

DIPLOMARBEIT

Entwicklung von Transformationsalgorithmen innenmaßbezogener 3D-Raummodelle zur Anwendung in baurechtlichen Nachweisverfahren

*Development of transformation algorithms for interior-dimension-based
3D room models for use in building code verification procedures*

Name: Lisa Sophie Ahnert

Matrikelnummer: 4879967

Verantwortlicher Hochschullehrer: Prof. Dr.-Ing. John Grunewald

Betreuer: Dipl.-Ing. Dirk Weiß

Dresden, den 18. Februar 2026

Inhaltsverzeichnis

Abbildungsverzeichnis.....	V
Tabellenverzeichnis	VII
Formelverzeichnis	VIII
Abkürzungsverzeichnis.....	IX
1 Einleitung	1
1.1 Hintergrund	1
1.2 Ziele und Abgrenzung der Arbeit.....	2
2 Grundlagen.....	3
2.1 GEG und DIN V 18599	3
2.2 DIN EN 12831.....	5
2.3 VDI 2078 und VDI 6020	6
2.4 Thermische Gebäudesimulation.....	7
2.4.1 NANDRAD	7
2.4.2 VICUS.....	7
2.5 Vorliegende Algorithmen/Stand der Forschung.....	9
2.5.1 Nachbarflächen finden	10
2.5.2 Verschieben und Verschneiden der Flächen	13
2.5.3 Korrektur.....	13
2.5.4 Ein- und Ausgabe der Daten	14
2.6 Datenstruktur, Festlegungen und Definitionen.....	14
3 Methodik	16
3.1 Erweiterungspotential	16
3.2 Überarbeitungen in der Funktion <code>findNeighbouringSurfaces</code>	18
3.2.1 Hilfspunktmethode	18
3.2.2 Sortierung	19
3.2.3 Punktgleichheitsmethode	20
3.3 Ergänzungen in der Funktion <code>createExtrudedPoints</code>	22

3.3.1	Parallele Flächen	22
3.3.2	Linear Abhängige Flächen	23
3.4	Einbau einer Iterationsstufe in den Extrusionsprozess	27
3.4.1	Erweiterung des Extrusionsalgorithmus	28
3.4.2	Markierung fehlerhafter Punkte	29
3.4.3	Gegenläufige Markierung und zweiter Extrusionsdurchlauf.....	30
4	Visuelle Validierung	32
4.1	Einraummodelle	33
4.1.1	Einfache Raummodelle	33
4.1.2	Modelle mit parallelen Flächen	34
4.1.3	Typische Dachformen	35
4.1.4	Zwerchgauben	36
4.1.5	Dachgauben.....	37
4.1.6	Weitere und kombinierte Dachformen	38
4.2	Mehrraummodelle	39
4.3	Einschränkungen und verbleibende Fehler	39
4.3.1	Eingangsdaten	40
4.3.2	Löcher im Ausgangskörper	40
4.3.3	Umsetzung der Normen und Mehrraummodelle	41
4.3.4	Lücken in Transformationskörpern und Versatz von Firstlinien.....	42
5	Schnittstelle zu baurechtlichen Nachweisen	44
6	Schlussbetrachtung	46
6.1	Zusammenfassung.....	46
6.2	Ausblick	46
	Literaturverzeichnis.....	48
	Anlagenverzeichnis	50
	Erklärung zur selbständigen Erstellung der Arbeit.....	51

Abbildungsverzeichnis

Abbildung 2.1: horizontale Maßbezüge nach DIN V 18599-1 [9, S. 64]	4
Abbildung 2.2: vertikale Gebäudemaßbezüge nach DIN V 18599-1 [9, S. 65].....	5
Abbildung 2.3: Maßbezüge nach VDI 2078 und VDI 6020	6
Abbildung 2.4: Beispiel eines Raummodells in VICUS.....	8
Abbildung 2.5: Bauteilverknüpfungen in VICUS	9
Abbildung 2.6: Ablauf des Algorithmus von Toni Bartel [17, S. 16].....	10
Abbildung 2.7: mögliche Lage zweier Flächen zueinander	11
Abbildung 2.8: Punkt-Methode	11
Abbildung 2.9: Hilfspunktmethode.....	12
Abbildung 2.10: Reparatur eines verwundenen Polygons	13
Abbildung 3.1: Testgeometrie 1: L-Dach mit Gaube	16
Abbildung 3.2: Ausgangsstand der Transformation des L-Dachs mit Gaube	16
Abbildung 3.3: potentielle Lagebeziehungen zwischen 3 Ebenen [18]	17
Abbildung 3.4: linear abhängige Flächen bei einem Dach mit Zwerchgaube	18
Abbildung 3.5: Fläche, die mehrfach an andere Fläche grenzt	19
Abbildung 3.6: Sortierung der <code>pointInfo</code> -Objekte.....	19
Abbildung 3.7: Veranschaulichung Funktionsweise <code>findNextSurface</code>	21
Abbildung 3.8: fehlerhafte Transformation durch unangepasste Hilfsebene	22
Abbildung 3.9: mögliche Positionierung linear abhängiger Flächen	24
Abbildung 3.10: Transformationspunkte von linear abgängigen, nicht parallelen Flächen.....	25
Abbildung 3.11: Fehlerhafte Punktbildung durch unverschobene Hilfsebene.....	26
Abbildung 3.12: Verschiebung der Hilfsebene	27
Abbildung 3.13: Veränderung der Ursprungsgeometrie infolge Extrusion	28
Abbildung 3.14: Fehlerhafte Geometrie durch unterschiedliche Dachschrägen	28
Abbildung 3.15: neuer Ablauf des Algorithmus.....	31
Abbildung 4.1: Verkleinerung eines Loches infolge der Transformation	41

Abbildung 4.2: Fehler in Transformation adiabater Bauteile	42
Abbildung 4.3: Lücke in Transformationskörper bei Zwerggauben	42
Abbildung 4.4: Versatz in der Firstlinie bei großer Schleppgaube	43

Tabellenverzeichnis

Tabelle 1: Bauteilkonstruktionen	33
Tabelle 2: Transformation einfacher Test-Raummodelle.....	34
Tabelle 3: Transformation von Räumen mit parallelen Flächen	34
Tabelle 4: Transformation von Modellen typischer Dachformen	36
Tabelle 5: Transformation von Dächern mit Zwerchgauben.....	37
Tabelle 6: Transformation von Dächern mit Dachgauben	38
Tabelle 7: Transformation komplexer Dachformen.....	38
Tabelle 8: Transformation von Mehrraummodellen.....	39

Formelverzeichnis

(1) Punkt- und Kantengleichheit	11
(2) lineare Abhängigkeit von Vektoren	23
(3) Schnittpunktberechnung aus drei Ebenen	25
(4) Winkel zwischen zwei Vektoren	27
(5) Verschiebungslänge 1	27
(6) Verschiebungslänge 2	27
(7) Punktverschiebung	27

Abkürzungsverzeichnis

EEWärmeG	Erneuerbare-Energien-Wärmegesetz
EnEG	Energieeinspargesetz
EnEV	Energieeinsparverordnung
GEG	Gebäudeenergiegesetz
IFC	Industry Foundation Classes
TWh	Terrawattstunde
VDI	Verein Deutscher Ingenieure

1 Einleitung

1.1 Hintergrund

Der Endenergieverbrauch privater Haushalte lag 2024 bei 625 Terrawattstunden (TWh) und entsprach damit einem Viertel des gesamten Endenergieverbrauchs in Deutschland. Über 60 Prozent dieses Verbrauches entfallen auf das Heizen [1]. Mit dem Gebäudeenergiegesetz (GEG), welches 2020 in Kraft getreten ist, wurde die Senkung des Energieverbrauchs von Gebäuden und die zunehmende Nutzung erneuerbarer Energieträger ein primäres Ziel der deutschen Politik [2, S. 8 f.].

Angesichts eines Wohngebäudebestandes, von dem 60 Prozent als energetisch unzureichend bewertet werden, besteht ein dringender Handlungsbedarf. Um das Ziel eines klimaneutralen Gebäudebestandes bis zum Jahr 2045 zu erreichen, müssen daher energetische Sanierungen stärker in den Fokus gerückt werden [3]. Die hierfür erforderlichen Transformationen von Gebäuden und der Wärmeversorgung sind jedoch mit vielfältigen und komplexen Herausforderungen verbunden, insbesondere im Bereich der Gebäudeenergieplanung.

Traditionelle stationäre Rechenverfahren, welche in Zeiten begrenzter Rechenleistungen und günstiger fossiler Energiequellen entwickelt wurden, stoßen bei der Planung effizienter und ressourcenschonender Anlagenkonzepte zunehmend an ihre Grenzen. Vereinfachungen und konventionelle Planungsansätze führen häufig zu einer Überdimensionierung gebäudetechnischer Anlagen, was sowohl erhöhte Betriebskosten als auch energetische Ineffizienzen zur Folge hat [4, S. 3 f.]. Um sämtliche relevante Einflussgrößen auf ein Gebäude, wie Nutzung, Klima, Konstruktion und Kubatur realitätsnah abbilden zu können, stellt die Gebäudesimulation ein geeignetes Planungswerkzeug dar. Ein wesentlicher Nachteil dabei besteht jedoch in den meist hohen Modellierungskosten.

In den vergangenen Jahren wurden bedeutende Fortschritte beim Import und Export von Modellen zwischen verschiedenen Modellierungsprogrammen durch das IFC-Datenformat erzielt. Für Bestandsgebäude liegt ein entsprechendes digitales Modell jedoch in der Regel nicht vor. Zudem existiert derzeit keine geeignete Schnittstelle zu Fachplanersoftware, wie Energieberaterprogrammen zur Erstellung von Energieausweisen. Aufgrund abweichender Bauteilmaßbezüge erfordern diese Programme eine erneute Dateneingabe, was mit einem erheblichen Zeit- und Kostenaufwand verbunden ist.

Energetische Gebäudesimulationsprogramme, wie beispielsweise VICUS [5] oder EnergyPlus [6], berechnen Raumtemperaturen und Energieströme anhand

innenmaßbezogener Gebäudegeometriemodelle. Baurechtliche und bautechnische Nachweise, wie der GEG-Nachweis und die Heizlastberechnung nach DIN EN 12831, erfordern hingegen außenmaßbezogene Bezugsflächen. Zur Sicherstellung einer durchgängigen digitalen Prozesskette und zur Steigerung der Effizienz des Datentransfers ist daher ein automatisierter und verlustfreier Übergang zwischen den innen- und außenmaßbezogenen Geometriemodellen erforderlich.

1.2 Ziele und Abgrenzung der Arbeit

Grundlegende Ansätze zur algorithmischen Transformation von Gebäudemodellen wurden bereits in früheren Arbeiten entwickelt. Diese Ansätze liefern für die geometrische Extrusion einfacher Raummodelle bereits gute Ergebnisse, stoßen jedoch bei komplexeren Raumgeometrien an ihre Grenzen. Insbesondere für komplexere Modelle stellt jedoch die Optimierung des erforderlichen Arbeitsaufwands das größte Potenzial dar.

Die Diplomarbeit schließt am erreichten Stand auf diesem Gebiet an und adressiert die weiterhin bestehenden Lücken bei der Extrusion komplexer Raumgeometrien. Im Fokus steht dabei die Gebäudesimulationssoftware VICUS. Ziel der Arbeit ist die Entwicklung eines robusten Transformationsalgorithmus, der auch komplexe Simulationsmodelle zuverlässig vom Innenmaß- in den Außenmaßbezug überführt. Darüber hinaus werden Überlegungen zu einem Workflow angestellt, der die Extraktion relevanter Daten aus den transformierten Gebäudemodellen zur Anwendung in baurechtlichen Nachweisen ermöglicht.

2 Grundlagen

Die Grundlage des Transformationsalgorithmus bilden verschiedene Normen und Richtlinien zur energetischen Bewertung von Gebäuden sowie zur thermischen Gebäudesimulation. Diese Regelwerke definieren Bauteilbezugsmaße für die jeweiligen Rechenverfahren und bilden damit die fachliche Grundlage für den Transformationsalgorithmus. Die Erweiterung des Algorithmus basiert auf einem bestehenden, in Kapitel 2.5 ausführlich beschriebenen Ansatz und ist für Raummodelle des Gebäudesimulationsprogramms VICUS konzipiert.

2.1 GEG und DIN V 18599

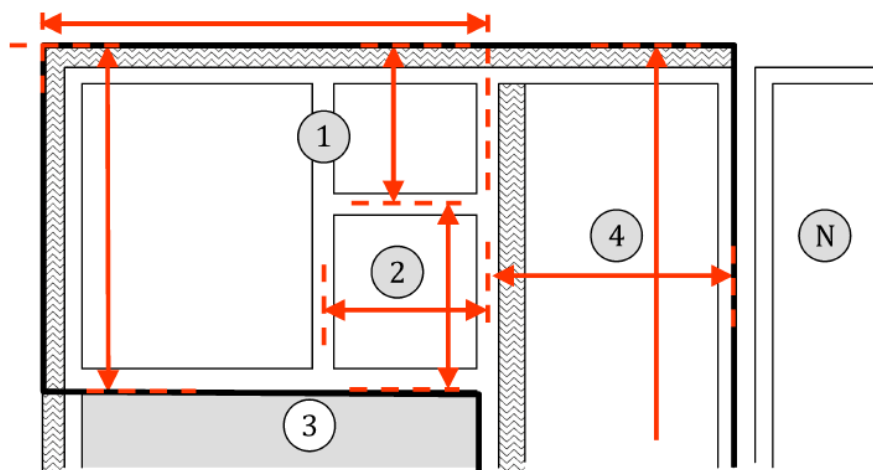
Das Gesetz zur Einsparung von Energie und zur Nutzung erneuerbarer Energien zur Wärme- und Kälteerzeugung in Gebäuden oder auch Gebäudeenergiegesetz ist ein deutsches Gesetz, welches am ersten November 2020 in Kraft getreten ist. Das Gesetz löst das Energieeinspargesetz (EnEG), die Energieeinsparverordnung (EnEV) und das Erneuerbare-Energien-Wärmegesetz (EEWärmeG) ab und hat zum Ziel, einen wesentlichen Beitrag zur Erreichung der nationalen Klimaschutzziele durch Maßnahmen zur Einsparung von Treibhausgasemissionen und der zunehmenden Nutzung erneuerbarer Energien zu leisten [2, S. 8 f.]. Es „enthält Anforderungen an die energetische Qualität von Gebäuden, die Erstellung und die Verwendung von Energieausweisen sowie an den Einsatz erneuerbarer Energien bei der Wärmeversorgung von Gebäuden.“ [7] Dafür verweist das Gesetz auf verschiedene nationale Normen. Als Berechnungsgrundlage für den Jahres-Primärenergiebedarf für neu zu errichtende Wohn- und Nichtwohngebäude ist dabei die DIN V 18599: 2018-09 aufgeführt [2, S. 18 f.]. Auch für Bestandsgebäude, bei denen Außenbauteile erneuert, ersetzt oder ergänzt werden, muss in der Regel eine erneute energetische Bewertung nach DIN V 18599 durchgeführt werden [2, S. 27 f.].

Die DIN V 18599 ist eine Normenreihe zur energetischen Bewertung von Gebäuden, wie der Berechnung des Nutz-, End- und Primärenergiebedarfes für Heizung, Kühlung, Lüftung, Trinkwarmwasser und Beleuchtung für Wohn- und Nichtwohngebäude. Im ersten Teil der Normenreihe werden die Grundlagen für die Berechnungen in den weiteren Teilen definiert. Dazu gehören auch die Bezugsmaße für die einzelnen Bauteile und Zonen eines Gesamtgebäudes.

Nach DIN V 18599-1 sind horizontale und vertikale Maßbezüge definiert. Abbildung 2.1 und Abbildung 2.2 zeigen diese Maßbezüge. Für horizontale Außenbauteile sind die Außenmaße nach DIN EN ISO 13789 zu verwenden. Also das an der Außenseite

eines Gebäudes gemessene Maß, sofern vorhanden einschließlich der Wärmedämmung und dem Putz [8, S. 13]. Bei Innenbauteilen zwischen einer temperierten und nicht temperierten Zone gilt ebenfalls das Außenmaß der temperierten Zone. Für Innenbauteile zwischen zwei ggf. unterschiedlich temperierten Zonen, wird das Achsmaß verwendet. Dieses beschreibt die Mitte des Rohbauteils ohne Berücksichtigung eventueller Innendämmschichten. Für die Maße in vertikaler Richtung stellt die Oberkante der Rohdecke das Bezugsmaß in allen Ebenen des Gebäudes dar. Dies gilt unabhängig von der Lage von potentiellen Dämmschichten und auch für den unteren Gebäudeabschluss. Eine Ausnahme bildet der obere Gebäudeabschluss, also die oberste Geschossdecke bei einem unbeheizten Dach oder das Dach selbst. Für diese Bauteile wird die Oberkante der obersten wärmetechnisch wirksamen Schicht (in der Regel die Dämmschicht) als Außenmaß verwendet. [9, S. 63]

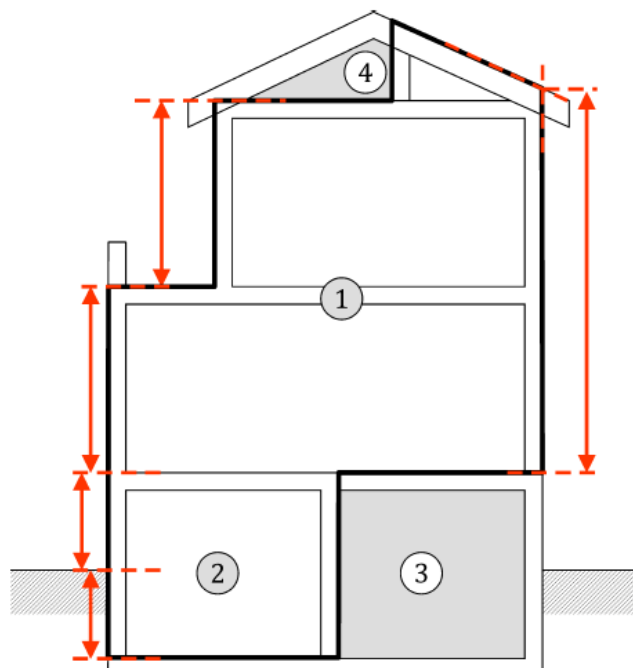
DIN/TS 18599-1:2025-10



Legende

- - - - Bemaßungshilfslinie
- wärmeübertragende Umfassungsfläche (Bilanzgrenze der Zonen 1 + 2 + 4)
- Zone 1 Innenbegrenzungen (zu Zonen 4 und 2) sowie Außenbegrenzungen (zur nicht thermisch konditionierten Zone 3 und zur Außenluft)
- Zone 2 Innenbegrenzungen nach oben, links und rechts; nach rechts zur Zone 4 hin gilt das Achsmaß (ohne Berücksichtigung der Dämmung); zur nicht temperierten Zone 3 hin gilt das Außenmaß
- Zone 3 nicht thermisch konditionierte Zone
- Zone 4 Es gilt zur Hinterlüftung hin die Oberkante der letzten wärmetechnisch wirksamen Schicht.
- Zone N Nachbargebäude

Abbildung 2.1: horizontale Maßbezüge nach DIN V 18599-1 [9, S. 64]



Legende

- - - - Bemaßungshilfslinie
- wärmeübertragende Umfassungsfläche (Bilanzgrenze der Zonen 1 + 2)
- Zone 1** Obere Begrenzung zur nicht thermisch konditionierten Zone 4 ist die Oberkante der obersten wärmetechnisch wirksamen Schicht der obersten Geschossdecke; die obere Begrenzung an Außenluft im Dach bildet die äußerste wärmetechnisch wirksame Schicht des Steildaches; die untere Begrenzung zur Zone 2 sowie zur nicht thermisch konditionierten Zone 3 ist die Oberkante der untersten Rohdecke. Die Maßbezüge gelten auch, wenn 3 Außenluft ist (Decke nach unten gegen Außenluft).
- Zone 2** Sowohl die obere Begrenzung an die Zone 1 als auch die untere Begrenzung an das Erdreich ist jeweils die Oberkante der Rohdecke.
- Zone 3** nicht thermisch konditionierte Zone
- Zone 4** nicht thermisch konditionierte Zone

Abbildung 2.2: vertikale Gebäudemessbezüge nach DIN V 18599-1 [9, S. 65]

2.2 DIN EN 12831

Die DIN EN 12831 ist eine europäische Norm, die wie die DIN V 18599 der energetischen Bewertung von Gebäuden dient. Sie umfasst Verfahren zur Berechnung der Norm-Heizlast von Gebäuden, welche angibt, wie hoch die Wärmezufuhr sein muss, um die geforderte Norm-Innentemperatur unter Norm-Außenbedingungen zu erreichen [10, S. 8].

