINGHAUS, Klaus: *Realisierung und Optimierung eines Neuro-Fuzzy Systems zur Erkennung rhythmischer Muster*. Osnabrück, Universität, Institut für Kognitionswissenschaft, Mag.-Arb., 2005
- [23] DELAUNAY, Boris N.: Sur la sphère vide. In: Bulletin of Academy of Sciences of the USSR 7 (1934), Nr. 6, S. 793-800
- [24] DONATH, Dirk ; RICHTER, Katharina ; PETZOLD, Frank ; THUROW, Torsten: A building information system based on a planning-relevant surveying system – a module in a comprehensive computer-aided project planning. In: HSIEH, Shang-Hsien (Hrsg.): *Proceedings of the 9th International Conference on Computing in Civil and Building Engineering (ICCCBE)*. Taipei (Taiwan) : ICCBE, 2002, S. 1291-1296
- [25] DRIXLER, Erwin: *Analyse der Form und Lage von Objekten im Raum*. München : Verlag der Bayerischen Akademie der Wissenschaften, 1993
- [26] EGGER, Martin ; VAN TREECK, Christoph ; RANK, Ernst ; STEGER, Martin: Produktmodell-basierte ökologische Lebenszyklus- und Kostenanalyse für Bauwerke. In: Koenigsdorff, Roland (Hrsg.) ; van Treeck, Christoph (Hrsg.): *BauSim 2006 : Tagungsband*. München : IBPSA Germany, 2006, S. 28-30
- [27] Ekkerlein, Christian: *Ökologische Bilanzierung von Gebäuden in frühen Planungsphasen auf Basis der Produktmodellierung*. München, Technische Universität, Lehrstuhl für Bauinformatik, Diss., 2004
- [28] ENEV (idF v. 2.12.2004)
- [29] GODFREY, Michael D. ; HENDRY, David F.: The computer as von Neumann planned it. In: *IEEE Annals of the History of Computing* 15 (1993), Nr. 1, S. 11-21
- [30] HAYKIN, Simon S.: *Neural Networks : A Comprehensive Foundation*. New York : Macmillan College Publishing, 1994
- [31] HOFFMANN, Gernot: *Circle in 3D*. URL <http://www.fh-empden.de/~hoffmann/circle3d06092005.pdf>. – Aktualisierungsdatum: 03.03.2007

– [mailto: hoffmann@fho-emden.de](mailto:hoffmann@fho-emden.de). – Fachhochschule Oldenburg / Ostfriesland / Wilhelmshaven

- [32] HOFFMANN, Stefan: *Nachweise der Energieeinsparverordnung auf der Grundlage eines IFC basierten Produktmodells*. München, Technische Universität, Lehrstuhl für Bauinformatik, Dipl.-Arb., 2001
- [33] HOPPE, Hugues: *Surface Reconstruction from Unorganized Points*. Seattle, University of Washington, Department of Computer Science and Engineering, Diss., 1994
- [34] HOPPE, Hugues ; DEROSE, Tony ; DUCHAMP, Tom ; McDONALD, John ; STUETZLE, Werner: Surface Reconstruction from Unorganized Points. In: *ACM SIGGRAPH Computer Graphics* 26 (1992), Nr. 2, S. 71-78
- [35] HORNIK, Kurt ; STINCHCOMBE, Maxwell ; WHITE, Halbert: Multilayer Feedforward Networks are Universal Approximators. In: *Neural Networks* 2 (1989), S. 359-366
- [36] HUSH, Don R. ; HORNE, Bill G.: Progress in Supervised Neural Networks : What's New Since Lippmann? In: *IEEE signal processing magazine* 10 (1993), S. 8-39
- [37] INDUSTRIEALLIANZ FÜR INTEROPERABILITÄT (Hrsg.): *buildingSmart – IFC*. URL http://www.buildingsmart.de/1/1_00_03.htm. – Aktualisierungsdatum: 17.2.2006 – [mailto: info@buildingsmart.de](mailto:info@buildingsmart.de). – Industrieallianz für Interoperabilität e. V.
- [38] INSTITUTE OF ROBOTICS AND MECHATRONICS (Hrsg.): *Status Report 1997-2004 : Part 1*. Oberpfaffenhofen : Deutsches Zentrum für Luft- und Raumfahrt e. V. in der Helmholtz-Gemeinschaft, 2004
- [39] KAHMEN, Heribert: *Vermessungskunde*. Berlin : de Gruyter, 1997
- [40] KELA, Ajay: Hierarchical octree approximations for boundary representation-based geometric models. In: *Computer-Aided Design* 21 (1989), Nr. 6, S. 355-362
- [41] KERN, Fredie: *Automatisierte Modellierung von Bauwerksgeometrien aus 3D-Laserscanner-Daten*. Braunschweig, Technischen Universität, Institut für Geodäsie und Photogrammetrie, Diss., 2003
- [42] KITWARE INC. (Hrsg.): *The VTK user's guide : install, use and extend the visualization toolkit*. New York : Kitware, 2006

