



Technische Universität München
Ingenieurfaculty Bau Geo Umwelt
Lehrstuhl für Geoinformatik

Prozedurale Generierung von 3D-Straßennetzwerken unter Berücksichtigung von Brücken und Tunneln am Beispiel Vorarlberg

Roman Esch

Masterarbeit

Bearbeitung: 07.07.2015 - 06.01.2016
Studiengang: Geodäsie und Geoinformation (Master)
Betreuer: Univ.-Prof. Dr. rer. nat. Thomas H. Kolbe
Caroline Marx, M.Sc.

2015 / 2016

Eidesstattliche Erklärung

Erklärung gemäß §18 Absatz 9 APSO der Technischen Universität München:

„Ich versichere, dass ich diese Masterarbeit selbstständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.“

München, den 29.12.2015

Roman Esch

Kurzfassung

Die Nachfrage an 3D-Landschafts- und Stadtmodellen steigt stetig. So werden diese für Landschafts- und Stadtplanungen, von der Film- und Spieleindustrie sowie für Navigations- und Routingfragestellungen verwendet. Hierbei sind auch die Überlegungen eines Übergangs von topografischen und Katasterkarten zu erwähnen. Für das Erscheinungsbild dieser 3D-Landschaftsmodelle spielen auch Straßennetzwerke mit Straßen, Brücken- und Tunnelbauwerken eine wichtige Rolle. Neben der reinen Visualisierung, besteht auch ein Interesse am semantischen Informationsgehalt, ebendieser Straßennetzwerke. Im Rahmen dieser Masterarbeit werden Konzepte zur prozeduralen Generierung von semantisch auswertbaren 3D-Straßennetzwerken entwickelt und bewertet. Die Entwicklung der Konzepte erfolgt anhand eines Datensatzes des österreichischen Bundeslandes Vorarlberg. Diese sollen jedoch auch allgemeingültig und für andere Datensätze implementierbar sein. Das Straßennetzwerk wird auf Basis von Punkt- und Liniengeometrien, gemäß dem CityGML-Schema im LOD 0 und 1 für Straßen und im LOD 2, für die Brücken- und Tunnelmodelle erzeugt. Die größten Herausforderungen stellen die Erzeugung eines durchgängigen Straßennetzwerkes, die Erzeugung semantisch und geometrisch repräsentativer 3D-Brücken- und Tunnelmodelle, die DGM-Korrekturen, sowie den Abgriff der Höheninformationen zum Übergang des Straßennetzwerkes in die dritte Dimension dar. Dazu werden die Konzepte in die vier Teilbereiche Straßen, Brücken, Tunnel und DGM unterteilt. Hierbei sind die Teilbereiche nicht unbedingt getrennt voneinander zu betrachten, weil diese zum Teil aufeinander aufbauen. Neben der Entwicklung und Bewertung der Konzepte, sollen diese auch für den konkreten Datensatz implementiert werden, sodass ein 3D-Straßennetzwerk für den Datensatz von Vorarlberg vorliegt.

Abstract

The demand for 3D landscape and urban models is rising steadily. So these are used for landscape and urban planning, from the film and games industry and for navigation and routing issues. Here the reflections of a transition from topographic and cadastral maps are to be mentioned. For the appearance of this 3D landscape models and road networks play an important role. Besides pure visualization is also an interest in the semantic information content of this road networks. Within the framework of this thesis concepts for procedural generation of semantically evaluable 3D road networks are developed and evaluated. The development of concepts is carried out using a data set of the Austrian province of Vorarlberg. However, these should be implemented universally and also could be used for other records. The road network is created on the basis of point and line geometries according to the CityGML schema in LOD 0 and 1 on roads and in LOD 2 for the bridge and tunnel models. The biggest challenges are the creation of an integrated road network, the creation of semantically and geometrically representative 3D Bridge and Tunnel models DGM-corrections, as well as tap the height information about the transition of the road network in the third dimension. For this purpose, the concepts are divided in the four subregions roads, bridges, tunnels and DGM. Here, the sub-areas are to be considered not strictly separated from one another, because these partially build on each other. In addition to the development and evaluation of concepts, these should also be implemented for the specific data set, so that a 3D road network for the record of Vorarlberg present.

Inhaltsverzeichnis

Eidesstattliche Erklärung	III
1 Motivation	1
1.1 Zielsetzung	1
1.2 Problemstellung	1
2 Verwandte Arbeiten	3
3 Theoretische Grundlagen	5
3.1 Prozedurale Modellierung	5
3.1.1 Prozedurale Modellierung - Definition	5
3.1.2 Shapes	5
3.1.3 CGA-Shape Grammatik	6
3.2 CityGML	7
3.2.1 Kernmodell	7
3.2.2 Brückenmodell	9
3.2.3 Tunnelmodell	11
3.2.4 Transportmodell	12
3.2.5 Digitales Geländemodell	13
3.3 3DCityDB	15
3.4 3D-WebClient	15
3.5 Software	16
3.5.1 Feature Manipulation Engine	16
3.5.2 CityEngine	17
4 Datengrundlage	19
5 Straßen	21
5.1 Extraktion der Straßenmittelachse	21
5.2 Höhenanpassung	22
5.3 Erzeugung der Straßenflächen	23
5.4 Übergang in das CityGML-Schema	28
6 Brücken	29
6.1 Extraktion der Brückenmittelachse	29
6.2 Höhenanpassung an bestehendes Straßennetzwerk	31
6.2.1 Brückenkörper	31
6.2.2 Brückenpfeiler	33
6.3 Modellierung der Brücken	34

6.4	Übergang in das CityGML-Schema	45
7	Tunnel	51
7.1	Höhenanpassung an bestehendes Straßennetzwerk	52
7.2	Modellierung der Tunnel	52
7.3	Übergang in das CityGML-Schema	55
8	Digitales Geländemodell	59
8.1	DGM-Korrektur: Straßen	59
8.1.1	Längsneigung	60
8.1.2	Querneigung	61
8.2	DGM-Korrektur: Brückenkörper	64
8.2.1	Brückenumfeld ohne Höhenvergleich	64
8.2.2	Brückenumfeld mit Höhenvergleich	67
8.2.3	Klassifikation der Punktwolke	72
8.3	DGM-Korrektur: Tunnelportale	75
8.4	DGM-Aufbereitung	76
8.5	Übergang in das CityGML-Schema	78
9	Visualisierung und Bewertung der Ergebnisse	79
9.1	Kombination zum Straßennetzwerk	79
9.2	Implementierung und Daten	88
9.3	Verbesserungsmöglichkeiten	92
10	Fazit und Ausblick	93
	Literaturverzeichnis	95
	Abbildungsverzeichnis	98
	Tabellenverzeichnis	103
	Abkürzungsverzeichnis	105
	Anhang	107
10.1	Tabellen	107
10.2	Visualisierungen	115
10.3	FME Workspaces	126

1 Motivation

1.1 Zielsetzung

Das österreichische Bundesland Vorarlberg plant den Aufbau eines semantischen dreidimensionalen (3D) Landschaftsmodells, unter anderem für die Bereiche Verkehr, Gebäude, Seilbahnen, Stromleitungen, Vegetation und Gewässer. Auf Basis dieses Landschaftsmodells sollen dann räumliche und thematische Abfragen und Anwendungen möglich sein. Daneben spielen die Überlegungen eines Übergangs der topographischen und der Katasterkarten in die dritte Dimension eine große Rolle. Bei diesem Projekt arbeitet der Lehrstuhl für Geoinformatik der Technischen Universität München mit dem Bundesland Vorarlberg und der Firma Virtual City Systems (VCS) zusammen. Bei einem Landschaftsmodell spielen Straßennetzwerke mit den Straßen sowie den Ingenieurbauwerken, wie Brücken und Tunnel, eine wichtige Rolle. Speziell dazu werden in dieser Arbeit Lösungsansätze zur prozeduralen Erstellung von 3D-Straßennetzwerken erarbeitet und bewertet. Die Straßen sollen im Level of Detail (LOD) 0 und LOD 1 erzeugt werden. Die Brücken und Tunnel sollen im LOD 2 als konkrete Modelle erzeugt werden. Ein Übergang der erzeugten Daten, gemäß dem City Geography Markup Language (CityGML)-Datenmodell, ist ebenfalls Bestandteil der Arbeit. Dieser Übergang von den derzeit überwiegend zweidimensionalen (2D) Daten in 3D-Daten beinhaltet viele ungelöste Probleme.

Im Rahmen dieser Arbeit wird ein Testgebiet von 175 km² bearbeitet. Dieses Testgebiet umfasst den mittleren Westen des österreichischen Bundeslandes Vorarlberg. Das Testgebiet lässt sich in 28 Kacheln zu je 2,5 km² unterteilen. In Abbildung 10.1 sind die Kachelgrenzen, die Landesstraßen, die Ortsstraßen, die Tunnel sowie die Brücken dargestellt.

1.2 Problemstellung

Der Übergang von Straßennetzwerken in die dritte Dimension umfasst vier Bereiche. Die erste Problemstellung betrifft die Straßen ohne Ingenieurbauwerke. Diese sollen im LOD 0 als Straßenmittelachsen und im LOD 1 als Straßenflächen erzeugt werden. Dabei sind die Straßen gemäß den korrekten Straßenbreiten und Höhenlagen zu erstellen.

Der zweite Problembereich umfasst die Ingenieurbauwerke Brücken, welche den unterbrechungsfreien Verlauf des Straßennetzwerkes gewährleisten. Die Brücken sollen im LOD 2 als Brückenkörper

erzeugt werden. Die konkreten Brücken sind auf Basis der jeweiligen Brückentypen, -positionen, -breiten und -höhen umzusetzen. Diese Umsetzung ist in dieser Form jedoch nur im LOD 2 möglich, da zum Beispiel Brückenpfeiler oder Fahrbahnplatten im LOD 1 nicht semantisch differenziert werden können.

Den dritten Problembereich stellen die Ingenieurbauwerke Tunnel dar, welche wiederum den unterbrechungsfreien Verlauf des Straßennetzwerkes gewährleisten sollen. Die Tunnel werden im LOD 2 erzeugt. Den letzten Bereich stellt das Digitale Geländemodell (DGM) dar, welches in den Bereichen der Straßen, Brücken und Tunnel korrigiert werden muss. Die gesamte Erzeugung aller vier Teilbereiche sollte möglichst automatisch und wenn möglich in prozeduraler Struktur erfolgen.

Dazu sind allgemeingültige Lösungskonzepte zu erarbeiten. Die aufgeführten Probleme weisen verschiedene Schwierigkeiten auf. Um die Straßen in korrekter Breite erzeugen zu können, bedarf es Informationen zur Straßenbreite. Die Anpassung der Straßen in der Höhe weist eine weitere Schwierigkeit auf. Die Straßen müssen so in das DGM integriert werden, dass die Straßenoberflächen nicht über und nicht unter dem Gelände liegen. Da die Straßen in der Realität unter Beachtung verschiedener Parameter wie Querneigung, Längsneigung und Ebenheit gebaut werden, müssen diese Aspekte auch Einzug in die Modellierung nehmen. Damit die Brücken korrekt erzeugt werden können, müssen ebenfalls Schwierigkeiten überwunden werden. So müssen die jeweiligen Brückenbauweisen aus den entsprechenden Zusatzinformationen extrahiert werden. Die korrekten Brückenpositionen und Brückenbreiten sind analog der Straßenbreiten zu ermitteln. Damit die Brücken und die Brückenpfeiler auch in der Höhe wie gewünscht platziert werden können, muss das oftmals fehlerhafte DGM korrigiert werden. Da die Brückengrundflächen, wie auch die Straßenflächen, in Kurven durch Geradenstücke approximiert werden, entstehen bei einer simplen Brückenerzeugung Fehler. Bei der Modellierung der Tunnel muss beachtet werden, dass die Straßen innerhalb der Tunnel ohne Unterbrechung an das übertägige Straßennetzwerk anschließen.

Bei der bisherigen Generierung von 3D-Daten mittels prozeduraler Modelle, werden nur 3D-Geometrien ohne die dazugehörigen semantischen Informationen erzeugt. Für diese Thematik gilt es einen Ansatz zu erarbeiten, welcher die Erzeugung von 3D-Geometrien inklusive semantischer Informationen ermöglicht.

Die vorliegende Arbeit ist in mehrere Bereiche untergliedert. Zu Beginn der Arbeit werden zunächst einige theoretische Grundlagen und Datengrundlagen erläutert. Im Hauptteil der Arbeit, welcher aus den vier Teilen Straßen, Brücken, Tunnel und digitales Geländemodell besteht, werden die Konzepte und Ansätze dargelegt. Den letzten Teil der Arbeit bilden die Visualisierung und Bewertung der Ergebnisse sowie ein kurzer Ausblick.

2 Verwandte Arbeiten

In diesem Kapitel werden verschiedene Arbeiten vorgestellt, welche sich ebenfalls mit Themen dieser Masterarbeit beschäftigen.

Procedural Modeling of Cities

In *Procedural Modeling of Cities* aus dem Jahr 2001 beschreiben Parish und Müller erstmals, wie sich mit prozeduraler Modellierung, basierend auf dem Lindenmayer-System (L-System), ganze Stadtmodelle effizient modellieren lassen. Dabei werden auf Basis von Eingabedaten, wie Gewässergrenzen und Bevölkerungsdichten Straßennetze erzeugt, welche das Land in Grundstücke aufteilen. Die Gebäude- und auch die Straßenmodellierung erfolgt dann durch eine Erweiterung des L-Systems. Dabei werden globale Ziele und lokale Zwangsbedingungen eingeführt, um die Modellierung zu vereinfachen. Die konkrete Umsetzung erfolgt mit dem erstmals vorgestellten System CityEngine, welches in dieser Masterarbeit in der aktuelleren Version *2014.1* zum Einsatz kommt (vgl. Kapitel 3.5.2) [Parish u. Müller (2001)].

Procedural Modeling of Buildings

In *Procedural Modeling of Buildings* beschreiben Müller et al. 2006 die Möglichkeiten der 3D-Modellierung mit Hilfe der regelbasierten Shape-Grammatik Computer Generated Architecture (CGA) (vgl. Kapitel 3.1.3). Diese Regeln erlauben auf Basis des Ansatzes von Parish u. Müller (2001) einen regelbasierten und hierarchischen Aufbau von Gebäude- beziehungsweise Stadtmodellen. Neben der Möglichkeit, die Modelle in verschiedenen Detailstufen erzeugen zu können, erlauben die generalisierten CGA-Regeln die Verarbeitung von großen Stadtgebieten [Müller u. a. (2006)].

Interactive Procedural Street Modeling

In *Interactive Procedural Street Modeling* aus dem Jahr 2008 befassen sich Chen et al. mit den Problemen der interaktiven Modellierung großer Straßennetze. Dabei stellen sie ein Werkzeug vor, mit dem ein Benutzer ein völlig neues Straßennetzwerk erstellen oder ein bestehendes modifizieren kann. Hierbei werden die den Straßen zu Grunde liegenden Mittelachsen mit Hilfe eines Tensorfeldes angepasst. Die Modellierung kann zudem durch globale und lokale Operationen wie Glättung oder Zwangsbedingungen angepasst werden [Chen u. a. (2008)].

Procedural Modeling of Urban Road Network

In *Procedural Modeling of Urban Road Network* aus dem Jahr 2010 beschreiben (Gang u. Guangshun, 2010) ein Konzept um städtische Straßennetze auf Basis von Luftbildern, Vektordaten sowie manueller Markierungen von Straßenmittelachsen zu modellieren. Dabei werden prozedurale

Methoden angewandt, welche eine effiziente Erzeugung eines 3D-Straßennetzwerks ermöglichen. Konkret wird untersucht, wie sich mit Straßenmittelachsen und Luftbildern Straßenpolygone ermitteln lassen. Außerdem werden die an Kreuzungen zusammentreffenden Straßenübergänge durch approximierte Kurven aneinander angepasst [Gang u. Guangshun (2010)].

Procedural Generation of Roads

In *Procedural Generation of Roads* aus dem Jahr 2010 beschreiben Galin et al. eine automatische Methode zur Generierung von Straßen, basierend auf dem *gewichteten anisotropen kürzester Weg Algorithmus*. Der Verlauf einer Straße, zwischen zwei Punkten, wird durch die Minimierung einer Kostenfunktion bestimmt. Hierbei werden verschiedene Parameter, wie zum Beispiel Geländeneigung oder natürliche Hindernisse wie Flüsse ausgewertet. In einem zweiten Schritt erfolgt die prozedurale Generierung der Straßen und falls notwendig von zusätzlichen Brücken und Tunneln. Im Gegensatz zur vorliegenden Masterarbeit, basiert die Platzierung und Anzahl der Straßen und Bauwerke auf einer Optimierung und entspricht nicht der Realität [Galin u. a. (2010)].

Automated 3D road and building reconstruction using airborne laser scanner data and topographic maps

In seiner Dissertation *Automated 3D road and building reconstruction using airborne laser scanner data and topographic maps* aus dem Jahr 2010 beschreibt und evaluiert Oude Elberink Methoden und Konzepte, um Straßen und Gebäude aus Laserscanning-Daten und 2D topographischen Karten zu erzeugen. Dabei erläutert er wie mit Hilfe der Punktwolken, selbst stark überlappende 2D Straßennetze in 3D hergestellt werden können. Konkret nutzt er die vergleichsweise gleichmäßige Steigung und Glätte der Straßen, um deren Verlauf, unter zum Beispiel Brücken, wiederherzustellen. Neben der Zuweisung der Höheninformation, werden die Straßen auch bezüglich ihrer topologischen Korrektheit untersucht [Oude Elberink (2010)].

Semantische Integration von zweidimensionalen GIS-Daten und Digitalen Geländemodellen

In seiner Dissertation aus dem Jahr 2006 beschreibt Koch ein neues Verfahren um zweidimensionale Daten semantisch in Digitale Geländemodelle zu integrieren. Die konkrete Integration, also die Anpassung der Höheninformationen der 2D-Daten, erfolgt durch die Umsetzung von Bedingungs-gleichungen und -ungleichungen. So werden die Anforderungen an Quer- und Längsneigung von Straßen in einer globalen Ausgleichung gelöst [Koch (2006)].

3D Spatial Data Infrastructures for Web-Based Visualization

In seiner Dissertation *3D Spatial Data Infrastructures for Web-Based Visualization* aus dem Jahr 2014 beschreibt Schilling Methoden und Ansätze für die Entwicklung von Geodateninfrastrukturen im Hinblick auf 3D Landschafts- und Stadtmodelle. Dabei ist besonders die Integration von Straßennetzwerken und Geländeoberflächen interessant. Die Glättung der Straßenbereiche des Digitalen Geländemodells wird mit Hilfe einer Constrained Delaunay Triangulation (CDT) durchgeführt. Die zugrundeliegende Zwangsbedingung ist dabei, dass die Kanten möglichst horizontal sein sollen [Schilling (2014)].

3 Theoretische Grundlagen

3.1 Prozedurale Modellierung

3.1.1 Prozedurale Modellierung - Definition

Die Prozedurale Modellierung beschäftigt sich mit (halb-) automatischer Generierung von Modellen mit Hilfe eines Programms oder einer Prozedur. Dabei ist die Datenkompression und das Potential, eine Vielzahl von komplexen Geometrien, ausgehend von wenigen Eingangsinformationen, ohne menschliche Eingriffe erzeugen zu können, für die Erstellung von virtuellen Umgebungen sehr gut geeignet. Diese virtuellen Umgebungen finden zunehmend in Filmen, Spielen und Simulationen Verwendung [Smelik u. a. (2014)]. Die Schlüsseleigenschaft der prozeduralen Generierung ist, dass die Entitäten, ob Geometrie oder Textur, als Abfolge von Produktionsregeln dargestellt werden und nicht als statische Objekte. Diese Produktionsregeln können eine hierarchische Struktur bilden, welche es erlaubt, die Bestandteile der Modelle zu differenzieren. Zudem müssen die Modelle nicht zwingend mit ihrer expliziten Geometrie abgespeichert werden, was den Speicherbedarf senkt [Kelly u. McCabe (2006)]. Des Weiteren ist eine automatische Erzeugung verschiedener LOD möglich, da die konkrete Modellierungsgenauigkeit eingeschränkt werden kann [Parish u. Müller (2001)].

3.1.2 Shapes

Basis der prozeduralen Modellierung und der Shape-Grammatik sind die Shapes. Ein Shape besteht aus einem Symbol, einer Geometrie und numerischen Attributen. Die Geometrie, welche Punkte, Linien oder Polygone beschreiben kann, besteht aus geometrischen Attributen. Dabei sind die wichtigsten geometrischen Attribute die Position P , welche durch die drei orthogonalen Vektoren X , Y , und Z beschrieben wird und der Größen-Vektor S . Diese Attribute definieren einen im Raum orientierten Begrenzungsrahmen (*bounding box*), welcher *scope* genannt wird und die Geometrie enthält. Neue Shapes entstehen durch die Anwendung von Produktionsregeln auf das Vorgänger-Shape. Durch dieses System ergibt sich ein hierarchischer Zusammenhang zwischen den Shapes [Müller u. a. (2006)]. Ein Beispiel eines Shapes mit Position P und dem Größen-Vektor S ist in Abbildung 3.1 dargestellt.

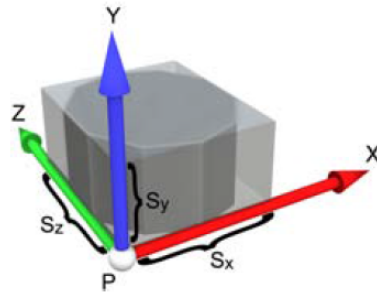


Abb. 3.1 – *scope* eines Shapes mit Position P und dem Größen-Vektor S [Müller u. a. (2006)]

3.1.3 CGA-Shape Grammatik

Die CGA ist eine Programmiersprache, mit der sich architektonische 3D-Gebäude- und Stadtmodelle erzeugen und visualisieren lassen. Die Primitive dieser Sprache sind die räumlich definierten Shapes. Die Beziehungen und Operationen der Sprache sind ebenfalls räumlich, wie zum Beispiel Rotation oder Translation [Gips (1999)]. Die Shape Grammatik ermöglicht mit ihren Produktionsregeln ein iteratives entwickeln von immer detaillierteren Modellen [Müller u. a. (2006)]. So kann zum Beispiel ein 3D-Modell, ausgehend von einem Primitiv, durch verschiedene Operationen in der Shape Grammatik in seine einzelnen Teile unterteilt werden. Dazu ist in Abbildung 3.2 eine beispielhafte CGA-Regel eines 3D-Modells aufgeführt. Darin wird das initiale Shape in der übergeordneten Regel *Gesamt* in X-Richtung aufgeteilt. Hierbei entstehen zwei Teile mit einmal 40 % und einmal 60 % der ursprünglichen Gesamtlänge in X-Richtung. Diese Teile werden den untergeordneten Regeln *Teil_a* und *Teil_b* übergeben. In der Regel *Teil_a* erfolgt eine Extrusion um 1m in positiver Z-Richtung, sowie die Einfärbung des entstehenden Volumenkörpers mit der Farbe blau. In der zweiten untergeordneten Regel *Teil_b* erfolgt eine Extrusion um 2m in positiver Z-Richtung sowie die Einfärbung des entstehenden Volumenkörpers mit der Farbe gelb. Um diese CGA-Regel zu verdeutlichen, ist in Abbildung 3.2 die Hierarchie sowie das resultierende 3D-Modell dargestellt

```
Gesamt -->
split(x){'0.4 : Teil_a |
         '0.6 : Teil_b }

Teil_a -->
extrude(1)
color("#0000ff")

Teil_b -->
extrude(2)
color("#ffff00")
```

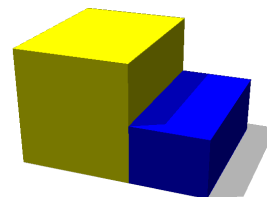
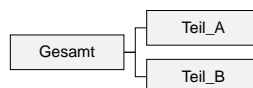


Abb. 3.2 – Beispielhafte CGA-Regel eines 3D-Modells (Links), Hierarchie (Mitte) sowie resultierendes 3D-Modell (Rechts)

3.2 CityGML

CityGML ist ein offenes Datenmodell und XML-basiertes Format für die Repräsentation, Speicherung und den Austausch semantischer 3D-Stadt- und Landschaftsmodelle. Es ist ein Anwendungsschema für die Geography Markup Language (GML) version 3.1.1, einem erweiterbaren internationalen Standard für den Austausch räumlicher Daten, welcher vom Open Geospatial Consortium (OGC) und dem ISO TC211 veröffentlicht wurde. Das Ziel der Entwicklung von CityGML ist es, eine gemeinsame Definition der grundlegenden Instanzen, Attribute und Beziehungen für 3D-Stadtmodelle bereitzustellen. Dies ist besonders wichtig im Hinblick auf die kosteneffiziente, nachhaltige Instandhaltung von 3D-Stadtmodellen und erlaubt zudem die Wiederverwendbarkeit der gleichen Daten in verschiedenen Anwendungsbereichen [Gröger u. a. (2012)]. CityGML unterstützt verschiedene LODs, welche der jeweiligen Detailtiefe der Modelle entsprechen. Das LOD 0 dient der Darstellung von Geländemodellen mit Gebäuden als Grundrissflächen und Straßen als Liniennetzwerke. Die nächste Stufe stellt das LOD 1 dar, welches die Gebäude als Klötzchenmodelle und Straßen in Flächendarstellung beinhaltet. Im LOD 2 besitzen die Gebäude verschiedene Dachformen und grobe Oberflächenstrukturen. Die Straßen, beziehungsweise die Verkehrsflächen, sind ab LOD 2 in verschiedene Verkehrsteilbereiche unterteilt. Das LOD 3 beinhaltet feine architektonische Strukturen der Gebäude. Im LOD 4 sind z.B. innere Bestandteile von Gebäuden, wie Räume und Türen, modelliert [Gröger u. a. (2012)]. Ein Überblick über die verschiedenen LODs ist in Abbildung 3.3 aufgeführt.