Die Gebäude- und Bauteilmaßbezüge sind sehr ähnlich zu denen in der DIN V 18599 definiert. Für Bauteile, bei denen Transmissionswärmeverluste stattfinden (Außenbauteile und Innenbauteile zu unbeheizten Räumen) werden die Außenmaße verwendet. Innenwände und Decken werden jeweils zur halben

Bauteildicke gemessen. Als Höhe der Außenwände gilt die jeweilige Geschosshöhe, gemessen von Oberkante zu Oberkante des jeweiligen Fußbodenaufbaus. [11, S. 47 f.]

2.3 VDI 2078 und VDI 6020

Der Verein Deutscher Ingenieure gibt verschiedene Richtlinien zur thermischen Gebäudesimulation heraus. Die VDI 2078 ist die Richtlinie zur Berechnung der thermischen Lasten und Raumtemperaturen für die Auslegung der Kühllast und Jahressimulation für Gebäude und die VDI 6020 ist die Richtlinie für Anforderungen an thermisch-energetische Rechenverfahren zur Gebäude- und Anlagensimulation. Beide Richtlinien haben nahezu identische Definitionen für das Bauteilverhalten und die Maße von Innen- und Außenbauteilen. Für das Bauteilverhalten wird in adiabate und nicht adiabate Bauteile unterschieden. Adiabate Bauteile sind dabei Bauteile, die auf beiden Seiten die gleichen Randbedingungen besitzen (in der Regel Innenwände), während die Randbedingungen von nicht adiabaten Bauteilen auf Innen- und Außenseite unterschiedlich sind (Außenwände, Innenwände zu anders beheizten Räumen und ggf. Kellerdecken). Bei den Bauteilmaßen wird jedoch nur zwischen Innenbauteilen (adiabat und nicht adiabat) und Außenbauteilen unterschieden. Für Innenbauteile sind die lichten Innenmaße des Bauteils (Nettomaße) zu verwenden. Für Außenbauteile sind die Bruttomaße der Außenseite zu verwenden. [4, S. 18] [12, S. 13]

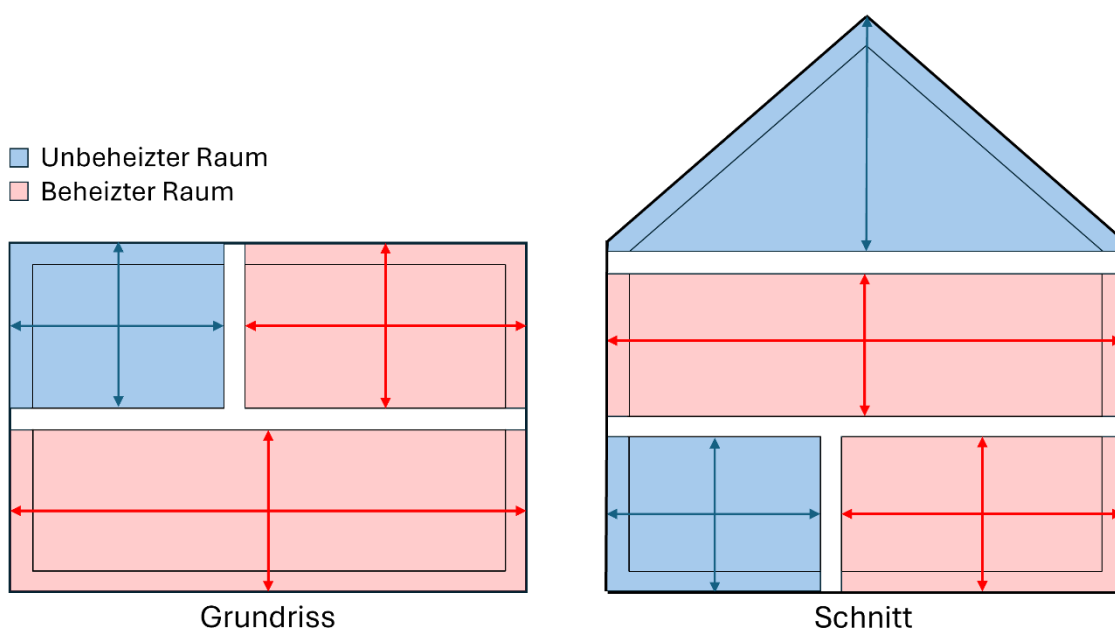


Abbildung 2.3: Maßbezüge nach VDI 2078 und VDI 6020

2.4 Thermische Gebäudesimulation

Die thermische Gebäudesimulation dient der dynamischen Berechnung des thermisch-energetischen Verhaltens von Gebäuden. Grundlage der Berechnung sind unter anderem das Außenklima, innere Lasten (z.B. durch Personen oder Geräte), sowie nutzungsabhängige Betriebszeiten.

Für die Abbildung des Außenklimas werden in der Regel die Wetterdatensätze eines Testreferenzjahres verwendet, die stündliche Wetterdaten für einen bestimmten Standort enthalten. Im Gegensatz zu statischen Berechnungsverfahren ermöglicht die dynamische Simulation eine zeitabhängige Betrachtung von Temperaturverläufen und Energieflüssen. Dadurch können Raumtemperaturen und Energiebedarfe für Heizen, Kühlen und Lüften deutlich präziser bestimmt werden [12, S. 11].

Ein Gebäudesimulationsprogramm besteht in der Regel aus einer Eingabemaske, über die das Gebäude erstellt und Parameter zugewiesen werden können und einem Rechenkern, der für die Berechnungen in der Simulation zuständig ist.

2.4.1 NANDRAD

Ein leistungsfähiger Solver zur thermischen Gebäudesimulation ist NANDRAD [13]. Er wurde insbesondere für die Berechnung großer und komplexer Gebäude entwickelt und ermöglicht eine detailgetreue dynamische Bewertung der Energieeffizienz von Gebäuden.

Eine Besonderheit von NANDRAD ist die schnelle und sehr genaue Berechnung durch die Finite Volumen Methode [14]. Neben der Berechnung von Jahresenergiebedarfen erlaubt der Solver auch die Bestimmung von Jahreslastkennlinien sowie normbasierte Heiz- und Kühllastberechnungen.

2.4.2 VICUS

Im Rahmen dieser Arbeit werden Modelle mit der Gebäudesimulations- und Wärmenetzplanungssoftware VICUS [5] erstellt und verwendet. VICUS stellt dabei die geometrische Eingabe- und Modellierungsmaske dar und nutzt für die thermischen Berechnungen den Solver NANDRAD.

Die Modellerstellung in VICUS erfolgt im Innenmaßbezug. Das Simulationsmodell besteht aus einzelnen Räumen, die durch Bauteilverknüpfungen miteinander verbunden sind. Jeder Begrenzungsfläche eines Raumes (Wände, Bodenplatte,

Decken und Dachflächen) ist ihr entsprechender Bauteilaufbau zugewiesen. Zusätzlich ist über Randbedingungen festgelegt, ob das entsprechende Bauteil an Außenluft, Erdreich oder an einen benachbarten Innenraum grenzt. Außerdem wird jedem Raum ein Nutzungsprofil zugeordnet, über das Informationen zur Nutzung und der Konditionierung hinterlegt werden. Abbildung 2.4 zeigt beispielhaft ein Raummodell in VICUS.

Die Modellerstellung in VICUS kann händisch in der 3D-Eingabemaske erfolgen. Es können aber auch Zeichenvorlagen über DXF- oder PDF-Dateien und fertige Modelle als IFC-Datei (Industry Foundation Classes) eingeladen werden.

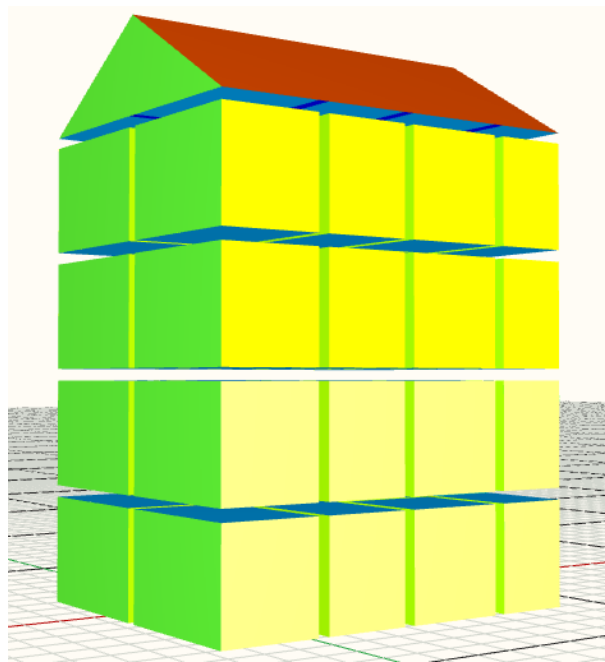


Abbildung 2.4: Beispiel eines Raummodells in VICUS

Die Verknüpfung aneinander grenzender Räume erfolgt über die Funktion „Clippen“. Dabei wird zu jeder Innenfläche eines Raumes die gegenüberliegende Fläche eines anderen Raumes gefunden. Treffen mehrere kleinere Flächen auf eine größere Fläche, wird diese in entsprechende Teilflächen zerlegt. Teilflächen ohne gegenüberliegende Fläche werden in VICUS als adiabat gekennzeichnet. Abweichend von der Definition adiabater Bauteilen in der VDI 2078 und VDI 6020 bedeutet dies hier, dass einer Bauteilseite keine Randbedingung zugewiesen ist.

Abbildung 2.5 veranschaulicht dieses Prinzip: Die Innenwand des großen Raumes wird durch das Clippen in drei Teilflächen zerteilt. Zwei Wandteilen haben je eine gegenüberliegende Wand der anderen Räume als Partner, während das mittlere, hellblau markierte Wandstück keinen direkten Partner besitzt.

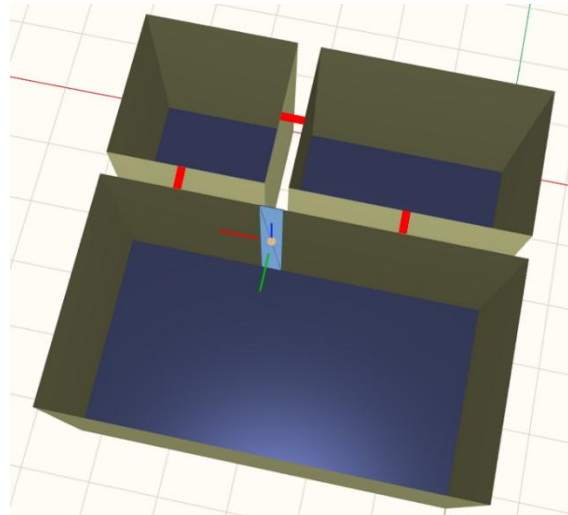


Abbildung 2.5: Bauteilverknüpfungen in VICUS

In Kombination mit dem NANDRAD-Solver ermöglicht VICUS:

- die Berechnung des Jahresenergieverbrauch
- die Ermittlung der Jahreslastkennlinien
- die Heizlastberechnung in Anlehnung an die DIN EN 12831
- die Kühllastberechnung in Anlehnung an die VDI 2078
- Komfortanalysen in Anlehnung an die DIN 16798 sowie
- den Nachweis des Sommerlichen Wärmeschutzes nach DIN 4108-2

Dadurch können energetische Maßnahmen bewertet, Risikobewertungen durchgeführt und Kosten-Nutzen-Analysen erstellt werden. [15]

2.5 Vorliegende Algorithmen/Stand der Forschung

Mit der Transformation innenmaßbezogener Raummodelle in der Gebäudesimulation haben sich bereits andere wissenschaftliche Arbeiten beschäftigt. Besonders hervorgehoben werden kann hier die Bachelorarbeit zur Transformation der Positionierung von IFC-Raumbegrenzungsflächen, die 2022 von Alex Junck an der RWTH Aachen geschrieben wurde [16] und die Projektarbeit zur Erarbeitung eines Algorithmus zur Transformation innenmaßbezogener 3D-Gebäudemodelle zum Außenmaßbezug aus 2024 von Toni Bartel (TU Dresden) [17]. Die geometrischen Ansätze beider Arbeiten sind sehr ähnlich. Allerdings arbeitet Junck in seinem Algorithmus, welcher in Python programmiert wurde, direkt mit IFC-Dateien und den darin enthaltenen Space Boundaries, während Bartels C++-Algorithmus VICUS-Dateien verarbeitet.

Diese Arbeit wird auf Bartels Algorithmus aufbauen, daher wird dieser nachfolgend genauer beschrieben:

Der Algorithmus setzt sich aus 3 Teilen zusammen, die der Reihe nach durchlaufen werden. Zuerst werden zu jeder Fläche die Nachbarflächen bestimmt, dann werden die extrudierten Flächen erstellt. Zuletzt werden die Flächen geprüft und gegebenenfalls korrigiert. Eine grobe Übersicht über den Ablauf des gesamten Algorithmus kann Abbildung 2.6 entnommen werden.

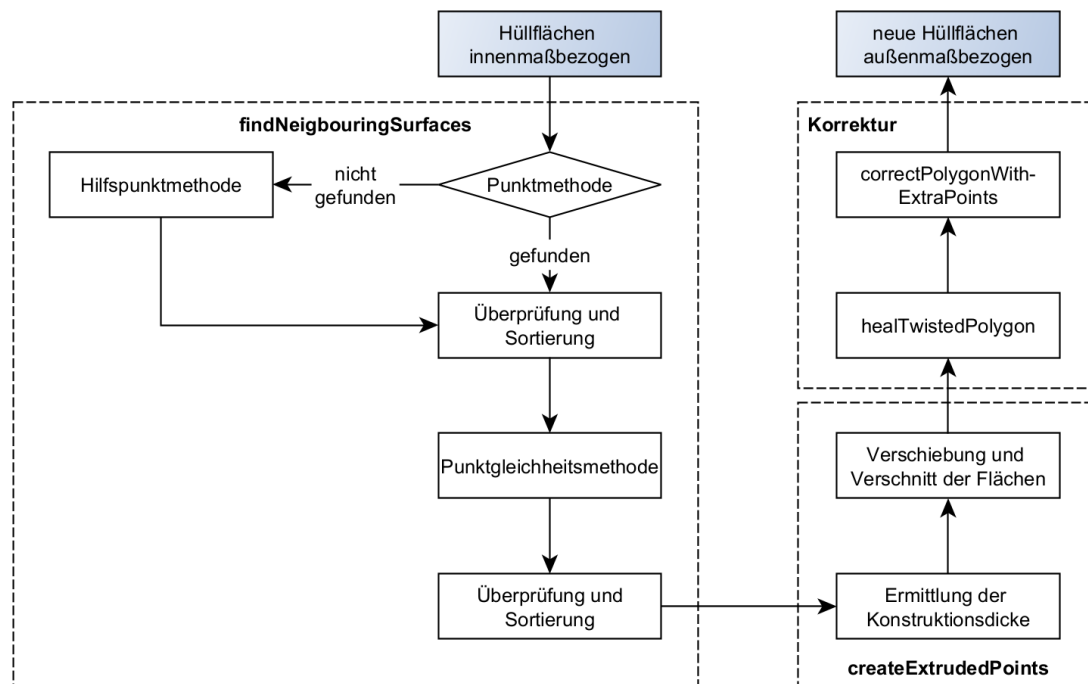


Abbildung 2.6: Ablauf des Algorithmus von Toni Bartel [17, S. 16]

2.5.1 Nachbarflächen finden

Zuerst werden in der Funktion `findNeighbouringSurfaces` für alle Flächen die angrenzenden Nachbarflächen gesucht, da diese später für die Bildung der Transformationspunkte benötigt werden. Dabei werden die verschiedenen möglichen Lagebeziehungen von zwei Flächen untereinander berücksichtigt. Potentielle Lagebeziehungen zwischen zwei Flächen sind in Abbildung 2.7 dargestellt.

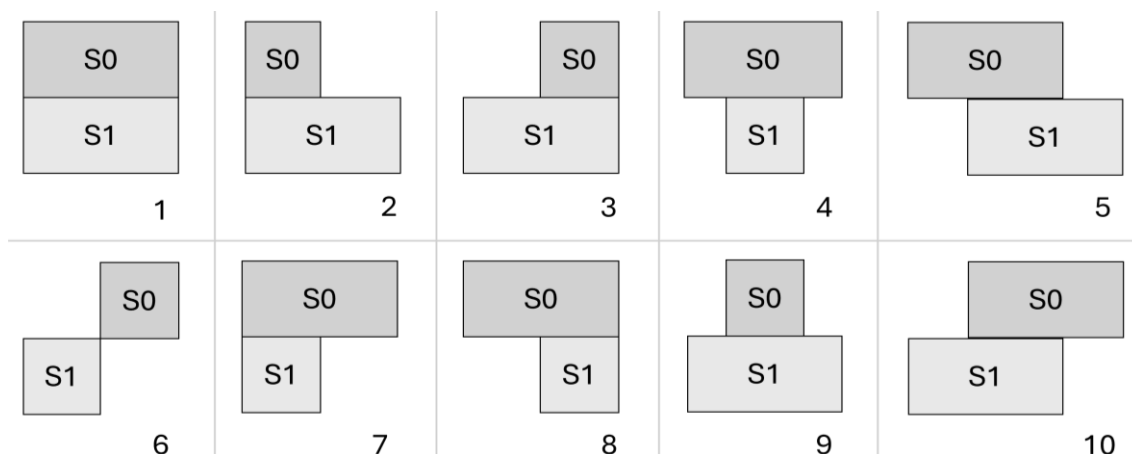
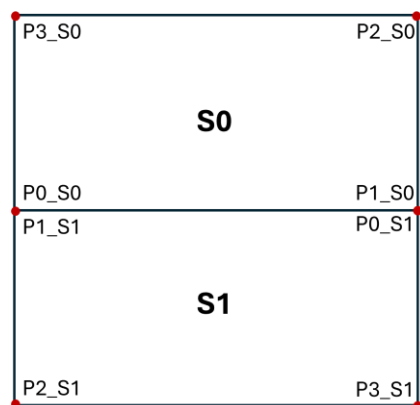


Abbildung 2.7: mögliche Lage zweier Flächen zueinander

Flächen, die sich genau zwei Eckpunkte bzw. eine Kante teilen, werden mit der **Punkt-Methode** ermittelt (Abbildung 2.7, Feld 1). In dieser Funktion wird für jeweils zwei Flächen über alle Punkte iteriert, um identische Kanten zu ermitteln. Ist der Abstand zwischen dem aktuellen Punkt der Ausgangsfläche S0 und dem Nachfolgepunkt der anderen Fläche S1 und der Abstand des Nachfolgepunktes von S0 und dem aktuellen Punkt der Fläche S1 Null, teilen beide Flächen die Kante (Abbildung 2.8 und Formel **(1)** [17, S. 17]). Die Nachbarflächen werden über die, in Kapitel 2.6 näher beschriebenen `pointInfo`-Objekte gespeichert. Findet die Funktion eine übereinstimmende Kante, wird ein `pointInfo`-Objekt mit den Informationen zu der Nachbarfläche angelegt und gespeichert. [17, S. 17 f.]



$$\overrightarrow{S0_i S1_{i+1}} = 0 \quad \&\& \quad \overrightarrow{S0_{i+1} S1_i} = 0 \quad (1)$$

Abbildung 2.8: Punkt-Methode

Stimmt nur ein Teil der Kanten von zwei Flächen überein, wird die **Hilfspunkt-Methode** verwendet (Abbildung 2.7, Feld 2-5 und 7-10). Dafür muss mindestens ein Teil der Kante der Ursprungsfläche mit einem Teil der Kante einer anderen Fläche übereinstimmen. Sobald eine Nachbarfläche an der Ausgangskante gefunden wird, wird ein neues `pointInfo`-Objekt für die Nachbarfläche angelegt. Es wird überprüft, ob die Nachbarfläche einen Hilfspunkt produziert, einen Hilfspunkt

besitzt oder sowohl einen Hilfspunkt produziert als auch einen besitzt. Die Koordinaten der entsprechenden Hilfspunkte werden in den `pointInfo`-Objekten gespeichert. Zusätzlich wird die bereits abgearbeitete aufaddierte Länge der Kante mit der Gesamtlänge verglichen. Die Funktion iteriert über die anderen Flächen des Raumes bis zur gesamten Länge der Ausgangskante die Nachbarflächen gefunden wurden. [17, S. 18 ff.]