-
- [43] Kós, Géza: An algorithm to triangulate surfaces in 3D using unorganised point clouds. In: *Computing Supplementum* 14 (2001), S. 219-232
- [44] KRÄMER, Jörg: *Delaunay-Triangulierungen in zwei und drei Dimensionen*. Tübingen, Eberhard-Karls-Universität, Wilhelm-Schickard-Institut Graphische Interaktive Systeme, Dipl.-Arb., 1995
- [45] KÜHNER, Siegfried: *Virtual Reality basierte Analyse und interaktive Steuerung von Strömungssimulationen im Bauingenieurwesen*. München, Technische Universität, Lehrstuhl für Bauinformatik, Diss., 2003
- [46] LAL, Jagdish: *Recognition of 3D Settlement Structures for Generalization*. München, Technische Universität, Lehrstuhl für Kartographie, Diss., 2005
- [47] LAWSON, Charles L.: Properties of n-dimensional triangulations. In: *Computer aided geometric design* 3 (1986), Nr. 4, S. 231-246
- [48] LEHRSTUHL FÜR TRAGWERKSPLANUNG ; BAYERISCHER FORSCHUNGSVERBUND ABFALLFORSCHUNG UND RESTSTOFFVERWERTUNG (Hrsg.): *Ressourcenschonende Instandsetzung alter Dachtragwerke : Nutzung des Laseraufmasses und objektorientierter Modellierung*. München : Verbundprojekt Stoffflussmanagement Bauwerke, 2005 (BayFORREST F230) – Abschlussbericht
- [49] LEICA GEOSYSTEMS (Hrsg.): *Leica Cyclone 5.6*. URL http://www.leica-geosystems.com/corporate/de/ndef/lgs_6515.htm. – Aktualisierungsdatum: 1.6.2007 –Leica Geosystems AG, Schweiz
- [50] LINKWITZ, Klaus: Über einige Ausgleichungsprobleme und ihre Lösung mit Hilfe von Matrizen-Eigenwerten. In: ACKERMANN, Friedrich: *Wissenschaftliche Beiträge aus dem Kreise der Schüler von Ernst Gotthardt anlässlich seiner Emeritierung am 31. März 1976*. München : Verlag der Bayerischen Akademie der Wissenschaften, 1976, S. 111-126
- [51] LORENSEN, William E. ; CLINE, Harvey E.: Marching Cubes : A high resolution 3D surface construction algorithm. In: *ACM SIGGRAPH Computer Graphics* 21 (1987), Nr. 4, S. 163-169