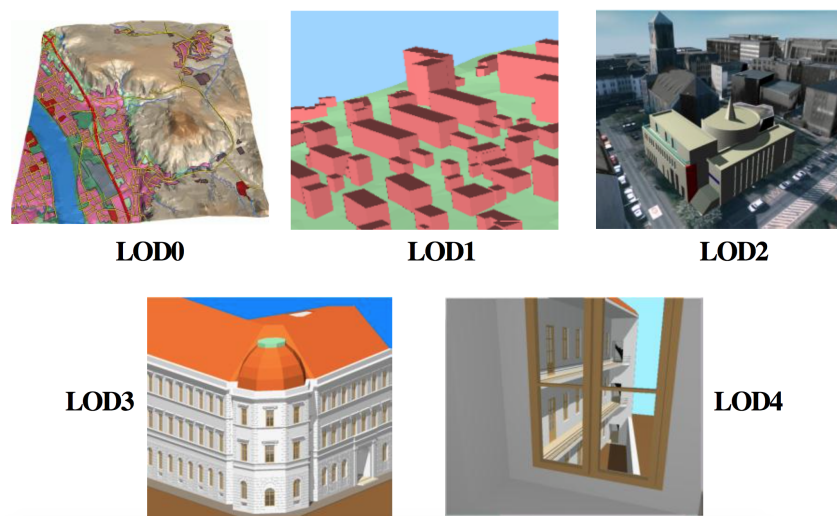


Abb. 3.3 – LOD Überblick [Gröger u. a. (2012)]

3.2.1 Kernmodell

Die Basisklasse aller thematischen Klassen innerhalb des CityGML-Datenmodells ist die abstrakte Klasse *_CityObject*. Diese abstrakte Klasse *_CityObject* enthält zum Beispiel ein Erstellungs- und ein Ablaufdatum (*creationDate* und *terminationDate*) der Features, sowie generische Attribute und externe Referenzen zu entsprechenden Objekten in anderen Datensätzen. Da *_CityObject* eine Un-

terklasse der GML-Klasse `_Feature` ist, kann es mehrere Namen des Features erben, was eine Unterscheidung in verschiedenen Sprachen möglich macht. Mit Hilfe der Generalisierungs-Eigenschaft, `generalizesTo` der Klasse `_CityObject`, ist es zudem möglich, Features darzustellen, welche die gleichen realen Objekte in verschiedenen LOD repräsentieren. Die Features von `_CityObject` und deren spezielle Unterklassen können zu einem `CityModel` zusammengefasst werden. Dies ist dann eine Feature Collection mit optionalen Metadaten. Generell hat jedes Feature die Attribute `class`, `function` und `usage`, es sei denn es ist anders deklariert. Das Attribut `class` kann nur einmal vorkommen, während die Attribute `function` und `usage` mehrfach verwendet werden können. Mit dem Attribut `class` wird die Klassifikation eines Objekts festgelegt. So kann ein Objekt zum Beispiel als Straße, Gleise oder Platz definiert werden. Das Attribut `function` enthält den originären Zweck des Objekts wie zum Beispiel Autobahn oder Gemeindestraße, wohingegen das Attribut `usage` die aktuelle Nutzung des Objekts, wie zum Beispiel Auto- oder Fußgängerbrücke, enthält. Die Unterklassen von `_CityObject` enthalten die verschiedenen Themenfelder eines Stadtmodells, welche wiederum durch gesonderte thematische Modelle abgedeckt werden. Dazu zählen: das Gebäudemodell `_AbstractBuilding`, das Tunnelmodell `_AbstractTunnel`, das Brückenmodell `_AbstractBridge`, das Stadtmöbelmodell `CityFurniture`, das DGM `ReliefFeature`, das Landnutzungsmodell `LandUse`, das Transportmodell `TransportationObject`, das Vegetationsmodell `_VegetationObject`, das Gewässermodell `WaterObject` und das generische Stadtobjektmodell `GenericCityObject` [Kolbe u. a. (2015b)]. Die für diese Masterarbeit relevanten Modelle sind das Brückenmodell, das Tunnelmodell, das Digitale Geländemodell und das Transportmodell. Die Besonderheiten und Aufgaben dieser Modelle werden im Folgenden erläutert.

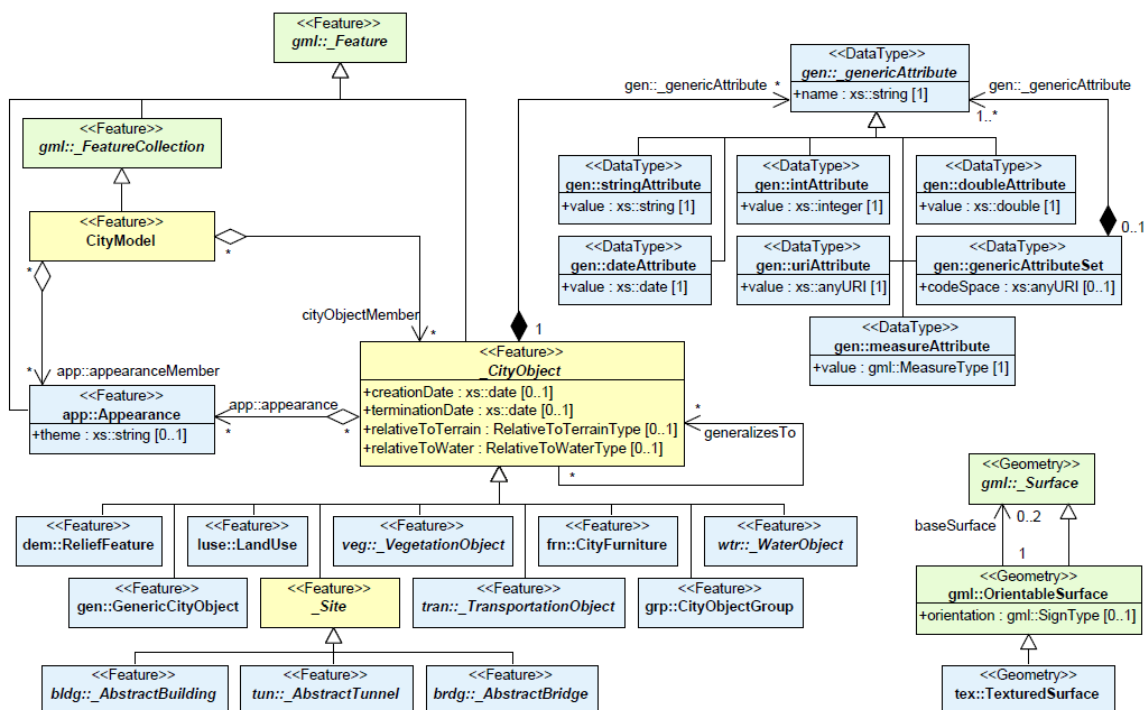


Abb. 3.4 – UML-Diagramm des CityGML Kernmodells mit den Unterklassen von `_CityObject` [Gröger u. a. (2012)]

3.2.2 Brückenmodell

Das Brückenmodell ermöglicht die Repräsentation der thematischen, räumlichen und visuellen Aspekte von Brücken und Brückenteilen in vier LODs (LOD 1-4). Eine Brücke (*Bridge*) kann aus mehreren Brückenteilen (*BridgePart*) bestehen. Wie *Bridge*, so ist auch *BridgePart* eine Unterklasse von *_AbstractBridge* und erbt damit die Attribute und Beziehungen der Elternklasse *_AbstractBridge* [Kolbe u. a. (2015b)]. Das gesamte UML-Diagramm des Brückenmodells ist in Abbildung 3.5 visualisiert.

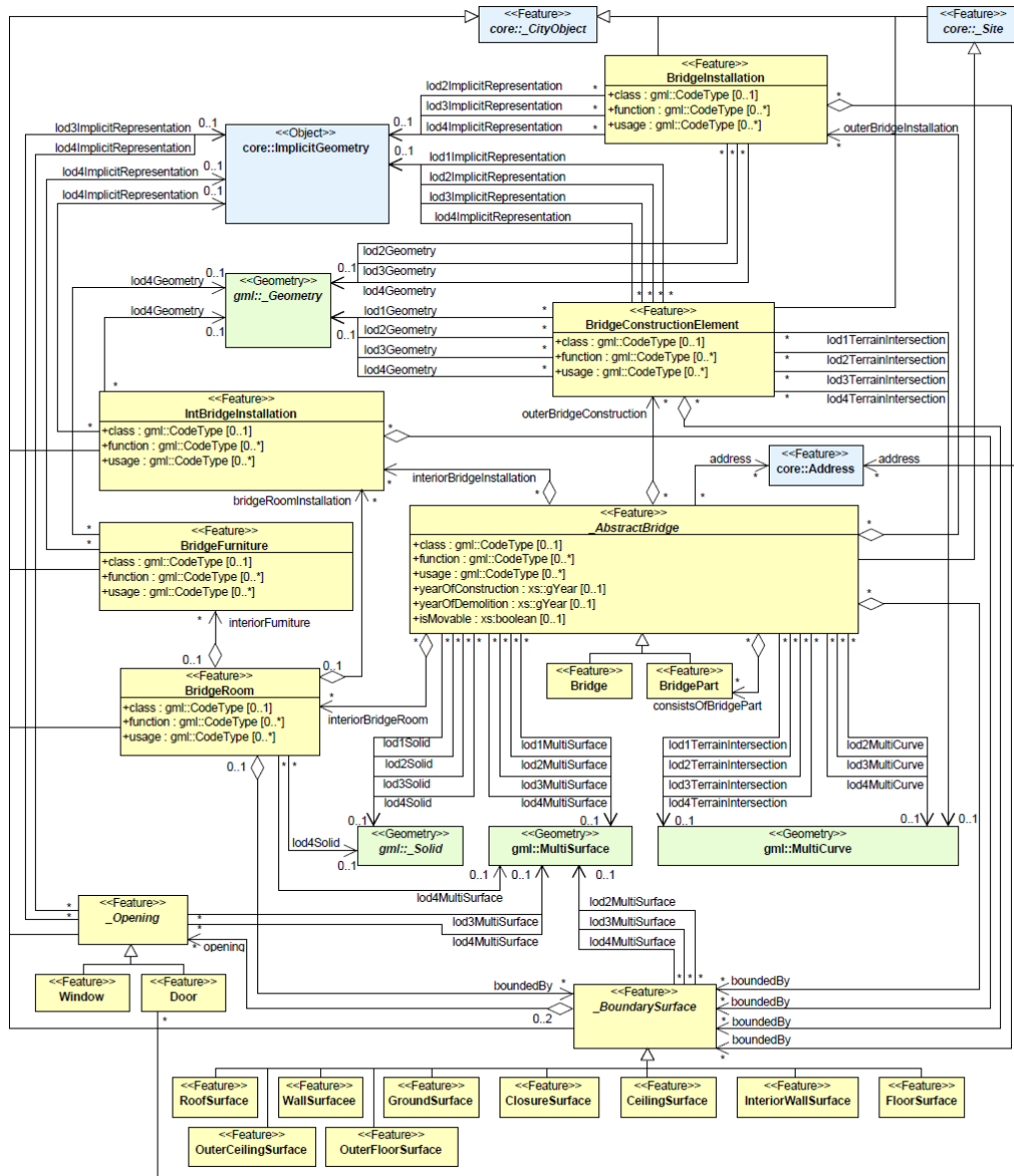


Abb. 3.5 – UML-Diagramm des Brückenmodells [Gröger u. a. (2012)]

Elemente, welche für die Struktur einer Brücke unverzichtbar sind, werden als *BridgeConstructionElement* modelliert. Dazu zählen unter anderem Fundamente (*fundaments*), Pylone (*pylon*) oder

Fahrbahn (*lane of traffic*), was in Abbildung 3.6 an einer Hängeseilbrücke beispielhaft dargestellt ist [Kolbe u. a. (2015b)].

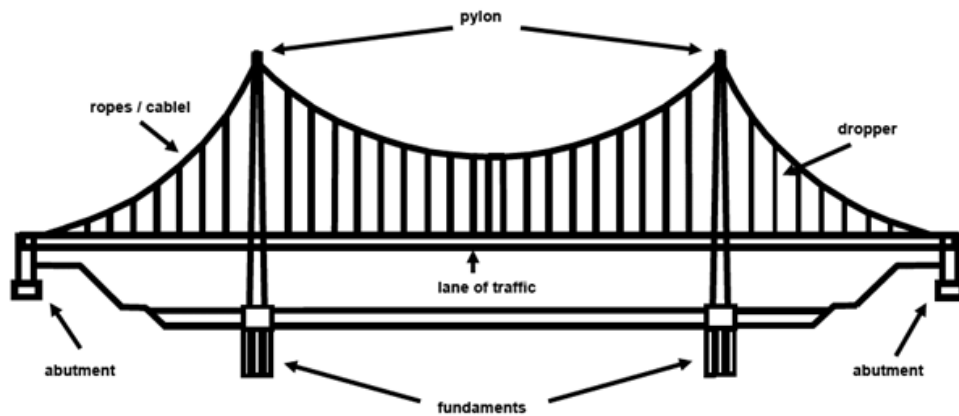


Abb. 3.6 – Verschiedene *BridgeConstructionElements* einer Hängeseilbrücke [Gröger u. a. (2012)]

Sind Bauteile einer Brücke nicht relevant für die Konstruktion, so werden diese als Brückeneinrichtung (*BridgeInstallation*) definiert. Zu *BridgeInstallation* zählen zum Beispiel Lampen, Treppenhäuser oder Geländer. Die Begrenzungsflächen der Brücken werden mit *_BoundarySurface* definiert. *_BoundarySurface* ist die abstrakte Basisklasse für mehrere thematische Klassen, welche die Außenhaut sowie die sichtbaren Oberflächen von Räumen, Bauteilen und Brückeneinrichtung einer Brücke beinhalten. Zu den thematischen Klassen zählen zum Beispiel Dachflächen (*RoofSurface*), Wandflächen (*WallSurface*) oder Bodenflächen (*FloorSurface*) was in Abbildung 3.7 an einem Brückenmodell visualisiert ist [Kolbe u. a. (2015b)].

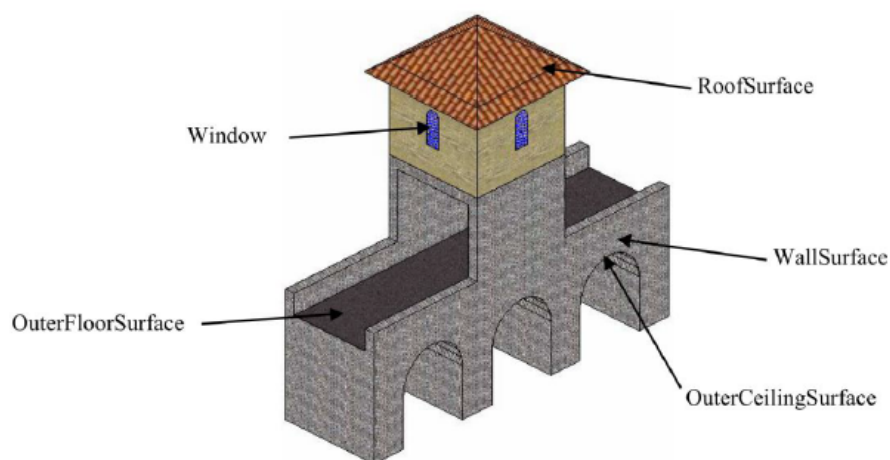


Abb. 3.7 – Verschiedene *_BoundarySurfaces* einer Brücke [Gröger u. a. (2012)]

3.2.3 Tunnelmodell

Analog dem Brückenmodell, unterstützt das Tunnelmodell die Repräsentation der thematischen, räumlichen und visuellen Aspekte von Tunneln und Tunnelkörpern in vier LODs (LOD 1-4). Hierbei ist die abstrakte Klasse `_AbstractTunnel` die Elternklasse der Klassen `Tunnel` und `TunnelPart`. Ein Tunnel kann ebenfalls aus mehreren Tunnelteilen bestehen. Die Attribute und Beziehungen von `_AbstractTunnel` werden den Klassen `Tunnel` und `TunnelPart` vererbt [Kolbe u. a. (2015b)]. Das gesamte UML-Diagramm des Tunnelmodells ist in Abbildung 3.8 visualisiert.

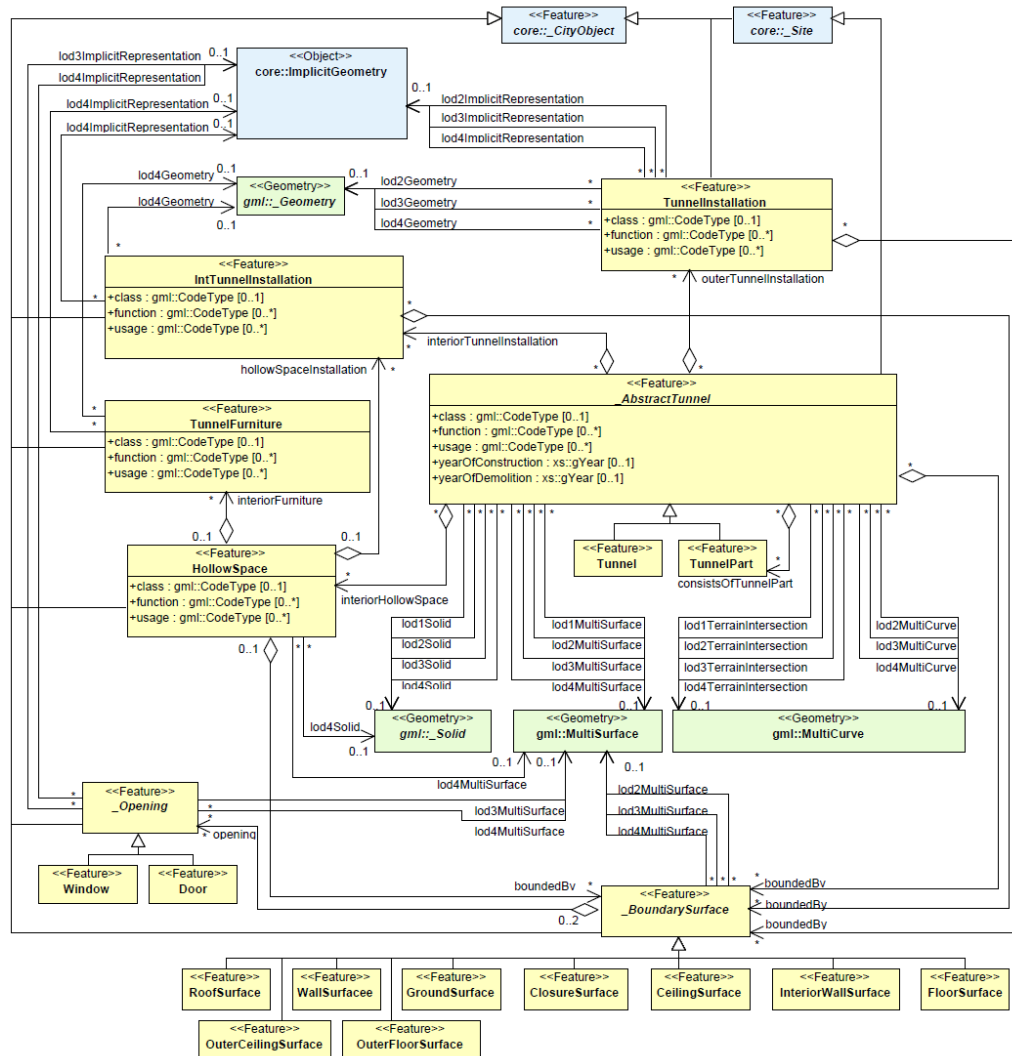


Abb. 3.8 – UML-Diagramm des Tunnelmodells [Gröger u. a. (2012)]

Im LOD 1 besteht das Tunnelmodell aus einer geometrischen Darstellung des Tunnelvolumens. Die geometrische Darstellung kann im LOD 2 durch zusätzliche *MultiSurface* und *MultiCurve* Geometrien verfeinert werden. Hierbei repräsentiert eine *MultiSurface* einen Verbund mehrerer Polygone (*Polygons*). Eine *MultiCurve* ist ein Zusammenschluss mehrerer Kurven (*Curves*). Im LOD 2 und in höheren LOD kann die äußere Struktur des Tunnels semantisch durch die Klassen `_BoundarySur-`

face und *TunnellInstallation* unterschieden werden. *_BoundarySurface* ist eine abstrakte Basisklasse für mehrere thematische Klassen. Zu den thematischen Klassen zählen zum Beispiel Dachflächen (*RoofSurface*), Wandflächen (*WallSurface*) oder Bodenflächen (*FloorSurface*), was in Abbildung 3.9 an mehreren Tunnelquerschnitten dargestellt ist. Die Klasse *TunnellInstallation* definiert Tunnelelemente, wie Außentreppen, welche das äußere Erscheinungsbild eines Tunnels stark beeinflussen [Kolbe u. a. (2015b)].

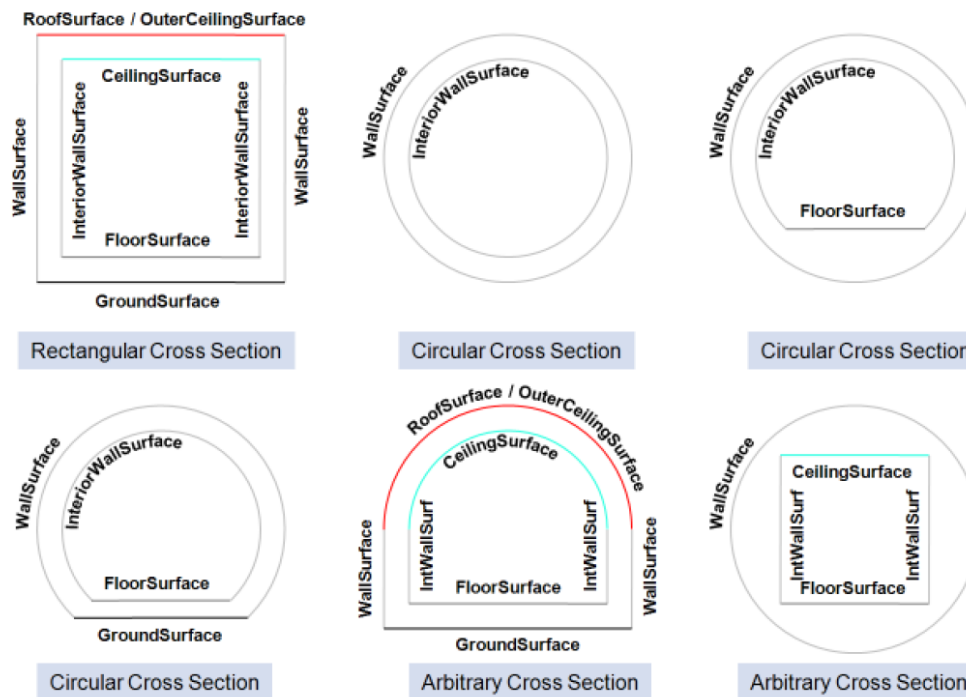


Abb. 3.9 – Verschiedene *_BoundarySurfaces* von unterschiedlichen Tunnelquerschnitten [Gröger u. a. (2012)]

3.2.4 Transportmodell

Das Transportmodell *transportation model* ist ein multifunktionales Modell, welches thematische, funktionale, geometrische und topologische Aspekte vereint. Dabei werden die Features im LOD 0, als lineare Netzwerke repräsentiert. Ab dem LOD 1 werden alle Features geometrisch durch 3D-Oberflächen beschrieben. Die Hauptklasse ist *TransportationComplex*, welche zum Beispiel eine Straße oder Gleise repräsentiert. Ein *TransportationComplex* setzt sich aus *TrafficArea* und *AuxiliaryTrafficArea* zusammen. Dabei beschreibt *TrafficArea* die Flächen, welche für die konkrete Nutzung relevant sind. So zum Beispiel Fahrbahn, Gehweg oder Radweg. *AuxiliaryTrafficAreas* definieren hingegen Flächen, welche weniger relevant für die Nutzung sind. Dazu zählen zum Beispiel Bordsteine, Mittelstreifen oder Grünflächen [Kolbe u. a. (2015b)]. Das gesamte UML-Diagramm des Transportmodells ist in Abbildung 3.10 dargestellt.

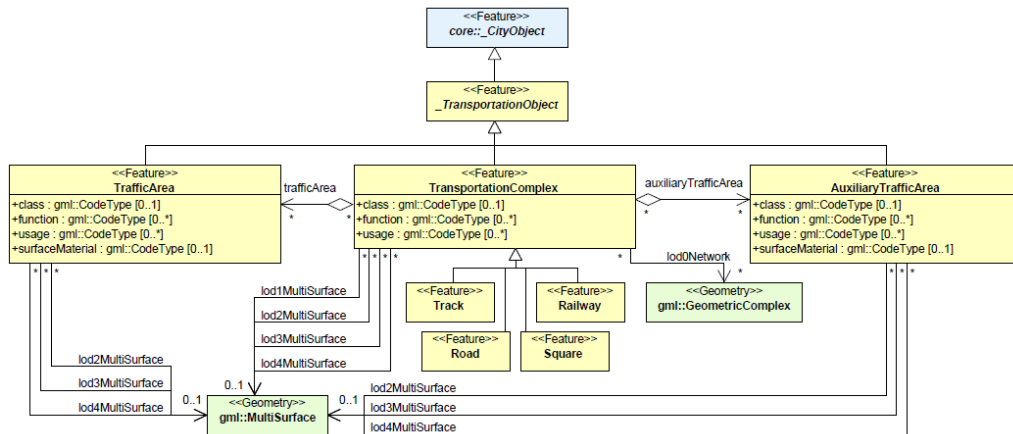


Abb. 3.10 – UML-Diagramm des Transportmodells [Gröger u. a. (2012)]

3.2.5 Digitales Geländemodell

CityGML unterstützt ein sehr anpassungsfähiges DGM, welches die Kombination von heterogenen DGM-Typen in verschiedenen LOD ermöglicht. Ein konkretes DGM wird durch die Klasse *ReliefFeature* beschrieben. Ein Relief besteht aus mehreren *ReliefComponents*. *ReliefComponents* hat die vier geometrischen Unterklassen *RasterRelief*, *Triangulated Irregular Network (TIN) (TINRelief)*, *Bruchkanten (BreaklineRelief)* und *Massepunkte (MassPointRelief)*. Die Klassen sind geometrisch wie folgt definiert: Bruchkanten durch die Geometrie *MultiCurve*, TINs durch die Geometrie *TriangulatedSurface*, Massepunkte durch die Geometrie *MultiPoint* und Raster durch die Klasse *Rectified-GridCoverage* [Kolbe u. a. (2015b)]. Das gesamte UML-Diagramm des Digitalen Geländemodells ist in Abbildung 3.11 veranschaulicht.