Anhand Abbildung 2.9. kann die Bildung der Hilfspunkte nachvollzogen werden. Fläche S1 produziert einen Hilfspunkt P_Aux1 auf die Ausgangsfläche S0. Fläche S2 besitzt diesen Hilfspunkt auch und produziert einen weiteren Hilfspunkt P_Aux2. Die Fläche S3 hat nur den Hilfspunkt P_Aux2 und produziert keinen neuen.

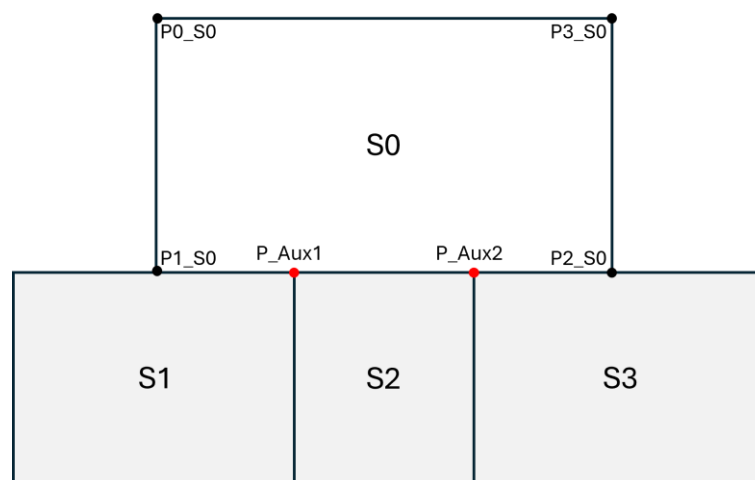


Abbildung 2.9: Hilfspunktmethode

Bereits nach der Punkt- und Hilfspunktmethode erfolgt eine Sortierung der gefundenen Nachbarflächen über die Funktion `sortUsedElements`. In dieser wird zunächst geprüft, dass in den Flächen zu jeder Kante alle Nachbarn gefunden wurden. Daraufhin werden die `pointInfo`-Objekte geordnet. Die Objekte werden ausgehend von einem Startpunkt in Drehrichtung umlaufend sortiert.

Auch die Flächen, die keine Kante sondern nur einen Punkt gemeinsam haben, müssen eingeordnet werden (Abbildung 2.7, Feld 6). Dafür gibt es die **Punktgleichheitsmethode**. In dieser werden die Nachbarflächen der direkten Nachbarn auf Punktgleichheit mit der Ausgangsfläche untersucht und deren `pointInfo`-Objekte ebenfalls angelegt und gespeichert. [17, S. 21 ff.]

Anschließend müssen auch die `pointInfo`-Objekte der punktgleichen Nachbarflächen in die bereits durch die Funktion `sortUsedElements` erstellte Reihenfolge einsortiert werden. Die richtige Abfolge der Nachbarflächen ist wichtig, damit im darauf folgenden Verschnitt-Algorithmus die richtigen Punkte gebildet werden. [17, S. 24]

2.5.2 Verschieben und Verschneiden der Flächen

In der Funktion `createExtrudedPoints` wird für den Verschnitt zuerst die Verschiebungslänge anhand der Bauteildicke ermittelt. Die Berechnung der geforderten Bauteildicke erfolgt nach den Anforderungen des GEG bzw. der DIN V 18599. Die Verschneidung erfolgt aus jeweils drei benachbarten Flächen: der Ausgangsfläche und zwei aufeinanderfolgend eingeordneten Nachbarflächen. Die Flächen werden entlang ihrer Normalenvektoren um die zuvor ermittelte Länge verschoben und daraufhin der Schnittpunkt zwischen den Flächen berechnet. Dabei wird berücksichtigt, dass die Flächen untereinander parallel sein können und gegebenenfalls eine Hilfsebene verwendet werden muss, um einen Punkt zu bilden. [17, S. 25 ff.]

2.5.3 Korrektur

Wenn eine neue Fläche aus den Transformationspunkten gebildet wurde, wird direkt im Anschluss mit der Funktion `healTwistedPolygon` überprüft, ob die Fläche verwunden ist. In einem verwundenen Polygon überschneiden sich Kanten, wodurch keine zusammenhängende Fläche gebildet wird. Ein Beispiel dafür zeigt Abbildung 2.10. Verwundene Polygone bilden keine validen Flächen, deshalb müssen sie korrigiert werden. Es wird geprüft, ob die Kanten zwischen jeweils vier aufeinanderfolgenden Punkten einen Schnittpunkt bilden. Wird ein Schnittpunkt gefunden, wird der zweite der vier Punkte gelöscht und der dritte Punkt durch den ermittelten Schnittpunkt ersetzt. [17, S. 28]

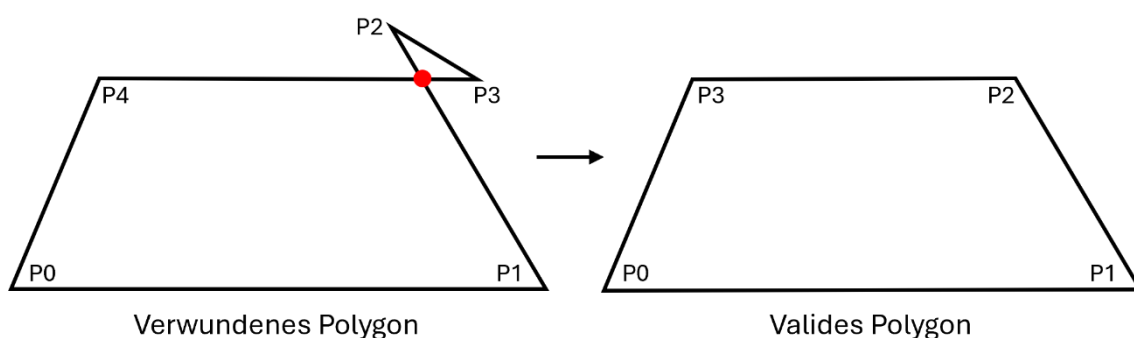


Abbildung 2.10: Reparatur eines verwundenen Polygons

Nachdem alle Transformationsflächen gebildet wurden erfolgt die letzte Überprüfung mit der Funktion `correctPolygonWithExtraPoints`. Dieser kontrolliert, dass jeder Punkt in mindestens einer weiteren Fläche vorhanden ist. Wenn ein Punkt nur an einer Stelle vorkommt, kann er nicht an der Bildung eines geschlossenen Körpers beteiligt sein. Der Punkt muss entfernt und durch den richtigen Punkt, welcher aus dem Vorgänger- und Nachfolger-Punkt und den jeweiligen Nachbarflächen ermittelt wird, ersetzt werden. [17, S. 28 ff.]

Damit ist der Transformationsprozess abgeschlossen. Zuletzt wird die Ausgabedatei erstellt.

2.5.4 Ein- und Ausgabe der Daten

Um ein Raummodell zu transformieren, wird eine VICUS-Datei in den Algorithmus eingegeben. Nach Durchlaufen des beschriebenen Algorithmus wird eine Kopie der ursprünglichen Datei angelegt und die erzeugten Transformationsflächen als zusätzliche Flächenelemente hinzugefügt.

Der Algorithmus wurde für verschiedene Einraum- und einfache Mehrraummodelle getestet und zeigt dafür gute Ergebnisse. Es gibt jedoch noch einige Grenzen und Fälle, in denen keine sinnvollen Ergebnisse erzeugt werden können. Diese werden in Kapitel 3.1 näher beschrieben.

2.6 Datenstruktur, Festlegungen und Definitionen

Für die nachfolgenden Ausführungen zur Erweiterung von Bartels Algorithmus sind einige grundlegende Kenntnisse über die Datenstruktur des Algorithmus und den Aufbau einer VICUS-Datei von Vorteil.

Eine VICUS-Datei ist vom groben ins feine strukturiert (Gebäude – Geschoss – Raum – Fläche – Konstruktion – Material). Den einzelnen Räumen sind Nutzungsprofile zugeordnet und sie sind über Bauteilverknüpfungen miteinander verbunden. Den Bauteilen sind Konstruktionen und Übergangskoeffizienten zugeordnet. Die Übergangskoeffizienten beschreiben die stoffliche Grenze zwischen dem Bauteil und der Umwelt. Darüber wird definiert, ob das Bauteil an Außenluft, Erdreich oder einen Innenbereich grenzt. Mithilfe der Übergangsbedingungen und der Nutzungsprofile kann daher bestimmt werden, welche Bauteilart vorliegt und ob das Bauteil zwischen zwei unterschiedlich temperierten Räumen liegt. Aus den Konstruktionsaufbauten kann die jeweilige Bauteildicke bestimmt werden. Jedes

Element eines Modells besitzt eine eigene ID, über die es eindeutig identifiziert werden kann.

Der Algorithmus iteriert über das gesamte Modell, welches auch aus mehreren Gebäuden bestehen kann. Dabei werden in jedem Gebäude die Räume etagenweise durchlaufen. Für jeden Raum werden die einzelnen Begrenzungsflächen betrachtet (Boden, Wände und Decke) und die entsprechende extrudierte Fläche berechnet. Diese wird als neues geometrisches Flächenelement erstellt. Alle Flächen sind ebene Polygone „Polylines“, die aus 3D-Punkten bestehen. Jeder Fläche ist ihr Normalenvektor zugeordnet, der immer nach außen bzw. vom Raum weg zeigt. Ein Normalenvektor steht stets senkrecht zu seiner entsprechenden Fläche. Damit ist vordefiniert, in welche Richtung die Ausgangsfläche verschoben wird, weil es sich bei der Extrusion um eine Parallelverschiebung in Normalenrichtung handelt.

Für die Verarbeitung der Flächen im Algorithmus werden für jeden Punkt Metainformationen gespeichert. Diese `pointInfo`-Objekte enthalten folgende Informationen, deren Werte teilweise erst im Verlauf des Algorithmus hinzugefügt werden:

- ID der Ausgangsfläche
- Index des Punktes in der Polyline der Ausgangsfläche
- ID der Nachbarfläche
- Index des Punktes in der Polyline der Nachbarfläche
- Wie stehen Ausgangs- und Nachbarfläche zueinander?
 - o in Winkel zueinander
 - o komplanar in einer Ebene
 - o komplanar aber nicht in der selben Ebene, aufgrund unterschiedlicher Verschiebungsdicken
- Produziert die Fläche einen Hilfspunkt auf der Nachbarfläche? (Wahr/Falsch)
- 3D-Koordinaten des produzierten Hilfspunktes
- Hat die Fläche einen Hilfspunkt (der durch eine Nachbarfläche produziert wurde)? (Wahr/Falsch)
- 3D-Koordinaten des Hilfspunktes
- Wurde die komplette Kantenlänge betrachtet? (Wahr/Falsch)
- Verwendete Kantenlänge (Teil der Ausgangskante, welcher von einer anderen Kante verwendet wird)

Die in der Funktion `createExtrudedPoints` erstellten Transformationsflächen „*newPolylines*“ werden als Vektoren von 3D-Punkten gespeichert.

3 Methodik

Der vorliegende Algorithmus wird schrittweise auf potentielle Schwachstellen und Unvollständigkeiten geprüft und gegebenenfalls angepasst oder erweitert. Zur Durchführung der Tests und zur Darstellung des Bearbeitungsfortschrittes, werden in VICUS Testmodelle erstellt. Diese entsprechen typischen und komplexeren Dachgeometrien, da in diesem Bereich das größte Verbesserungspotential erwartet wird.

Das erste Testmodell ist ein L-Dach mit einer großen Zwerchgaube, welches in Abbildung 3.1 dargestellt wird. Der Transformationskörper dieses Modells weist zu Beginn des Arbeitsprozesses, wie in Abbildung 3.2 gezeigt, noch zahlreiche Unstimmigkeiten auf und eignet sich somit gut zur Identifikation von Problemstellen und zur Dokumentation des Fortschrittes am Algorithmus.

Im weiteren Verlauf der Arbeit werden zusätzliche Testmodelle erstellt, auf die in Kapitel 4 näher eingegangen wird. Diese Modelle dienen dazu, verbleibende Inkonsistenzen zu identifizieren und den Ablauf der modifizierten bzw. neu hinzugefügten Algorithmen zu überprüfen.

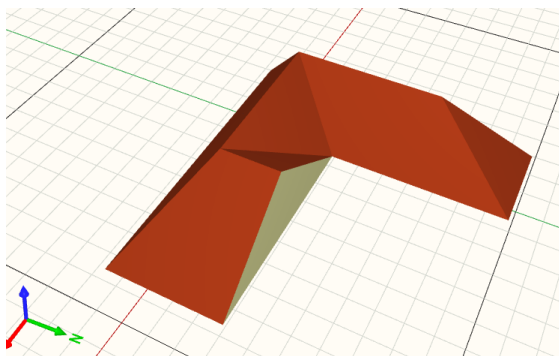


Abbildung 3.1: Testgeometrie 1: L-Dach mit Gaube

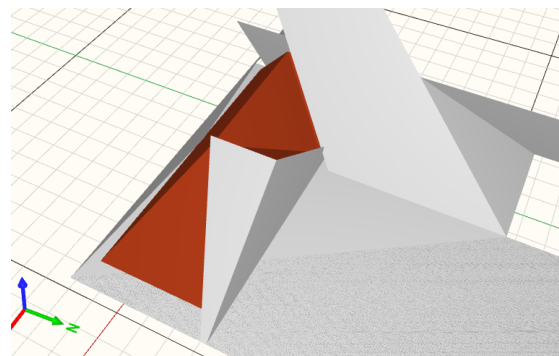


Abbildung 3.2: Ausgangsstand der Transformation des L-Dachs mit Gaube

3.1 Erweiterungspotential

Das Ziel der Erweiterung des Algorithmus ist, die zuverlässige Transformation komplexer Raumgeometrien zu ermöglichen. Hierfür müssen zunächst jene Fälle identifiziert werden, die im bisherigen Algorithmus entweder nicht berücksichtigt oder nur unvollständig behandelt werden.

In der Funktion `findNeighbouringSurfaces` arbeitet die Punktmethode bereits zuverlässig und bedarf keiner Anpassung. Optimierungsbedarf besteht jedoch in der Hilfspunktmethode sowie der anschließenden Sortierung

(sortUsedElements). Insbesondere bei Räumen mit vielen parallelen Flächen treten Inkonsistenzen bei der Zuordnung der Nachbarschaftsbeziehungen auf.

Auch die Punktgleichheitsmethode ist bislang auf einfache Fälle beschränkt. Der Algorithmus ordnet pro Punkt maximal eine Fläche zu. In komplexeren Modellen können jedoch mehrere Flächen an einem gemeinsamen Punkt zusammentreffen, deren gegenseitige Lagebeziehungen für die korrekte Transformation berücksichtigt werden müssen.

Darüber hinaus zeigt sich Erweiterungsbedarf in der Funktion `createExtrudedPoints`, in der die Schnittpunkte aus jeweils drei Flächen gebildet werden.

Abbildung 3.3 stellt potentielle Lagebeziehungen zwischen drei Ebenen dar. Während für Standardfälle, wie der Bildung eines eindeutigen Schnittpunktes (Fall 5) und der vollständigen Parallelität der Flächen (Fall 1) bereits die zuverlässige Berechnung korrekter Transformationspunkte möglich ist, werden andere Fälle bisher nur unzureichend oder gar nicht berücksichtigt.

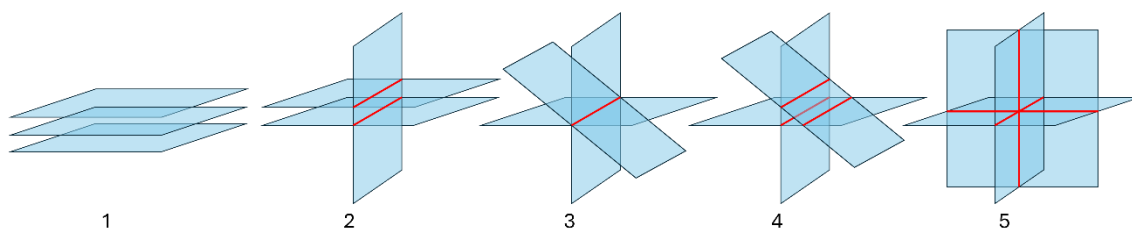


Abbildung 3.3: potentielle Lagebeziehungen zwischen 3 Ebenen [18]

Für den Fall, dass sich zwei parallele und eine nicht parallele Fläche (Fall 2) schneiden, wurde bereits ein algorithmischer Ansatz entwickelt. Hier muss ermittelt werden, ob die Erstellung fehlerhafter Transformationskörper bereits durch die Überarbeitung der Hilfspunktmethode und der Sortierung unterbunden werden kann, oder weitere Anpassungen nötig werden.

Bisher wird nur die Lage von drei direkt aneinandergrenzende Flächen betrachtet. Wenn auch die eingeordneten Flächen berücksichtigt werden, kann es vorkommen, dass die Flächen nicht parallel aber trotzdem linear abhängig sind (Fall 3 und 4). Bei Gebäuden tritt dieser Fall vor allem bei Zwerchgauben auf und wird in Abbildung 3.4 durch die beiden hellblau markierten Flächen und den Boden gezeigt. Um auch für entsprechende Raumgeometrien korrekte Transformationsmodelle zu erzeugen, muss hier eine weitere Fallunterscheidung ergänzt werden.

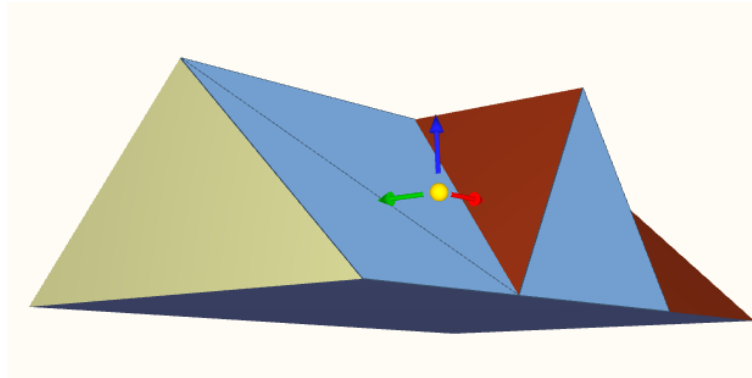


Abbildung 3.4: linear abhängige Flächen bei einem Dach mit Zwerchgaube

Die folgenden Abschnitte dokumentieren die vorgenommenen algorithmischen Änderungen und Ergänzungen.

3.2 Überarbeitungen in der Funktion `findNeighbouringSurfaces`

Wie bereits in Kapitel 2.5.1 beschrieben, ermittelt die Funktion `findNeighbouringSurfaces` im Rahmen der Punkt- und Hilfspunktmethode die jeweiligen Nachbarflächen einer Ausgangsfläche. Für jede gefundene Nachbarfläche wird ein `pointInfo`-Objekt erzeugt und in einem Vektor der jeweiligen Ausgangsfläche zugeordnet. Die Lagebeziehung der Nachbarfläche wird dabei zunächst über den Index des Eckpunkts der Ausgangsfläche (`idxOriginalPolygonPoint`) gespeichert.

In der Punkt-Methode ist diese Zuordnung eindeutig, da Kantenlänge und Eckpunkte von Ausgangs- und Nachbarfläche genau übereinstimmen. In der Hilfspunktmethode können hingegen mehrere Nachbarflächen an dieselbe Kante der Ausgangsfläche angrenzen, sodass mehrere `pointInfo`-Objekte dem selben `idxOriginalPolygonPoint` zugeordnet sind.