- [52] MALLOT, Hanspeter A.: *Kognitive Neurobiologie : Grundlagen der Neuro- und Verhaltensbiologie III*. Tübingen, Universität, Lehrstuhl für Kognitive Neurowissenschaft, 2005 – Vorlesungsskript Wintersemester 2005 / 2006
- [53] MAYER, Andreas ; MECHLER, Bernhard ; SCHLINDWEIN, Andreas ; WOLKE, Rainer: *Fuzzy-Logic : Einführung und Leitfaden zur praktischen Anwendung*. Bonn : Addison-Wesley, 1993
- [54] MUNDANI, Ralf-Peter: *Hierarchische Geometriemodelle zur Einbettung verteilter Simulationsaufgaben*. Stuttgart, Universität, Institut für Parallele und Verteilte Systeme, Diss., 2005
- [55] NEITZEL, Frank: *Identifizierung konsistenter Datengruppen am Beispiel der Kongruenzuntersuchung geodätischer Netze*. München : Verlag der Bayerischen Akademie der Wissenschaften, 2004
- [56] NICHOLLS, John G.; MARTIN, A. Robert; WALLACE, Bruce G.: *Vom Neuron zum Gehirn : Zum Verständnis der zellulären und molekularen Funktion des Nervensystems*. Heidelberg : Spektrum Akademischer Verlag, 2002
- [57] MINSKY, Marvin L. ; PAPERT, Seymour A.: *Perceptrons. An introduction to computational geometry*. Cambridge : MIT Press, 1988
- [58] NIEMANN, Heinrich: *Klassifikation von Mustern*. Berlin : Springer, 1983
- [59] NIEMEIER, Wolfgang: Modellierung – Ableitung geometrischer Informationen aus Punktwolken. In: BARTH, Wolf ; FOPPE, Karl ; SCHÄFER, Thomas (Hrsg.): *Terrestrisches Laserscanning (TLS) : Ein geodätisches Messverfahren mit Zukunft*. Augsburg : Wißner, 2005, S. 81-94
- [60] NIEMEIER, Wolfgang ; KERN, Fredie: Anwendungspotentiale von scannenden Messverfahren. In: WEFERLING, Ulrich (Hrsg.): *Von Handaufmaß bis High Tech : messen, modellieren, darstellen ; Aufnahmeverfahren in der historischen Bauforschung*. Mainz am Rhein : von Zabern, 2001
- [61] OLEXIK, William A.: *Spinal Neuron #2 - 100x : Microscope Pictures*. URL http://www.montgomerycollege.edu/faculty/~wolexik/public_html/Neuron-100x-2.jpg. – Aktualisierungsdatum: 12.12.2003 – mailto: william.olexik@montgomerycollege.edu. – Montgomery College, USA

-
- [62] OKABE, Atsuyuki ; BOOTS, Barry ; SUGIHARA, Kokichi ; Chiu, Sung Nok: *Spatial tessellations : Concepts and applications of Voronoi diagrams*. Chichester : Wiley, 2000
- [63] O'ROURKE, Joseph: *Computational geometry in C*. Cambridge : Cambridge Univ. Press, 1994
- [64] O'ROURKE, Joseph: Finding minimal enclosing boxes. In: *International journal of computer and information sciences* 14 (1985), Nr. 3, S. 183-199
- [65] PETERSEN, Michael: Ist Laserscanning wirtschaftlich einsetzbar? In: SCHÄFER, Markus ; NIEMEIER, Wolfgang (Hrsg.): *Terrestrisches Laser-Scanning (TLS 2006) : Beiträge zum 72. DVW-Seminar am 9. und 10. November 2006 in Fulda*. Augsburg : Wißner, 2006, S. 161-174
- [66] PETZOLD, Frank: *Computergestützte Bauaufnahme als Grundlage für die Planung im Bestand : Untersuchungen zur digitalen Erfassung und Modellbildung*. Weimar, Bauhaus-Universität, Professur Informatik in der Architektur, Diss., 2001
- [67] PUPPE, Frank: *Einführung in Expertensysteme*. Berlin : Springer, 1991
- [68] RANK, Ernst ; WUNDERLICH, Thomas A. ; SCHÄFER, Thomas ; SCHLEINKOFER, Matthias ; BAYERISCHER FORSCHUNGSVERBUND ABFALLFORSCHUNG UND RESTSTOFFVERWERTUNG (Hrsg.): *Vom Laseraufmaß zum Stoffflussmanagement für Altbauten*. München : Verbundprojekt Stoffflussmanagement Bauwerke, 2005 (BayFORREST F248) – Abschlussbericht
- [69] REIF, Gerald: *Moderne Aspekte der Wissensverarbeitung : Ein interaktiver Lernbehelf für das Web Based Training*. Graz, Technische Universität, Institut für Informationsverarbeitung und Computergestützte neue Medien, Dipl.-Arb., 2000
- [70] RIETDORF, Andreas: *Automatisierte Auswertung und Kalibrierung von scannenden Messsystemen mit tachymetrischem Messprinzip*. Berlin, Technischen Universität, Institut für Geodäsie und Geoinformationstechnik, Diss., 2005
- [71] ROJAS, Raúl: *Theorie der neuronalen Netze : Eine systematische Einführung*. Berlin : Springer, 1993