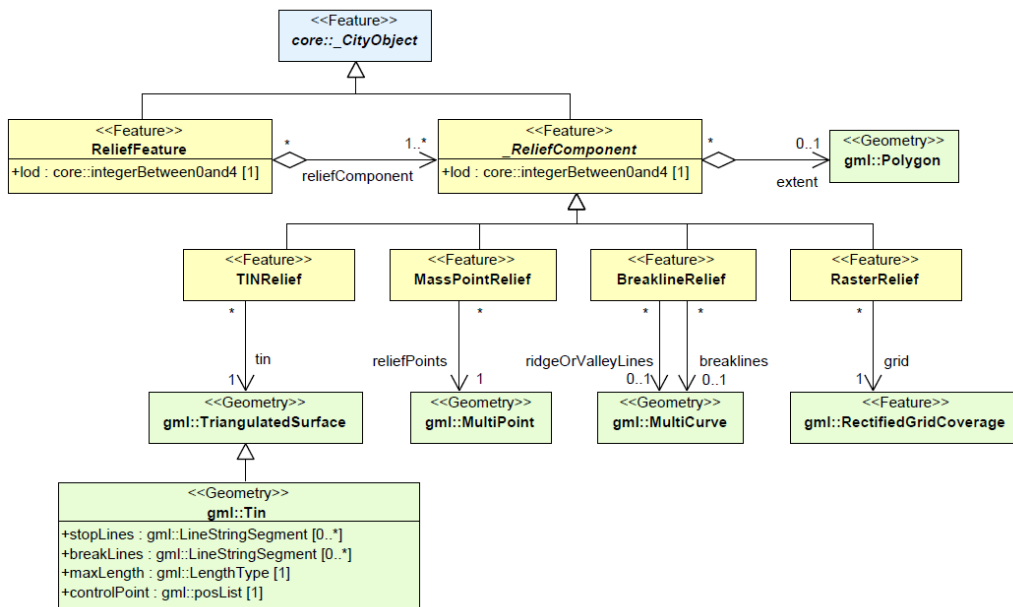


Abb. 3.11 – UML-Diagramm des Digitalen Geländemodells [Gröger u. a. (2012)]

Bei der Integration von 3D-Modellen in Geländemodelle ergeben sich oftmals Überschneidungen, weil die 3D-Modelle das Gelände berühren oder durchdringen. Dies ist besonders bei Tunnelmodellen der Fall, welche sich zum größten Teil unterhalb der Geländeoberfläche befinden, jedoch an den Portalen die Geländeoberfläche durchstoßen. Um diesen Umstand beschreiben zu können, kann die *TerrainIntersectionCurve* verwendet werden. Die *TerrainIntersectionCurve* beschreibt die genaue Position, wo das Gelände das 3D-Modell berührt. Ein weiteres Problem stellen Modelle dar, welche an der Geländeoberfläche enden, so zum Beispiel U-Bahneingänge oder Tunnelportale. Das Problem an den Modellen ist, dass sie als geschlossene Körper modelliert werden. Die Geländemodelle sind problematisch, weil diese als durchgängige Fläche vorliegen müssen. Diese Kontaktflächen können mit Hilfe der *ClosureSurface* als „offen“ beziehungsweise „durchlässig“ definiert werden [Gröger u. a. (2012)]. Ein Beispiel zweier *TerrainIntersectionCurves* und einer *ClosureSurface* ist in Abbildung 3.12 dargestellt.

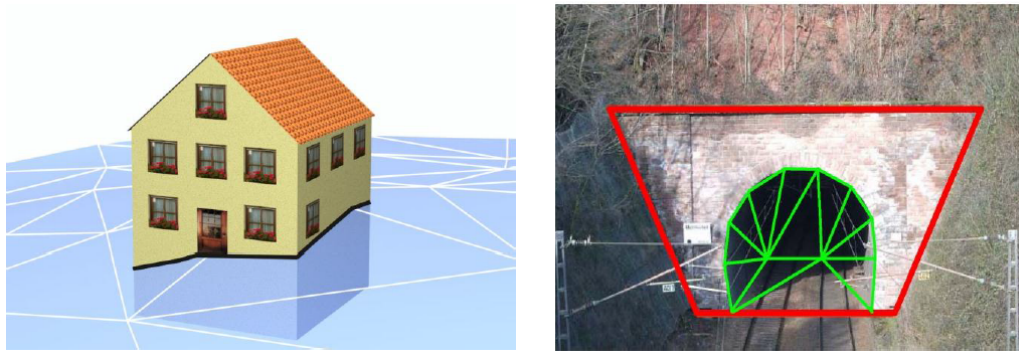


Abb. 3.12 – *TerrainIntersectionCurve* eines Gebäudes und eines Tunnels (links in schwarz, rechts in rot) und eine *ClosureSurface* eines Tunnels (rechts in grün) [IGG Uni Bonn]

3.3 3DCityDB

Die 3D City Database (3DCityDB) 3.0.0 ist eine frei zugängliche Datenbankstruktur. Sie wurde in Kooperation mit dem *Lehrstuhl für Geoinformatik* der *Technischen Universität München* sowie den Unternehmen *virtualcitySystems GmbH* und *M.O.S.S. GmbH* entwickelt. Sie besteht aus einem Datenbankschema und einer Reihe von Software Werkzeugen zum Importieren, Verwalten, Analysieren, Visualisieren und Exportieren von 3D Stadtmodellen gemäß dem CityGML-Standard. Das Datenbankschema ergibt sich aus der Projektion des objektorientierten Datenmodells von CityGML 2.0 auf die relationale Struktur eines Spatially-enhanced Relational Database Management System (SRDBMS). Die 3DCityDB unterstützt ORACLE- und PostgreSQL-Datenbanken. Sie ist in der Lage, sehr große Modelle in verschiedenen LODs mit Millionen von 3D-Objekten, bestehend aus hunderten Millionen Geometrien und Texturen, zu verwalten [Kolbe u. a. (2015b)].

3.4 3D-WebClient

Der 3D-WebClient, welcher vom *Lehrstuhl für Geoinformatik* der *Technischen Universität München* entwickelt wurde, ermöglicht die Nutzung, Bearbeitung und Analyse semantischer 3D-Stadtmodelle (vgl. Abbildung 3.13). Hauptunterschied zu bisherigen Anwendungen ist, dass mehrere Nutzer gleichzeitig mit den Stadtmodellen arbeiten können. Die Datenhaltung des 3D-WebClient, wird mit einer 3DCityDB realisiert. Damit trotz der großen Datenmengen ein Arbeiten mit der cloud-basierten Anwendung möglich ist, werden nur die relevanten Regionen und die relevanten Objektarten und Attribute exportiert. Dabei wird ein 3D-Modell im KML/COLLADA-Format zur visuellen Darstellung und eine Tabelle (Spread sheet) je Objekt erzeugt. Im konkreten Webclient können dann räumliche, zeitliche und semantische Abfragen durchgeführt werden [Yao u. Kolbe (2014)].

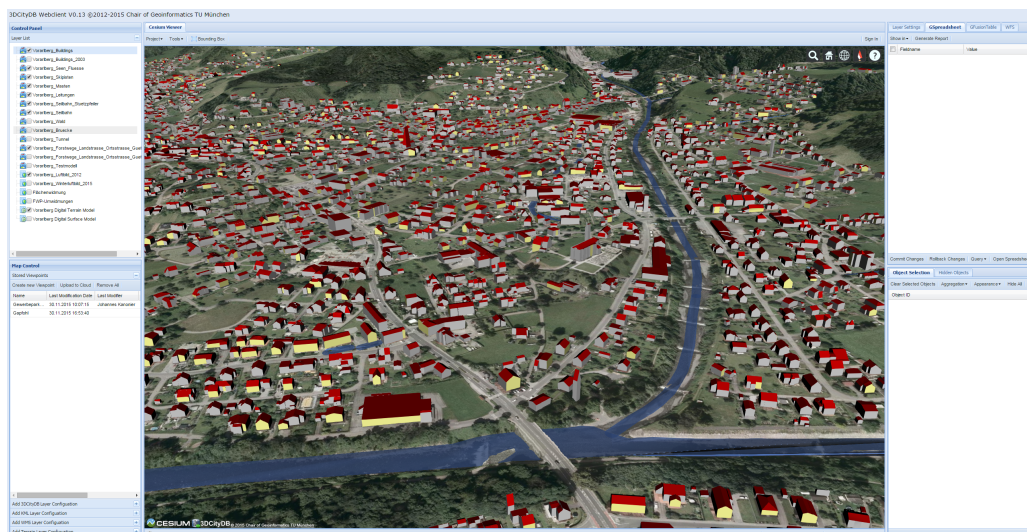


Abb. 3.13 – Ausschnitt aus dem 3D-WebClient. Bildquelle: Lehrstuhl für Geoinformatik TU München (2015)

3.5 Software

3.5.1 Feature Manipulation Engine

Bei der Datenvorbereitung und Erarbeitung von Lösungskonzepten werden in dieser Arbeit verschiedene Software-Pakete benutzt. Die Software Feature Manipulation Engine (FME) der Firma Safe Software Inc. wird zur Datenvorbereitung, -bearbeitung und -konvertierung verwendet. Sie ist ein professionelles ETL-Werkzeug (Extract, Transform, Load) und unterstützt die meisten GIS-Formate. Zudem ist eine Nutzung mit verschiedenen Datenbanken möglich [Jahnke u. a. (2011)]. Konkret werden zum Einlesen der Daten sogenannte *Reader* verwendet. Das Gegenstück bilden die *Writer*, welche die Daten wiederum außerhalb des Programms speichern. Zwischen diesen Schritten können die Daten durch bereits implementierte Transformatoren und Werkzeuge verarbeitet werden. Neben den bereits vorhandenen Werkzeugen können auch individuelle Transformatoren erstellt werden. Die *Reader*, *Writer* und Werkzeuge werden im Programm als Formelemente abgebildet. Um den Datenfluss zwischen diesen Formelementen herzustellen, werden die Eingänge und Ausgänge mit Linien verbunden, wodurch sich ein Prozessablauf der Datenbearbeitung ergibt [ESRI (2015a)]. Ein Beispiel eines Prozessablaufs in FME ist in Abbildung 3.14 aufgeführt. Darin wird ein ESRI Shape mit Straßenmittelachsen einer Kreuzung eingeladen, welche um einen bestimmten Wert gepuffert werden. Anschließend werden diese wiederum als ESRI Shape abgespeichert.

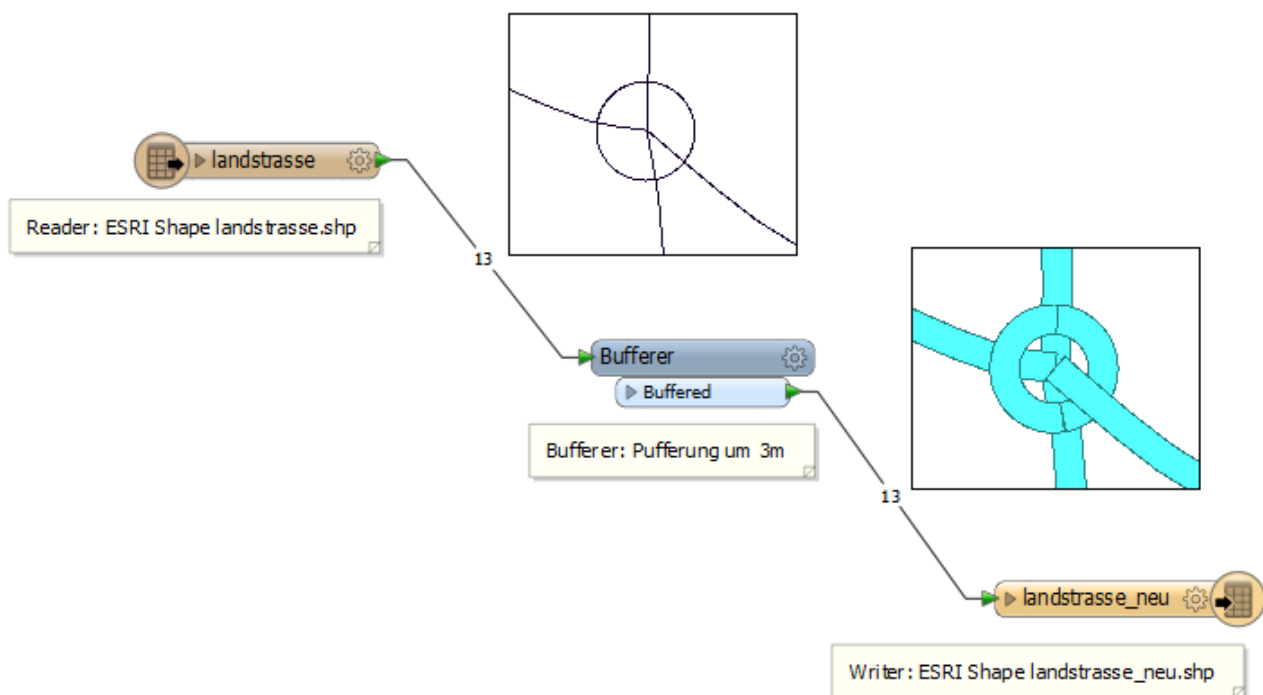


Abb. 3.14 – Beispielhafter FME Prozessablauf mit *Reader landstrasse*, *Transformer Bufferer* und *Writer landstrasse_neu*

3.5.2 CityEngine

Die prozedurale Modellierung der 3D-Modelle sowie die Erzeugung der Straßenflächen, wird mit der Software CityEngine durchgeführt. Die Software ist vom, an der ETH Zürich gegründeten, Spin-Off *Procedural Inc.* entwickelt worden. Dieses Unternehmen wurde 2011 von der Firma Environmental Systems Research Institute (ESRI Inc.) aufgekauft [(Hofmann u. Heller, 2013)]. CityEngine ist in der Lage großflächige Stadtmodelle mit vergleichsweise wenigen Eingangsdaten zu erzeugen und bieten dem Benutzer zugleich eine Vielzahl an Einstellungsmöglichkeiten. Die konkrete Modellierung basiert auf Produktionsregeln, welche eine hierarchische Struktur der Modelle ermöglicht (vgl. Kapitel 3.1.3) [Parish u. Müller (2001)]. Als Eingabedaten können Vektordaten verschiedener Formate verwendet werden. Zudem ist die Verwendung von Geländemodellen und Texturen möglich. Neben der Möglichkeit eigene Produktionsregeln und Einstellungen zu verwenden, bietet CityEngine eine Vielzahl an vorinstallierten Modellierungsumgebungen. So können zufällige Straßennetze und Grundstücksflächen in verschiedenen Stilen erzeugt werden (vgl. Abbildung 3.15).

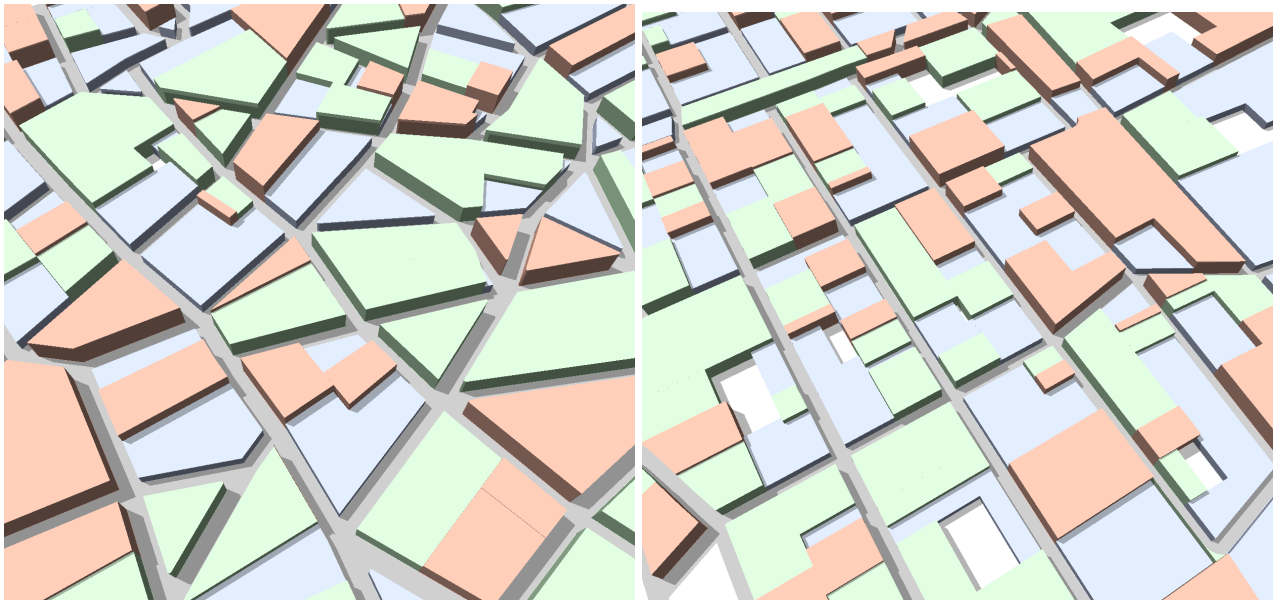


Abb. 3.15 – Zufällig erzeugtes Stadtmodell vom Typ *organic* (links) und *raster* (rechts)

4 Datengrundlage

Die Straßennetzwerke werden auf Basis der, vom Bundesland Vorarlberg, bereitgestellten Daten für ein Teilgebiet von Vorarlberg erstellt. Die Daten lassen sich in vier Blöcke einteilen.

Den ersten Block bilden verschiedene 2D-Vektordaten, wie zum Beispiel Straßen, Tunnel und Brücken. Diese Daten liegen im Shapefile-Format vor, welches aus mehreren Unterdateien besteht. Die Unterdatei *.shp enthält die Geometriedaten, *.shx dient der Indexierung, *.dbf enthält die Sachdaten und *.prj enthält die Georeferenzierung. Die Straßen, welche in Landesstraßen und Ortsstraßen unterteilt sind, liegen als Linien-Objekte vor. Deren Attribute sind in den Tabellen 10.8 und 10.9 aufgeführt. Dabei liegen für die Landesstraßen die sechs Straßenkategorien A für Autobahnen, F für Straßen in Planung, L für Landesstraßen, P für Privatstraßen, S für Schnellstraßen und G für Gemeindestraßen vor. Bei den Ortsstraßen liegen die vier Straßenkategorien O für Ortsstraßen, F für Straßen in Planung, L für Landesstraßen und Z für Zufahrten vor. Bei der weiteren Bearbeitung werden alle Kategorien bis auf die Straßen in Planung verwendet. Die Tunnel liegen ebenfalls als Linien-Objekte vor. Die Attribute der Tunnel sind in Tabelle 10.11 aufgelistet. Die Brücken liegen als Punkt-Objekte vor. Die Attribute, welche direkt mit den Brückenpunkten verknüpft sind, sind in Tabelle 10.6 aufgezeigt.

Den zweiten Block stellen die zusätzlichen Brückeninformationen dar, welche zur Anreicherung der Brückenpunkte dienen. Diese liegen in verschiedenen Excel-Tabellen vor. Die Attribute der Brückeninformationen sind in Tabelle 10.10 aufgeführt.

Der dritte Block beinhaltet Digitale Oberflächenmodelle (DOM) sowie DGM, welche jeweils aus Laserscanning-Befliegungen abgeleitet sind. Diese Höhenmodelle liegen im IMAGINE Image Format vor. Dieses Format speichert die vollständige Datei in *.img und die Kalibrierungsinformationen in *.img.aux.xml. Die originale Punktwolke ist ebenfalls verfügbar.

Im vierten und letzten Block sind die Orthofotos zusammengefasst, welche in den Farbkanälen Rot, Grün und Blau (RGB) aufgenommen sind. Diese Fotos liegen im Enhanced Compressed Wavelet Format vor, welches eine verlustbehaftete Komprimierung von Rastergrafiken darstellt [vgl. ESRI (2015b)]. Das Format besteht aus zwei Unterdateien. Zum einen, die Enhanced Compressed Wavelet Datei *.ecw, welche die Rastergrafik enthält und zum anderen, die sogenannte World-Datei *.eww, welche die Georeferenzierung des Orthofotos definiert.

5 Straßen

Den größten Anteil an Straßennetzwerken stellen, neben den Ingenieurbauwerken, die Straßen dar, welche auf konventionelle Weise auf der Erdoberfläche verlaufen. In einem 3D-Landschaftsmodell sollten Straßen visuell ansprechend dargestellt und semantisch nutzbar sein. Um nun von reinen Straßenmittelachsen zu einer flächigen Darstellung von Straßen zu kommen, bedarf es einiger Arbeitsschritte. Hierzu müssen die jeweiligen Straßenmittelachsen zunächst extrahiert und auf Überlagerungen überprüft werden. Da die Straßenmittelachsen oftmals keine Höheninformationen beinhalten, müssen diese dem jeweiligen korrigierten DGM (vgl. Kapitel 8.2) angepasst werden. Eine Modellierung der Straßenflächen und der Übergang in das CityGML-Schema ermöglichen dann eine semantische Nutzung innerhalb eines 3D-Landschaftsmodells.

5.1 Extraktion der Straßenmittelachse

Die Straßenmittelachsen sind die Basis für mehrere Schritte innerhalb dieser Arbeit, daher ist ihre korrekte Extraktion wichtig. Dabei ist es wichtig Straßenmittelachsen zu identifizieren, welche, abgesehen von Kreuzungen, überlagert vorliegen. In dieser Arbeit werden zwei Typen von Straßenmittelachsen verarbeitet. Zum einen die Landesstraßen und zum anderen die Ortsstraßen, welche jeweils mehrere Straßenkategorien enthalten (vgl. Kapitel 4). Tritt eine Überlagerung von Straßenmittelachsen auf, so werden die Ortsstraßen für den durchgängigen Verlauf der Landesstraßen entfernt. Diese Einstufung folgt der hierarchischen Klassifizierung der Straßen. Ein Beispiel einer Überlagerung von Straßenmittelachsen ist in Abbildung 5.1 dargestellt.

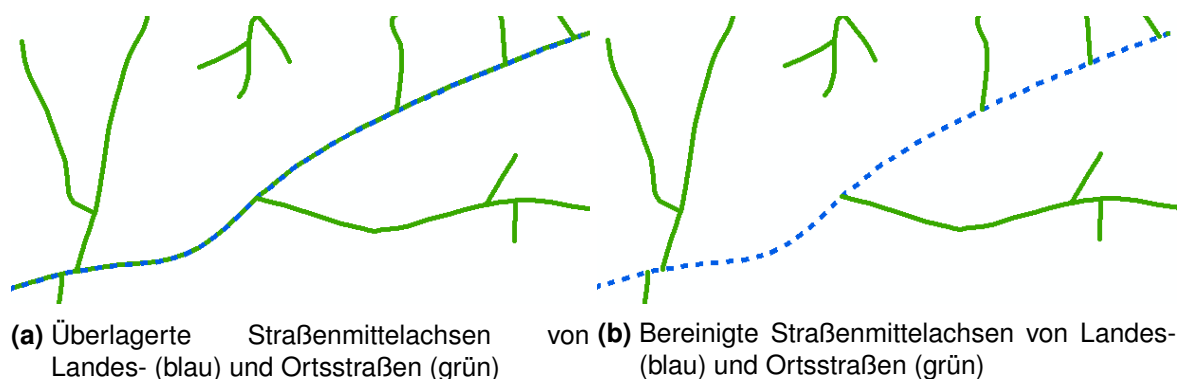


Abb. 5.1 – Beispiel einer Überlagerung von Straßenmittelachsen

Die konkrete Extraktion der Straßenmittelachse erfolgt in dieser Arbeit mit der Software *FME* und dem Workspace *1_Vektor_in_Kachel.fmw* (vgl. Abbildung 10.18). Hierbei werden die Straßen durch Verschneidungen von überlagernden Straßen sowie Brücken und Tunneln bereinigt. Sind überlagernde Straßenmittelachsen jedoch zu weit von einander entfernt, ist es sehr schwierig, diese zu identifizieren. Neben der geometrischen Entfernung der untergeordneten Straßenmittelachse ist es zudem nicht eindeutig, welche der Straßenmittelachsen die korrekte der tatsächlichen Straße ist. Dieses Problem wird im Rahmen dieser Arbeit nicht weiter untersucht.

5.2 Höhenanpassung

Die, der Höhe des DGM angepassten, Straßenmittelachsen dienen neben der Erzeugung der Straßenfläche, auch als Referenz für die Höhenanpassung der Brücken- und Tunnelmittelachsen (vgl. Kapitel 6.2.1 und 7.1). Die konkrete Höhenanpassung erfolgt mit den, zuvor auf Überlagerung überprüften, Straßenmittelachsen sowie einem gemäß Kapitel 8.1 korrigierten DGM. Die Straßenmittelachsen werden an die Höhen des DGM angepasst und erhalten somit neue Höhenwerte. Ein Beispiel einer solchen Anpassung ist in Abbildung 5.2 dargestellt. Dabei sind die einheitlichen Höhen der Straßenmittelachsen für die Darstellung hinzugefügt, da die zu Grunde liegenden Straßenmittelachsen bis zur Höhenanpassung keine Höheninformationen besitzen. Bei der Höhenanpassung sind auch sich kreuzende Verkehrswege zu beachten. Mit Hilfe der Höheninformationen können diese, ausgehend von 2D-Daten, korrekt in die dritte Dimension überführt werden [Oude Elberink (2010)].

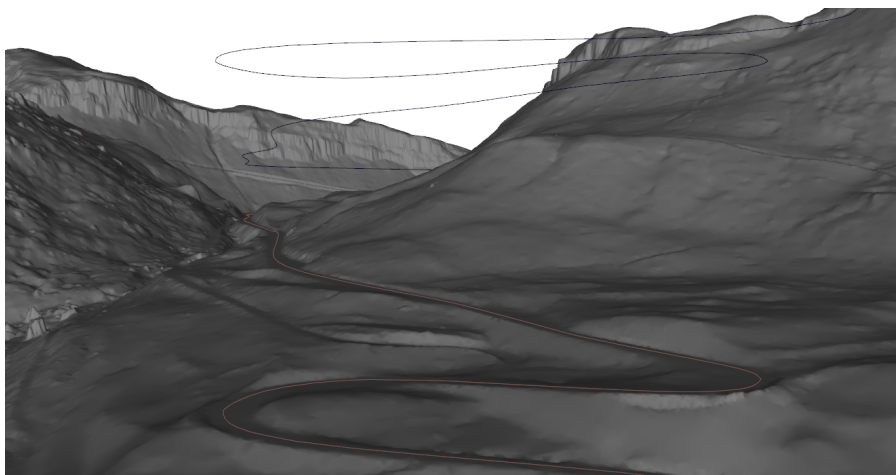


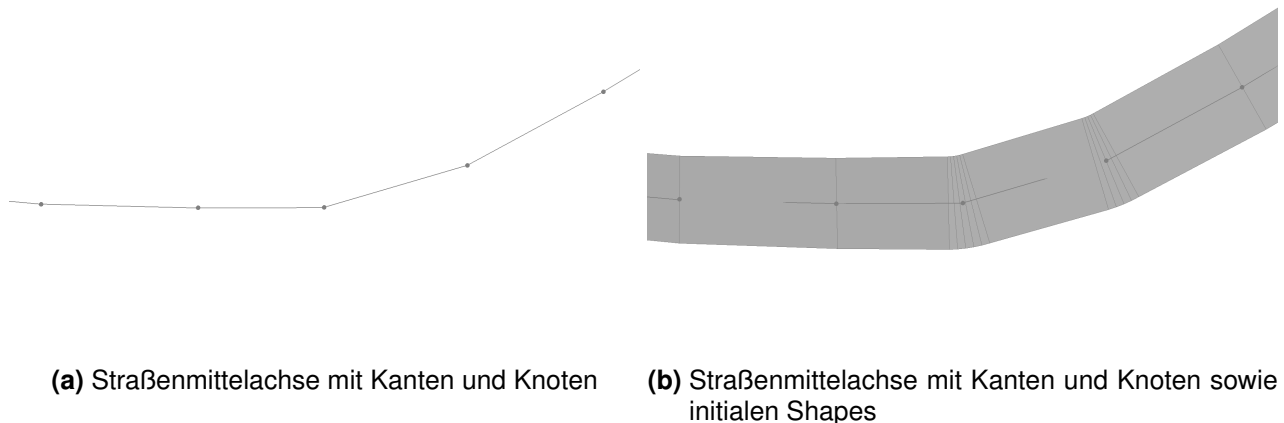
Abb. 5.2 – Beispiel einer Höhenanpassung der Straßenmittelachsen. Straßenmittelachsen mit einheitlichen Höhen (schwarz) und angepasste Straßenmittelachsen (orange)

Die Höhenanpassung wird mit der Software *FME* und dem konkreten Workspace *4_Linienanpassung.fmw* durchgeführt (vgl. Abbildung 10.26). Dabei werden die Höhen durch einen Abgriff vom zugrundeliegenden DGM übertragen. Nach der Anpassung der Höhenlage der Straßenmittelachsen, können diese, für das LOD0 Netzwerk, im CityGML-Schema gespeichert werden (vgl. Kapitel

5.4). Um ebenfalls Straßenflächen erzeugen zu können, dienen die Straßenmittelachsen im nächsten Schritt der Modellierung der Straßenflächen.