3.2.1 Hilfspunktmethode

Zur Überprüfung, ob für eine Kante alle Nachbarflächen erfasst wurden, wird die bereits abgearbeitete Kantenlänge gespeichert und mit der Gesamtkantenlänge verglichen. Die Suche nach Nachbarflächen erfolgt durch Iteration über alle übrigen Flächen des Raums. Wird eine passende Nachbarkante in einer Fläche gefunden, wird mit der nächsten Fläche fortgefahren.

In Folge der Flächenzerlegung durch das Bauteilclipping (siehe Kapitel 2.4) kann es vorkommen, dass eine Fläche mehrfach an eine Kante der Ausgangsfläche angrenzt (siehe Abbildung 3.5). In diesem Fall kann mit dem vorliegenden Algorithmus nur

eins der gemeinsamen Kantenstücke identifiziert werden, wodurch nicht die gesamte Kantenlänge erfasst wird.

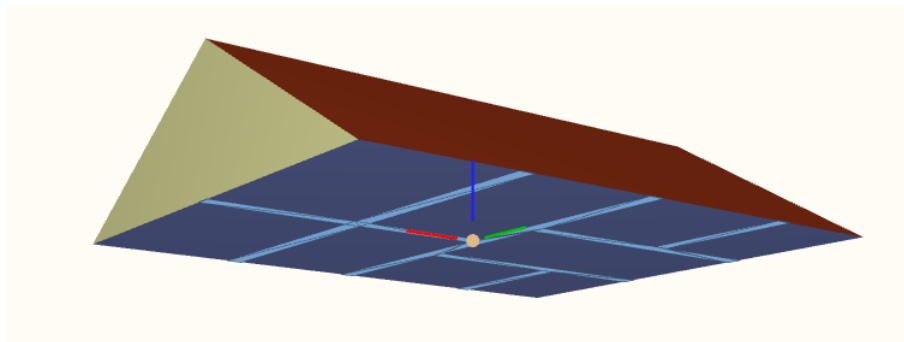


Abbildung 3.5: Fläche, die mehrfach an andere Fläche grenzt

Um dennoch alle relevanten Nachbarflächen zu finden, werden in Folge sämtliche Kanten der übrigen Flächen betrachtet. Die Iteration wird erst beendet, wenn entweder alle Kanten geprüft oder Nachbarflächen über die gesamte Kantenlänge gefunden wurden.

3.2.2 Sortierung

In der Funktion `sortUsedElements` wird zunächst überprüft, ob für jede Kante Nachbarflächen über deren komplette Länge gefunden wurden. Nach der Überarbeitung der Hilfspunktmethode treten in diesem Schritt keine Fehler mehr auf.

Anschließend erfolgt die Sortierung der `pointInfo`-Objekte einer Fläche in umlaufender Drehrichtung. Dabei dienen der `idxOriginalPolygonPoint` sowie die produzierten und vorhandenen Hilfspunkte als Orientierung (siehe Abbildung 3.6).

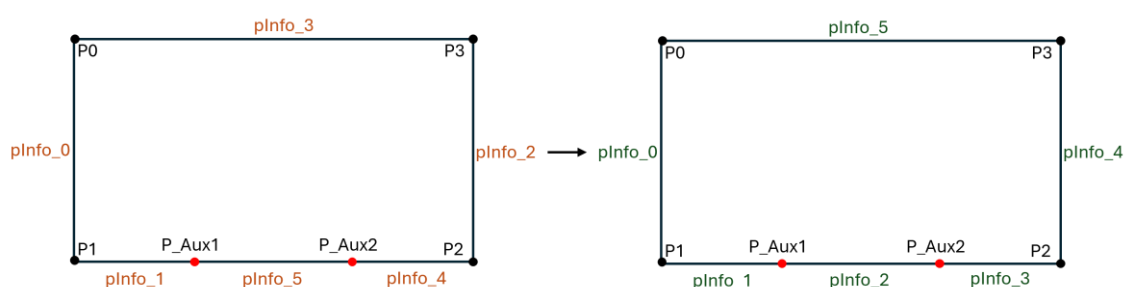


Abbildung 3.6: Sortierung der `pointInfo`-Objekte

In der ursprünglichen Implementierung wurden die Objekte nacheinander durchlaufen und anhand zweier Bedingungen einsortiert:

- direkte Zuordnung für Objekte ohne Hilfspunkt (Punktmethode) sowie

- Übereinstimmung der Koordinaten des vorhandenen Hilfspunkts mit dem produzierten Hilfspunkt des zuvor eingeordneten Objekts

Diese Vorgehensweise ist fehleranfällig, sobald mehr als zwei `pointInfo`-Objekte einem Kantenindex zugeordnet sind, da ein vorzeitiger Abbruch der Sortierung erfolgen kann.

In der überarbeiteten Version erfolgt zunächst eine Vorsortierung der `pointInfo`-Objekte nach den `idxOriginalPolygonPoint(s)`. Ist einem Index nur ein Objekt zugeordnet, wird es direkt einsortiert. Sind mehrere Objekte vorhanden, wird zunächst das Objekt gewählt, welches nur einen Hilfspunkt produziert, aber keinen besitzt. Anschließend werden die Objekte, die sowohl einen Hilfspunkt besitzen, als auch einen produzieren, anhand der Übereinstimmung ihrer Hilfspunkte mit dem jeweils produzierten Hilfspunkt des Vorgängerobjekts sortiert. Das verbleibende Objekt, welches einen Hilfspunkt besitzt, jedoch keinen produziert, wird abschließend einsortiert. Auf diese Weise wird eine robuste und vollständige Sortierung der `pointInfo`-Objekte sichergestellt.

3.2.3 Punktgleichheitsmethode

Bei ersten Tests des Algorithmus fällt auf, dass die Ermittlung der Flächen mit Punktgenauigkeit nicht korrekt erfolgt. Es wird maximal eine Fläche einsortiert, bevor bereits der nächste Punkt betrachtet wird. Da jedoch sämtliche Flächen, die an einem Punkt an die Ausgangsfläche angrenzen, potentiell Einfluss auf die spätere Transformationsfläche haben können, ist eine Überarbeitung des Algorithmus erforderlich.

Zur Lösung der Problems wird die Funktion `findNextSurface` eingeführt, welche die Aufgabe der Punktgleichheitsmethode übernimmt. Sie ist in die Funktion `findNeighbouringSurfaces` integriert und wird nach Abschluss der Sortierfunktion `sortUsedElements` aufgerufen.

Zunächst werden die Punkte einer Ausgangsfläche S_0 nacheinander betrachtet. Zu einem Punkt P_{0,S_0} wird die Nachbarfläche S_1 bestimmt. Die Zuordnung der direkten Nachbarn zu den Punkten ist bereits in der Punkt- und Hilfspunktmethode erfolgt, sodass die Flächen direkt abgerufen werden können.

Anschließend wird für den folgenden Punkt P_{1,S_0} eine weitere Nachbarfläche ermittelt. Diese gefundene Fläche muss ungleich der zuvor bestimmten Nachbarfläche S_1 sein und definiert die Abbruchfläche S_{End} . Ergibt sich erneut die

Fläche S1, wird direkt der nächste Punkt betrachtet, da keine weitere Einsortierung möglich ist.

In der Fläche S1 wird außerdem der Punkt P0_S1 gesucht, der mit dem Punkt P1_S0 übereinstimmt. Zusätzlich werden die Normalenvektoren der Ausgangsfläche und der beiden direkten Nachbarflächen an diesem Punkt gespeichert. Diese Informationen bilden die Eingangsparameter der rekursiven Funktion `findNextSurface`.

Innerhalb der Funktion wird zunächst der Vorgängerpunkt P_Pre_S1 bestimmt, der entweder auf eine neue Fläche SX oder auf die Abbruchfläche verweist. In SX wird der Punkt gesucht, der mit P0_S1 identisch ist. Anschließend wird geprüft, ob SX der Abbruchfläche entspricht. Ist dies nicht der Fall, wird zusätzlich überprüft, ob SX parallel zu der Ausgangsfläche oder einer der beiden direkten Nachbarflächen ist. Parallele Flächen werden nicht eingeordnet, da sie keinen neuen relevanten Punkt erzeugen. Nicht parallele Flächen werden hingegen einsortiert.

Die Funktion wird rekursiv aufgerufen, bis die Abbruchfläche erreicht ist. Mit Erreichen dieser Fläche ist das Abbruchkriterium erfüllt und die Rekursion wird beendet.

Der Ablauf der Funktion ist in Abbildung 3.7 exemplarisch dargestellt. Die rote Fläche S0 stellt die Ausgangsfläche dar, die grüne Fläche S1 die erste direkte Nachbarfläche und die gelbe Fläche die Abbruchfläche. Ausgehend von diesen Flächen werden die angrenzenden Flächen sukzessive überprüft und (sofern sie nicht parallel sind) eingeordnet, bis die Abbruchfläche erreicht wird.

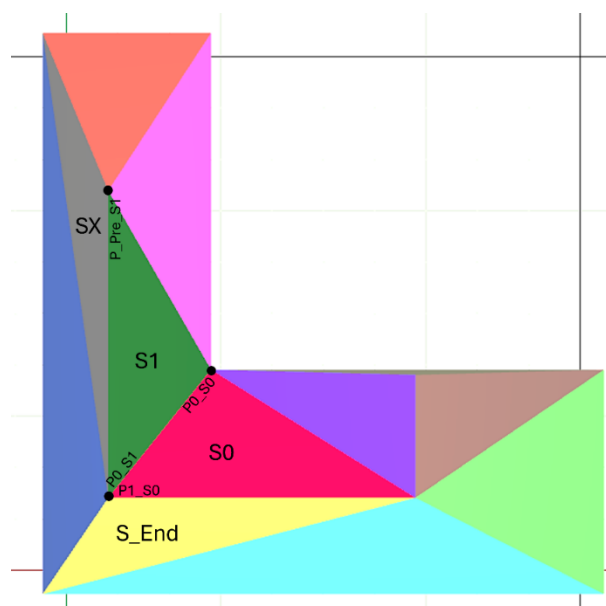


Abbildung 3.7: Veranschaulichung Funktionsweise `findNextSurface`

3.3 Ergänzungen in der Funktion `createExtrudedPoints`

Da Bartels Algorithmus für bestimmte Lagebeziehungen dreier Flächen zueinander bislang keine korrekten Transformationspunkte bilden kann, wurden zusätzliche Berechnungsfälle entwickelt und Ergänzungen in bestehenden Fällen vorgenommen. Die folgenden Unterkapitel beschreiben die vorgenommenen Änderungen und Erweiterungen.

3.3.1 Parallele Flächen

In der Funktion `createExtrudedPoints` werden jeweils drei Flächen betrachtet, die gemeinsamen einen Schnittpunkt bilden. Zunächst erfolgt die Prüfung auf Parallelität der beteiligten Flächen. Sind alle drei Flächen parallel, wird der Schnittpunkt entlang der gemeinsamen Normalenrichtung um die jeweilige Bauteildicke nach außen verschoben.

Sind hingegen nur zwei der drei Flächen parallel, wird zur Bestimmung eines eindeutigen Transformationspunktes eine Hilfsebene eingeführt. Deren Normalenvektor wird aus dem Kreuzprodukt der Normalenvektoren der beiden nicht parallelen Flächen gebildet. Als Stützpunkt dient der Schnittpunkt der Flächen.

Treffen genau drei Flächen aufeinander, liefert der Algorithmus korrekte Transformationspunkte, da die Verschiebung ausschließlich innerhalb der Hilfsebene erfolgt. Grenzen an den Schnittpunkt jedoch noch weitere Flächen, kann es zu Unstimmigkeiten in den entstehenden Transformationsflächen kommen.

Ein typisches Beispiel ist in Abbildung 3.8 dargestellt. Damit durch die Gaube kein Loch in der Dachfläche entsteht, wird die Fläche in zwei Stücke geteilt. Dadurch treffen an dem rot markierten Schnittpunkt vier Flächen in unterschiedlichen Neigungen aufeinander. Die Hilfsebene wird in diesem Fall ungeeignet positioniert, wodurch ein seitlich versetzter Transformationspunkt und eine Lücke im Transformationskörper entsteht.

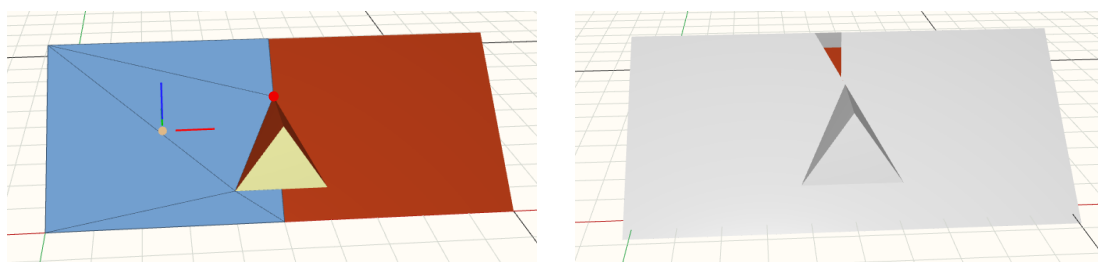


Abbildung 3.8: fehlerhafte Transformation durch unangepasste Hilfsebene

Um diese Fälle korrekt zu behandeln, werden die Regeln zur Bestimmung der Hilfsebene überarbeitet. Der Stützpunkt der Hilfsebene bleibt weiterhin der ursprüngliche Schnittpunkt der Flächen.

Fall A: Ausgangsfläche und erste Nachbarfläche sind parallel

Der Normalenvektor der Hilfsebene wird durch das Kreuzprodukt des Normalenvektors der Ausgangsfläche mit dem Vektor aus dem aktuellen Schnittpunkt und seinem Vorgängerpunkt gebildet.

Fall B: Ausgangsfläche und zweite Nachbarfläche sind parallel

Sind die Flächen keine direkten Nachbarn erfolgt die Punktbildung wie im ersten Fall. Sind die Flächen direkte Nachbarn, wird zur Bildung des Normalenvektors der Hilfsebene statt dem Vektor zwischen dem aktuellen Schnittpunkt und dem Vorgängerpunkt der Vektor zwischen dem aktuellen Schnittpunkt und dem nachfolgende Punkt in der Fläche verwendet.

Fall C: Beide Nachbarflächen sind parallel

Für diesen Fall wird die Normale der Hilfsebene wie bisher aus dem Kreuzprodukt der Normalenvektoren der Ausgangsfläche und einer der Nachbarflächen gebildet.

3.3.2 Linear Abhängige Flächen

Wie bereits in Kapitel 3.1 beschrieben, existiert bislang kein Ansatz, um linear abhängige, nicht parallele Flächen miteinander zu verschneiden. Damit auch für diesen Fall korrekte Schnittpunkte gebildet werden können, wird die Funktion `createExtrudedPoints` um eine entsprechende Fallunterscheidung erweitert.

Schneiden sich drei Ebenen in einer oder mehreren Schnittgeraden, liegt eine lineare Abhängigkeit zwischen den Normalenvektoren dieser Ebenen vor. Diese Abhängigkeit kann mithilfe der Determinante der Matrix überprüft werden, die aus den drei Normalenvektoren der Flächen gebildet wird. Ist die Determinante gleich Null, sind die Flächen linear abhängig, und der beschriebene Fall tritt ein (siehe Formel (2) [19, S. 251]).

$$\det \begin{pmatrix} x_{n0} & y_{n0} & z_{n0} \\ x_{n1} & y_{n1} & z_{n1} \\ x_{n2} & y_{n2} & z_{n2} \end{pmatrix} = 0 \quad (2)$$

Um dennoch einen Schnittpunkt aus diesen Flächen zu erzeugen, wird eine Hilfsebene eingeführt. Vor der Bestimmung der Hilfsebene anhand ihrer Normalen und ihres Stützpunktes muss jedoch zunächst die relative Lage der beteiligten Flächen analysiert werden.

Es sind zwei Fälle möglich. Im ersten Fall grenzt nur eine der Nachbarflächen direkt an die Ausgangsfläche. Durch den Einbau einer Iterationsstufe in den Extrusionsprozess (siehe Kapitel 3.4), können jedoch auch Flächen von der Punktbildung ausgeschlossen werden. Im zweiten Fall grenzen daher beide Nachbarflächen direkt an die Ausgangsfläche, weil die zwischen ihnen liegende Fläche von der Punktbildung ausgeschlossen wurde. Abbildung 3.9 zeigt ein entsprechendes Beispiel. Wenn Fläche A die Ausgangsfläche ist, stellt Fläche B den ersten (direkten) Nachbarn dar, während Fläche C als zweiter Nachbar über Punktgleichheit zugeordnet wird. Wird hingegen Fläche B als Ausgangsfläche betrachtet und wurde Fläche D zuvor von der Punktbildung ausgeschlossen, sind Fläche A und C die beiden direkten Nachbarflächen, mit denen ein oder zwei Schnittpunkte zu erzeugen sind.

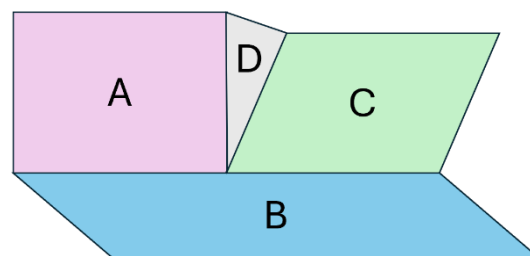


Abbildung 3.9: mögliche Positionierung linear abhängiger Flächen

Fall 1: Nur eine Nachbarfläche ist direkter Nachbar

Ist nur eine der beiden Nachbarflächen ein direkter Nachbar der Ausgangsfläche, wird die Normale der Hilfsebene aus dem Kreuzprodukt des Normalenvektors der Ausgangsebene und dem Vektor zwischen den beiden zuletzt erzeugten Transformationspunkten berechnet. Dieses Vorgehen setzt voraus, dass bereits mindestens zwei korrekte Transformationspunkte existieren. Daher wird zunächst überprüft, aus wie vielen Punkten der neue Polygonzug bereits besteht.

Ist noch kein Punkt vorhanden, wird das `pointInfo`-Objekt der Ausgangsfläche um eine Position rotiert und die Funktion `createExtrudedPoints` erneut aufgerufen. Existiert bereits ein Punkt, erfolgt die Rotation um zwei Positionen, da eine Rotation um nur eine Position erneut den Fall ohne vorhandenen Punkt erzeugen würde.

Die Berechnung des Schnittpunktes erfolgt durch die Lösung des in Formel (3) [19, S. 225 ff.] dargestellten linearen Gleichungssystems. Die Matrix $\mathbf{A}$ besteht aus den Normalenvektoren der drei beteiligten Flächen. Es werden zwei Schnittpunkte mit jeweils der Ausgangsfläche, der Hilfsebene und einer der beiden Nachbarflächen erzeugt. Dafür wird in der Matrix $\mathbf{A}$ der jeweilige Normalenvektor einer der Nachbarflächen mit dem Normalenvektor der Hilfsebene ersetzt. Der Vektor $\mathbf{b}$

ergibt sich aus dem Produkt des Normalenvektors der Hilfsebene mit dem zuletzt gebildeten Transformationspunkt des neuen Polygonzuges. Vor der Einordnung der beiden neuen Punkte wird überprüft, welcher näher an dem letzten Punkt des Polygonzuges liegt, um eine korrekte Punktreihenfolge in der Fläche sicherzustellen.

$$A * x = b \quad \text{bzw.} \quad \begin{pmatrix} x_{n0} & y_{n0} & z_{n0} \\ x_{n1} & y_{n1} & z_{n1} \\ x_{n2} & y_{n2} & z_{n2} \end{pmatrix} * \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} b1 \\ b2 \\ b3 \end{pmatrix} \quad (3)$$

Ein Beispiel dazu ist in Abbildung 3.10 dargestellt. Die Ausgangsfläche stellt die orange gefärbte Dachfläche dar und die rote Markierung zeigt den aktuell betrachteten Punkt. Die erste Nachbarfläche an diesem Punkt ist der dunkelblau gefärbte Boden und die zweite Nachbarfläche die gelb gefärbte Wand der Zwerchgaube. Die Hilfsebene steht in diesem Fall im rechten Winkel zum Boden und der Gaubenwand und verläuft direkt durch den Schnittpunkt der drei Flächen. Nach der Verschiebung der Flächen wird durch Dachfläche, Boden und Hilfsebene der blau markierte Transformationspunkt gebildet. Durch die Dachfläche, die Wand und die Hilfsebene wird der grün markierte Transformationspunkt gebildet.