- [72] ROMBERG, Richard: *Gebäudemodell-basierte Strukturanalyse im Bauwesen*. München, Technische Universität, Lehrstuhl für Bauinformatik, Diss., 2005
- [73] ROSENBLATT, Frank: The perceptron : A probabilistic model for information storage and organization in the brain. In: *Psychological Reviews* 65 (1958), S. 386-408
- [74] RUMELHART, David E. ; HINTON, Geoffrey E. ; WILLIAMS, Ronald J.: Learning representations by back-propagating errors. In: *Nature* 323 (1986), S. 533-536
- [75] RUNKLER, Thomas A.: *Information mining : Methoden, Algorithmen und Anwendungen intelligenter Datenanalyse*. Braunschweig : Vieweg, 2000
- [76] SABELKA, Rainer: *Dreidimensionale Finite Elemente Simulation von Verdrahtungsstrukturen auf Integrierten Schaltungen*. Wien, Technischen Universität, Institut für Mikroelektronik, Diss., 2001
- [77] SAMET, Hanan: The Quadtree and Related Hierarchical Data Structures. In: *Computing Surveys* 16 (1984), Nr. 2, S. 187-260
- [78] SCHÄFER, Thomas ; SCHULZ, Thorsten: Kalibrierung, Einflussgrößen und Genauigkeit von Terrestrischen Laserscannern. In: BARTH, Wolf ; FOPPE, Karl ; SCHÄFER, Thomas (Hrsg.): *Terrestrisches Laserscanning (TLS) : Ein geodätisches Messverfahren mit Zukunft*. Augsburg : Wißner, 2005, S. 29-48
- [79] SCHÄFER, Thomas; WEBER, Thomas: *Ingenieurvermessung 2004 – 14th International Conference on Engineering Surveying (Zürich, 16. März 2004) : Von der Punktwolke zum CAD*. URL http://www.geo.bv.tum.de/de/mitarb/weber/Von_der_Punktwolke_zum_CAD.pdf. – Aktualisierungsdatum: 18.07.2005 – [mailto: th.schaefer@bv.tum.de](mailto:th.schaefer@bv.tum.de). – Technische Universität München
- [80] SCHIELE, Oliver Jens: *Ein operationelles Kalibrierverfahren für das flugzeuggetragene Laserscannersystem ScaLARS*. Stuttgart, Universität, Institut für Navigation, Diss., 2005
- [81] SCHLENDER, Dirk: *Anforderungen an die Detailstufenverwaltung in virtuellen Simulationsumgebungen*. Düsseldorf : VDI, 1999