5.3 Erzeugung der Straßenflächen

Die Erzeugung der Straßenflächen basiert auf den in ihrer Höhenlage angepassten Straßenmittelachsen. Die konkreten Flächen entsprechen den initialen Shapes, welche durch die Pufferung der Straßenmittelachsen entstehen. Dabei werden für die Kanten im Liniennetzwerk Geraden-Stücke erzeugt. Die Verbindungen der Kanten sind die Knotenpunkte. Verläuft eine Straße absolut gerade, sind keine weiteren Knotenpunkte notwendig. Sind jedoch Richtungs- oder Steigungsänderungen im Straßennetz enthalten, so werden zusätzliche Knotenpunkte benötigt. An diesen Stellen werden die Flächenstücke der Kanten aneinander angepasst, um die Kurve nachzubilden. Ist der Kurvenradius jedoch zu klein, so werden kleinere Teilflächen erzeugt, um die Kurve zu bilden. Dazu ist in Abbildung 5.3 ein Beispiel eines Straßenabschnittes dargestellt. In diesem Straßenabschnitt sind drei leichte sowie zwei starke Richtungsänderungen zwischen Kanten enthalten. Die konkrete Durchführung der Erzeugung der Straßenflächen wird in dieser Arbeit mit der Software *CityEngine* durchgeführt.



(a) Straßenmittelachse mit Kanten und Knoten

(b) Straßenmittelachse mit Kanten und Knoten sowie initialen Shapes

Abb. 5.3 – Beispiel einer Erzeugung der Straßenflächen

Um die Straßenflächen erzeugen zu können, müssen jedoch einige Punkte beachtet werden. Da in einem großen Straßennetzwerk mehrere Straßenklassifizierungen vorkommen, müssen auch verschiedene Straßenbreite modelliert werden. In dieser Arbeit wird zwischen Landesstraßen und allen restlichen Straßen unterschieden. Die Landesstraßen werden mit einer Breite von 7 m modelliert und erzeugt. Die restlichen Straßen, wie zum Beispiel Ortsstraße, Zufahrten oder Privatstraßen, werden mit einer Breite von 5 m Breite modelliert und erzeugt. Neben den initialen Shapes, welche die konkrete Fahrbahn darstellen, werden in der Software *CityEngine* auch Gehwegflächen erzeugt. Diese müssen für die weitere Erzeugung deaktiviert werden.

Ein weiterer Punkt, welcher beachtet werden muss, sind die Kreuzungen, beziehungsweise die Endpunkte von Straßen in Form einer Sackgasse. An diesen Knotenpunkten können verschiedene Kreuzungstypen ausgegeben werden, wobei in dieser Arbeit der Typ *Crossing* gewählt wird, weil so der ursprüngliche Straßenverlauf bewahrt wird (vgl. Abbildung 5.4).

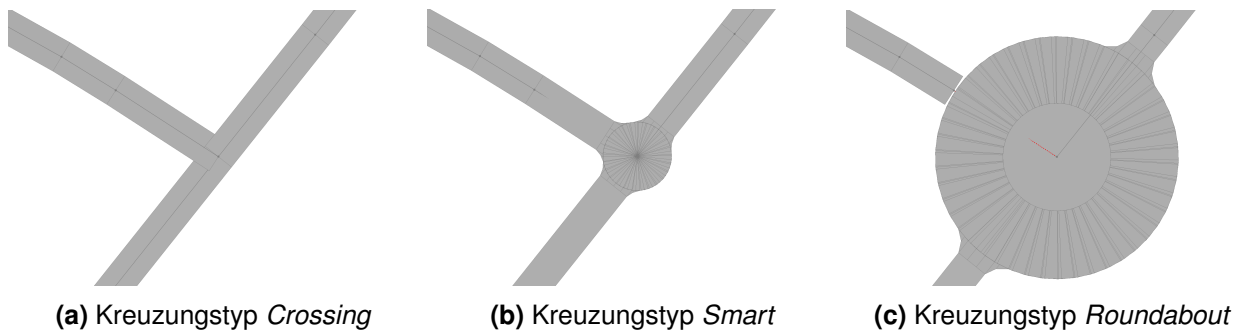


Abb. 5.4 – Verschiedene Kreuzungstypen bei der Straßenflächenmodellierung

Ein konkret erzeugtes Straßennetzwerk, bestehend aus Straßenflächen, ist in Abbildung 5.5 dargestellt. Dabei liegen die Straßen im Bildvordergrund in einem ebenen Gelände. Die Straßen im Hintergrund führen vom ebenen Gelände auf eine Anhöhe.



Abb. 5.5 – Ausschnitt eines Straßennetzwerkes aus Straßenflächen

Bei der Erzeugung der Straßenflächen treten in einigen Bereichen Probleme auf. Diese Bereiche sind zum Beispiel sehr steile Straßenabschnitte, sehr enge Kurven oder die Kombination aus beidem. Das Problem äußert sich dadurch, dass keine durchgehenden Straßenflächen erzeugt werden. Somit können horizontale und vertikale Lücken zwischen den einzelnen Straßenflächen entstehen (vgl. Abbildung 5.6 und 5.7). Um diese Problembereiche korrigieren zu können, kann die Operation *Cleanup Graph* angewendet werden. Hierbei werden Probleme mit überlappenden und nicht verbundenen Straßenmittelachsen gelöst. Jedoch ergeben sich durch die Korrektur neue Probleme. Zum einen wird die Lage und Höhe der Straßen sehr stark verändert, zum anderen, gehen Informationen über zum Beispiel die Straßenbreite verloren. Aus diesem Grund werden die nicht-korrigierten Straßenflächen weiterverarbeitet (vgl. Abbildung 5.7).

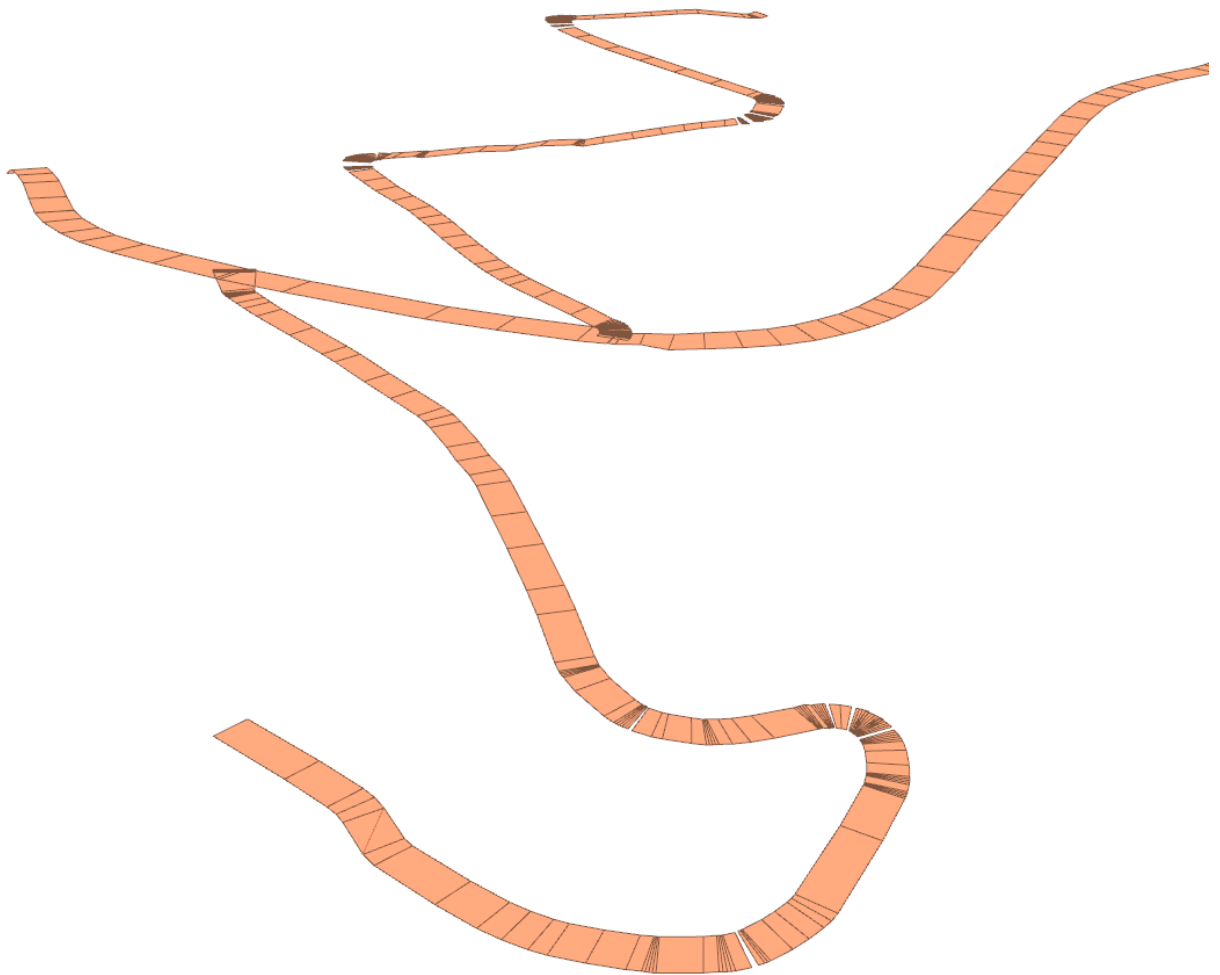
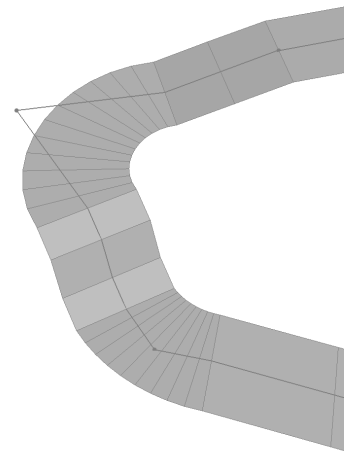
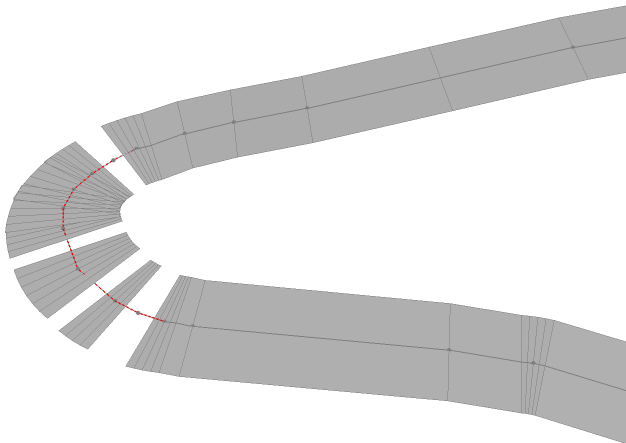


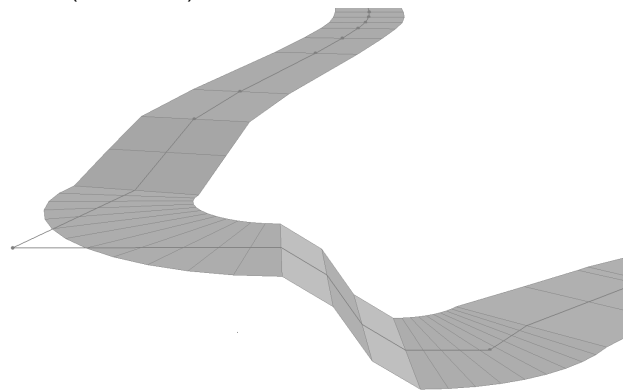
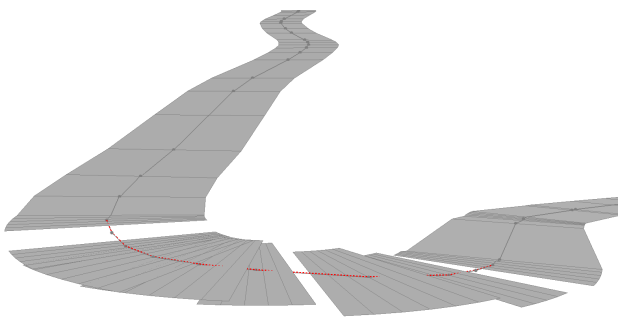
Abb. 5.6 – Ausschnitt eines fehlerhaften Straßennetzwerkes aus Straßenflächen

Die Basis der Straßenflächen, die Straßenmittelachsen, weisen keine Sprünge in der Höhe oder der Lage auf (vgl. Abbildung 5.7). Aus diesem Grund liegt der Fehler der Straßenflächenerzeugung bei der Software *CityEngine*.



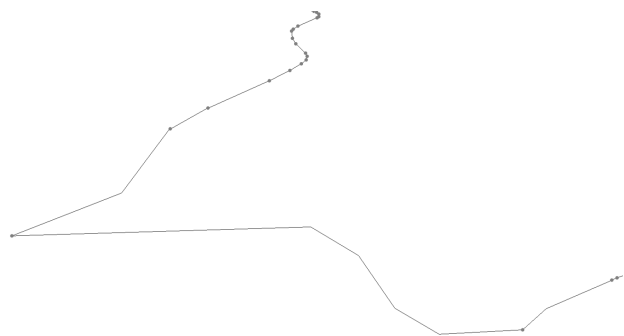
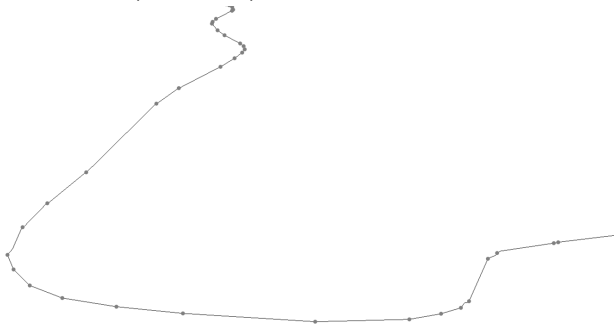
(a) Nicht-korrigierte Straßenflächen und Straßenmittelachse (Ansicht 1)

(b) Korrigierte Straßenflächen und Straßenmittelachse (Ansicht 1)



(c) Nicht-korrigierte Straßenflächen und Straßenmittelachse (Ansicht 2)

(d) Korrigierte Straßenflächen und Straßenmittelachse (Ansicht 2)



(e) Nicht-korrigierte Straßenmittelachse (Ansicht 2)

(f) Korrigierte Straßenmittelachse (Ansicht 2)

Abb. 5.7 – Ausschnitt einer Straße vor und nach einer Korrektur

Weitere Schwierigkeiten treten an Kreuzungen auf. Treffen Straßen mit unterschiedlichen Straßenbreiten aufeinander, so kann es passieren, dass eine Straßenbreite verloren geht. Dann wird im Kreuzungsbereich nur eine einzige Straßenbreite modelliert (vgl. Abbildung 5.8).

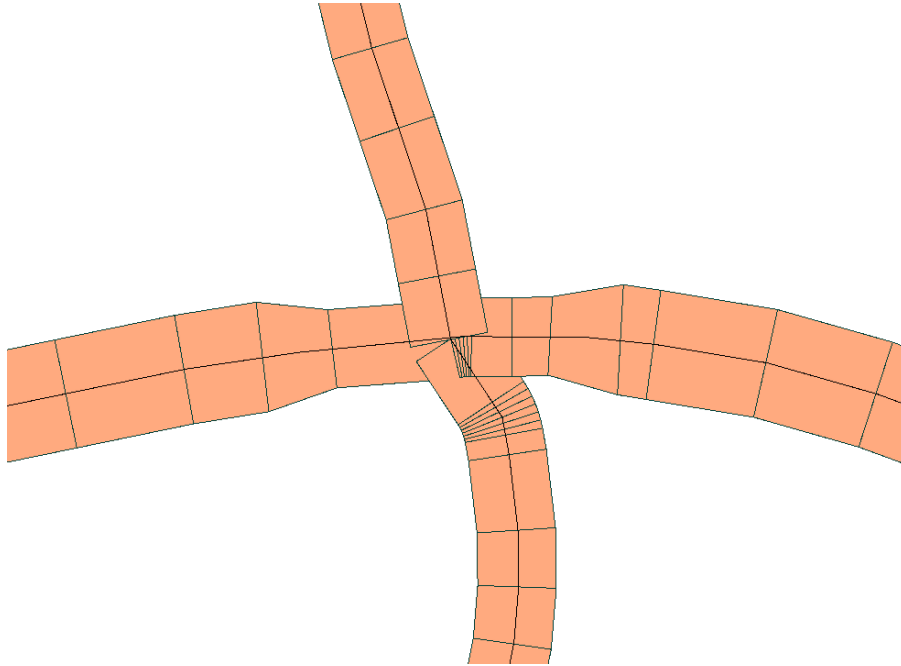


Abb. 5.8 – Ausschnitt einer fehlerhaft erzeugten Kreuzung

5.4 Übergang in das CityGML-Schema

Die Straßen werden in zwei LOD gemäß dem CityGML-Schema gespeichert. Zum einen als Liniennetzwerk im LOD 0, zum anderen als Straßenflächen im LOD 1. Hierbei wird die konkrete Klasse *TransportationComplex* für beide LOD verwendet. Für den Übergang in das CityGML-Schema benötigen die Straßen noch verschiedene spezifische Attribute. Dazu zählen die Attribute *class*, *function* und *usage*. Diese beschreiben die Verkehrsart, den jeweiligen Zweck und die konkrete aktuelle Nutzung (vgl. Tabelle 10.5 und 10.7). Zwei weitere wichtige Attribute sind der *CityGML Lod Name* und die *Feature Role*. Der *CityGML Lod Name* enthält bei dem Liniennetzwerk den Wert *lod0Network* und bei den Straßenflächen den Wert *lod1MultiSurface*. Die *Feature Role* wird jeweils mit dem Wert *CityObjectMember* geschrieben.

Zusätzliche Attribute sind die *gml_id* und die *gml_parent_id*. Die *gml_id* setzt sich aus der Straßenklassifizierung *STR_KATEGORIE* und einem Zählattribut zusammen. Das Attribut *gml_parent_id* beinhaltet den Wert *Vorarlberg*. Um den Aufbau eines CityGML-Schemas zu verdeutlichen, ist in Abbildung 5.9 ein Auszug eines CityGML Dokumentes einer Straße aufgeführt. Die Speicherung sowie die Anreicherung mit Attributen erfolgt mit der Software *FME* und dem Workspace *CityGML_Trans.fmw* (vgl. Abbildung 10.27).

```
<?xml version="1.0" encoding="UTF-8"?>
<core:CityModel xmlns:smil20="http://www.w3.org/2001/SMIL20/" ... >
  <gml:boundedBy>
    <gml:Envelope srsName="EPSG:31254" srsDimension="3">
      <gml:lowerCorner>-52502.494140625 232495.53125 0</gml:lowerCorner>
      <gml:upperCorner>-34997.0146484375 242500.920410156 1757.89947509766</gml:upperCorner>
    </gml:Envelope>
  </gml:boundedBy>
  re:cityObjectMember
  <tran:TransportationComplex gml:id="L_2725">
    <gml:name>Walgaustraße</gml:name>
    ...
    <tran:class>1040</tran:class>
    <tran:lod2MultiSurface>
      <gml:MultiSurface srsName="EPSG:31254" srsDimension="3">
        <gml:surfaceMember>
          <gml:Polygon>
            <gml:exterior>
              <gml:LinearRing>
                <gml:posList>-51844.7236328125 241715.812988281 506.169647216797 ... </gml:posList>
              </gml:LinearRing>
            </gml:exterior>
          </gml:Polygon>
        </gml:surfaceMember>
        ...
        <gml:surfaceMember>
          <gml:Polygon>
            <gml:exterior>
              <gml:LinearRing>
                <gml:posList>-51845.150390625 241705.902832031 505.671325683594 ... </gml:posList>
              </gml:LinearRing>
            </gml:exterior>
          </gml:Polygon>
        </gml:surfaceMember>
      </gml:MultiSurface>
    </tran:lod2MultiSurface>
  </tran:TransportationComplex>
</core:cityObjectMember>
```

Abb. 5.9 – Auszug eines CityGML Dokumentes einer Straße

6 Brücken

Brücken spielen in Straßennetzwerken eine entscheidende Rolle. Sie ermöglichen es, den angeschlossenen Straßen Täler, Gewässer und auch andere Verkehrsnetzwerke zu überqueren. In 3D-Landschaftsmodellen sollten Brücken neben einer, für das Erscheinungsbild wichtigen, visuellen Darstellung, auch semantisch nutzbar sein. Um dies zu erreichen, müssen die vorhandenen Daten mehrere Schritte durchlaufen. Dazu sind in diesem Kapitel Ansätze aufgeführt, um zu semantisch nutzbaren 3D-Brückenmodellen zu gelangen. Ausgehend von der Extraktion einer Brückenmittelachse, über die Höhenanpassung, die Modellierung der Brücken bis hin zum Übergang in das CityGML-Schema, welches für die semantische Nutzbarkeit entscheidend ist.

6.1 Extraktion der Brückenmittelachse

In diesem Abschnitt der Arbeit wird das Konzept zur Extraktion der Brückenmittelachsen vorgestellt. Ausgangspunkt der Extraktion bilden die Brückenpunkte, welche jeweils einer Brücke entsprechen. Für eine Weiterverarbeitung ist es jedoch notwendig, die Brückenmittelachse jeder Brücke zu ermitteln. Dazu werden die Landesstraßen verwendet, welche als Linien-Objekte vorliegen. Das Vorgehen zur Extraktion der Brückenmittelachsen ist in Abbildung 6.1 skizzenhaft dargestellt. Anhand von Entfernungsinformationen der jeweiligen Brücken (vgl. Tabelle 10.6), werden die Brückenlängen ermittelt. Die Brückenlängen werden dann entlang der Straßenmittelachse angelegt. Somit ergibt sich die jeweilige resultierende Brückenmittelachse.

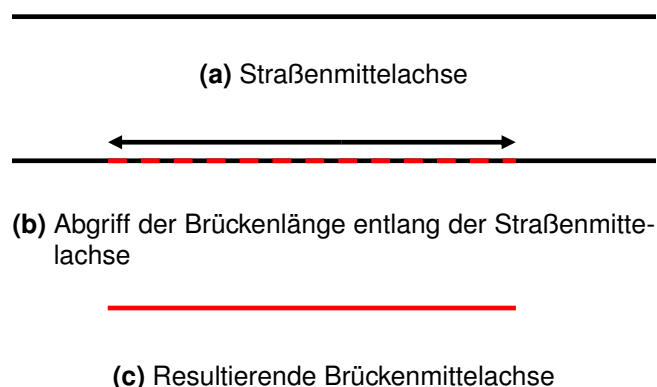


Abb. 6.1 – Konzept der Extraktion von Brückenmittelachsen

Die praktische Umsetzung erfolgt in dieser Arbeit mit der Software FME. Dabei kann die Extraktion in die zwei Blöcke, *Brückenmittelachsen extrahieren und kombinieren* und *Brückenmittelachsen mit Attributen anreichern*, unterteilt werden, welche in Abbildung 6.2 dargestellt sind. Im ersten Block werden die Eingangsdaten validiert und kombiniert. Dies dient als Vorbereitung für die dann folgende Extraktion der Brückenmittelachsen, auf Basis der Brückenpunkte sowie der Landesstraßen. Im zweiten Block werden die zuvor extrahierten Brückenmittelachsen mit zusätzlichen Informationen, gemäß Tabelle 10.10, angereichert. Die Extraktion der Brückenmittelachse ist Teil des Workspaces *1_Vektor_in_Kachel.fmw* (vgl. Abbildung 10.18).

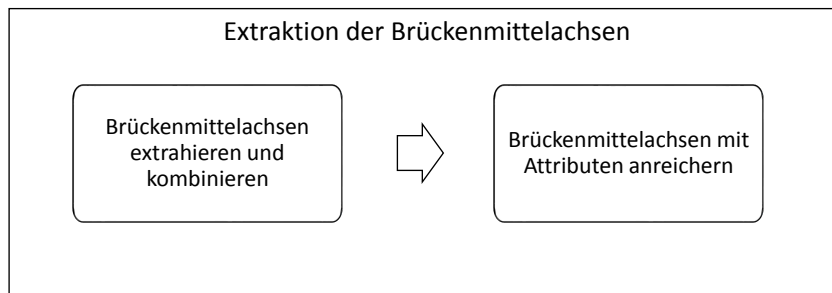
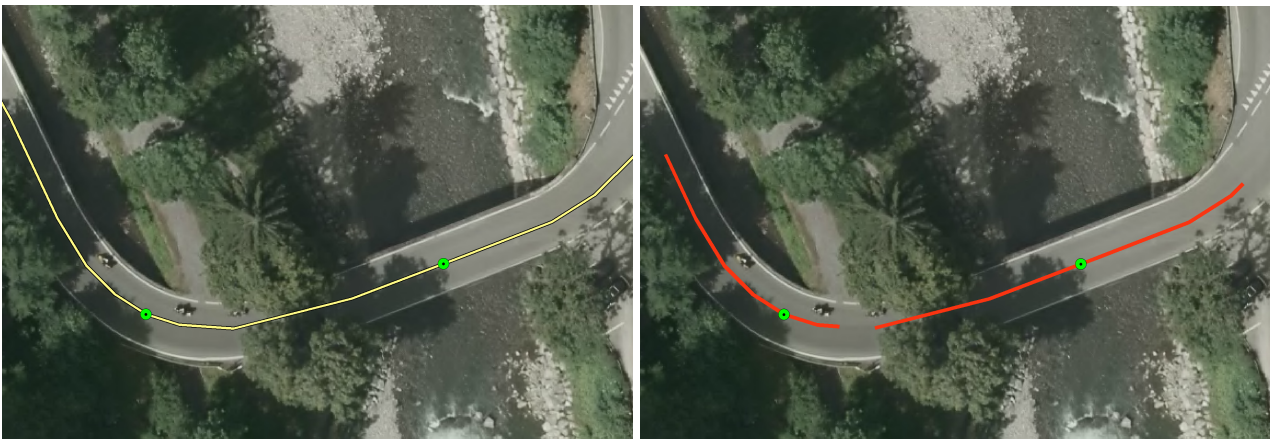


Abb. 6.2 – Unterteilung des Konzeptes der Extraktion von Brückenmittelachsen

Eine in FME durchgeführte Extraktion ist in Abbildung 6.3 dargestellt. Darin sind in der Ausgangssituation zunächst die Straßenmittelachsen und die Brückenpunkte sichtbar. Im rechten Bild sind zwei extrahierte Brückenmittelachsen in rot dargestellt. Es ist deutlich zu erkennen, dass die Brückenpunkte nicht dem Brückenmittelpunkt entsprechen. Die Brückenmittelachsen können demnach nicht durch eine runde Pufferung um die Brückenpunkte extrahiert werden. Zudem würden die Kurven das Ergebnis verfälschen.