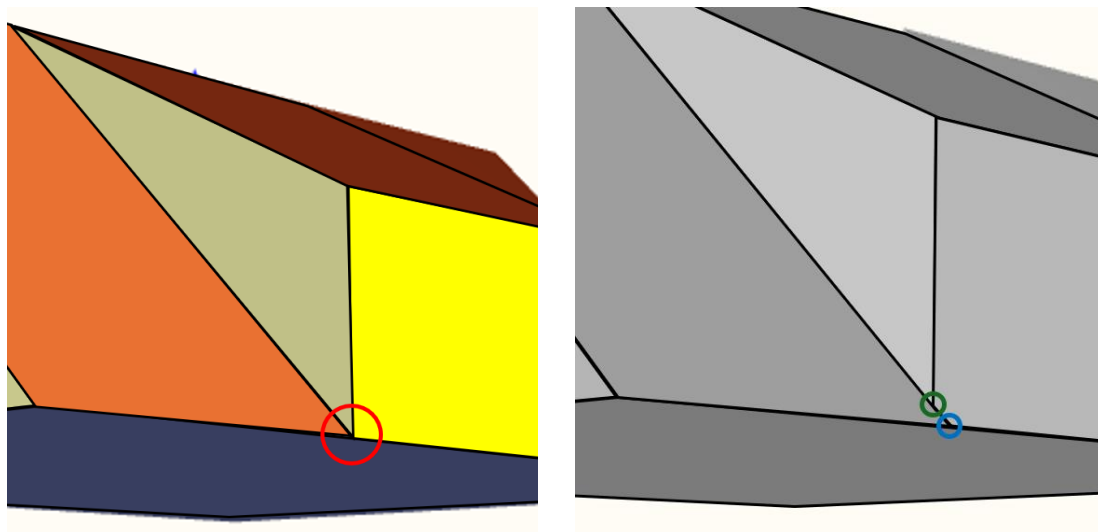


Abbildung 3.10: Transformationspunkte von linear abgängigen, nicht parallelen Flächen

Fall 2: Beide Nachbarflächen sind direkte Nachbarn

Grenzen beide Nachbarflächen direkte an die Ausgangsfläche, muss die zwischen ihnen liegende Fläche berücksichtigt werden, weil diese die Lage der Hilfsebene um ihre Bauteildicke verschiebt. Abbildung 3.11 zeigt ein entsprechendes Beispiel anhand des Transformationskörpers, der in Abbildung 3.4 dargestellten Dachgeometrie. Die Hilfsebene zur Erzeugung der Transformationspunkte für die Bodenfläche wurde hier exakt durch den ursprünglichen Schnittpunkt gelegt.

Aufgrund der Verschiebung des Daches der Zwerggaube liegt die Hilfsebene jedoch nicht mehr an der korrekten Position.

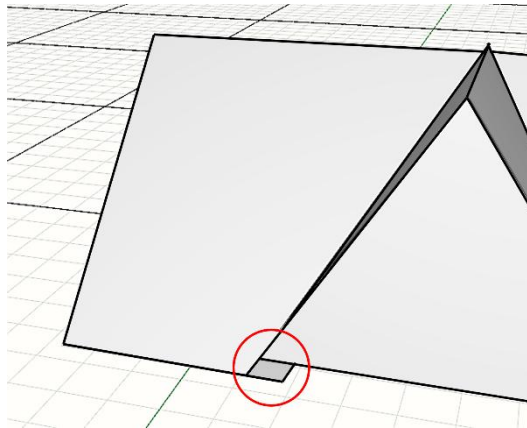


Abbildung 3.11: Fehlerhafte Punktbildung durch unverschobene Hilfsebene

In diesem Fall erfolgt die Punktbildung abweichend vom zuvor beschriebenen Vorgehen. Zuerst wird die Fläche S3 identifiziert, die zwischen den beiden direkt angrenzenden Nachbarflächen S1 und S2 liegt. Von dieser Fläche werden sowohl die Bauteildicke als auch der Normalenvektor benötigt. Anschließend wird der aktuelle Schnittpunkt der Ausgangsfläche mit den beiden Nachbarflächen bestimmt.

Die Normale der Hilfsebene $\vec{n}_H$ wird in diesem Fall aus dem aktuellen Punkt P und seinem Vorgänger P_{Pre} der Ausgangsfläche S0 bestimmt. Der Stützpunkt der Hilfsebene wird um den Einfluss der Bauteildicke d_3 der Fläche S3 verschoben. Wie in Abbildung 3.12 dargestellt, ist dafür auch die Dicke der Ausgangsfläche zu berücksichtigen.

Der Winkel α ergibt sich aus dem Arkuskosinus des Skalarproduktes zwischen den Normalenvektoren der Fläche S3 und der Hilfsebene. Die Verschiebungsdistanz x_1 ergibt sich aus dem Produkt der Bauteildicke d_0 der Ausgangsfläche mit dem Tangens des Winkels α , während x_2 mit dem Quotienten aus der Dicke d_3 der Fläche S3 mit dem Kosinus von α berechnet wird. Ist der Winkel α kleiner als 90 Grad erfolgt die Verschiebung des Ausgangspunktes P in Richtung der Normalen der Hilfsebene, ist der Winkel größer erfolgt die Verschiebung in die entgegengesetzte Richtung der Normalen (siehe Formeln **(4)** bis **(7)**).

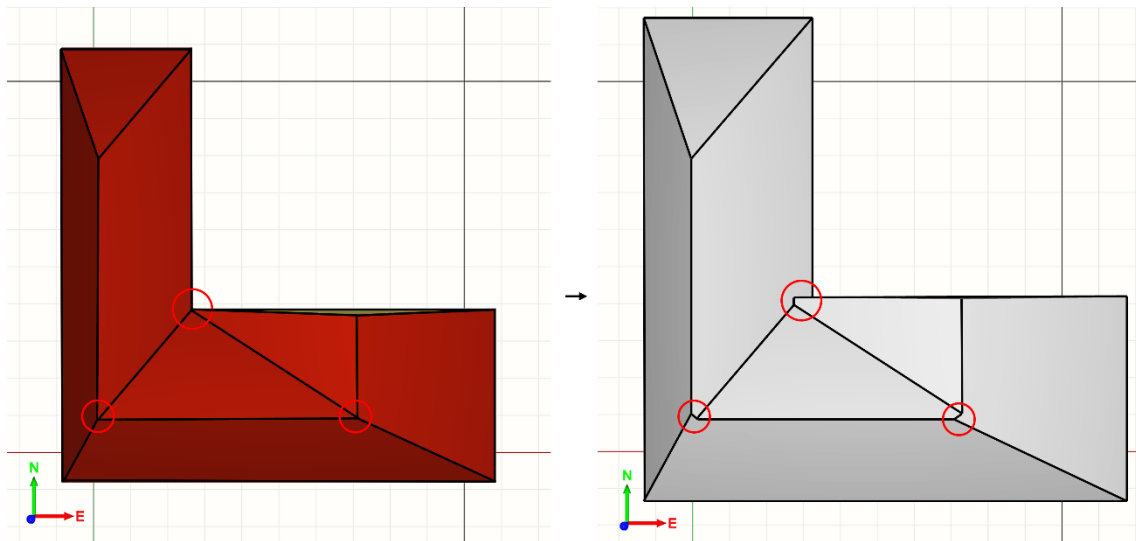


Abbildung 3.13: Veränderung der Ursprungsgeometrie infolge Extrusion

Die Funktionen `healTwistedPolygon` und `correctPolygonWithExtraPoints` können viele der durch diese Verschiebungen entstehenden Fehler korrigieren. In bestimmten Fällen bleiben Fehler jedoch unerkannt, etwa wenn die neu entstehende Fläche nicht verwunden ist und alle Punkte scheinbar korrespondierende Punkte in benachbarten Flächen besitzen. Ein entsprechendes Beispiel ist in Abbildung 3.14 dargestellt.

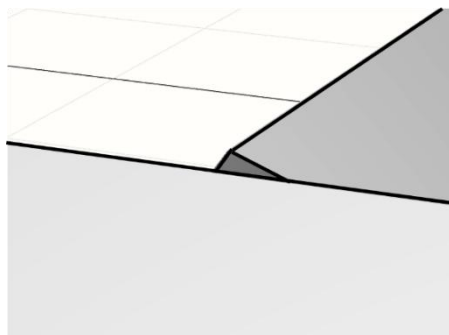


Abbildung 3.14: Fehlerhafte Geometrie durch unterschiedliche Dachschrägen

3.4.1 Erweiterung des Extrusionsalgorithmus

Um punktgenaue Nachbarflächen zu identifizieren, die nicht zur Bildung der Transformationspunkte benötigt werden, wird eine Iterationsstufe in die Funktion `createExtrudedPoints` integriert.

Die Funktion wird zunächst wie bisher einmal vollständig durchlaufen. Zu jedem neu erzeugten Punkt werden zusätzliche Metainformationen in neu eingeführten `newPointInfo`-Objekten gespeichert. Diese Objekte werden flächenweise in Matrizen `vecNewPointInfos` abgelegt, deren Indizes den jeweiligen

Ausgangsflächen entsprechen. Auf diese Weise kann die Punktentstehung nach dem ersten Extrusionsdurchlauf vollständig nachvollzogen und für einen zweiten Durchlauf gezielt ausgewertet werden.

Ein `newPointInfo`-Objekt enthält unter anderem folgende Informationen:

- Index des korrespondierenden `pointInfo`-Objektes der Ausgangsfläche
- ID der ersten Nachbarfläche
- Kennzeichnung, ob es sich um eine direkte Nachbarfläche handelt
- ID der zweiten Nachbarfläche
- Index des erzeugten Punktes in der Transformationsfläche
- Markierung zum Löschen
- Kennzeichnung, ob Punkt durch die Funktion `healTwistedPolygon` erzeugt oder verändert wurde

3.4.2 Markierung fehlerhafter Punkte

Nach dem ersten Durchlauf von `createExtrudedPoints` werden in der Funktion `healTwistedPolygon` verwundene und damit geometrisch fehlerhafte Flächen identifiziert. Wie bereits in Kapitel 2.5 beschrieben, werden dabei Punkte, die eine Verwindung einer Fläche verursachen, erkannt. Diese Punkte werden gelöscht oder durch einen neuen Schnittpunkt ersetzt. Zusätzlich werden Punktduplikate und kollineare Punkte entfernt, da diese für die Bildung der Transformationsfläche nicht erforderlich sind

Alle durch diese Prozesse entfernten oder ersetzten Punkte werden in den zugehörigen `newPointInfo`-Objekten markiert. Punkte, deren erste Nachbarfläche ein direkter Nachbar der Ausgangsfläche ist, dürfen nicht markiert werden, da die direkten Nachbarflächen stets für die Bildung der Transformationsfläche erforderlich sind.

Da `healTwistedPolygon` immer nur zwei aufeinander folgende Kanten betrachtet, können Fälle mit mehr als zwei aufeinanderfolgenden fehlerhaften Punkten nicht abgefangen werden. Aus diesem Grund wird am Ende der Funktion überprüft, ob die resultierende Transformationsfläche weiterhin verwunden ist. Ist dies der Fall, werden alle `newPointInfo`-Objekte der Fläche, deren erste Nachbarfläche kein direkter Nachbar ist, zur Löschung markiert. Dadurch wird sichergestellt, dass im zweiten Extrusionsdurchlauf ausschließlich direkte Nachbarflächen berücksichtigt verwendet.

3.4.3 Gegenläufige Markierung und zweiter Extrusionsdurchlauf

Da geometrische Fehler in Körpern häufig paarweise in korrespondierenden Flächen auftreten, muss die Markierung stets auch in der zugehörigen Nachbarfläche erfolgen. Ein zweiter Extrusionsdurchlauf wird jedoch nur dann durchgeführt, wenn in mindestens einer Fläche markierte `newPointInfo`-Objekte existieren. Andernfalls führt ein erneuter Durchlauf zu keiner Veränderung des Transformationskörpers und ist nicht erforderlich.

Vor dem zweiten Durchlauf wird die Funktion `markCorrespondingPoints` ausgeführt. In dieser Funktion werden zu bereits markierten `newPointInfo`-Objekten die entsprechenden Objekte in den korrespondierenden Flächen identifiziert und ebenfalls zur Löschung markiert. Der Abgleich erfolgt über die Flächen-IDs und die gespeicherten Nachbarschaftsinformationen.

Für den zweiten Durchlauf der Extrusion werden anschließend neue `pointInfo`-Objekte aus den unmarkierten `newPointInfo`-Objekten erzeugt. Dabei ist stets die erste Nachbarfläche ausschlaggebend.

Der neue schematische Ablauf des Algorithmus ist in Abbildung 3.15 dargestellt. Die beschriebenen Anpassungen des Algorithmus (einschließlich der in Kapitel 3.3 erläuterten Erweiterungen der Funktion `createExtrudedPoints`) liefern bereits so robuste Ergebnisse, dass die Funktion `correctPolygonWithExtraPoints` nicht mehr erforderlich ist.

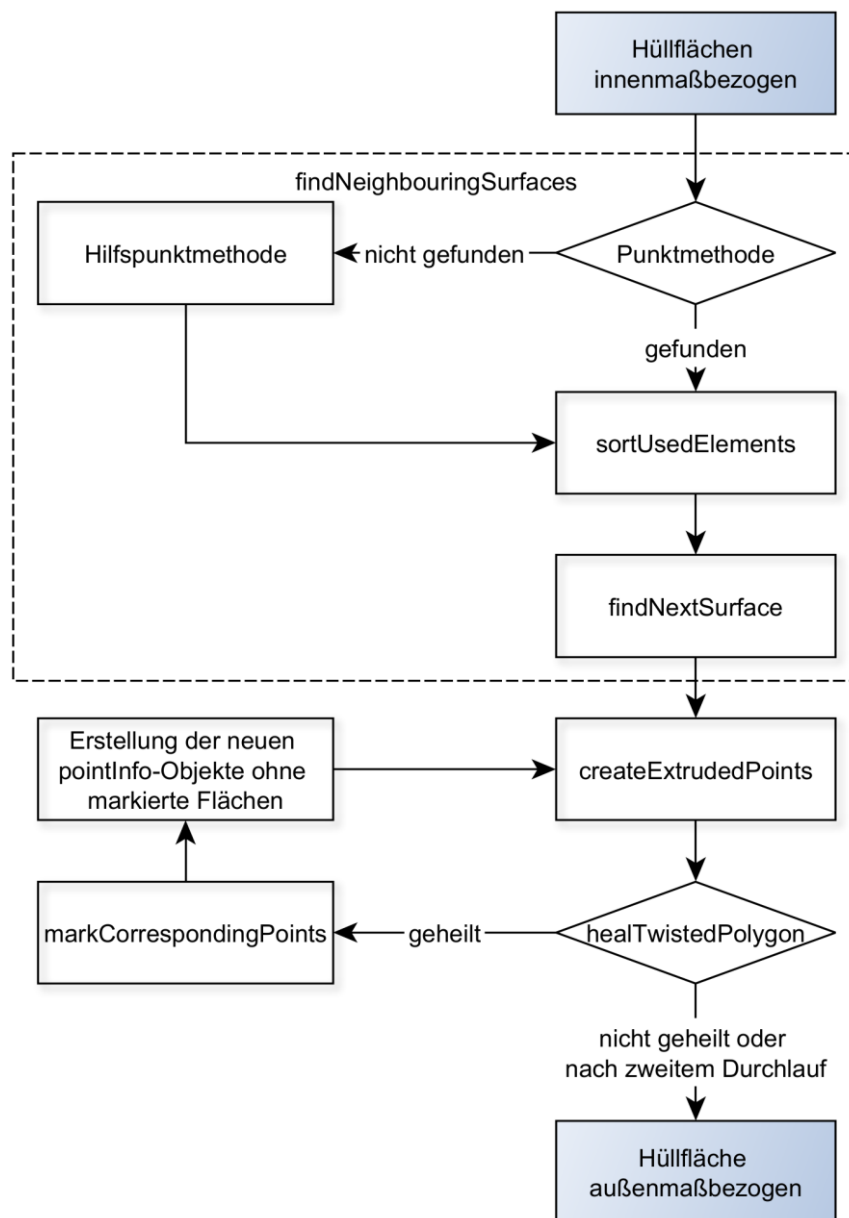


Abbildung 3.15: neuer Ablauf des Algorithmus

4 Visuelle Validierung

Eine visuelle Validierung des Algorithmus wird anhand von verschiedenen Testfällen durchgeführt. Ein großer Teil davon erfolgt an Einraummodellen, welche im Innenmaßbezug in VICUS erstellt werden. Anschließend werden auch einzelne Mehrraummodelle getestet. Insgesamt werden 48 Testgeometrien angelegt und mithilfe des Algorithmus zum Außenmaßbezug umgerechnet.

Für die Testmodelle wurden die folgenden, in Tabelle 1 aufgelisteten Konstruktionen verwendet. Die verschiedenen Konstruktionen besitzen unterschiedliche Bauteildicken. Die Räume erhalten Nutzungsprofile, wobei vorerst allen Räumen ein Heizprofil zugeordnet wird.

Bauteil	Material	Dicke [cm]
Außenwand		34
	Gipsputz	2
	Mauerwerk	30
	Kalkputz	2
Innenwand		18
	Gipsputz	1,5
	Mauerwerk	15
	Gipsputz	1,5
Bodenplatte		25
	Beton	15
	Dämmung	6
	Estrich	4
Geschossdecke		24,3
	Gipskarton	1,25
	Holzbalken mit Isolierung	20
	Hartholz	3

Dach		15,3
	Gipskarton	1,25
	Dachsparren mit Isolierung	14

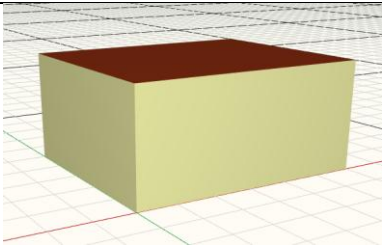
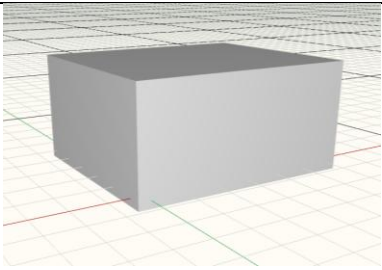
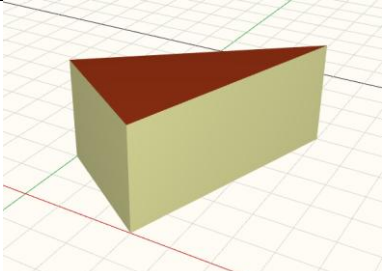
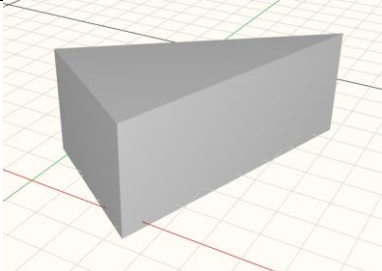
Tabelle 1: Bauteilkonstruktionen

4.1 Einraummodelle

Bartels Algorithmus erzielte bereits für einfache Einraummodelle, wie Quader, Prismen und einfache Dachformen gute Ergebnisse. Diese werden zur Vollständigkeit erneut getestet und aufgeführt. Weiterhin werden Raummodelle mit parallelen Flächen getestet. Der Fokus liegt auf der Betrachtung von Modellen typischer und komplexer Dachformen und Gauben.

4.1.1 Einfache Raummodelle

Zuerst werden einfache Räume mit paralleler Boden- und Deckenfläche modelliert und getestet. Diese Körper waren bereits vor den Anpassungen des Algorithmus umsetzbar. Die in den folgenden Kapiteln abgebildeten Tabellen nennen den Namen des Ausgangskörpers und zeigen Abbildungen von dem in VICUS modellierten Raum im Innenmaßbezug und den Transformationskörper im Außenmaßbezug als VICUS-Modell.