-
- [82] SCHRÖDER, Eberhard: *Kartenentwürfe der Erde : Kartographische Abbildungsverfahren aus mathematischer und historischer Sicht*. Thun : Deutsch, 1988
- [83] SCHWEIBENZ, Bernd ; WEBER, Thomas: Dokumentation des Baufortschritts mit Laserscanning-Technologie. In: *Baumarkt + Bauwirtschaft* (2003), Nr. 12, S. 70-73
- [84] SESTER, Monika: *Lernen struktureller Modelle für die Bildanalyse*. München Verlag der Bayerischen Akademie der Wissenschaften, 1995
- [85] STÖCKER, Horst (Hrsg.): *Taschenbuch mathematischer Formeln und moderner Verfahren*. Thun : Deutsch, 1999
- [86] STRANG, Gilbert: *Linear algebra and its applications*. Toronto : Thomson Learning, 1988
- [87] THUROW, Torsten: *Digitaler Architekturbestand : Untersuchungen zur computergestützten, schrittweisen Erfassung und Abbildung der Geometrie von Gebäuden im Kontext der planungsrelevanten Bauaufnahme*. Weimar, Bauhaus-Universität, Professur Informatik in der Architektur, Diss., 2004
- [88] TÖLKE, Jonas: *Gitter-Boltzmann-Verfahren zur Simulation von Zweiphasenströmungen*. München, Technische Universität, Lehrstuhl für Bauinformatik, Diss., 2001
- [89] UNIQA VERSICHERUNGEN: *Der UNIQA Tower*. URL http://tower.uniq.at/index_tow.php. – Aktualisierungsdatum: 01.06.2007 – mailto: internet@uniq.at. – UNIQA Versicherungen AG
- [90] VAN TREECK, Christoph Alban: *Gebäudemodell-basierte Simulation von Raumluftrömungen*. München, Technische Universität, Lehrstuhl für Bauinformatik, Diss., 2004
- [91] VARADY, Tamas ; MARTIN, Ralph: Reverse Engineering. In: FARIN, Gerald E. (Hrsg.) ; HOSCHEK, Josef (Hrsg.) ; KIM, Myung-Soo (Hrsg.): *Handbook of computer aided geometric design*. Amsterdam : Elsevier, 2002, S. 651-681
- [92] VORONOI, Georges: Nouvelles applications des paramètres continus à la théorie des formes quadratiques : Recherches sur les paralléloèdres primitifs. In: *Journal für die reine und angewandte Mathematik* 134 (1908), S. 198–287

- [93] WASSOUF, Ziad ; VAN TREECK, Christoph ; RANK, Ernst: Ein Objekt- und Informationsmodell der technischen Gebäudeausstattung zur Ökobilanzierung von Gebäuden. In: Koenigsdorff, Roland (Hrsg.) ; van Treeck, Christoph (Hrsg.): *BauSim 2006 : Tagungsband*. München : IBPSA Germany, 2006, S. 210-212
- [94] WEFERLING, Ulrich ; PETZOLD, Frank ; DONATH, Dirk: Neue Techniken in der Bestandserfassung. In: KIRSCHKE, Heiko (Hrsg.): *Digital Proceedings des Internationalen Kolloquiums über Anwendungen der Informatik und Mathematik in Architektur und Bauwesen (IKM)*. Weimar : Bauhaus-Universität Weimar, 2003
- [95] WELSTEAD, Stephen T.: *Neuronal network and fuzzy logic applications in c/c++*. New York : Wiley, 1994
- [96] WENISCH, Petra: *Computational Steering of CFD Simulations on Teraflop-Supercomputers*. München, Technische Universität, Lehrstuhl für Bauinformatik, Diss., 2008
- [97] WHITE, Halbert: Connectionist Nonparametric Regression : Multilayer Feedforward Networks Can Learn Arbitrary Mappings. In: *Neural Networks 3* (1990), S. 535-549
- [98] WIORA, Georg: *Optische 3D-Messtechnik : Präzise Gestaltvermessung mit einem erweiterten Streifenprojektionsverfahren*. Heidelberg, Universität, Interdisziplinäres Zentrum für Wissenschaftliches Rechnen, Diss., 2001
- [99] WUNDERLICH, Thomas A.: Der Anwendungsreichtum des terrestrischen Laserscannings. In: *Flächenmanagement und Bodenordnung* (2006), Nr. 4, S. 170-174
- [100] WYVILL, Geoff ; MCPHEETERS, Craig ; WYVILL, Brian: Data structure for soft objects. In: *The Visual Computer 2* (1986), Nr. 4, S. 227-234
- [101] YANG, Hyun S. ; KAK, Avi C.: Determination of the identity, position and orientation of the topmost object in a pile. In: *Computer Vision, Graphics, and Image Processing 36* (1986), S. 229-255
- [102] ZADEH, Lotfi A.: Fuzzy Sets. In: *Information and Control 8* (1965), Nr. 3, S. 338-353
- [103] ZADEH, Lotfi A.: The concept of linguistic variable and its application to approximate reasoning. In: *Information Sciences 8* (1975), Nr. 3, S. 199-249

[104] ZELL, Andreas: *Simulation neuronaler Netze*. München : Oldenbourg, 2000