(a) Straßenmittelachsen mit Brückenpunkten

(b) Brückenmittelachsen mit Brückenpunkten

Abb. 6.3 – Beispiel einer Extraktion von Brückenmittelachsen

6.2 Höhenanpassung an bestehendes Straßennetzwerk

In diesem Abschnitt werden die Konzepte der Höhenanpassung der Brücken dargelegt. Da die Brücken aus einem Überbau mit der Fahrbahnplatte und dem tragenden Unterbau mit Pfeilern, Bögen oder Gewölben bestehen, müssen verschiedene Höhenanpassungen vorgenommen werden. Dazu ist die Höhenanpassung in die Abschnitte *Brückenkörper* und *Brückenpfeiler* unterteilt, welche im Folgenden erläutert werden.

6.2.1 Brückenkörper

Die Höhenanpassung der Brückenkörper hat das Ziel, den Überbau der Brücken samt Fahrbahnplatte auf das Niveau der angrenzenden Straßen zu bringen. Hierbei wird ein Konzept benötigt, welches eine Höhenanpassung ohne Sprünge an den Übergängen von Brücken und Straßen ermöglicht. Konkret werden hierzu die Höhen der Brückenanfangs- und Brückenendpunkte den angrenzenden Straßenstücken angepasst. Die eigentlichen Brückenachsen zwischen Anfangs- und Endpunkten werden dann linear interpoliert. Die Höhenanpassung erfolgt vor der Modellierung und wird an den Brückenmittellachsen durchgeführt. Dieses Vorgehen ist in Abbildung 6.4 an einem Straßen- und Brückenlängsschnitt skizzenhaft dargestellt.

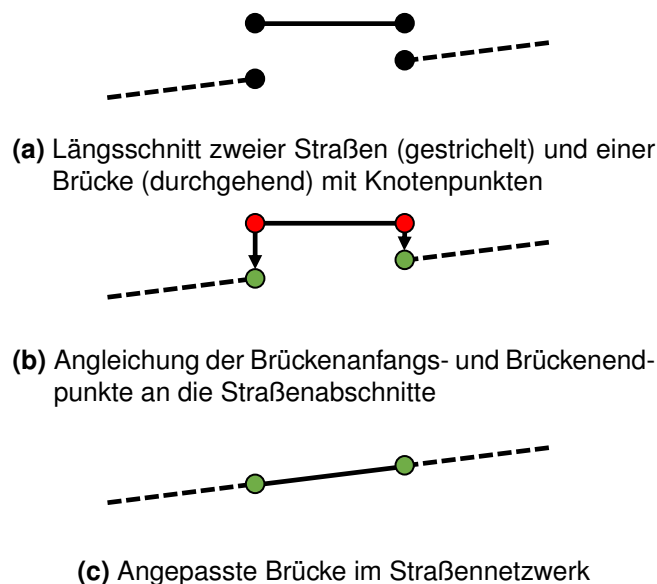


Abb. 6.4 – Konzept der Höhenanpassung der Brückenkörper

Die Höhenanpassung der Brückenkörper wird mit FME durchgeführt. Dabei lässt sich die Höhenanpassung der Brückenkörper in die drei Blöcke *Straßenstücke und Brückenpunkte extrahieren*, *Zuweisung der Nachbarpunkte* und *Interpolation der Brückenhöhen* einteilen. Die Blöcke sind in Abbildung 6.5 aufgeführt. Im ersten Block werden die Brückenanfangs- und Brückenendpunkte sowie die

Straßenanfangs- und Straßenendpunkte extrahiert und gespeichert. Auf Basis dieser Punkte werden im zweiten Block, die nächstgelegenen Straßenpunkte der jeweiligen Brückenpunkte gesucht. Die Brückenpunkte beinhalten nach einer erfolgreichen Suche dann die zur Interpolation benötigten Höheninformationen, welche in Block 3 durchgeführt wird. Hierbei werden die Brückenpunkte mittels einer linearen Interpolation zwischen den extrahierten Höhen verbunden. Die Höhenanpassung der Brückenkörper ist Teil des Workspaces *4_Linienanpassung.fmw* (vgl. Abbildung 10.26).

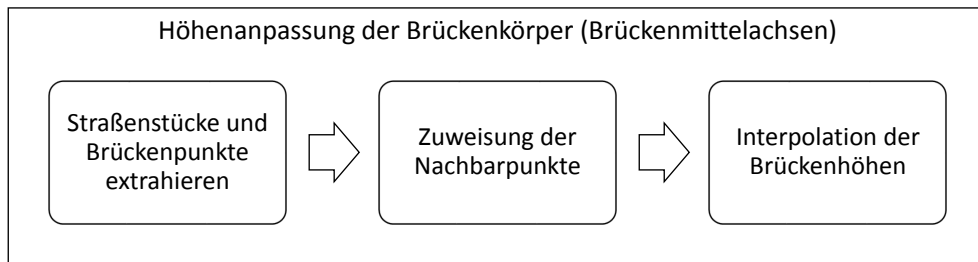
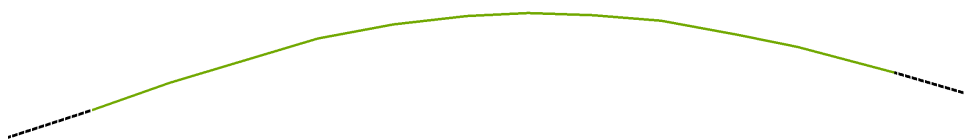


Abb. 6.5 – Unterteilung des Konzepts der Höhenanpassung der Brückenkörper

Der verwendete Ansatz zur Höhenanpassung der Brückenkörper verwendet eine Interpolation zwischen Anfangs- und Endpunkten der Brücken. Dabei erhalten die Brücken eine konstante Steigung zwischen eben diesen Punkten, was in Abbildung 6.6 an einem Beispiel dargestellt ist. Eine individuelle Steigungsänderung kann mit diesem Ansatz nicht umgesetzt werden. Für eine solche Anpassung bedarf es bauwerksspezifischen Höhenangaben, welche als Stützpunkte für die Höhenanpassung verwendet werden können.



(a) Straßenmittelachsen (schwarz) und Brückenmittelachse (rot) mit DGM Höhen



(b) Straßenmittelachsen (schwarz) und Brückenmittelachse (grün) mit angepassten Höhen

Abb. 6.6 – Beispiel einer Höhenanpassung eines Brückenkörper

6.2.2 Brückenpfeiler

Die Höhenanpassung der Brückenpfeiler hat das Ziel, Höheninformationen für eine spätere Modellierung zu extrahieren. Dabei müssen die Pfeiler, Bögen oder Gewölbe den Brückenkörper, also den Überbau, bis zum DGM abstützen. Da die genaue Anzahl und Position der Pfeiler, Bögen oder Gewölbe in den aktuellen Datensätzen nicht ausreichend dokumentiert sind und die Modellierung eine exakte Platzierung nicht unterstützt, ist keine individuelle Höhenplatzierung notwendig. Aus diesem Grund wird die maximale Höhe über Grund je Brücke für alle Pfeiler, Bögen oder Gewölbe übernommen. Die Höhenanpassung erfolgt, wie die Höhenanpassung der Brückenkörper, vor der Modellierung und wird an den Brückenmittelachsen durchgeführt. Die Brückenachsen werden dabei mit den Höheninformationen für die spätere Modellierung der Pfeiler, Bögen oder Gewölbe angereichert. Um den größten Höhenunterschied finden zu können, werden in kleinen Abständen die Höhenunterschiede abgegriffen und dann verglichen (vgl. Abbildung 6.7)

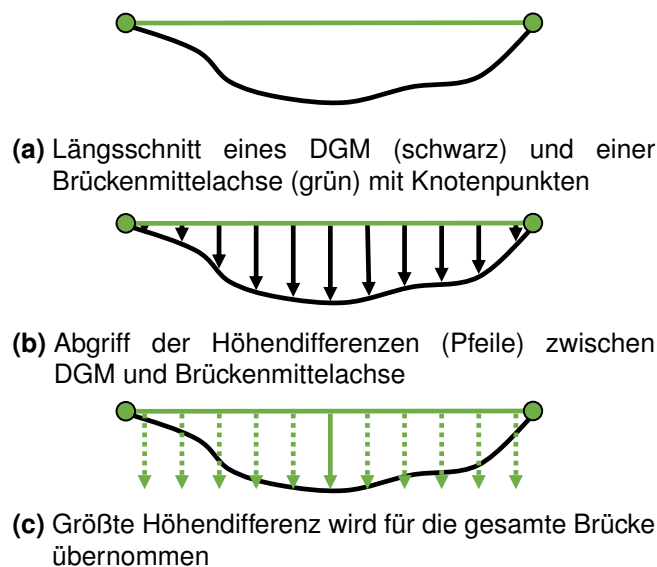


Abb. 6.7 – Konzept der Höhenanpassung der Brückenpfeiler

Die Höhenanpassung der Brückenpfeiler wird mit FME durchgeführt. Hierbei werden die Höhen der interpolierten Brückenmittelachse mit einer auf das DGM gelegten Brückenmittelachse verglichen. Der Abgriff der Höhen erfolgt an den Knotenpunkten der Linien. Der größte Höhenunterschied entlang der Brücke wird dann in der interpolierten Brückenmittelachse gespeichert. Um dieses Vorgehen zu verdeutlichen, ist in Tabelle 10.12 ein Beispiel einer Höhenanpassung aufgeführt. In dieser Tabelle sind die Informationen je Punkt abgebildet. Darin sind neben der Bauwerksnummer die zwei Höhen aufgelistet, welche verglichen werden sollen. Die Differenzen daraus und die größte Differenz sind am Ende der Tabelle dargestellt. Dabei beträgt in diesem Beispiel die größte Differenz 2,00m. Auch die Höhenanpassung der Brückenpfeiler ist Teil des Workspaces *4_Linienanpassung.fmw* (vgl. Abbildung 10.26).

6.3 Modellierung der Brücken

Die geometrische Modellierung der Brücken erfolgt auf Basis der, in der vorgesehenen Bauwerksbreite gepufferten, Linienelemente. Diese Flächen (initiale Shapes) werden in CGA-Regeln weiterverarbeitet um 3D-Modelle zu erzeugen. Mit Hilfe der CGA-Regeln werden die Bauwerke für die Erzeugung in eine hierarchische Struktur unterteilt. Dabei bilden die Bauwerke selbst das Wurzelement in der Hierarchie. Die weitere Unterteilung der Bauwerke kann anhand der Bauteile erfolgen. Hierbei kann zum Beispiel eine Brücke in einen Überbau mit Fahrbahnplatte und Geländer sowie einen Unterbau mit Pfeiler untergliedert werden (vgl. Abbildung 6.12). Zudem können durch zusätzliche Informationen individuelle Eigenschaften von Bauwerken, wie zum Beispiel Brückenbauweise oder Pfeilerhöhen, modelliert werden (vgl. Abbildung 6.8). Die eigentlichen Modellierungsschritte basieren jeweils auf den Primitiven. Dabei werden geometrische Operationen wie Extrusionen, Translationen und Zerteilungen angewendet.

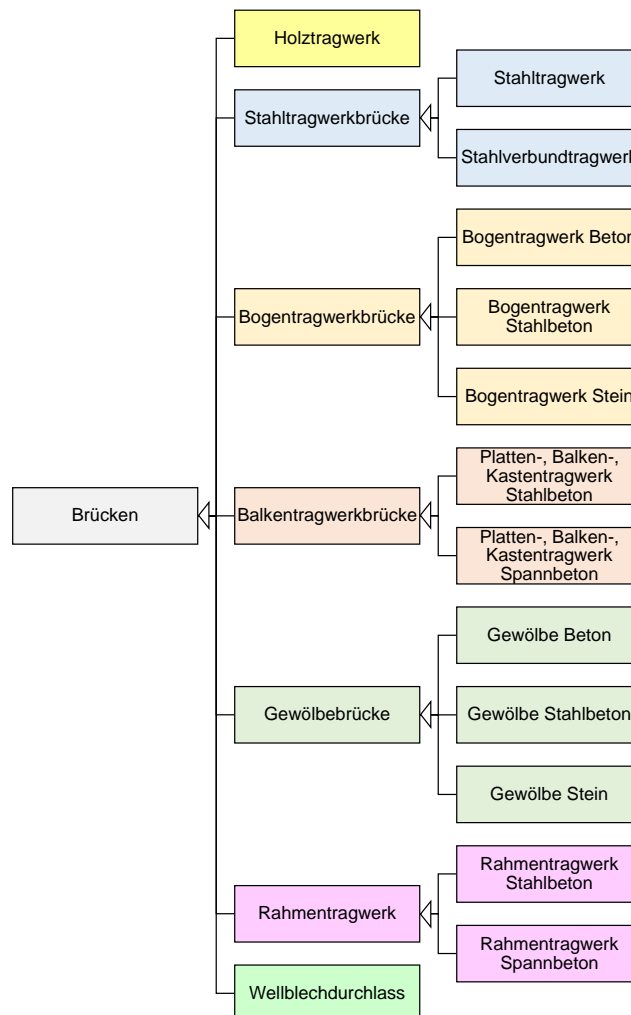


Abb. 6.8 – UML-Diagramm der Brückenbauweisen mit Unterkategorien nach Projektvorgaben Vorarlberg (2014)

Der Testdatensatz, welcher für diese Arbeit verwendet wird, beinhaltet Brücken mit unterschiedlichen Brückenbauweisen. Aus diesem Grund wird für jede Brückenbauweise ein Modellierungskonzept in Form einer CGA-Regel entwickelt. Es lassen sich sieben Brückenbauweisen identifizieren, welche teilweise nochmals untergliedert sind. Diese weitere Unterteilung beschreibt das verwendete Baumaterial, welches für die Modellierung im LOD2 jedoch nicht relevant ist. Aus diesem Grund werden die in Abbildung 6.8 dargestellten übergeordneten Brückenbauweisen *Holztragwerk*, *Stahltragwerkbrücke*, *Bogentragwerkbrücke*, *Balkentragwerkbrücke*, *Gewölbebrücke*, *Rahmentragwerk* und *Wellblechdurchlass* modelliert. Diese übergeordneten Brückenbauweisen werden, in Anlehnung an die gängigen Bauweisen aus Geißler u. Geißler (2014) und FII (2010), modelliert. Dabei hat die Modellierung nicht den Anspruch statisch korrekte 3D-Modelle von Bauwerken nachzubilden. Die für die Modellierung benötigten CGA-Regeln unterscheiden sich je nach Brückenbauweise, jedoch wird im Folgenden nur die konkrete CGA-Regel *Balkentragwerkbrücke* erläutert. Alle weiteren CGA-Regeln werden mit ihren Unterschieden dargelegt. Die Brückenbauweise *Balkentragwerkbrücke* und deren CGA-Regel lässt sich in die zwei Teile *Überbau* und *Unterbau* unterteilen (vgl. Abbildung 6.9). Diese Unterteilung wird mit der Regel `Bruecke` durchgeführt. Dabei wird das zugrunde liegende Shape an die Regeln `Ueberbau` und `Unterbau` übergeben. Neben der Definition von Regeln sind zudem die Attribute wichtig, welche als Eingabeparameter für geometrische Operationen dienen. Diese können entweder direkt in der Regel definiert werden oder mit einem Wert aus einem Attribut, des zugrunde liegenden Shapes, überschrieben werden. In Abbildung 6.9 entsprechen die Attribute `Abstand` und `elevation` dem Wert 15m. Diese dienen der Festlegung von Pfeilerabständen in Fahrtrichtung sowie der Pfeilerhöhen von der Brücke zum DGM. Das Attribut `elevation` wird jedoch bei der konkreten Regelanwendung von den jeweiligen Pfeilerhöhen aus den Shape Attributen überschrieben, welche gemäß Kapitel 6.2.2 ermittelt werden.

```
attr Abstand = 15
attr elevation = 15

@StartRule
Bruecke --> Ueberbau Unterbau
```

Abb. 6.9 – Ausschnitt einer CGA-Regel für Balkentragwerkbrücken (1 von 3)

Der erste Brückenteil *Überbau* wird im weiteren Verlauf der CGA-Regel ebenfalls unterteilt (vgl. Abbildung 6.10). Hierbei besteht der *Überbau*, gemäß der Regel `Ueberbau`, aus den zwei Teilen *Fahrbahnplatte* und *Geländer*. Dabei wird das zugrundeliegende Shape an die Regeln `Fahrbahnplatte` und `Gelaender` übergeben. In der Regeln `Fahrbahnplatte` wird das Shape zunächst um 1m in negative Z-Richtung extrudiert. Nach dieser Operation liegt die *Fahrbahnplatte* als Volumenkörper vor. Danach wird dieser Volumenkörper in seine Begrenzungsflächen unterteilt. Die unterteilten Begrenzungsflächen werden dann in einem festgelegten Schema eingefärbt. Diese Einfärbung dient der semantischen Identifizierung beim Übergang in das CityGML-Schema. Das Farbschema und die Identifizierung der Flächen werden in Kapitel 6.4 erläutert. In der Regel `Gelaender` wird das Shape quer zur Fahrtrichtung in drei Teile aufgeteilt. Dabei wird festgelegt, dass die Geländer 10cm breit sind. Die Fläche zwischen den Geländern wird für diese Regel nicht mehr benötigt und entfernt. Die

10cm breiten Geländergrundflächen werden dann der Regel `Gelaender2` übergeben. Darin werden die Geländergrundflächen in positive Z-Richtung extrudiert und ebenfalls eingefärbt. Nach diesen Operationen liegen die *Gelaender* als eingefärbte Volumenkörper vor.

```

Ueberbau --> Fahrbahnplatte Gelaender

Fahrbahnplatte -->
extrude(-1)

comp(f){
  bottom: color("#0000cd") x| # BLAU für lot_Floor |
  top:     color("#ff0000") x| # ROT für lot_Ceiling |
  front:   color("#00cd00") x| # GRÜN für lot_Wall |
  back:    color("#00cd00") x| # GRÜN für lot_Wall |
  left:    color("#00cd00") x| # GRÜN für lot_Wall |
  right:   color("#00cd00") x # GRÜN für lot_Wall
}

Gelaender --> split(v,unitSpace,0){0.1: Gelaender2 | ~6.8: NIL | 0.1: Gelaender2}

Gelaender2 -->
extrude(world.y, 1)
color("#8b4513") # BRAUN für BI_Wall

```

Abb. 6.10 – Ausschnitt einer CGA-Regel für Balkentragwerkbrücken (2 von 3)

Der zweite Brückenteil *Unterbau* wird im weiteren Verlauf der CGA-Regel ebenfalls unterteilt (vgl. Abbildung 6.11). Hierbei besteht der *Unterbau* aus dem Teil *Pfeiler*. Diese Zuordnung wird durch die Regel `Unterbau` definiert, in der das zugrundeliegende Shape an die Regel `Pfeiler` übergeben wird. In der Regel `Pfeiler` wird zunächst geprüft, ob das Shape mindestens 6m lang ist. Falls ja wird es an die Regel `Pfeiler2` übergeben. Darin wird das Shape in und quer zur Fahrtrichtung aufgeteilt, um die Pfeilerposition unter der *Fahrbahnplatte* festzulegen. Konkret wird definiert, dass die Pfeiler um den Wert des Attributes `Abstand` voneinander entfernt sind. Des Weiteren wird definiert, dass die Pfeiler 2,5m vom jeweiligen Brückenrand entfernt sind. Die resultierenden Pfeilergrundflächen werden der Regel `Pfeiler3` übergeben. In `Pfeiler3` werden die zuvor extrahierten Pfeilergrundflächen um 1m in negative Z-Richtung verschoben sowie um das Attribut `elevation` in negative Z-Richtung extrudiert. In einem weiteren Schritt werden die Pfeiler in ihre Begrenzungsflächen unterteilt und diese wiederum eingefärbt. Danach liegen die *Pfeiler* als eingefärbte Volumenkörper vor.

Um den hierarchischen Aufbau der CGA-Regel *Balkentragwerkbrücke* nochmals zu verdeutlichen, sind die resultierenden Modellteile der einzelnen Unterregeln in Abbildung 6.12 aufgeführt. Darin ist zu erkennen, dass der Ausgangspunkt des Modells das `initiale` Shape ist, welches auf Basis der Brückenmittelachse erzeugt wird. Das konkrete Modell der Regel `Bruecke` setzt sich aus den Modellteilen `Ueberbau` und `Unterbau` zusammen. Dabei ist der Modellteil `Ueberbau` nochmals in `Fahrbahnplatte` und `Gelaender` unterteilt.

```

Unterbau --> Pfeiler

Pfeiler -->
case geometry.du(0,unitSpace)>6: Pfeiler2
else : NIL

Pfeiler2 -->
split(u, unitSpace, 0) { ~(Abstand/2) : NIL | 2.5 : split(v, unitSpace, 0){ ~2.5 :
NIL | 2: Pfeiler3 | ~2.5: NIL} | ~(Abstand/2) : NIL}*

Pfeiler3 -->
t(0,-1,0)
extrude(y, -elevation)

comp(f) {
  bottom: color("#ffff00") x| # GELB für piers_Wall
  top: color("#ff00ff") x| # MAGENTA für piers_Ground
  front: color("#ffff00") x| # GELB für piers_Wall
  back: color("#ffff00") x| # GELB für piers_Wall
  left: color("#ffff00") x| # GELB für piers_Wall
  right: color("#ffff00") x # GELB für piers_Wall
}

```

Abb. 6.11 – Ausschnitt einer CGA-Regel für Balkentragwerkbrücken (3 von 3)

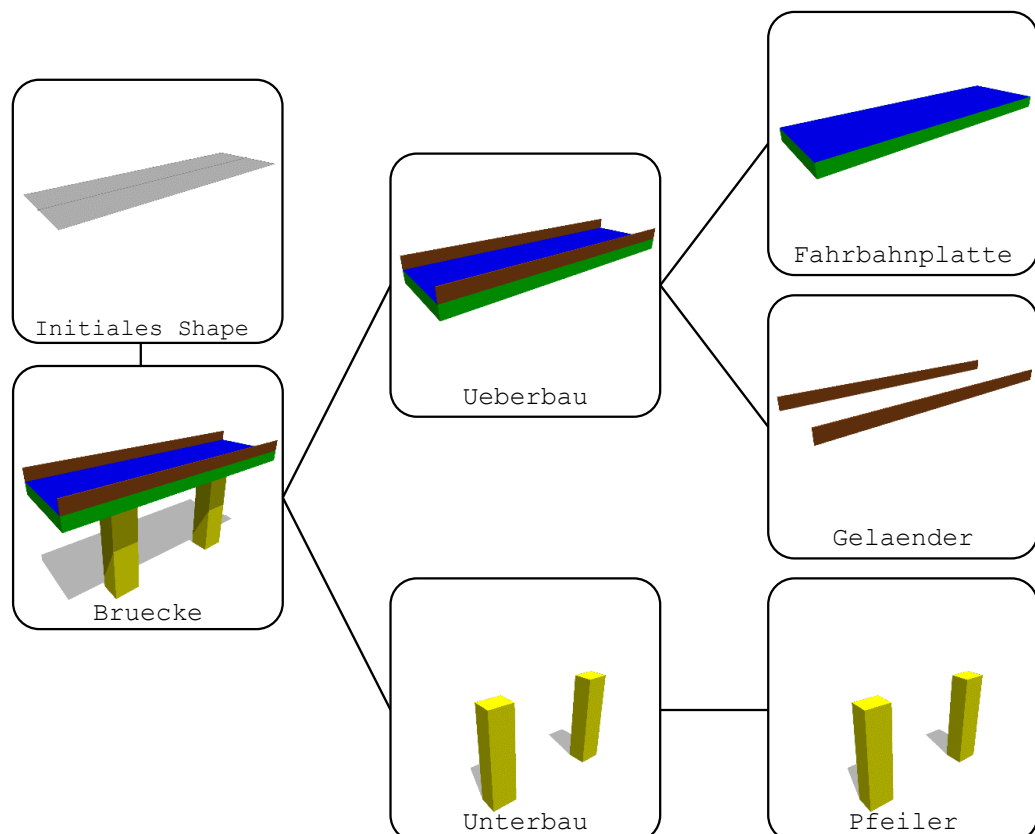


Abb. 6.12 – Hierarchie der CGA-Regel *Balkentragwerkbrücke* ausgehend vom initialen Shape

Bevor die modellierten CGA-Regeln angewandt werden können, müssen die Brückenmittelachsen überarbeitet werden. Dies ist notwendig, weil die CGA-Regeln für zusammenhängende Brückenabschnitte modelliert sind und die Brückenmittelachsen jeweils aus mehreren Linienstücken bestehen. Das geschieht da das, für die Generierung notwendige, Format *Shape* keine Kurven unterstützt. Kurven werden durch mehrere Linienstücke repräsentiert. Somit entstehen für eine Brückenmittelachse, mit Krümmungen oder Steigungen, mehrere initiale Shapes. Um diese Teilstücke der Brückenmittelachsen entfernen zu können, wird eine Generalisierung mit Hilfe der Software CityEngine durchgeführt. Diese Generalisierung mit dem Namen *Simplify Graph*, erzeugt für jede Brückenmittelachse eine neue durchgehende Brückenmittelachse. Das Ergebnis ist dann eine zusammenhängende Brückenmittelachse je Brücke und somit auch jeweils ein initiales Shape. Bei diesem Prozess wird jedoch die Lage und Höhe der Brückenmittelachsen leicht verändert. Nur durch diese Veränderungen können zusammenhängende Brückenmittelachsen entstehen. Die Lage-Veränderungen der Brückenmittelachsen sind dazu in 6.13 als Abweichungen von der ursprünglichen Brückenmittelachse dargestellt. Auf der Abszissenachse sind hierzu die Bauwerksnummern aufgeführt. Die Ordinatenachse entspricht den maximalen und mittleren Abweichungen in Meter. Alle Einzelwerte sind auch in Tabelle 10.13 aufgelistet. Es ist zu erkennen, dass eine Brücke eine um ein Vielfaches größere maximale Abweichung als die restlichen Brücken aufweist. Diese Brücke ist in Abbildung 6.14 nochmals dargestellt. Zudem ist in der Abbildung auch die Brücke mit den geringsten Abweichungen visualisiert. Grund für die starken Abweichungen der oberen Brücke ist die verhältnismäßig enge Kurve. Diese wird bei der Generalisierung stark vereinfacht, was wiederum zu den großen Abweichungen führt. Die Gewölbebrücken sind nicht enthalten, weil sie nicht generalisiert werden.