Name	Raummodell im Innenmaßbezug	Transformationsmodell im Außenmaßbezug
Quader		
Raum mit dreieckiger Grundfläche		

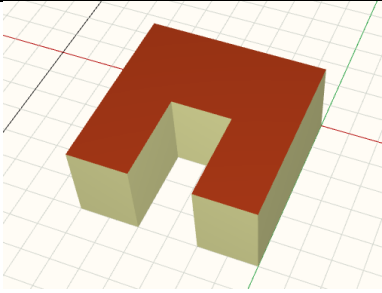
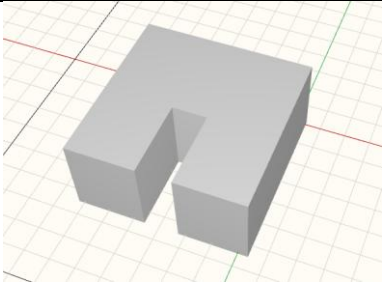
U-förmiger Raum		
-----------------	---	--

Tabelle 2: Transformation einfacher Test-Raummodelle

4.1.2 Modelle mit parallelen Flächen

Die Transformation von Räumen mit parallelen Flächen hat im ursprünglichen Algorithmus nur für maximal zwei parallele Flächen funktioniert. Durch die Überarbeitung der Hilfspunktmethode und der Sortierfunktion `sortUsedElements` (siehe Kapitel 3.2), ist nun auch die Transformation von Körpern mit mehreren parallelen Flächen möglich.

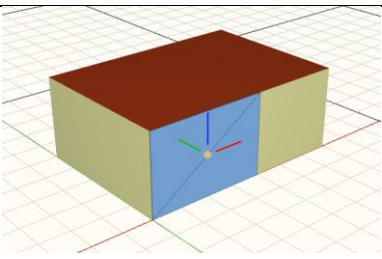
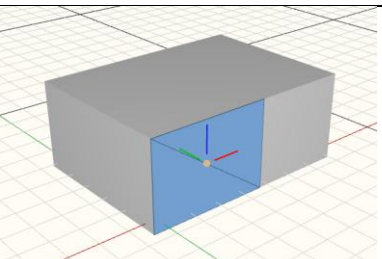
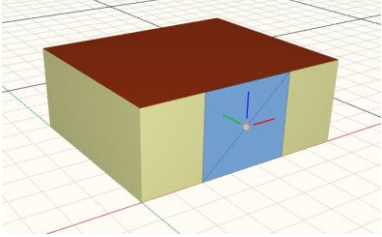
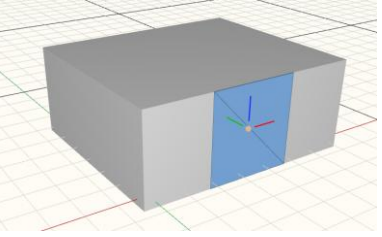
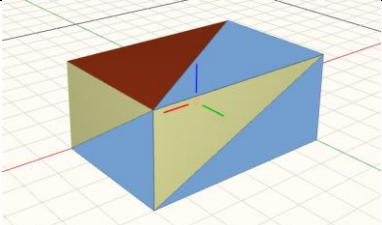
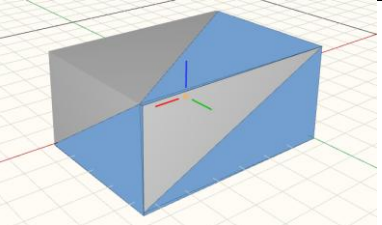
Name	Raummodell im Innenmaßbezug	Transformationsmodell im Außenmaßbezug
Quader mit zwei parallelen Flächen		
Quader mit drei parallelen Flächen		
Quader trianguliert		

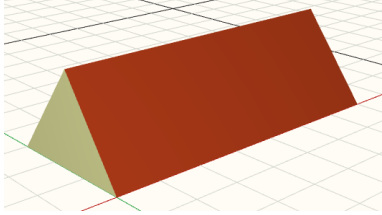
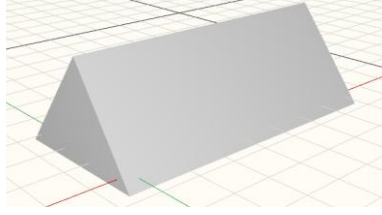
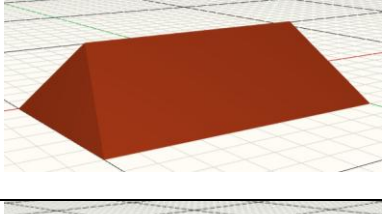
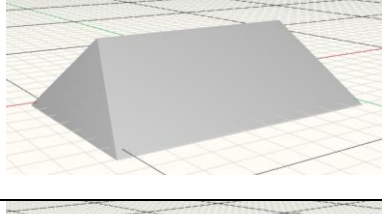
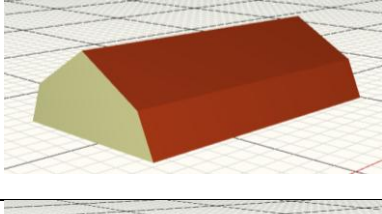
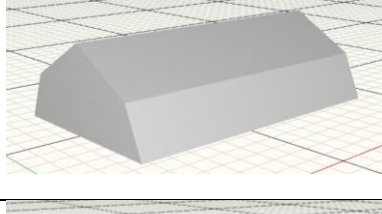
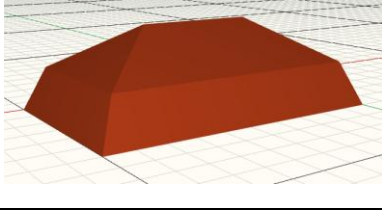
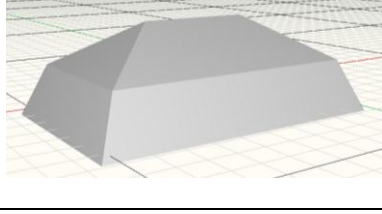
Tabelle 3: Transformation von Räumen mit parallelen Flächen

4.1.3 Typische Dachformen

Für Dächer wird das größte Verbesserungspotential durch die Anpassungen des Algorithmus vermutet, da viele der nachfolgend aufgeführten Fälle zuvor noch nicht betrachtet wurden und größtenteils nicht umsetzbar waren.

Tabelle 4 zeigt Abbildungen von einigen der durchgeführten Tests an Modellen verschiedener typischer Dachformen. Weitere Testfälle sind im Anhang abgebildet.

Von den abgebildeten Testfällen war nur das Kreuzdach vor den Änderungen des Algorithmus nicht umwandelbar. Die Transformation des Kreuzdachs ist erst durch das Ersetzen der Punktgleichheitsmethode mit der Funktion `findNextSurface` (siehe Kapitel 3.2.3) und die Ergänzung einer Iterationsstufe in den Extrusionsprozess möglich (siehe Kapitel 3.4).

Name	Raummodell Innenmaßbezug	im Transformationsmodell im Außenmaßbezug
Satteldach		
Walmdach		
Mansarddach		
Mansard-Walmdach		

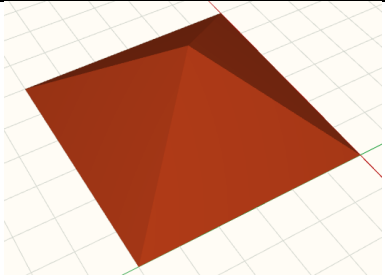
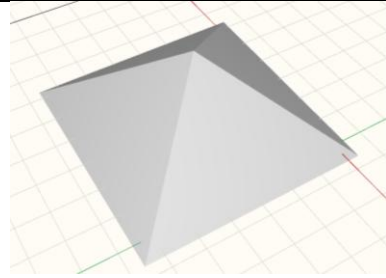
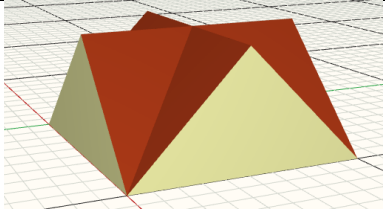
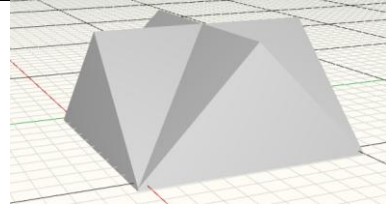
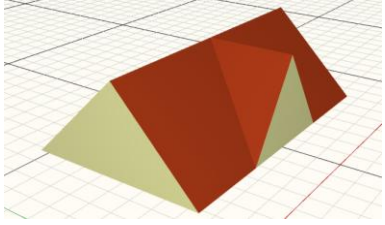
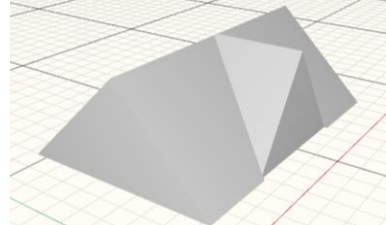
Zeltdach		
Kreuzdach		

Tabelle 4: Transformation von Modellen typischer Dachformen

4.1.4 Zwerchgauben

Zwerchgauben sind Gauben, die direkt am Übergang zwischen Decke und Dachkonstruktion beginnen. Sie können bis zum Dachfirst reichen aber auch niedriger sein. Als Grundkonstruktion für die Modelle wird ein Satteldach gewählt, in welches die Zwerchgauben integriert werden. Tabelle 5 zeigt ausgewählte Tests an Modellen mit Zwerchgauben. Weitere sind im Anhang dargestellt.

Vor der Erweiterung des Algorithmus konnten Zwerchgauben nicht sinnvoll umgerechnet werden. Durch die Erweiterung der Funktion createExtrudedPoints um einen Fall für linear abhängige, nicht parallele Flächen (siehe Kapitel 3.3) und die Einführung einer Iterationsstufe in den Extrusionsprozess (siehe Kapitel 3.4) ist die Transformation möglich.

Name	Raummodell im Innenmaßbezug	Transformationsmodell im Außenmaßbezug
Zwerchdach groß		

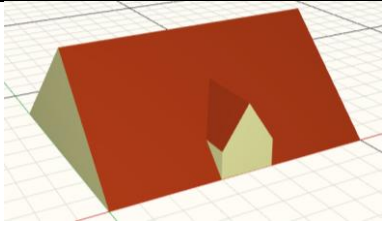
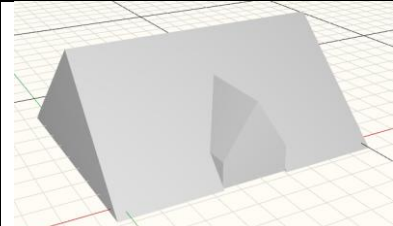
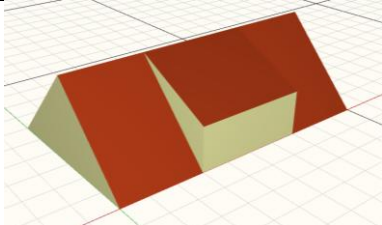
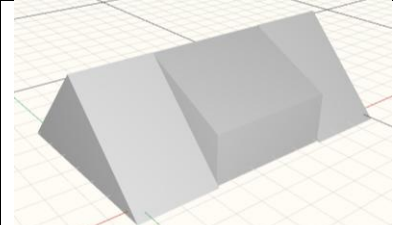
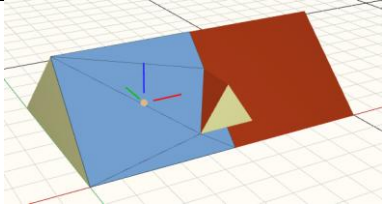
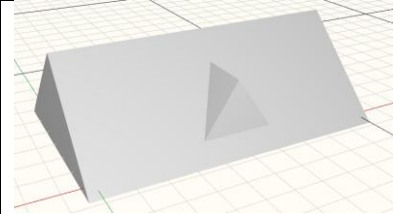
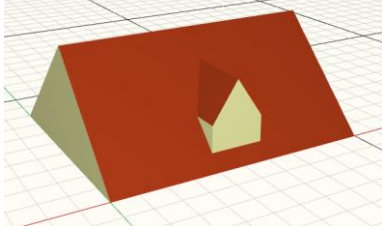
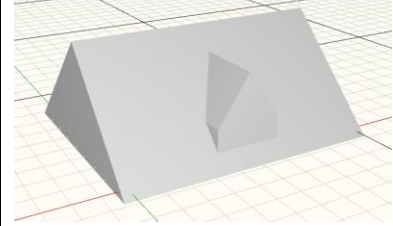
Zwerchgaube klein		
Schlepp-Zwerchgaube groß		

Tabelle 5: Transformation von Dächern mit Zwerchgauben

4.1.5 Dachgauben

Es werden Modelle von verschiedenen typischen Gaubenarten getestet, von denen einige in Tabelle 6 und weitere im Anhang abgebildet werden. Das Grunddach ist dabei immer ein Satteldach. Da der Algorithmus noch nicht in der Lage ist, Löcher in Ausgangsflächen zu berücksichtigen, muss die Dachfläche, in die die Gaube eingefügt wird, in mindestens zwei Flächen zerteilt werden.

Durch die Zerlegung der Dachfläche kann die Transformation dieser Modelle nach Anpassung der Hilfspunktmethode, der Funktion `sortUsedElements` und der Änderung in der Berechnung von Transformationspunkten paralleler Flächen korrekt durchgeführt werden.

Name	Raummodell im Innenmaßbezug	Transformationsmodell im Außenmaßbezug
Spitzgaube		
Giebelgaube		

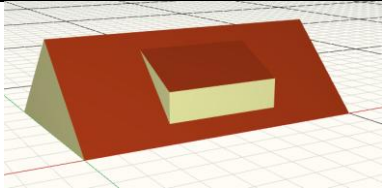
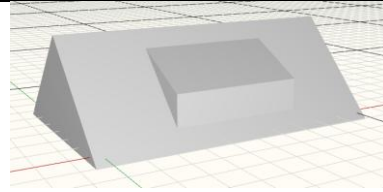
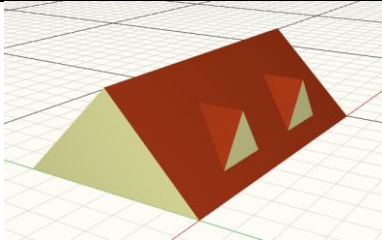
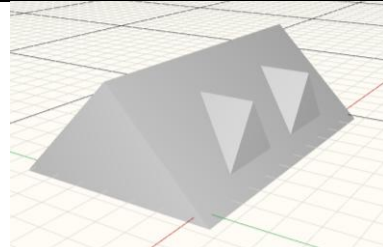
Schleppgaube		
Zwei Spitzgauben		

Tabelle 6: Transformation von Dächern mit Dachgauben

4.1.6 Weitere und kombinierte Dachformen

Zuletzt werden einige komplexe und kombinierte Dachmodelle getestet. Eins davon stellt das L-Dach mit einer großen Zwerchgaube dar, ein anderes ein Verwinkeltes Walmdach. Tabelle 7 zeigt Abbildungen der Testfälle.

Die Transformation dieser Modelle war zu Beginn des Arbeitsprozesses sehr fehlerhaft bzw. nicht möglich. Durch die getroffenen Veränderungen am Algorithmus ist sie nun möglich.

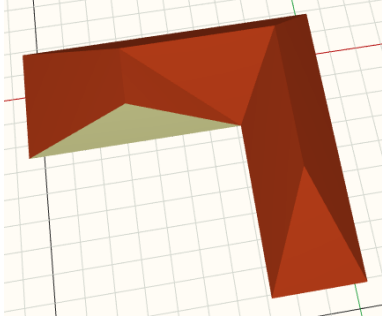
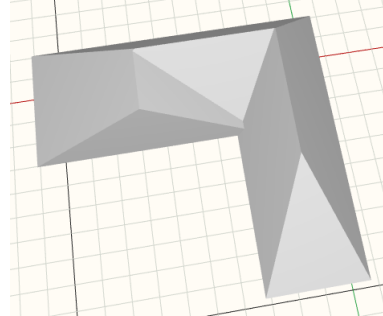
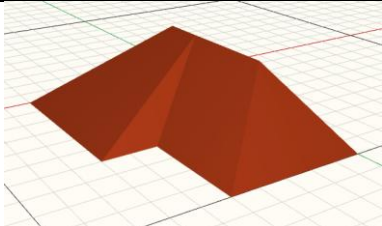
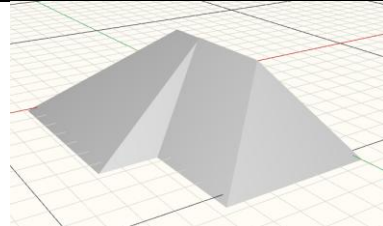
Name	Raummodell im Innenmaßbezug	Transformationsmodell im Außenmaßbezug
L-Dach mit Zwerchgaube		
Vieleckenwalm		

Tabelle 7: Transformation komplexer Dachformen

4.2 Mehrraummodelle

Zuletzt werden Mehrraummodelle getestet. Da der Fokus der Arbeit auf der verbesserten Darstellung von komplexen Einraummodellen liegt, werden nur einfache Tests durchgeführt. Neben einer Abbildung des Ausgangsmodells und des Transformationskörpers wird in Tabelle 8 eine weitere Abbildung mit einer kombinierten Ansicht eingeführt. Um komplexere Mehrraummodelle richtig transformieren zu können, müssen noch weitere Anpassungen am Algorithmus getätigt werden. Die bestehenden Probleme werden in Kapitel 4.3 erläutert.

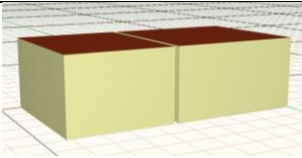
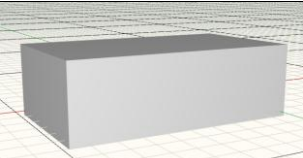
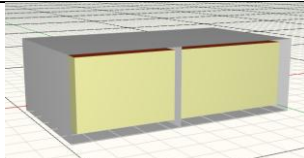
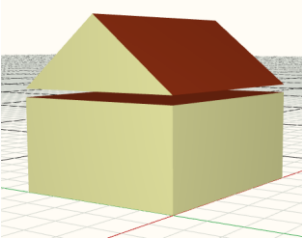
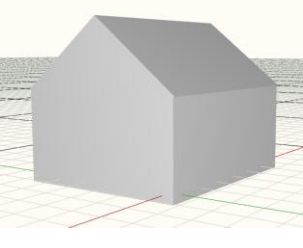
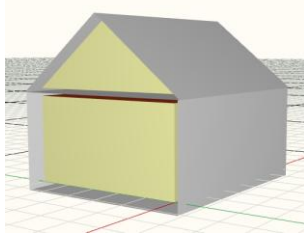
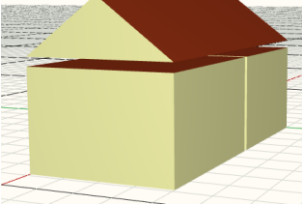
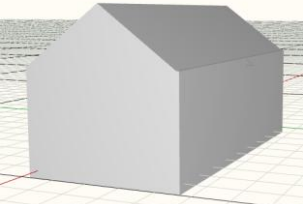
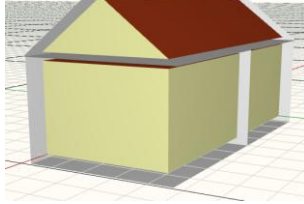
Name	Raummodell im Innenmaßbezug	Transformationsmodell im Außenmaßbezug	Kombinierte Ansicht Innen- und Außenmaßbezug
2 Räume			
Raum mit Dach			
2 Räume mit Dach			

Tabelle 8: Transformation von Mehrraummodellen

4.3 Einschränkungen und verbleibende Fehler

In diesem Abschnitt wird betrachtet, welche Vereinfachungen vorgenommen werden mussten, damit die Transformation mancher Testmodelle funktioniert, welche Testfälle noch nicht funktionieren und in welchen Bereichen des Algorithmus weiterhin Verbesserungspotential besteht.

4.3.1 Eingangsdaten

Zur Durchführung der Transformation ist ein fehlerfreies Eingabemodell notwendig. Da die Verschiebungsrichtung der Flächen in Richtung der Normalenvektoren erfolgt, dürfen diese nicht in den Raum hineinragen. Außerdem darf keine Umfassungsfläche des Raumes fehlen, weil sonst nicht alle Nachbarflächen korrekt ermittelt werden können, die für den späteren Verschnitt benötigt werden.