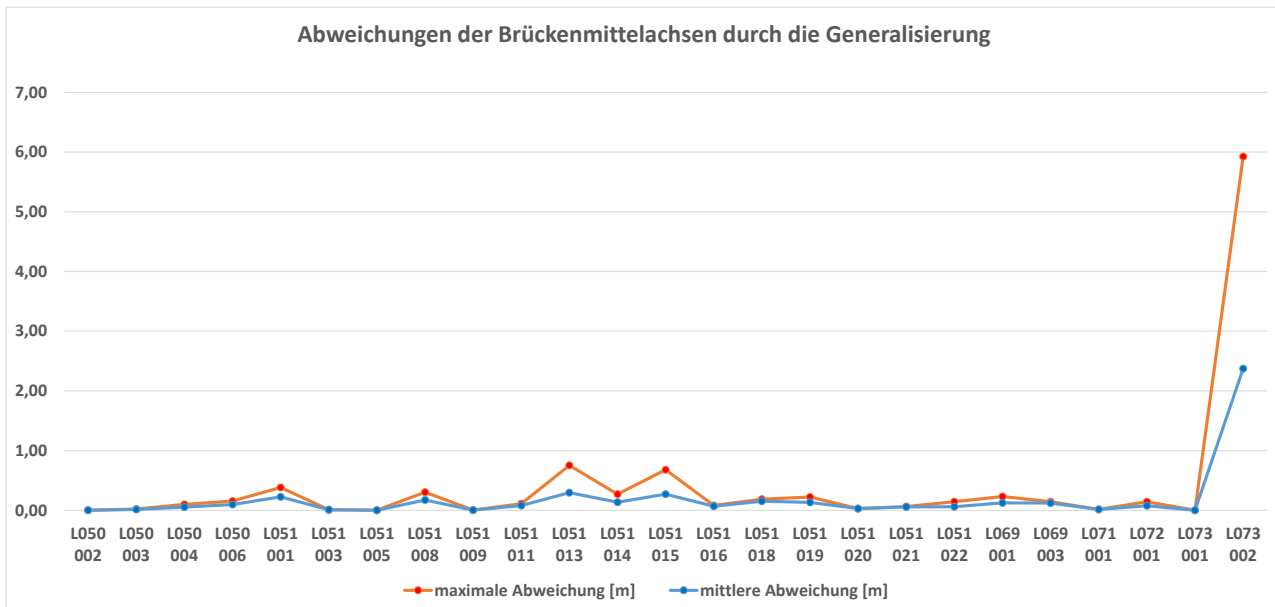


Abb. 6.13 – Abweichungen der Brückenmittelachsen durch die Generalisierung. Abszissenachse mit Bauwerksnummern. Ordinatenachse mit maximalen Abweichungen und mittleren Abweichungen je Brücke in Meter.

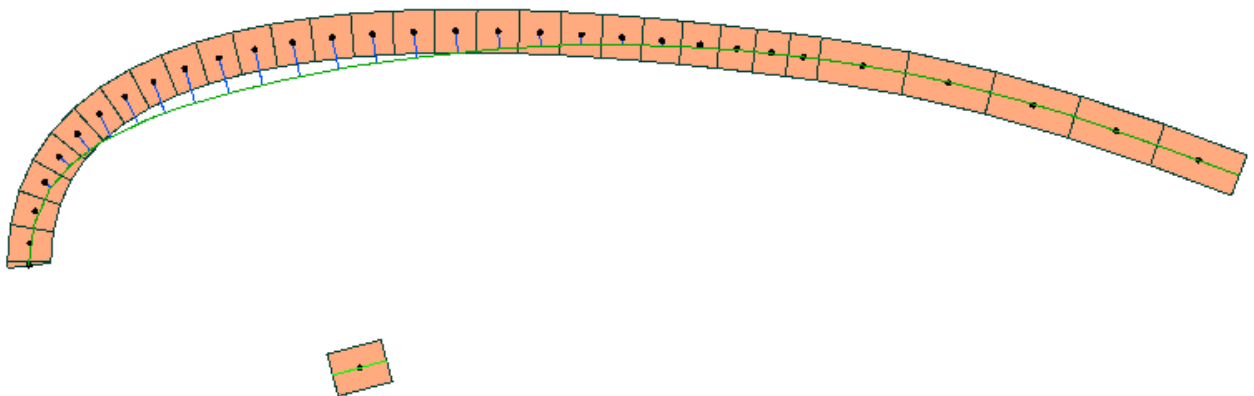


Abb. 6.14 – Brücken mit geringsten (L073 001, unten) und größten (L073 002, oben) Abweichungen. Verlauf der original Brückenmittelachse (Punkte und Flächen), Verlauf der generalisierten Brückenmittelachsen (grüne Linie) sowie Abweichungsvektoren (blaue Linien).

Nach der erfolgreichen Anwendung der CGA-Regel *Balkentragwerkbrücke* auf ein Shape, beziehungsweise auf die zugrunde liegende generalisierte Brückenmittelachse, ergibt sich ein 3D-Modell, wie in Abbildung 6.15 dargestellt. Dabei werden eine Fahrbahnplatte, Geländer am Brückenrand sowie Pfeiler erzeugt. Die Anzahl und Position der Pfeiler basieren hierbei nicht auf realen Daten. Wo und wie viele Pfeiler erzeugt werden, wird in den Regeln festgelegt. Dies liegt an der Tatsache, dass die verwendete Software eine genaue Modellierung der Anzahl und Position der Pfeiler nicht ermöglicht. Zudem fehlt es den zugrunde liegenden Daten an der für die Modellierung benötigten Struktur. Hierzu wären zum Beispiel genaue Angaben zu Bauteilpositionen notwendig. Die CGA-Regel *Balkentragwerkbrücke* gilt stellvertretend für die Brückenbauweisen *Platten-, Balken- und Kastentragwerk Stahlbeton* sowie *Spannbeton*.

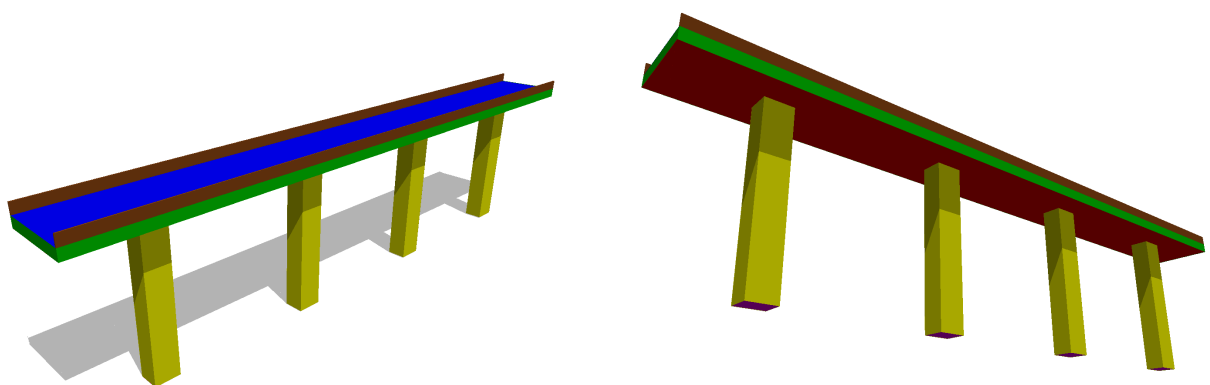


Abb. 6.15 – Eingefärbtes 3D-Modell der CGA-Regel *Balkentragwerkbrücke*

Die CGA-Regel für die Modellierung von Stahltragwerkbrücken heißt *Stahltragwerkbrücke* (vgl. Abbildung 6.16). Diese CGA-Regel, ist der CGA-Regel *Balkentragwerkbrücke* sehr ähnlich. Auf Basis des initialen Shapes werden eine Fahrbahnplatte, Geländer am Brückenrand sowie Pfeiler erzeugt. Der Unterschied zur CGA-Regel *Balkentragwerkbrücke* ist, dass die Pfeiler nicht direkt die Fahrbahnplatte abstützen. Der Kontakt zwischen Pfeiler und Fahrbahnplatte wird hier durch zwei parallel verlaufende Stahlträger hergestellt. Die CGA-Regel *Stahltragwerkbrücke* gilt stellvertretend für die Brückenbauweisen *Stahltragwerk* sowie *Stahlverbundtragwerk*.

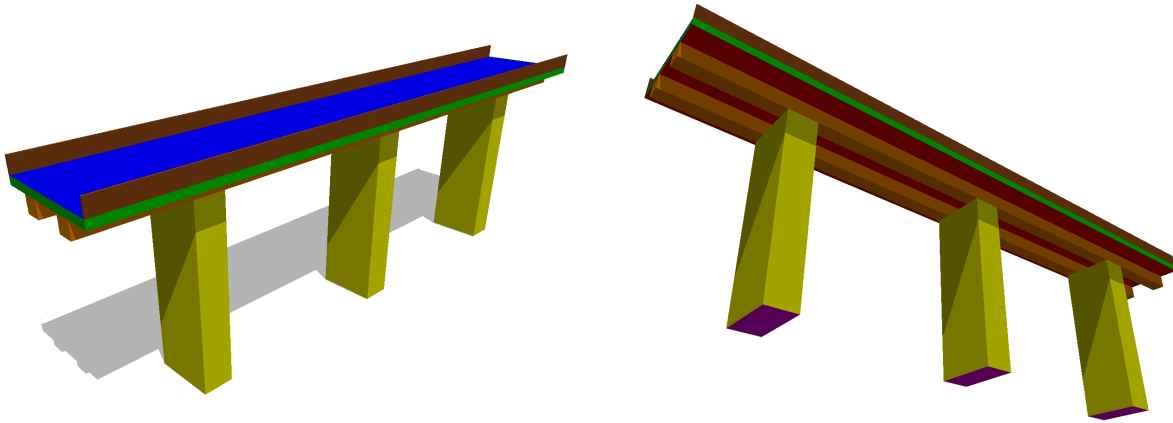


Abb. 6.16 – Eingefärbtes 3D-Modell der CGA-Regel *Stahltragwerkbrücke*

Eine weniger häufig vorkommende Brückenbauweise ist die Holztragwerkbrücke, welche durch die CGA-Regel *Holztragwerkbrücke* modelliert wird (vgl. Abbildung 6.17) [Päffgen (2015)]. Ähnlich der Balkenbrücke werden Fahrbahnplatte, Geländer und Pfeiler erzeugt. Der Unterschied zur CGA-Regel *Balkentragwerkbrücke* ist, dass ausgehend von der geringeren Tragfähigkeit des Baustoffes Holz, mehr Pfeiler benötigt werden [Päffgen (2015)]. Zudem werden die einzelnen Pfeiler durch Querträger versteift.

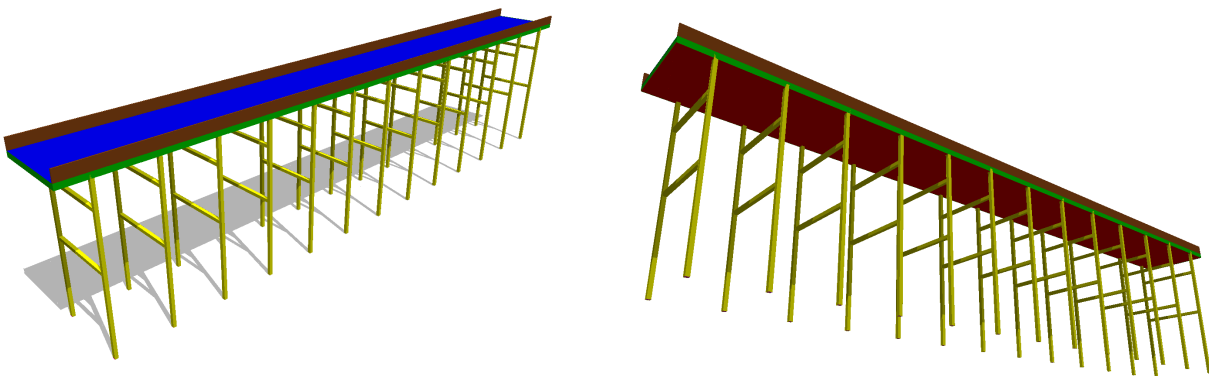


Abb. 6.17 – Eingefärbtes 3D-Modell der CGA-Regel *Holztragwerkbrücke*

Die CGA-Regel für die Modellierung von Bogentragwerkbrücken heißt *Bogentragwerkbrücke* (vgl. Abbildung 6.19). Diese CGA-Regel, ist der CGA-Regel *Balkentragwerkbrücke* nur noch im Überbau ähnlich. Fahrbahnplatte und Geländer am Brückenrand werden wie bei den bisherigen CGA-Regeln erzeugt. Im Unterschied zur CGA-Regel *Balkentragwerkbrücke*, wird der Unterbau, bestehend aus dem Bogen und dessen Stützpfählern, nicht auf Basis des initialen Shapes erzeugt. Der in Abbildung 6.19 gelb eingefärbte Unterbau, wird als 3D-Objekt importiert und auf die Brückenlänge und -breite skaliert. Das 3D-Objekt des Unterbaus wird mit der Software *Google SketchUp* erzeugt und für die Nutzung in *CityEngine* mit *FME* angepasst (vgl. Abbildung 6.18). Hierbei wird insbesondere das Datei-Format von **.dae* in **.obj* geändert.

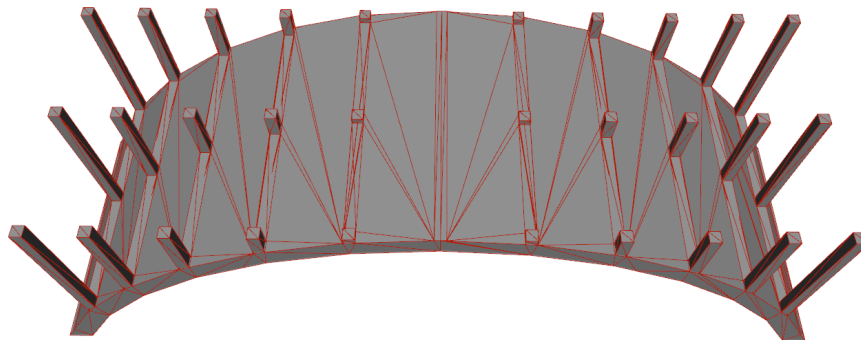


Abb. 6.18 – 3D-Objekt eines Bogentragwerkbrücken-Unterbaus aus *Google SketchUp*

Der Vorteil der Verwendung von vorgefertigten 3D-Objekten ist, dass komplexe Formen, wie die der Bogentragwerkbrücken, nachgebildet werden können. Der Nachteil dieser 3D-Objekte ist, dass sich diese zwar in x- und y-Richtung skalieren lassen, jedoch ist eine Anpassung an Kurven und Steigungen fehlerbehaftet. Grund hierfür ist die Orientierung der 3D-Objekte am jeweiligen Brückenkörper. Die 3D-Objekte werden starr an die Ausgangsrichtung und -steigung der jeweiligen Brücke angepasst. Beinhaltet eine Brücke eine Richtungs- oder Steigungsänderung, so wird das 3D-Objekt zwar so skaliert, dass es die Fahrbahn vollständig abstützt, allerdings bleibt die ursprüngliche Ausrichtung bestehen. Dieses Verhalten trifft auch zu, wenn eine Brücke aus mehreren Shapes besteht und betrifft die CGA-Regeln *Bogentragwerkbrücke*, *Gewölbebrücke*, *Rahmentragwerkbrücke* sowie *Wellblechdurchlass* (vgl. Abbildung 6.23). Die CGA-Regel *Bogentragwerkbrücke* gilt stellvertretend für die Brückenbauweisen *Bogentragwerk Beton*, *Stahlbeton* sowie *Stein*.

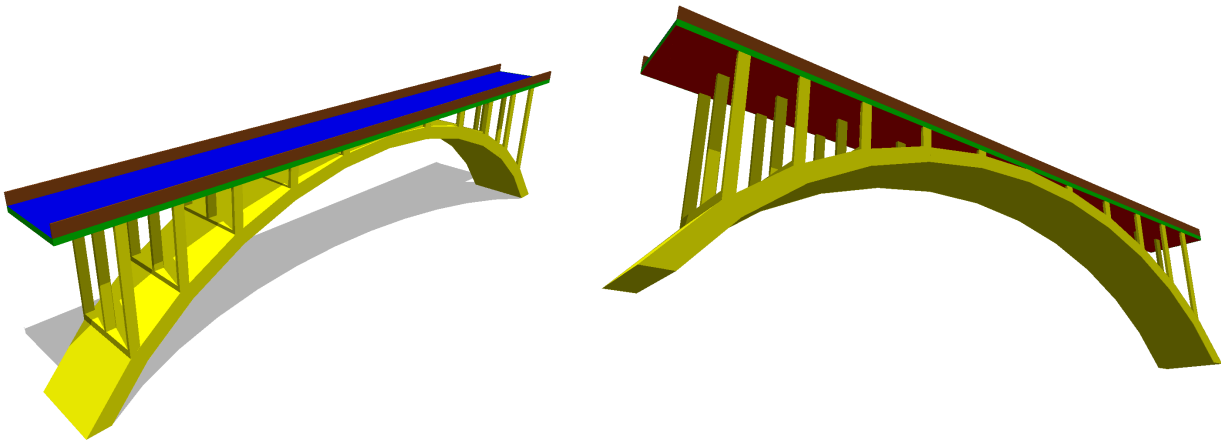


Abb. 6.19 – Eingefärbtes 3D-Modell der CGA-Regel *Bogentragwerkbrücke*

Die Modellierung von Gewölbebrücken ist sehr ähnlich der Modellierung der Bogentragwerkbrücken. Hierbei besteht der Überbau wiederum aus Fahrbahnplatte und Geländer. Der Unterschied zur CGA-Regel *Bogentragwerkbrücke* ist, dass für den Unterbau importierte 3D-Objekte. Dieses ist bei Gewölbebrücken ein massives Gewölbe, welches in Abbildung 6.20 gelb eingefärbt ist. Das Gewölbe wird nach dem Import auf die Brückenlänge und -breite skaliert. Die CGA-Regel *Gewölbebrücke* gilt stellvertretend für die Brückenbauweisen *Gewölbebrücke Beton*, *Stahlbeton* sowie *Stein*.

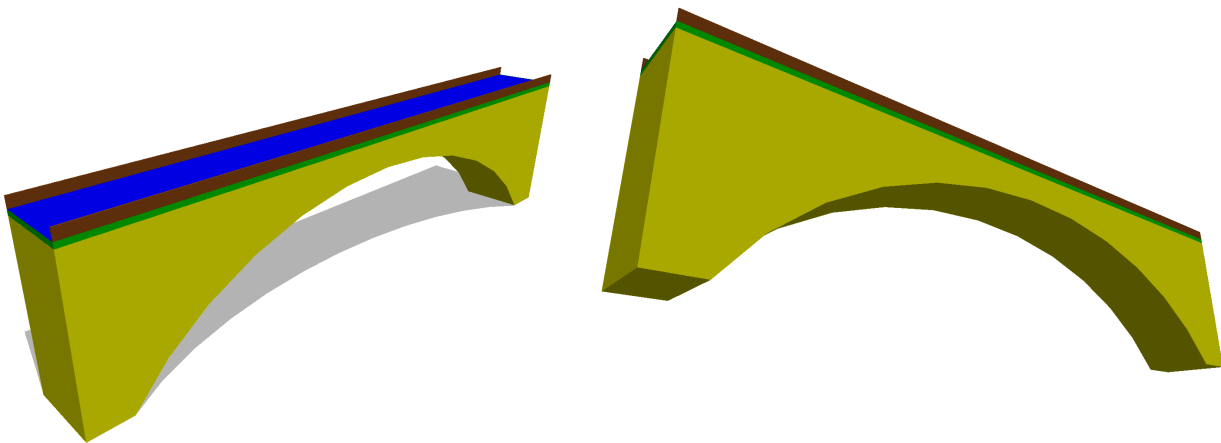


Abb. 6.20 – Eingefärbtes 3D-Modell der CGA-Regel *Gewölbebrücke*

Die CGA-Regel für die Modellierung von Rahmentragwerkbrücken heißt *Rahmentragwerkbrücke* (vgl. Abbildung 6.21). Diese verbindet Komponenten der CGA-Regeln *Balkentragwerkbrücke* und *Bogentragwerkbrücke*. Analog zur Bogentragwerk- und Gewölbebrücke wird der Überbau aus Fahrbahnplatte und Geländer modelliert. Ebenfalls wird für den Unterbau ein 3D-Objekt importiert, welches wiederum an das Shape angepasst wird. Allerdings wird, wie in der CGA-Regel *Balkentragwerkbrücke*, ein Abstand zwischen den schräg verlaufenden Pfeilern festgelegt. Konkret wird definiert,

dass für die erste und die zweite Hälfte eines Shapes jeweils ein 3D-Objekt eines Rahmentragwerkes importiert wird. Die CGA-Regel *Rahmentragwerkbrücke* gilt stellvertretend für die Brückenbauweisen *Rahmentragwerk Stahlbeton* sowie *Spannbeton*.

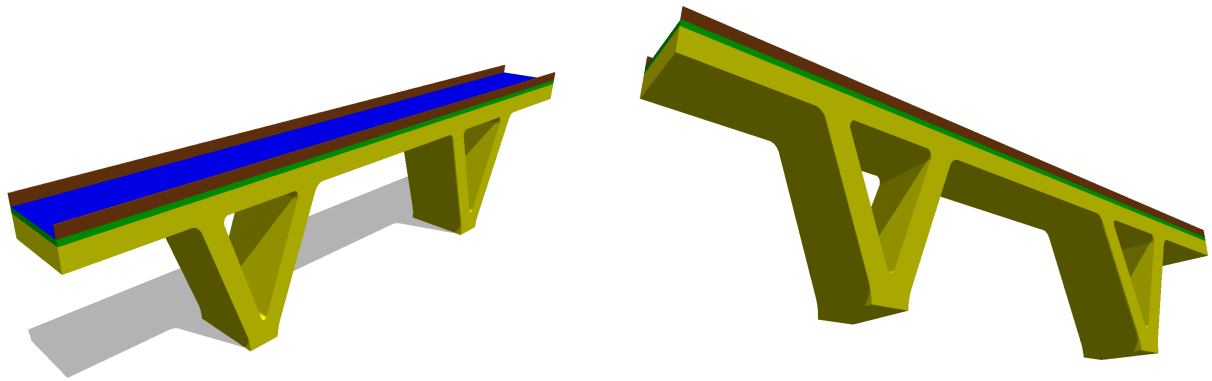


Abb. 6.21 – Eingefärbtes 3D-Modell der CGA-Regel *Rahmentragwerkbrücke*

Die letzte zu modellierende Brückenbauweise ist der Wellblechdurchlass (vgl. Abbildung 6.22). Diese Bauweise wird mit der CGA-Regel *Wellblechdurchlass* modelliert. Hierbei ist der Aufbau der CGA-Regel wieder sehr ähnlich der Bogentragwerk- und Gewölbebrücken. Der Überbau setzt sich aus Fahrbahnplatte und Geländer zusammen. Der Unterbau wird wiederum als 3D-Objekt importiert und an das Shape angepasst.

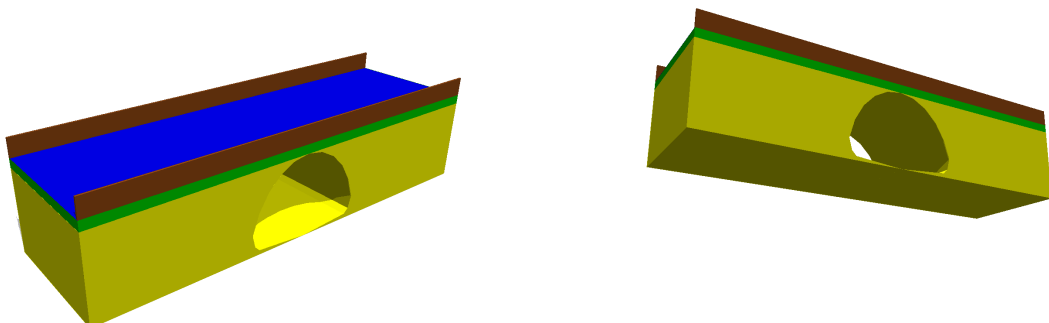


Abb. 6.22 – Eingefärbtes 3D-Modell der CGA-Regel *Wellblechdurchlass*

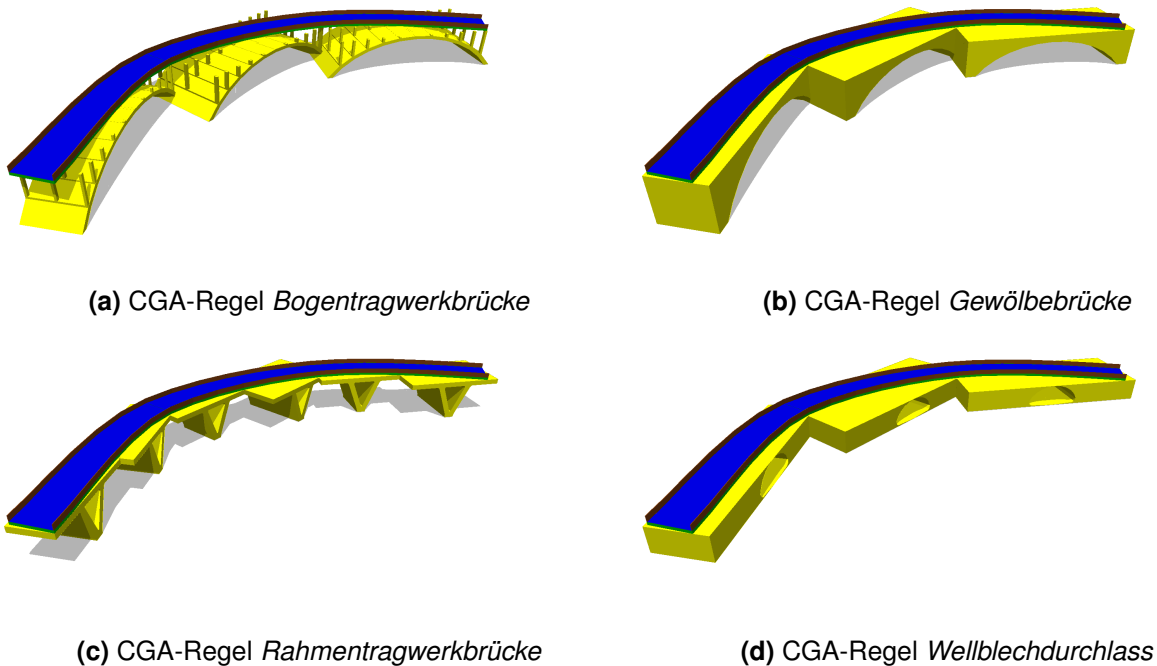


Abb. 6.23 – Fehlerhafte 3D-Brückenmodelle in Kurven

Das in Abbildung 6.23 dargestellte Problem der fehlerhaften 3D-Modelle kann im Rahmen dieser Arbeit nicht behoben werden. Für die konkrete Modellierung werden die abgebildeten Brückenbauweisen trotzdem auf die Brückenlinien angewandt. Jedoch werden bei den fehlerhaften CGA-Regeln (im konkreten Datensatz nur die Gewölbebrücken) die Brückenmittellachsen und somit die initialen Shapes nicht generalisiert. Somit werden jeweils kleine 3D-Objekte importiert und skaliert. Die Überhänge der 3D-Objekte unter den jeweiligen Überbauten werden so minimiert (vgl. Abbildung 6.24). Allerdings leidet die Ästhetik der 3D-Modell unter dieser Minimierung. Nach der Erzeugung der Brückenmodelle folgt der Export aus der verwendeten Software *CityEngine*. Dazu werden die 3D-Modelle im *Collaborative Design Activity (COLLADA)*-Format gespeichert. Hierbei erfolgt die Gruppierung der Flächen jedes 3D-Modells nach den Farben der Begrenzungsflächen (siehe Kapitel 6.4). Zudem werden die 3D-Modelle noch im *Esri Geodatabase*-Format als *geodb_multipatch*-Geometrien gespeichert. Dieser zusätzliche Export der Geometrie ist notwendig, weil die gruppierten 3D-Modelle im *COLLADA*-Format nicht mehr korrekt deaggregiert werden können. Dies ist allerdings notwendig, um alle Klassen der 3D-Modelle im CityGML-Schema abspeichern zu können.