4.3.2 Löcher im Ausgangskörper

Bisher kann der Algorithmus keine Löcher in Flächen der Ausgangskonstruktion berücksichtigen. Dieser Fall tritt zum Beispiel bei Dachgauben oder Fahrstuhlschächten auf. Durch eine Teilung der Fläche, in der sich das Loch befindet, kann die Transformation trotzdem durchgeführt werden. Das führt, wie in Kapitel 4.1 gezeigt, bereits zu richtigen Ergebnissen.

Um Löcher in der Transformation berücksichtigen zu können, müssten in den Algorithmus Fälle zur Berücksichtigung von Löchern eingeführt werden. Eine Möglichkeit wäre, den Körper an dem Loch in zwei Stücke zu zerlegen. Die Fläche, in der sich das Loch befindet, besitzt daraufhin kein Loch mehr. In das abgetrennte Teilstück wird anstelle des Loches eine Hilfsfläche eingeführt, welche die selbe Bauteildicke wie die Fläche, in der sich das Loch befindet, besitzt. Die beiden Teilkörper können daraufhin separat extrudiert werden. Nach der Transformation der Raumteile muss eine Wiedervereinigung stattfinden. Das könnte über Boolean-Operatoren, logischen Operatoren zur Bildung mengentheoretischer Verknüpfungen, durchgeführt werden. Diese Option stellt jedoch eine relativ große Veränderung des bestehenden Algorithmus dar.

Eine weitere Möglichkeit wäre, die Betrachtung von Löchern in die bestehenden Funktionen zu integrieren. Dafür müssen zuerst die Nachbarflächen des Loches in der Funktion `findNeighbouringSurfaces` gefunden werden. Für den späteren Verschnitt wird sowohl die Fläche benötigt, in der das Loch sich befindet, als auch die Flächen, welche an das Loch angrenzen. Gegebenenfalls wird dafür eine weitere Datenstruktur benötigt. Bei der Verschiebung und dem Verschnitt der Ebenen muss beachtet werden, dass die Lochfläche um die Dicke der Fläche, in der sie liegt verschoben werden muss und das Loch je nach Lage der angrenzenden Flächen kleiner oder größer werden kann. Wie in Abbildung 4.1 dargestellt, verkleinert sich das Loch infolge der Transformation, wenn eine Durchdringung des Ausgangskörpers stattfindet. Schließt an das Loch ein Körper an, wie es bei Dachgauben der Fall ist, wird das Loch durch die Transformation größer.

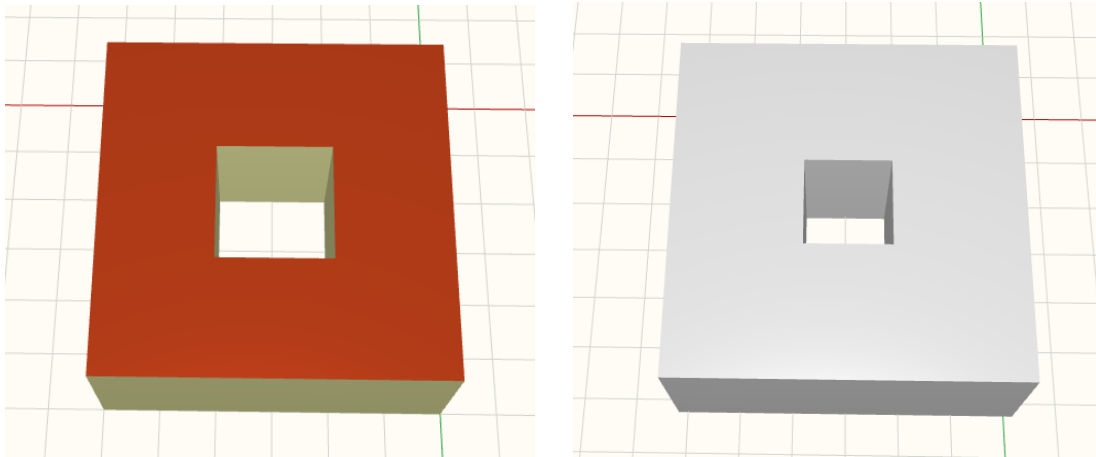


Abbildung 4.1: Verkleinerung eines Loches infolge der Transformation

4.3.3 Umsetzung der Normen und Mehrraummodelle

Die Berechnung der Verschiebungslängen der Bauteile erfolgt bisher auf Grundlage der in Kapitel 2.1 beschriebenen Bezugsbauteilmaße der DIN V 18599. Für die Umsetzung der Maßbezüge nach VDI 6020 und VDI 2078 wurden in Kapitel 2.3 die theoretischen Grundlagen geschaffen. Eine Umsetzung im Algorithmus wurde jedoch nicht durchgeführt.

Die Außenbauteile werden, gemäß den Maßbezügen der DIN V 18599 um die gesamte Bauteildicke verschoben, die Innenbauteile zwischen beheizten Räumen um die halbe Bauteildicke. Nicht berücksichtigt werden bisher jedoch die anderen Maßbezüge bei Geschossdecken, welche bisher ebenfalls um die halbe Bauteildicke verschoben werden. Außerdem werden adiabate Bauteile bisher wie Außenbauteile um die gesamte Bauteildicke verschoben. Die Fläche des adiabaten Bauteils sollte allerdings, wenn es sich um ein Innenbauteil handelt, entsprechend der umliegenden Bauteile verschoben werden. Abbildung 4.2 zeigt einen solchen Fall. Der Boden des Daches ist in drei Teile zerlegt, wobei die mittlere, hellblau markierte Teilfläche adiat ist, da sie auf die Wand zwischen den zwei darunter liegenden Räumen trifft. Da das adiabate Stück um die gesamte Bauteildicke, die anderen beiden Teile jedoch nur um die halbe Bauteildicke verschoben wurden, entsteht eine herausstehende Ecke im Transformationskörper des Daches.

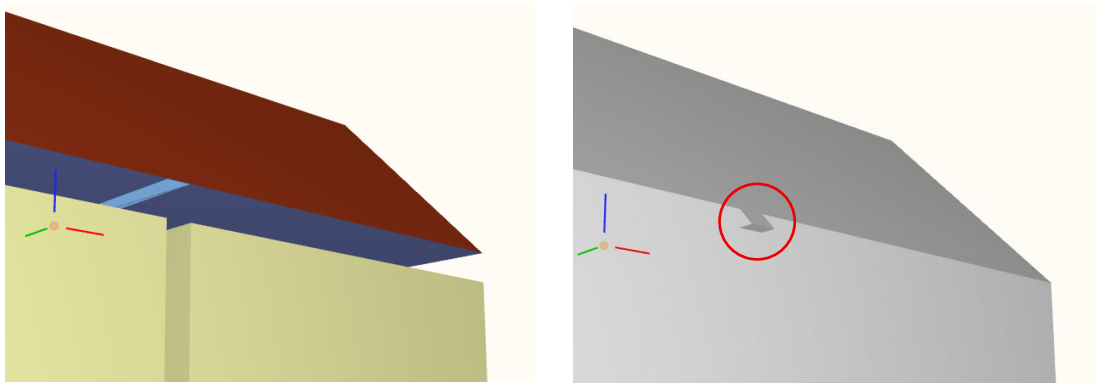


Abbildung 4.2: Fehler in Transformation adiabater Bauteile

4.3.4 Lücken in Transformationskörpern und Versatz von Firstlinien

Durch unterschiedliche Bauteildicken von parallelen Flächen können Lücken in dem Transformationskörper entstehen. Auch bei der Transformation der Testmodelle mit Zwerchgauben entstehen, wie in Abbildung 4.3 gezeigt, Lücken im Transformationskörper. Eine Funktion zum schließen der Lücken in den Transformationskörpern wurde in Rahmen dieser Arbeit nicht in den Algorithmus eingefügt. In den meisten Fällen besitzen die fehlenden Flächen des Transformationskörpers nur eine sehr geringe Fläche. Hier kann die Option diskutiert werden, diese Flächen für baurechtlichen Nachweise zu übermessen und daher zu vernachlässigen. Damit die Körper trotzdem ohne Lücken dargestellt werden, sollte in der Zukunft eine entsprechende Ergänzung im Algorithmus vorgenommen werden. Fraglich dabei ist jedoch, wie die Zuordnung zu einem bestimmten Bauteilaufbau aussehen könnte.

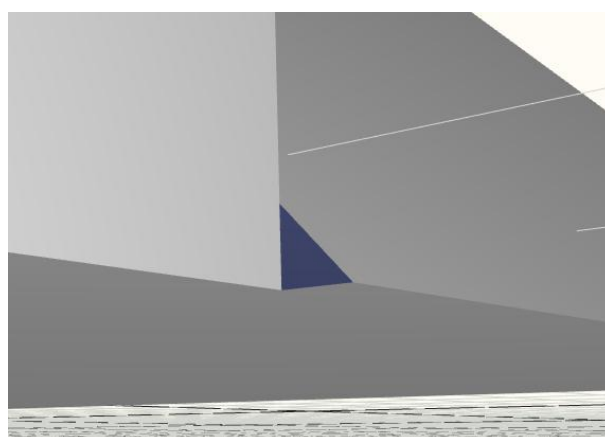


Abbildung 4.3: Lücke in Transformationskörper bei Zwerchgauben

Bei der Transformation der großen Zwerch-Schleppgaube kommt es außerdem aufgrund der unterschiedlichen Dachschrägen zu einem Versatz in der Firstlinie. Abbildung 2.1 zeigt ein Detail dieses Falles. Laut dem Algorithmus ist das korrekt, allerdings nicht realitätsnah. Der Versatz entsteht, weil das Modell im Innenmaßbezug erstellt wurde und die verschiedenen Dachschrägen bei der Erstellung nicht berücksichtigt wurden. Um für diesen Fall ein realistischeres Ergebnis zu erzielen, müssen schon bei der Erstellung des Modells im Innenmaßbezug, die Winkel zwischen den Dachflächen berücksichtigt werden.

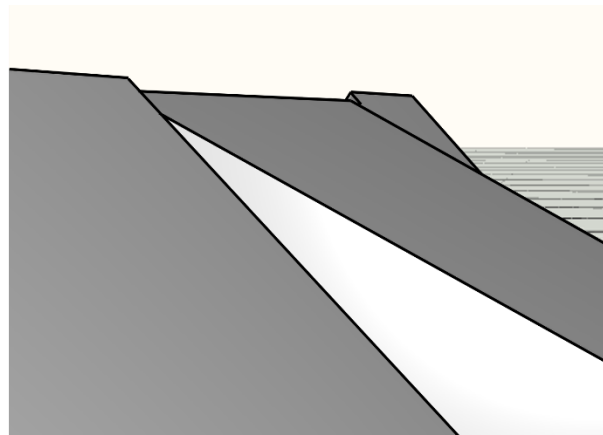


Abbildung 4.4: Versatz in der Firstlinie bei großer Schleppgaube

5 Schnittstelle zu baurechtlichen Nachweisen

Für Bauvorhaben sind in der Regel verschiedene baurechtliche bzw. bautechnische Nachweise durchzuführen. So sind laut § 66 der Sächsischen Bauordnung unter anderem „die Einhaltung der Anforderungen an die Standsicherheit, den Brand-, Schall-, und Erschütterungsschutz [...] nachzuweisen“ [20]. Ein weiterer obligatorischer Nachweis, der für jeden Neubau, aber auch für größere Sanierungsprojekte und Umnutzungen geführt werden muss, ist der GEG-Nachweis.

Der GEG-Nachweis beinhaltet, wie bereits in Kapitel 2.1 erläutert, die Vorschrift, dass für ein neu zu errichtendes Wohn- oder Nichtwohngebäude und auch nach Änderungen an den Außenbauteilen eines Bestandsgebäudes der Jahresprimärenergiebedarf nach DIN V 18599: 2018-09 zu ermitteln ist [2].

Die DIN/TS 18599-1 definiert in Kapitel 4.3 Ausgangsgrößen für die Berechnungen in den anderen Teilen Normenreihe [9, S. 27]. Darunter befinden sich unter anderem geometrische Größen, welche aus dem Simulationsmodell und dem Transformationsmodell extrahiert werden können. Folgende Werte können aus dem Innenmaßbezogenen Modell ermittelt werden:

- die **Nettogrundfläche** A_{NGF} als nutzbare Fläche der konditionierten Räume kann über die die Grundfläche der Räume, denen eine Nutzungszone mit Heizlastprofil zugewiesen ist ermittelt werden
- die **lichte Raumhöhe** h_R als Höhendifferenz zwischen der Oberkante des Fußbodens und der Unterkante der Geschosdecke kann über die Höhe der innenmaßbezogenen Räume ermittelt werden
- das **Nettoraumvolumen** V als Volumen der konditionierten Flächen kann über das Produkt der Nettogrundfläche und der lichten Raumhöhe berechnet werden
- die Anzahl der beheizten Geschosse n_G

Über das Transformationsmodell können die **wärmeübertragenden Umfassungsflächen** A ermittelt werden, welche nach den in Kapitel 2.1 beschriebenen Bezugsmaßen berechnet werden. Diese Flächen stellen die Grenze zwischen den konditionierten Räumen und der Außenluft dar und sind dementsprechend Außenwände, Dachflächen, Bauteile gegen das Erdreich und Innenwände zwischen unterschiedlich beheizten Zonen.

Damit die Maße der Umfassungsflächen für weitere Berechnungen nutzbar bleiben, sollten ergänzende Informationen zu ihnen ermittelt und gespeichert werden. Denkbar wäre die Ausgabe einer Tabelle, welche neben den Maßen des Bauteils

außerdem Informationen zur Lage im Gebäude (Geschoss, Raum und Ausrichtung) und der Bauteilart beinhaltet (Konstruktion, Dicke, U-Wert). Zusätzlich kann eine zusammenfassende Tabelle der verwendeten Konstruktionen mit Informationen zum Aufbau, den Materialparametern und der aufaddierten Fläche ausgegeben werden.

Dafür muss jedoch eine eindeutige Zuordnung der Transformationsfläche zur Ausgangsfläche gewährleistet sein.

6 Schlussbetrachtung

6.1 Zusammenfassung

In der vorliegenden Arbeit wurde ein Algorithmus für die Transformation innenmaßbezogener Raummodelle zum Außenmaßbezug entwickelt. Hierfür wurden zunächst die bauteilbezogenen Bezugsmaße relevanter Normen definiert und bestehende Lösungsansätze aus früheren Arbeiten analysiert. Als Ausgangspunkt diente ein vorhandener Algorithmus, der die Transformation einfacher Modelle gemäß den Vorgaben des GEG bzw. der DIN V 18599 bereits abbilden konnte.

Durch die gezielte Erweiterung dieses Algorithmus konnten zentrale Einschränkungen bei der Verarbeitung komplexer Raummodelle überwunden werden. Insbesondere wurden neue Berechnungsfälle zur Behandlung paralleler sowie linear abhängiger Flächen geschaffen und der Extrusionsprozess um eine Iterationsstufe erweitert, die eine robustere Identifikation und Korrektur fehlerhafter Flächen ermöglicht. Die entwickelten Anpassungen wurden anhand zahlreicher Testmodelle überprüft und führen zu geometrisch konsistenten Transformationskörpern auch bei komplexen Raum- und Dachgeometrien.

Darüber hinaus wurde exemplarisch untersucht, wie die für baurechtliche Nachweise erforderlichen Informationen extrahiert werden können. Am Beispiel des GEG-Nachweises zeigt sich, dass der entwickelte Algorithmus einen wichtigen Beitrag zur Annäherung von thermischer Gebäudesimulation und normbasierter Nachweisführung leisten kann.

6.2 Ausblick

Trotz der erzielten Verbesserungen bestehen weiterhin Grenzen des Algorithmus. Zentrale Ansatzpunkte liegen in der konsistenten Integration unterschiedlicher Normen sowie in der Überarbeitung der Berechnung von Bauteildicken für verschiedene Bauteilarten. Diese Aspekte sind insbesondere für die normkonforme Transformation von komplexeren Mehrraummodellen von entscheidender Bedeutung. Ergänzend sollten Mechanismen zur Prüfung und Validierung der Eingabedaten implementiert werden, um die Anwenderfreundlichkeit und Robustheit des Verfahrens weiter zu erhöhen. Ein weiterer wesentlicher Entwicklungsschritt besteht in der Ausgestaltung einer geeigneten Datenexportstruktur, wofür vertiefende Untersuchungen zu normabhängigen Anwendungsfällen und eingesetzten Softwarelösungen erforderlich sind.

Vor dem Hintergrund der Energiewende und dem hohen Sanierungsbedarfs leistet die vorliegende Arbeit einen Beitrag zur Weiterentwicklung automatisierter, konsistenter und normgerechter Planungsprozesse. Sie zeigt, dass eine automatisierte und normgerechte Transformation innenmaßbezogener Raummodelle auch für komplexe Geometrien möglich ist. Damit wird eine belastbare Grundlage für weiterführende Entwicklungen einer Schnittstelle zwischen Gebäudesimulation und baurechtlichen Nachweisen geschaffen.

Literaturverzeichnis

- [1] „Energieverbrauch privater Haushalte“, Umweltbundesamt. Zugriffen: 6. Februar 2026. [Online]. Verfügbar unter: <https://www.umweltbundesamt.de/daten/private-haushalte-konsum/wohnen/energieverbrauch-privater-haushalte>
- [2] *Gesetz zur Einsparung von Energie und zur Nutzung erneuerbarer Energien zur Wärme- und Kälteerzeugung in Gebäuden - Gebäudeenergiegesetz (GEG)*. 2020. Zugriffen: 5. Januar 2026. [Online]. Verfügbar unter: <https://www.gesetze-im-internet.de/geg/GEG.pdf>
- [3] „Sanierungsquote - BuVEG“, Bundesverbands energieeffiziente Gebäudehülle e.V. Zugriffen: 9. Februar 2026. [Online]. Verfügbar unter: <https://buveg.de/sanierungsquote/>
- [4] *VDI 2078:2015-06, Berechnung der thermischen Lasten und Raumtemperaturen (Auslegung Kühllast und Jahressimulation)*, Düsseldorf., 2015. Zugriffen: 5. Januar 2026. [Online]. Verfügbar unter: <https://www.dinmedia.de/de/technische-regel/vdi-2078/197958916>
- [5] *VICUS | Software für die Gebäude- und Wärmenetzplanung*. (20. Juni 2024). Zugriffen: 3. November 2025. [Online]. Verfügbar unter: <https://vicus-software.com/>
- [6] *EnergyPlus*. Zugriffen: 16. Februar 2026. [Online]. Verfügbar unter: <https://energyplus.net/downloads>
- [7] „BMWSB Gebäudeenergiegesetz - BMWSB“. Zugriffen: 5. Januar 2026. [Online]. Verfügbar unter: https://www.bmwsb.bund.de/DE/bauen/innovation-klimaschutz/gebaeudeenergiegesetz/gebaeudeenergiegesetz_node.html
- [8] *DIN EN ISO 13789:2018-04, Wärmetechnisches Verhalten von Gebäuden_- Transmissions- und Lüftungswärmetransferkoeffizient_- Berechnungsverfahren (ISO_13789:2017); Deutsche Fassung EN_ISO_13789:2017*. doi: 10.31030/2560190.
- [9] *DIN/TS 18599-1:2025-10, Energetische Bewertung von Gebäuden_- Berechnung des Nutz-, End- und Primärenergiebedarfs für Heizung, Kühlung, Lüftung, Trinkwarmwasser und Beleuchtung_- Teil_1: Allgemeine Bilanzierungsverfahren, Begriffe, Zonierung und Bewertung der Energieträger*. doi: 10.31030/3605938.
- [10] *DIN EN 12831-1:2017-09, Energetische Bewertung von Gebäuden_- Verfahren zur Berechnung der Norm-Heizlast_- Teil_1: Raumheizlast, Modul M3-3; Deutsche Fassung EN_12831-1:2017*. doi: 10.31030/2571775.
- [11] *DIN EN 12831-1:2025-06, Energetische Bewertung von Gebäuden_- Verfahren zur Berechnung der Norm-Heizlast_- Teil_1: Raumheizlast, Modul M3-3; Deutsche und Englische Fassung prEN_12831-1:2025*. doi: 10.31030/3613477.
- [12] *VDI 6020:2022-12, Anforderungen an thermisch-energetische Rechenverfahren zur Gebäude- und Anlagensimulation*, Düsseldorf, 2022. Zugriffen: 5. Januar 2026. [Online]. Verfügbar unter: <https://www.dinmedia.de/de/technische-regel/vdi-6020/352981659>
- [13] H. Fechner, „NANDRAD - Bauklimatik Dresden“. Zugriffen: 12. Februar 2026. [Online]. Verfügbar unter: <https://bauklimatik-dresden.de/download/nandrad/>

- [14] „Simulationswerkzeuge“, TU Dresden. Zugriffen: 12. Februar 2026. [Online]. Verfügbar unter: https://tu-dresden.de/bu/architektur-landschaft/ibk/varius/inhalte/verwendete-software?set_language=de
- [15] „VICUS BUILDINGS | Software für Nachhaltige Gebäudeplanung“. Zugriffen: 12. Februar 2026. [Online]. Verfügbar unter: <https://vicus-software.com/vicus-buildings/>
- [16] A. Junck, „Modellierungsansätze in der thermischen Gebäudesimulation: Transformation der Positionierung von IFC-Raumbegrenzungsflächen“, Energieeffizientes Bauen E3D, RWTH Aachen, Aachen, Bachelorarbeit, Apr. 2022.
- [17] T. Bartel, „Erarbeitung eines Algorithmus zur Transformation innenmaßbezogener 3D-Gebäudemodelle zu Außenmaßbezug“, Institut für Bauklimatik, TU Dresden, Dresden, Projektarbeit, März 2024.
- [18] „Lagebeziehungen zwischen 3 Ebenen - lernen mit Serlo!“, serlo.org. Zugriffen: 13. Januar 2026. [Online]. Verfügbar unter: <https://de.serlo.org/mathe/243174/lagebeziehungen-zwischen-3-ebenen>
- [19] M. Knorrenschild, *Mathematik für Ingenieure 1 - Grundlagen im Bachelorstudium*. München: Carl Hanser Verlag GmbH & Co. KG, 2009. Zugriffen: 20. Januar 2026. [Online]. Verfügbar unter: <https://www.hanser-elibrary.com/doi/book/10.3139/9783446420878>
- [20] *Sächsische Bauordnung - SächsBO*. 2026. Zugriffen: 12. Januar 2026. [Online]. Verfügbar unter: <https://www.revosax.sachsen.de/vorschrift/1779-SaechsBO#p66>

Anlagenverzeichnis

Anlage 1	Selbstständigkeitserklärung
Anlage 2	Kurzfassung der Diplomarbeit
Anlage 3	Abbildungen weiterer Testfälle

Erklärung zur selbständigen Erstellung der Arbeit

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus fremden Quellen, einschließlich des Internets, von digitalen Übersetzungshilfen oder sonstigen digitalen Quellen, direkt oder indirekt übernommenen Inhalte, Textpassagen und Gedanken sind ausnahmslos als solche kenntlich gemacht.