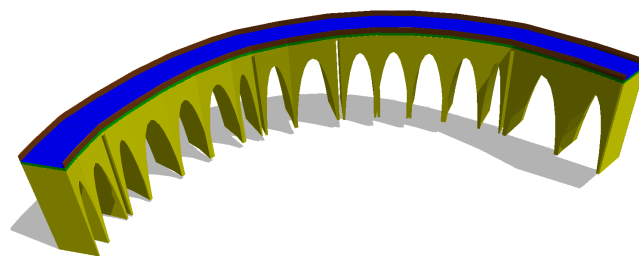


Abb. 6.24 – Anwendung der CGA-Regel *Gewölbebrücke* auf eine nicht-generalisierte Brückenmittellachse

6.4 Übergang in das CityGML-Schema

Um aus den erzeugten 3D-Modellen, semantisch angereicherte Modelle bilden zu können, bedarf es mehr als nur den jeweiligen Objektattributen. Hierbei ist es zum Beispiel für das CityGML-Schema sinnvoll, Bauwerke in Konstruktionselemente und Begrenzungsflächen zu unterteilen (vgl. Abbildung 6.25). Damit die Unterteilung der Modelle durchführbar ist, müssen oftmals die jeweiligen Modellteile oder -flächen mit eindeutigen IDs oder Farben deklariert werden (vgl. Abbildung 6.12 - 6.23). Diese Deklaration ist notwendig, weil die Modellteile und -flächen nach der geometrischen Generierung oftmals keine semantischen Informationen zu ihrer Funktion oder Bedeutung in den 3D-Modellen aufweisen. Auf Basis einer konkreten Deklaration, können die 3D-Modelle dann mit den gewünschten semantischen Informationen weiterverarbeitet und gespeichert werden. Die Tatsache, dass oftmals ein manueller Eingriff in Form einer Deklaration nötig ist, bezieht sich hierbei nicht nur auf die prozedurale Modellierung, sondern auch auf ähnliche Modellierungskonzepte.

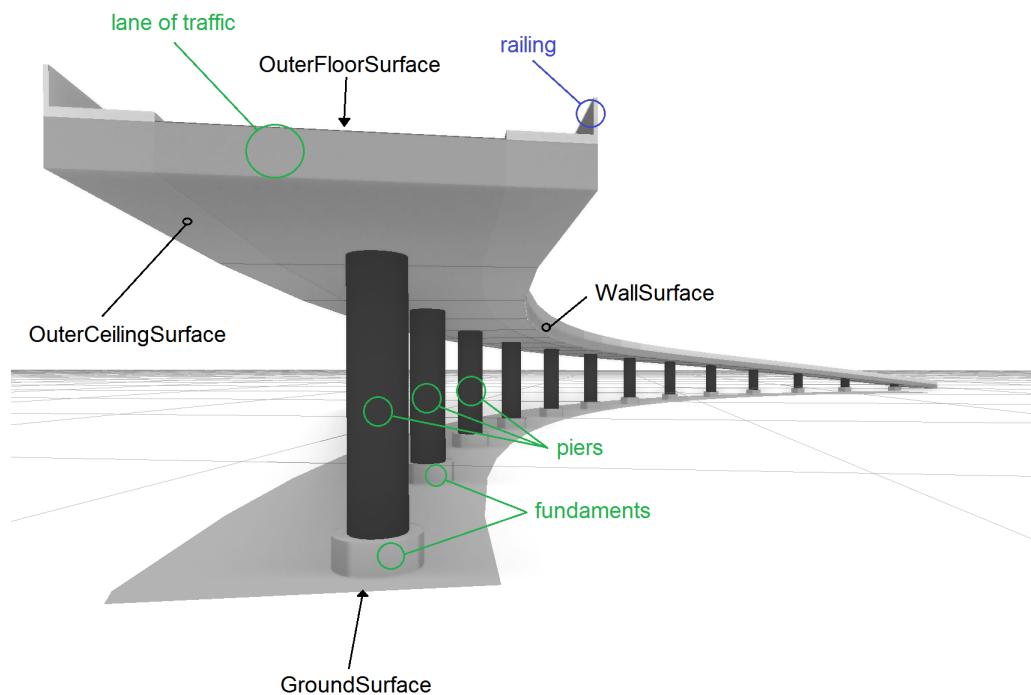


Abb. 6.25 – Unterteilungsmöglichkeiten einer Brücke gemäß dem CityGML-Schema (vgl. Kapitel 3.2.2): Begrenzungsflächen *_BoundarySurface* (schwarz), Konstruktionselemente *BridgeConstructionElement* (grün), Installationen *BridgeInstallation* (blau)

Das in dieser Arbeit verwendete Farbschema zur semantischen Identifikation der Bauteile und Begrenzungsflächen von Brücken besteht aus sieben Farben (vgl. Abbildung 6.26). Hierbei werden die Flächen mit den Farben blau, grün und rot der Klasse *BridgeConstructionElement* beziehungsweise dem konkreten Elementtyp *lane of traffic* zugewiesen. Die Flächen mit den Farben gelb und magenta werden ebenfalls der Klasse *BridgeConstructionElement* jedoch dem konkreten Elementtyp *piers* zugewiesen. Mit der Farbe orange wird auch die Klasse *BridgeConstructionElement* jedoch hier der konkreten Elementtyp *steel beam* definiert. Die Farbe Orange ist nur für die Brückenbauweise *Stahltragwerkbrücke* relevant. Die mit der letzten Farbe braun eingefärbten Flächen gehören zur Klasse *BridgeInstallationElement* und dem konkreten Elementtyp *railing*. Die Farben verweisen wiederum auf die verschiedenen Unterklassen der abstrakten Klasse *_BoundarySurface*. So gehören die Farben grün, gelb, orange und braun zur Klasse *WallSurface*. Blaue Flächen gehören zur Klasse *OuterFloorSurface*, rote Flächen gehören zur Klasse *OuterCeilingSurface* und magenta eingefärbte Flächen gehören zur Klasse *GroundSurface*.

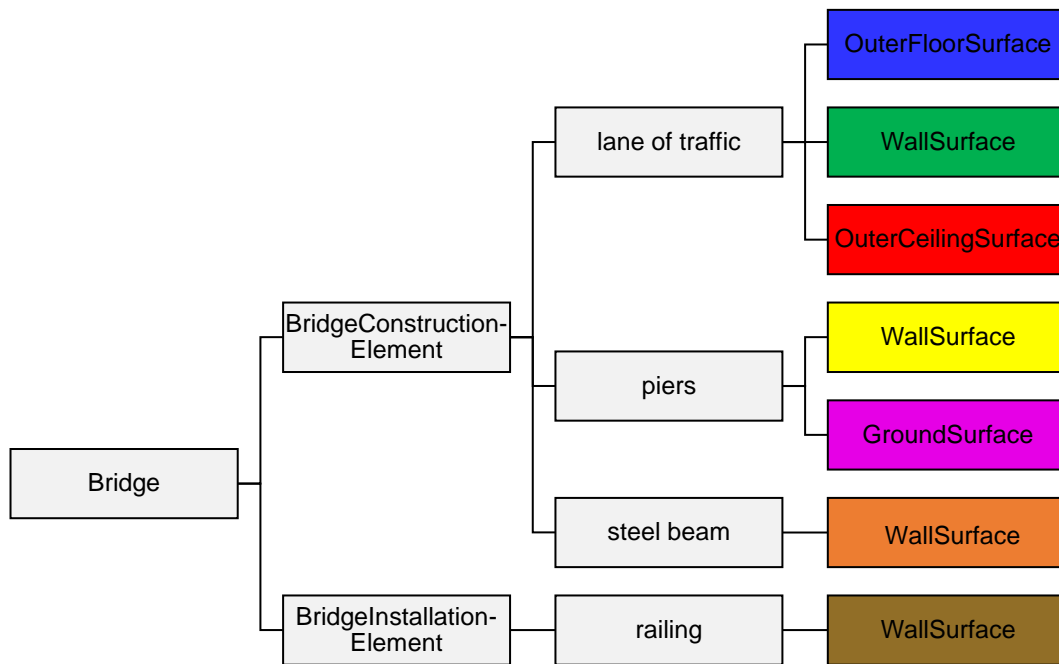


Abb. 6.26 – Hierarchischer Aufbau des Farbschemas zur semantischen Identifikation von Brücken

Die in diesem Farbschema gruppierten Flächen der 3D-Modelle aus Kapitel 6.3, müssen weitere Schritte durchlaufen, um in das gewünschte CityGML-Schema überführt werden zu können. Hierbei sind neben der Identifikation der Farben räumliche Operationen notwendig, um die semantische Struktur herstellen zu können. Dieser gesamte Prozess lässt sich in die vier Blöcke *Deaggregation der 3D-Modelle*, *Elemente vereinigen*, *3D-Modelle mit Attributen anreichern* und *Export im CityGML-Schema* aufteilen (vgl. Abbildung 6.27). Der Übergang in das CityGML-Schema wird mit dem FME Workspace *CityGML_Bridge.fmw* durchgeführt (vgl. Abbildung 10.28).

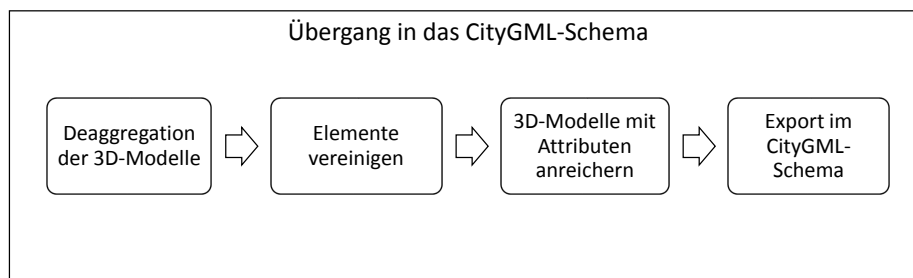


Abb. 6.27 – Unterteilung des Konzepts des Übergangs von Brücken in das CityGML-Schema

Hierbei werden im ersten Block die 3D-Modelle im COLLADA-Format eingelesen, deaggregiert und im *Esri Geodatabase*-Format als *geodb_multipatch*-Geometrien gespeichert. Dieser Deaggregationsvorgang wird im Anschluss nochmals ausgeführt, um alle Aggregationen zu trennen. Dies ist notwendig, um spätere räumliche Operationen zu ermöglichen. Die Informationen zur ursprünglichen farbenbasierten Gruppierung werden bei der Deaggregation als Attribut gespeichert.

Im zweiten Block werden die *geodb_multipatch*-Geometrien jeweils nach ihrer Gruppierung zu Elementen zusammengefasst. Dabei werden zum Beispiel die ursprünglich gelb eingefärbten Flächen der Pfeiler zu einzelnen Pfeilern vereint. Diese Kombination basiert auf der räumlichen Nähe der Flächen. Nach diesem Schritt beinhalten die Flächen, neben den Informationen, zu *BridgeConstructionElement* (*C_BridgeConstructionElement*), *BridgeInstallationElement* (*C_BridgeInstallationElement*) und *_BoundarySurface* (*C_BoundarySurface*), zusätzlich die Informationen zu welchem konkreten Bauteil sie zugeordnet sind (*C_Element_ID*).

Der dritte Block ist für die Anreicherung mit den ursprünglichen Attributen der Brücken verantwortlich. Diese Anreicherung ist notwendig, da durch das Exportieren der 3D-Modelle, deren Objekt-Attribute verloren gehen. Die konkrete Anreicherung wird durch die räumliche Überlagerung der 3D-Modelle, in Form der einzelnen Flächen, mit den gepufferten Brückenmittelachsen erreicht. Zudem werden den einzelnen Brückenflächen die CityGML-konformen Attribute für *C_BridgeConstructionElement*, *C_BridgeInstallationElement* und *C_BoundarySurface* zugeordnet.

Im vierten und letzten Block werden die Brückenflächen für den konkreten Export in das CityGML-Schema vorbereitet. Für die Klasse *Bridge* werden die 3D-Brückenmodelle zuvor noch als *geodb_multipatch*-Geometrien importiert. Die Klasse *Bridge* kann aus geometrischen Gründen nicht aus den Flächen des COLLADA-Formates gebildet werden. Grund dafür sind die in der Modellierung importierten 3D-Objekte, welche sich nicht in das Format *MultiSurface* umformen lassen. Weitere CityGML-spezifische Attribute sind der *CityGML Lod Name* und die *Feature Role*. Diese werden je nach Klasse speziell erzeugt und gespeichert (vgl. Tabelle 6.1). Falls CityGML-Attribute und Attribute der ursprünglichen Brückenmittelachsen die gleiche Bedeutung haben, werden diese einander zugeordnet. So wird zum Beispiel das CityGML-Attribut *citygml_year_of_construction* mit dem Wert des Attributes *Baujahr* der originalen Brückenmittelachsen versehen. Zusätzliche CityGML-spezifische Attribute sind *class*, *function* und *usage*. Im Fall von Brücken enthalten die Attribute die jeweilige Brückenbauweise, den jeweiligen Zweck und die jeweilige aktuelle Nutzung (vgl. Tabelle 10.1 und 10.2).

Klasse	CityGML Lod Name	Feature Role
Bridge	lod2MultiSurface	CityObjectMember
BridgeConstructionElement	lod2Geometry	outerBridgeConstruction
BridgeInstallationElement	lod2Geometry	outerBridgeInstallation
OuterFloorSurface	lod2MultiSurface	boundedBy
WallSurface	lod2MultiSurface	boundedBy
OuterCeilingSurface	lod2MultiSurface	boundedBy
GroundSurface	lod2MultiSurface	boundedBy

Tab. 6.1 – CityGML-Attribute der exportierten 3D-Brückenmodelle

Um die gewünschten 3D-Modelle und 3D-Modellteile der Brücken zu erhalten, müssen die Flächen je nach Klasse aggregiert werden. Die Geometrie der Klasse *Bridge* wird durch die Aggregation der Flächen nach dem eindeutigen Attribut *C_Bauwerksnummer* durchgeführt. Damit werden die Brücken jeweils als ganzes 3D-Modell exportiert. Die Geometrie der Klasse *BridgeConstructionElement* wird hingegen durch die Aggregation der Flächen nach den eindeutigen Attributen *C_Bauwerksnummer*, *C_BridgeConstructionElement* und *C_Element_ID* durchgeführt. Durch das Attribut *C_Bauwerksnummer* wird die jeweilige Brücke, mit dem Attribut *C_BridgeConstructionElement* der jeweilige Konstruktionselementtyp und mit dem Attribut *C_Element_ID* wird das jeweilige Konstruktionselement gewählt. Damit wird gewährleistet, dass zum Beispiel Brückenpfeiler als einzelne Objekte vorliegen. Die Aggregation der Klasse *BridgeInstallationElement* unterscheidet sich nur durch das Attribut *C_BridgeInstallationElement* statt *C_BridgeConstructionElement*. Die Flächen der abstrakten Klasse *_BoundarySurface* und deren Unterklassen *OuterFloorSurface*, *WallSurface*, *OuterCeilingSurface* sowie *GroundSurface* werden nicht aggregiert. Weitere wichtige CityGML Attribute sind die *gml_id* und die *gml_parent_id*, welche die hierarchische Zugehörigkeit der Klassen festlegen. Wie sich die *gml_id* und die *gml_parent_id* je Klasse zusammensetzen, ist in der folgenden Aufzählung aufgeführt:

- *Bridge*
 - *gml_parent_id*: „Vorarlberg“
 - *gml_id*: *C_Bauwerksnummer*
- *BridgeConstructionElement*
 - *gml_parent_id*: *C_Bauwerksnummer*
 - *gml_id*: *C_Bauwerksnummer* & *C_BridgeConstructionElement* & *C_Element_ID*
- *BridgeInstallationElement*
 - *gml_parent_id*: *C_Bauwerksnummer*
 - *gml_id*: *C_Bauwerksnummer* & *C_BridgeInstallationElement* & *C_Element_ID*

- *_BoundarySurface* (Und alle Unterklassen)
 - *gml_parent_id*: *gml_id*: *C_Bauwerksnummer* & *C_BridgeInstallationElement* & *C_Element_ID*
 - *gml_id*: *C_Bauwerksnummer* & *C_BridgeInstallationElement* & *C_Element_ID* & *C_BoundarySurface* & *C_Flächen_ID*

Ein Beispiel einer Brücke (*Bridge*) mit einem untergeordneten Pfeiler (*piers*) und einer untergeordneten Wandfläche (*WallSurface*) ist in nachfolgender Aufzählung dargelegt:

- *Bridge*
 - *gml_parent_id*: „Vorarlberg“
 - *gml_id*: „L051015“
- *BridgeConstructionElement*
 - *gml_parent_id*: „L051015“
 - *gml_id*: „L051015_piers_6“
- *_BoundarySurface* (*WallSurface*)
 - *gml_parent_id*: „L051015_piers_6“
 - *gml_id*: „L051015_piers_6_WallSurface_30“

Dieser hierarchische Aufbau, beziehungsweise die Untergliederung, lässt sich auch im CityGML-Dokument wiederfinden. Dazu ist in Abbildung 6.28 ein Auszug eines CityGML-Dokumentes einer Brücke aufgeführt. Es ist zu erkennen, dass die *WallSurface* über die Beziehung *boundedBy* dem *BridgeConstructionElement* untergeordnet ist. Das *BridgeConstructionElement* ist wiederum mit der Beziehung *outerBridgeConstruction* der Klasse *Bridge* untergeordnet. Diese Zugehörigkeiten können auch in der jeweiligen *gml_id* verfolgt werden.

```

<?xml version="1.0" encoding="UTF-8"?>
<core:CityModel xmlns:smil20="http://www.w3.org/2001/SMIL20/" ... >
  <gml:name>Vorarlberg</gml:name>
  <gml:boundedBy>
    <gml:Envelope srsName="EPSG:31254" srsDimension="3">
      <gml:lowerCorner>-46441.2881000002 235970.440124001 965.766059999994</gml:lowerCorner>
      <gml:upperCorner>-45213.8729858404 236304.813903297 1033.3603</gml:upperCorner>
    </gml:Envelope>
  </gml:boundedBy>
  <core:cityObjectMember>
    <bridge:Bridge gml:id="L051012">
      <gml:name>Gewölbedurchlass</gml:name>
      ...
      <bridge:class>1000</bridge:class>
      <bridge:lod2MultiSurface>
        <gml:MultiSurface srsName="EPSG:31254" srsDimension="3">
          <gml:surfaceMember>
            <gml:Polygon>
              <gml:exterior>
                <gml:LinearRing>
                  <gml:posList>-45798.9470199998 236127.22095 999.694 ... </gml:posList>
                </gml:LinearRing>
              </gml:exterior>
            </gml:Polygon>
          </gml:surfaceMember>
        </gml:MultiSurface>
      </bridge:lod2MultiSurface>
      <bridge:outerBridgeConstruction>
        <bridge:BridgeConstructionElement gml:id="L051012_piers_74">
          <gml:name>Gewölbedurchlass</gml:name>
          ...
          <bridge:class>1000</bridge:class>
          <bridge:lod2Geometry>
            <gml:MultiSurface srsName="EPSG:31254" srsDimension="3">
              <gml:surfaceMember>
                <gml:CompositeSurface>
                  <gml:surfaceMember>
                    <gml:Polygon gml:id="fme-gen-fe9567a5-efb3-4198-8de5-4de978560605">
                      <gml:exterior>
                        <gml:LinearRing>
                          <gml:posList>-45798.7913208014 236120.035888161 998.197 ... </gml:posList>
                        </gml:LinearRing>
                      </gml:exterior>
                    </gml:Polygon>
                  </gml:surfaceMember>
                </gml:CompositeSurface>
              </gml:surfaceMember>
            </gml:MultiSurface>
          </bridge:lod2Geometry>
          <bridge:boundedBy>
            <bridge:WallSurface gml:id="L051012_piers_74_WallSurface_7553">
              <gml:name>Gewölbedurchlass</gml:name>
              ...
              <bridge:class>1000</bridge:class>
              <bridge:lod2MultiSurface>
                <gml:MultiSurface srsName="EPSG:31254" srsDimension="3">
                  <gml:surfaceMember>
                    <gml:CompositeSurface>
                      <gml:surfaceMember>
                        <gml:Polygon gml:id="fme-gen-fe9567a5-efb3-4198-8de5-4de978560605">
                          <gml:exterior>
                            <gml:LinearRing>
                              <gml:posList>-45798.7913208014 236120.035888161 998.19' ... </gml:posList>
                            </gml:LinearRing>
                          </gml:exterior>
                        </gml:Polygon>
                      </gml:surfaceMember>
                    </gml:CompositeSurface>
                  </gml:surfaceMember>
                </gml:MultiSurface>
              </bridge:lod2MultiSurface>
            </bridge:WallSurface>
          </bridge:boundedBy>
        </bridge:BridgeConstructionElement>
      </bridge:outerBridgeConstruction>
    </bridge:Bridge>
  </core:cityObjectMember>

```

Abb. 6.28 – Auszug eines CityGML Dokumentes einer Brücke

7 Tunnel

Ebenso wie Brücken spielen auch Tunnel in Straßennetzwerken eine wichtige Rolle [Haack u. Schaefer (2015)]. Durch sie können Straßen Hindernisse wie Berge schnell unterqueren, ohne lange und teilweise steile Ausweichstrecken bauen zu müssen. Ebenfalls spielen Tunnel bei urbanen Verkehrsnetzen eine entscheidende Rolle, weil sie die Verkehrsbelastung von der Oberfläche in den Untergrund verlagern. In 3D-Landschaftsmodellen sollten Tunnel neben einer, für das Erscheinungsbild wichtigen, visuellen Darstellung, auch semantisch nutzbar sein. Um solche 3D-Tunnelmodelle generieren zu können, bedarf es mehrerer Schritte, welche in diesem Kapitel als Ansätze aufgeführt sind. Eine konkrete Extraktion der Tunnelmittelachsen, ähnlich Kapitel 6.1, ist für die Tunnel in dieser Arbeit nicht notwendig. Die Tunnel liegen bereits als Tunnelmittelachsen im Datenbestand vor und können direkt weiter verarbeitet werden (vgl. Abbildung 7.1).



Abb. 7.1 – Gegebene Tunnelmittelachsen (Rot) mit Orthofoto

7.1 Höhenanpassung an bestehendes Straßennetzwerk

In diesem Abschnitt wird das Konzept zur Höhenanpassung der Tunnel erklärt. Dabei ist es das Ziel, die Tunnel samt Fahrbahn auf das Niveau der angrenzenden Straßen zu bringen. Ähnlich wie bei der Höhenanpassung der Brückenkörper, werden hierzu die Höhen der Tunnelanfangs- und Tunnelendpunkte den angrenzenden Straßenstücken angepasst. Die Höhen des Tunnelverlaufs zwischen Anfang und Ende wird ebenfalls interpoliert. Anders als bei den Brücken, bestehen bei den Tunneln keinerlei Kontrollmöglichkeit der Höhen. Dieses Vorgehen ist in Abbildung 7.2 an einem Straßen- und Brückenlängsschnitt skizzenhaft dargestellt.

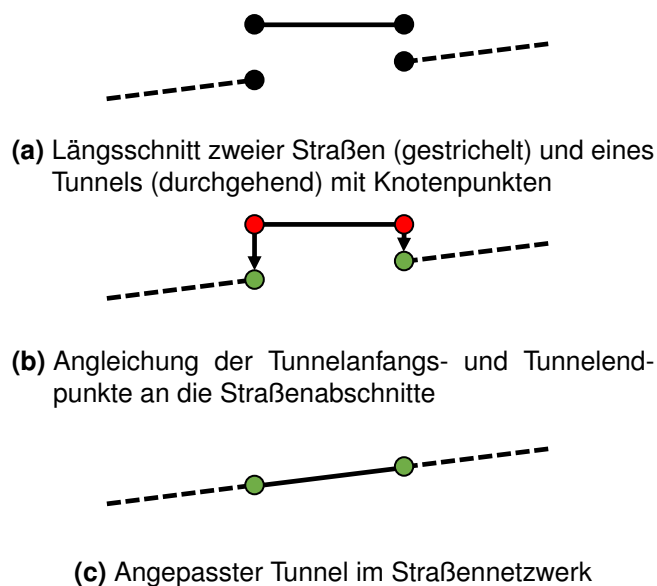


Abb. 7.2 – Konzept der Höhenanpassung von Tunneln

Analog dem Ansatz der Höhenanpassung der Brückenkörper, wird auch die Höhenanpassung der Tunnel durchgeführt (vgl. Kapitel 6.2.1). Daher wird dieses Vorgehen nicht nochmals erläutert. Im Unterschied zu den Brückenkörpern wird bei den Tunneln die Tunnelmittelachse statt der Brückenmittelachse angepasst. Die Höhenanpassung der Tunnelkörper ist Bestandteil des FME Workspace *4_Linienanpassung.fmw* (vgl. Abbildung 10.26).