Ferner versichere ich, Argumente, Textpassagen, Programmcodes, bildliche und andere Darstellungen (Abbildungen, Diagramme, Übersichten, Tabellen usw.) sowie anderweitig inhaltsbasierte Ergebnisse, die durch generative „Künstliche Intelligenz“ (KI) oder mit Hilfe von auf generativer KI basierenden Programmen erstellt wurden, von mir ausnahmslos markiert sowie die Quellen (u. a. Programm, Datum des Zugriffs, Eingangsdaten) angegeben wurden. Eine Ausnahme stellen digitale Rechtschreib-, Grammatik- und Formulierungshilfen dar, solange diese lediglich die von mir vorgegebenen Texte redigieren und darüber hinaus keine neuen Inhalte generieren.

Die „Leitlinien zur guten wissenschaftlichen Praxis an der TU Dresden“ wurden von mir beachtet.

Ich versichere auch, dass die von mir eingereichte papierförmige Version mit der digitalen Version übereinstimmt und die vorliegende Arbeit weder im Inland noch im Ausland in gleicher oder ähnlicher Form einer anderen Prüfungsbehörde vorgelegt oder bereits veröffentlicht wurde.

Dresden, 18. Februar 2026

Kurzfassung der Diplomarbeit

Schlagworte: Gebäudesimulation

Geometrische Transformation

Verfasser: Lisa Sophie Ahnert

Titel der Diplomarbeit: Entwicklung von Transformationsalgorithmen
innenmaßbezogener 3D-Raummodelle zur Anwendung
in baurechtlichen Nachweisverfahren

Standort: TU Dresden

Fakultät Architektur und Landschaft

Institut für Bauklimatik

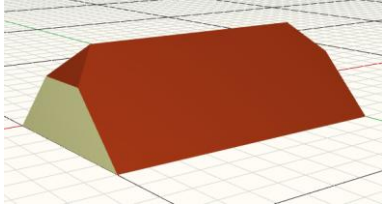
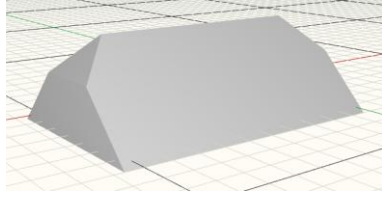
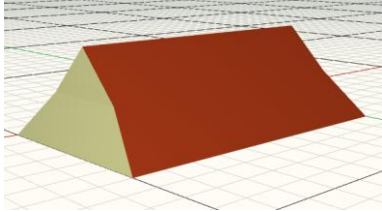
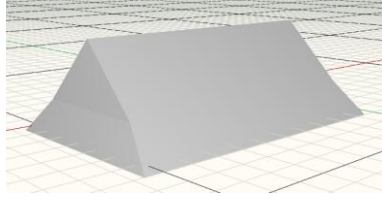
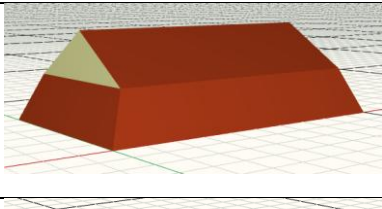
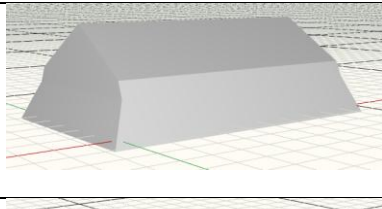
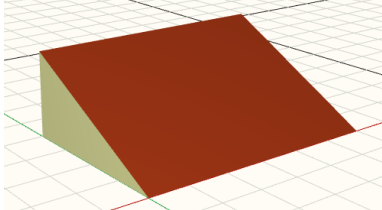
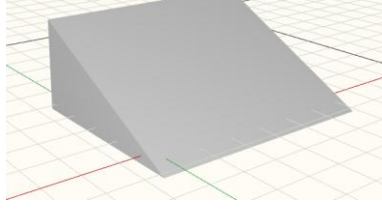
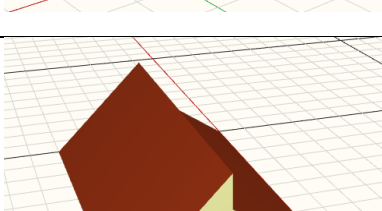
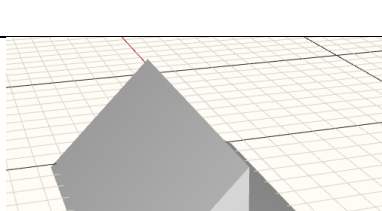
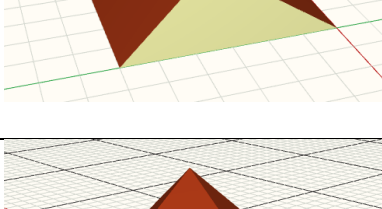
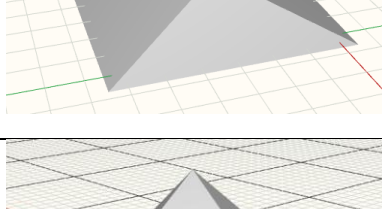
Bibliographische Angaben: 2026, 47 (65) Seiten

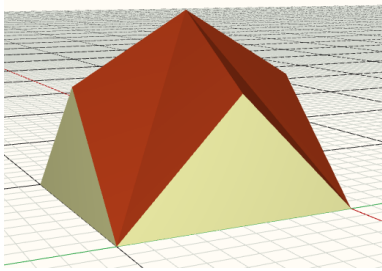
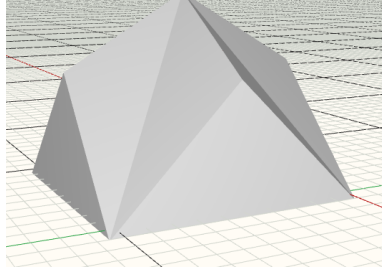
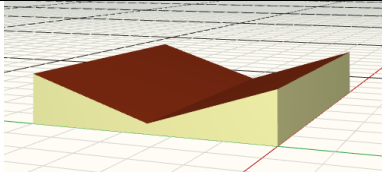
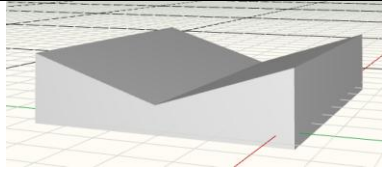
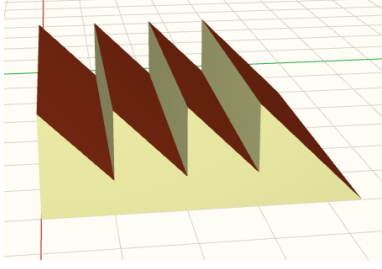
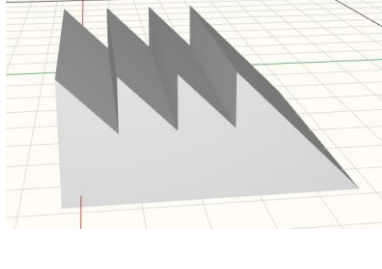
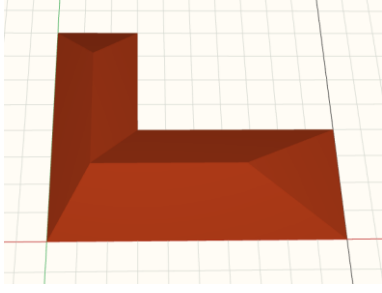
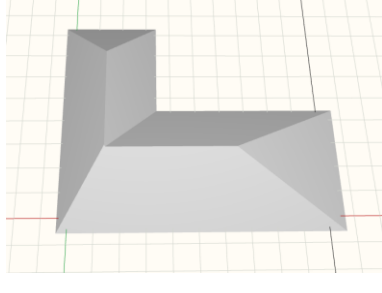
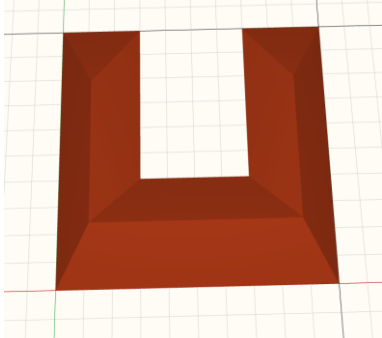
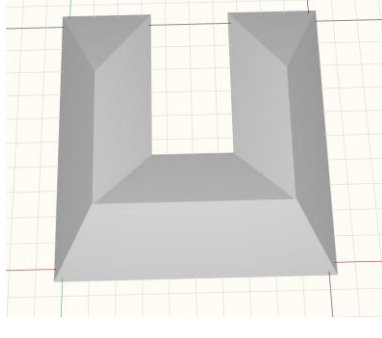
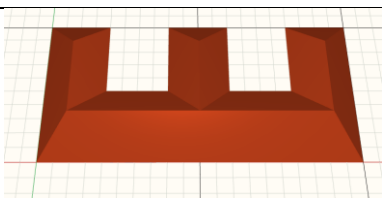
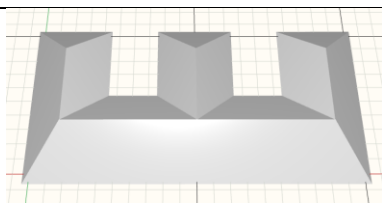
Vor dem Hintergrund steigender Anforderungen an die Energieeffizienz von Gebäuden gewinnt eine durchgängige digitale Planung zunehmend an Bedeutung. Während thermische Gebäudesimulationsprogramme mit innenmaßbezogenen Geometriemodellen arbeiten, erfordern baurechtliche Nachweise wie der GEG-Nachweis außenmaßbezogene Bezugsflächen. Der fehlende automatisierte Übergang zwischen diesen Maßbezügen führt bislang zu hohem Modellierungsaufwand und unnötigen Kosten.

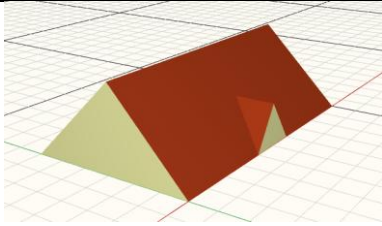
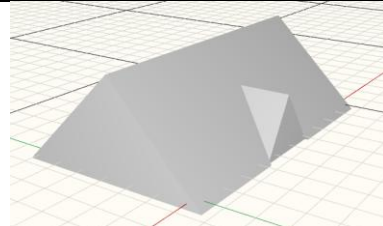
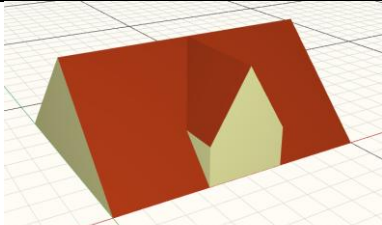
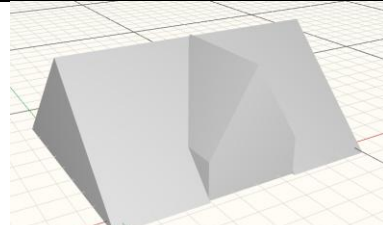
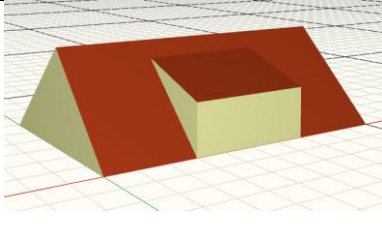
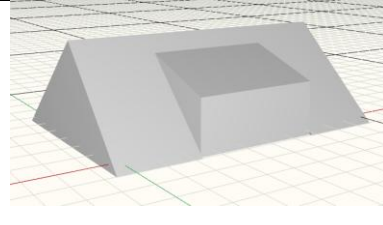
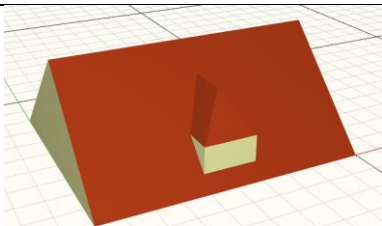
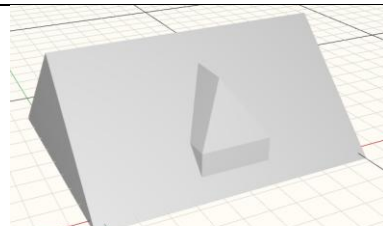
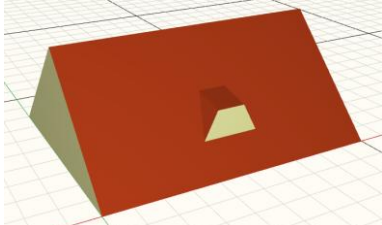
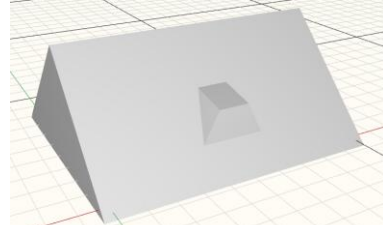
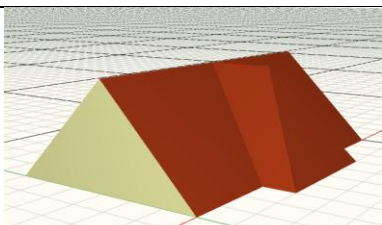
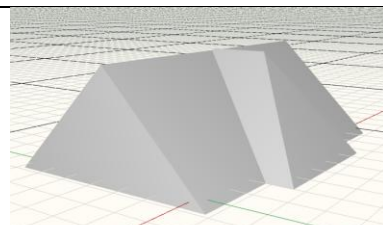
Ziel dieser Arbeit ist die Entwicklung eines robusten Algorithmus zur Transformation innenmaßbezogener Raummodelle in den Außenmaßbezug. Aufbauend auf einem bestehenden Ansatz wurde der Algorithmus erweitert, um auch komplexe Geometrien korrekt zu verarbeiten. Hierfür wurden neue Berechnungsfälle für parallele und linear abhängige Flächen implementiert sowie eine Iterationsstufe zur Identifikation und Korrektur fehlerhafter Transformationspunkte integriert.

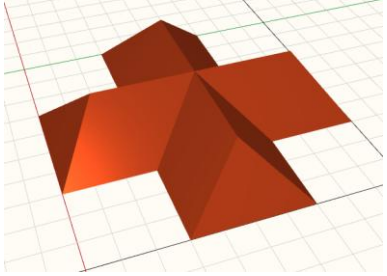
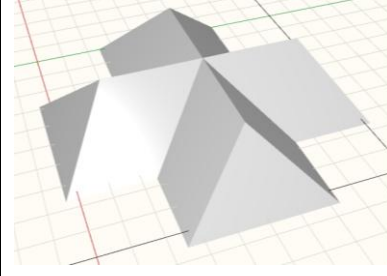
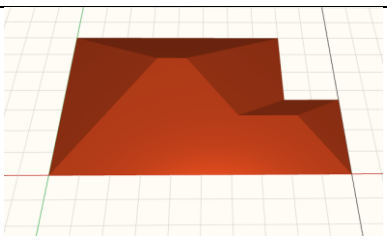
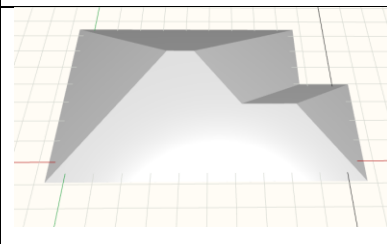
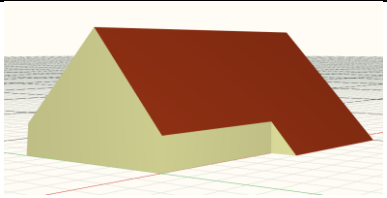
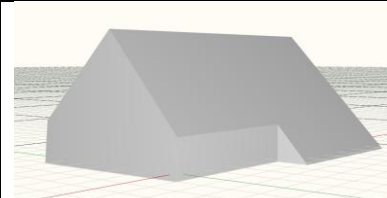
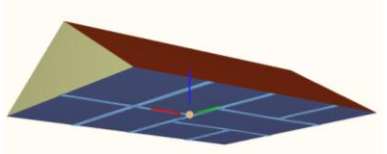
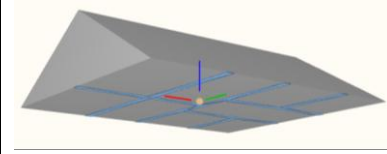
Die Validierung anhand verschiedener Testmodelle zeigt, dass konsistente Transformationskörper erzeugt und der manuelle Nachbearbeitungsaufwand reduziert werden kann. Damit wird eine Grundlage für eine effizientere Verknüpfung von Gebäudesimulation und normbasierter Nachweisführung geschaffen.

Abbildungen von weiteren Testfällen

Name	Raummodell im Innenmaßbezug	Transformationsmodell im Außenmaßbezug
Krüppelwalmdach		
Fußwalmdach		
Mansarddach mit Schopf		
Pultdach		
Pultdach versetzt		
Rhombendach		

Faltdach		
Schmetterlingsdach		
Sheddach		
L-Dach		
U-Dach		
E-Dach		

Zwerchdach klein		
Zwerchgaube groß		
Schlepp-Zwerchgaube klein		
Walmdachgaube		
Trapezgaube		
Sattel-Walmdach		

Walm-Kreuzdach		
Walmdach mit Wiederkehr		
Schleppdach		
Dach mit zerlegtem Boden		
Raum mit unterschiedlichen Waddicken	