7.2 Modellierung der Tunnel

Die geometrische Modellierung der Tunnel erfolgt auf Basis der, in der vorgesehenen Bauwerksbreite gepufferten, Tunnelmittelachsen. Diese Flächen (initiale Shapes) werden dann in CGA-Regeln weiterverarbeitet, um 3D-Tunnelmodelle zu erzeugen. Die CGA-Regeln ermöglichen auch die Aufteilung der Bauwerke, hier Tunnel, in eine hierarchische Struktur. Dabei bilden die Tunnel selbst das Wur-

zelement in der Hierarchie. Die Kindelemente bilden bei der Tunnelmodellierung die Begrenzungsflächen (vgl. Abbildung 7.3). Die Tunnelausmaße können mit zusätzlichen Attributen der gepufferten Linienelemente, individuell modelliert werden. Die Modellierung in dieser Arbeit unterstützt nur Tunnel mit rechteckigem Querschnitt gemäß der rechteckigen Tunnelbauweise aus Geißler u. Geißler (2014). Der hierarchische Aufbau und die Funktionsweise von CGA-Regeln werden in Kapitel 6.3 erläutert und hier nicht wiederholt. In der konkreten CGA-Regel *Tunnelkasten*, wird das initiale Shape zu einem Tunnelkasten extrudiert. Dieser wird dann in seine Begrenzungsflächen unterteilt, welche in verschiedene Farben eingefärbt werden (vgl. Abbildung 7.4). Die Einfärbung dient der semantischen Identifizierung beim Übergang in das CityGML-Schema. Dieser Übergang ist ähnlich dem der Brücken, allerdings ist hier die Farbeinteilung anders. Das Innere eines Tunnelmodells, ohne die Abschlussflächen, ist in Abbildung 7.5 dargestellt.

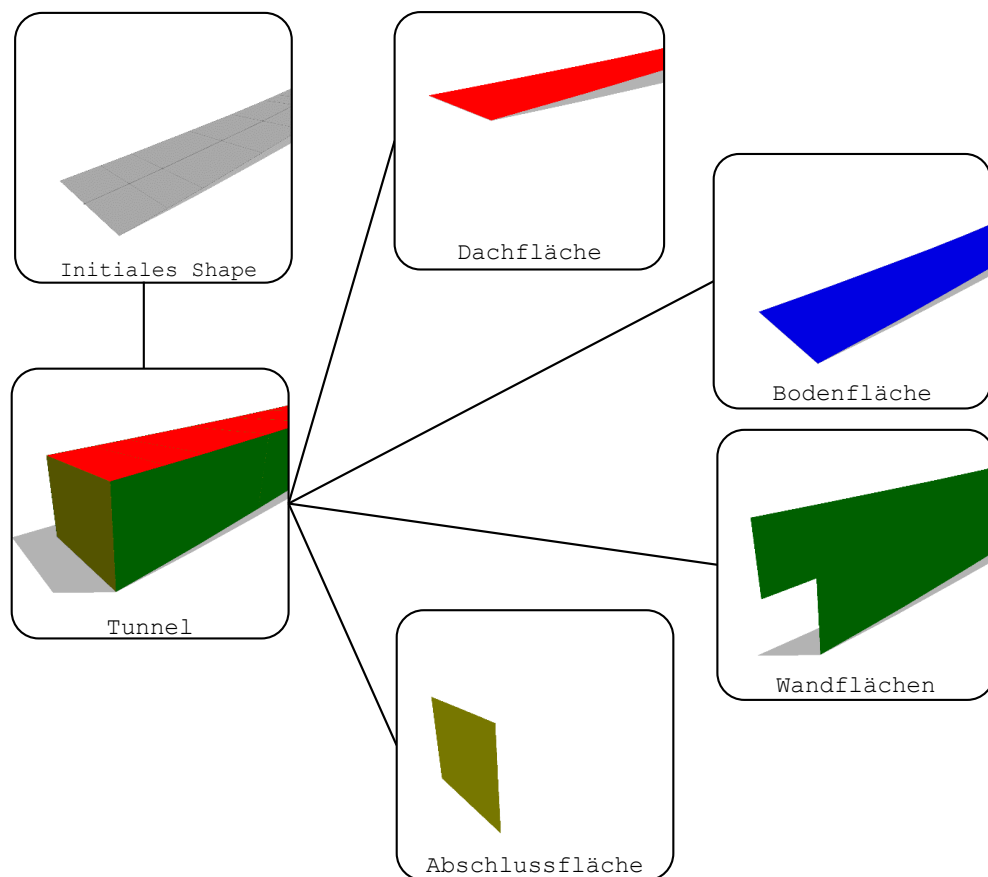


Abb. 7.3 – Hierarchie der CGA-Regel *Tunnelkasten* ausgehend vom initialen Shape

Die Begrenzungsflächen der Tunnel im LOD1 sind zugleich die Außen- und Innenflächen. Eine Unterscheidung der Außen- und Innenflächen gemäß Abbildung 3.9 ist somit nicht möglich. Die Wahl der Flächentypen folgt in dieser Arbeit der Definition des Autors.

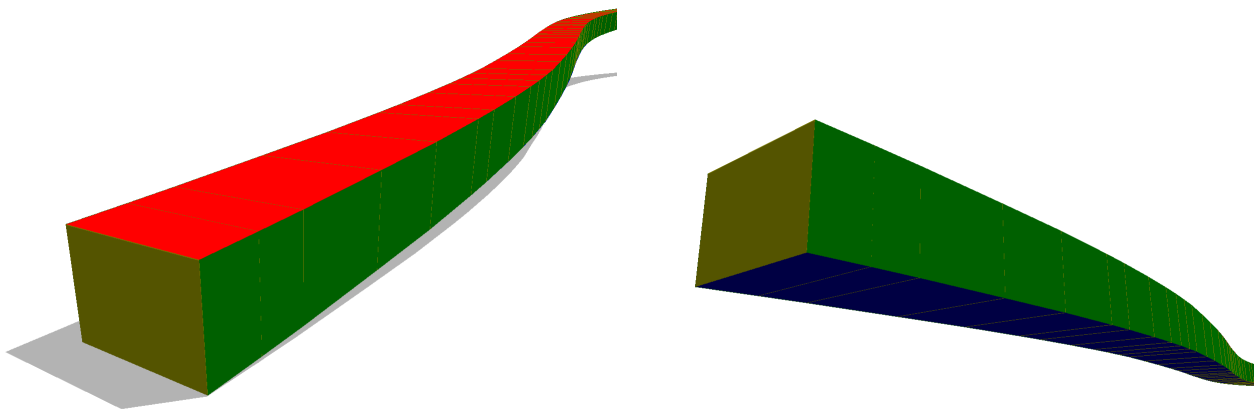


Abb. 7.4 – Ausschnitt eines eingefärbten 3D-Modells der CGA-Regel *Tunnelkasten*

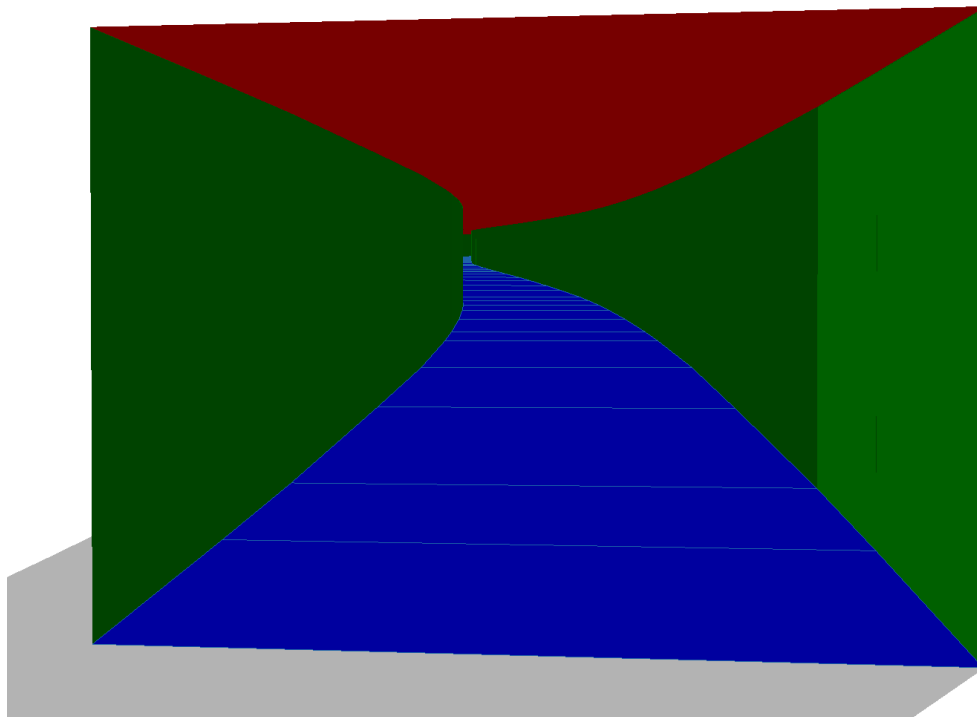


Abb. 7.5 – Ausschnitt eines eingefärbten 3D-Modells der CGA-Regel *Tunnelkasten* ohne Abschlussflächen

7.3 Übergang in das CityGML-Schema

Um die 3D-Tunnelmodelle semantisch anzureichern, können diese gemäß dem CityGML-Schema in Begrenzungsflächen unterteilt werden (vgl. Abbildung 7.6). Hierbei werden den Modellflächen, ähnlich den Brückenmodellen, Farben zugeordnet und diese damit deklariert. Auf Basis dieser Deklaration, können die 3D-Modelle dann mit den gewünschten semantischen Informationen weiterverarbeitet und gespeichert werden. Das in dieser Arbeit verwendete Farbschema zur semantischen Identifikation der Begrenzungsflächen von Tunneln besteht aus vier Farben (vgl. Abbildung 7.6). Die Farben verweisen auf die verschiedenen Unterklassen der abstrakten Klasse *_BoundarySurface*. So gehört die Farbe Grün zur Klasse *WallSurface*. Blaue Flächen gehören zur Klasse *GroundSurface*, rote Flächen gehören zur Klasse *RoofSurface* und gelbe Flächen gehören zur Klasse *ClosureSurface*. Die Klasse *ClosureSurface* entspricht den Abschluss- oder Portalflächen der Tunnelkörper.

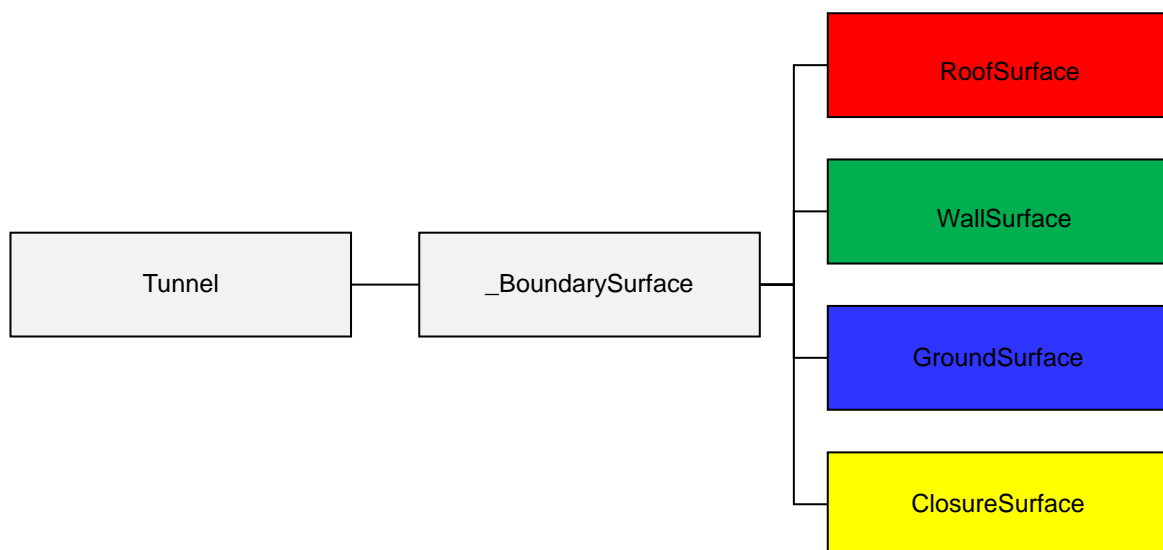


Abb. 7.6 – Hierarchischer Aufbau des Farbschemas zur semantischen Identifikation von Tunneln

Die, in diesem Farbschema, gruppierten Flächen der 3D-Modelle, aus Kapitel 7.2, müssen weitere Schritte durchlaufen, um in das gewünschte CityGML-Schema überführt werden zu können. Hierbei sind neben der Identifikation der Farben räumliche Operationen notwendig, um die semantische Struktur herstellen zu können. Dieser gesamte Prozess lässt sich in die vier Blöcke *Deaggregation der 3D-Modelle*, *3D-Modelle mit Attributen anreichern*, *ClosureSurface extrahieren* und *Export im CityGML-Schema* aufteilen (vgl. Abbildung 7.7). Der Übergang in das CityGML-Schema wird mit dem FME Workspaces *CityGML_Tunnel.fmw* durchgeführt (vgl. Abbildung 10.29).

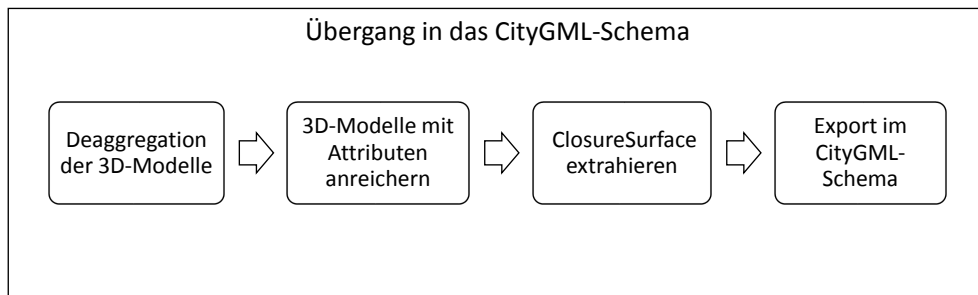


Abb. 7.7 – Unterteilung des Konzepts des Übergangs von Tunneln in das CityGML-Schema

Im ersten Block werden die Tunnelmodelle im COLLADA-Format eingelesen, zweimal deaggregiert und in das *Esri Geodatabase*-Format überführt. Anschließend folgt die Speicherung der einzelnen Flächen als *geodb_multipatch*-Geometrien. Die Gruppierungsinformationen werden im ersten Block je Fläche als Attribut gespeichert.

Im zweiten Block werden die Tunnelmodelle, ähnlich der Brückenmodelle, mit den ursprünglichen Attributen angereichert, welche beim Export aus *CityEngine* verloren gehen. Die einzelnen Flächen werden dann gemäß der Gruppierungsinformationen in die Unterklassen der abstrakten Klasse *_BoundarySurface* unterteilt.

Die *ClosureSurfaces* müssen jedoch im dritten Block zusätzlich bearbeitet werden, weil auch innerhalb der Tunnel *ClosureSurfaces* erzeugt werden. Konkret werden alle *ClosureSurfaces* gelöscht, welche keine Portalflächen sind.

Im vierten Block werden die Tunnel für den Export in das CityGML-Schema vorbereitet und angereichert. So werden auch die Attribute *CityGML Lod Name* und die *Feature Role* erzeugt. Die konkreten Werte dieser Attribute sind abhängig von der jeweiligen Klasse (vgl. Tabelle 7.1). Attribute der Tunnelmittelachsen und CityGML-Attribute mit der gleichen Bedeutung werden verknüpft, sodass die CityGML-Attribute reelle Werte enthalten. Weitere, für das CityGML-Schema, wichtige Attribute sind *class*, *function* und *usage*. Sie beinhalten im Falle von Tunneln die Kategorie (zum Beispiel Verkehr), den Zweck und die aktuelle Nutzung (zum Beispiel Straßentunnel) (vgl. Tabelle 10.3 und 10.4).

Klasse	CityGML Lod Name	Feature Role
Tunnel	lod2MultiSurface	CityObjectMember
RoofSurface	lod2MultiSurface	boundedBy
WallSurface	lod2MultiSurface	boundedBy
GroundSurface	lod2MultiSurface	boundedBy
ClosureSurface	lod2MultiSurface	boundedBy

Tab. 7.1 – CityGML-Attribute der exportierten 3D-Tunnelmodelle

Um die Klasse *Tunnel*, welche komplette Tunnelmodelle enthält, exportieren zu können, müssen die zugehörigen Begrenzungsflächen vereint werden. Dies wird durch eine Aggregation, mit Hilfe des eindeutigen Attributs *C_Bauwerksnummer*, durchgeführt. Alle Flächen der abstrakten Klasse *_BoundarySurface* und deren unter Unterklassen *RoofSurface*, *WallSurface*, *GroundSurface* sowie *CeilingSurface* werden nicht aggregiert.

Da alle Klassen eine *gml_id* und eine *gml_parent_id* aufweisen müssen, werden diese zusätzlich angefügt. Der Aufbau der *gml_id* sowie der *gml_parent_id* ist je Klasse in der folgenden Aufzählung aufgeführt:

- *Tunnel*
 - *gml_parent_id*: „Vorarlberg“
 - *gml_id*: *C_Bauwerksnummer*
- *_BoundarySurface* (Und alle Unterklassen)
 - *gml_parent_id*: *gml_id*: *C_Bauwerksnummer*
 - *gml_id*: *C_Bauwerksnummer* & *C_BoundarySurface* & *C_Flächen_ID*

Um den Aufbau der *gml_id* sowie der *gml_parent_id* bei Tunneln zu verdeutlichen ist in der folgenden Aufzählung ein Beispiel eines Tunnels aufgeführt. Darin ist die Elternklasse *Tunnel* und die Kindklasse (*WallSurface*).

- *Tunnel*
 - *gml_parent_id*: „Vorarlberg“
 - *gml_id*: „L051T01“
- *_BoundarySurface* (*GroundSurface*)
 - *gml_parent_id*: „L051015 “
 - *gml_id*: „L051T01_GroundSurface_548 “)

Die Zugehörigkeit der Klassen sowie die *gml_id* lassen sich auch im eigentlichen CityGML-Dokument erkennen. Dazu ist in Abbildung 7.8 ein Auszug des CityGML-Dokumentes des Tunnels *SchwarzeRüfeTunnel* aufgeführt. Darin ist zu erkennen das die Klasse *RoofSurface* über die Beziehung *boundedBy* der Klasse *Tunnel* untergeordnet ist. Dieser Zusammenhang ist auch in der *gml_id* festgehalten.

```

<?xml version="1.0" encoding="UTF-8"?>
<core:CityModel xmlns:smil20="http://www.w3.org/2001/SMIL20/" ... >
<gml:name>Vorarlberg</gml:name>
  <gml:boundedBy>
    <gml:Envelope srsName="EPSG:31254" srsDimension="3">
      <gml:lowerCorner>-49973.3602905273 236839.66809082 718.1512999999998</gml:lowerCorner>
      <gml:upperCorner>-49001.8192749023 237191.134887695 807.5807999999996</gml:upperCorner>
    </gml:Envelope>
  </gml:boundedBy>
<core:cityObjectMember>
  <tun:Tunnel gml:id="L051T03">
    <gml:name>SchwarzeRüfeTunnel</gml:name>
    ...
    <tun:class>1000</tun:class>
    <tun:lod2MultiSurface>
      <gml:MultiSurface srsName="EPSG:31254" srsDimension="3">
        <gml:surfaceMember>
          <gml:Polygon>
            <gml:exterior>
              <gml:LinearRing>
                <gml:posList>-49194.7015991211 236904.18951416 799.9633999999993 ... </gml:posList>
              </gml:LinearRing>
            </gml:exterior>
          </gml:Polygon>
        </gml:surfaceMember>
      </gml:MultiSurface>
    </tun:lod2MultiSurface>
    <tun:boundedBy>
      <tun:RoofSurface gml:id="L051T03_RoofSurface_185">
        <gml:name>SchwarzeRüfeTunnel</gml:name>
        ...
        <tun:class>1000</tun:class>
        <tun:lod2MultiSurface>
          <gml:MultiSurface srsName="EPSG:31254" srsDimension="3">
            <gml:surfaceMember>
              <gml:Polygon>
                <gml:exterior>
                  <gml:LinearRing>
                    <gml:posList>-49194.7015991211 236904.18951416 799.9633999999993 ... </gml:posList>
                  </gml:LinearRing>
                </gml:exterior>
              </gml:Polygon>
            </gml:surfaceMember>
          </gml:MultiSurface>
        </tun:lod2MultiSurface>
      </tun:RoofSurface>
    </tun:boundedBy>
    ...
  </tun:Tunnel>
</core:cityObjectMember>

```

Abb. 7.8 – Auszug eines CityGML-Dokumentes eines Tunnels

8 Digitales Geländemodell

Das DGM repräsentiert die natürliche Geländeoberfläche ohne Bebauung und Vegetation [Reinhardt (2015)]. Es spielt in einem 3D-Landschaftsmodell eine entscheidende Rolle. Zum einen dient das DGM der Visualisierung des Geländes durch die Darstellung ohne Textur oder mit einer optionalen Texturierung durch Orthofotos. Zum anderen ist das DGM die Grundlage der Höhenanpassung der Vektordaten. Dies sind, in dieser Arbeit, die Straßen-, Brücken- und Tunnelmittelachsen. Damit diese Höhenanpassung korrekt durchgeführt werden kann, muss das DGM in mehreren Bereichen korrigiert werden. Dazu müssen die Straßenbereiche geglättet werden und die Brückenkörper und die Tunnelportale entfernt werden. Nach den Korrekturen muss das DGM nochmals aufbereitet werden, bevor es dann gemäß dem CityGML-Schema gespeichert wird.

8.1 DGM-Korrektur: Straßen

In diesem Abschnitt der Arbeit werden die Konzepte zur DGM-Korrektur im Straßenbereich dargelegt. Eine Korrektur ist nötig, da Geländemodelle trotz hoher Qualität, im Straßenbereich unrealistische Quer- und Längsneigungen aufweisen können. Zum einen spielt die Längsneigung, beziehungsweise die Gradienten, eine wichtige Rolle, welche den Höhenverlauf eines Verkehrsweges in Fahrtrichtung wiedergibt. Diese muss begrenzt werden, damit Verkehrsteilnehmer mit ihren Fahrzeugen, in diesem Fall PKW, LKW sowie forst- und landwirtschaftliche Fahrzeuge, die Straßen befahren können. Zum anderen ist die Querneigung der Straßen wichtig, damit diese befahrbar sind. Die Längs- und Querneigung werden im modernen Straßenbau zumeist so geplant, dass das Oberflächenwasser effektiv und in befestigungsspezifischer Geschwindigkeit abfließt. So ist bei Asphalt- und Betonfahrbahnen ein schneller Abfluss vorgesehen, hingegen bei unbefestigten Verkehrswegen ein langsamer Abfluss erforderlich ist, damit die Fahrbahnerosion begrenzt wird. Dieses Zusammenspiel der Neigungen spielt in dieser Arbeit keine Rolle. Die Längsneigung bezieht sich in diesem Projekt auf das zugrunde liegende DGM. Die Querneigung wird für eine realistische Darstellung der Verkehrswege möglichst minimiert.

8.1.1 Längsneigung

Da die Längsneigung von untergeordneten Straßen teilweise relativ hoch ist, muss auch der Schwellwert einer Korrektur hoch angelegt werden. So sind gerade kleinere Straßen in alpinen Gebieten zum Teil extrem steil. Daher werden die Straßennetze in erster Linie auf Fehler untersucht. Das Konzept der Korrektur betrachtet dabei die Knotenpunkte entlang des Straßenverlaufs. Ausgehend von jedem Knotenpunkt werden die zwei Steigungen der Kanten vor und nach dem jeweiligen Knotenpunkt berechnet. Anhand eines Schwellwertes wird dann geprüft, ob mindestens eine der vier Steigungen jedes Knotenpunktes zu groß ist. Weist ein Knotenpunkt eine zu große Steigung auf, so gilt er als fehlerhaft und wird entfernt. Alle Knotenpunkte, die bestehen bleiben, werden zum Abschluss wieder verbunden, um das DGM zu schließen. Die Betrachtung des weiteren Umfeldes jedes Knotens hat das Ziel, fehlerhafte Gebiete komplett zu bereinigen. Das ist besonders von Nöten, wenn wie in Abbildung 8.1 dargestellt, Abschnitte innerhalb eines fehlerhaften Gebietes eine akzeptable Steigung aufweisen. Diese würden bei einer lokalen Betrachtung im Datensatz bestehen bleiben und diesen damit verfälschen.

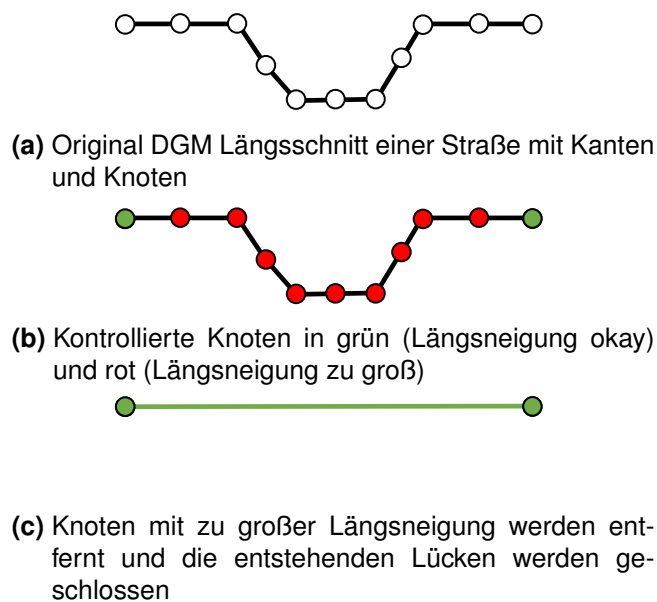


Abb. 8.1 – Konzept der Korrektur der Längsneigung

Die Korrektur der Längsneigung erfolgt auf Basis der gepufferten Straßenmittelachse mit der Software FME und dem Workspace *2_DGM_Korrektur.fmw* (vgl. Abbildung 10.20). Dabei werden die Straßenränder betrachtet, die je nach Straßentyp in verschiedenen Breiten erzeugt werden. Diese werden vor der eigentlichen Prüfung der Längsneigung auf das DGM gelegt, um die aktuellen Höheninformationen zu extrahieren. In den, in Abbildung 8.2, dargestellten Blöcken, erfolgt die Prüfung des rechten und des linken Fahrbahnrandes parallel. Darin werden jeweils zwei Knotenpunkte vor und nach jedem Punkt betrachtet. Liegt die Steigung mindestens einer der vier Kanten zwischen die-

sen Punkten oberhalb eines Grenzwertes, so gilt der mittlere Punkt als fehlerhaft. Nach der Prüfung jeder Seite werden die entstandenen Lücken in den Linien geschlossen (vgl. Abbildung 8.5).

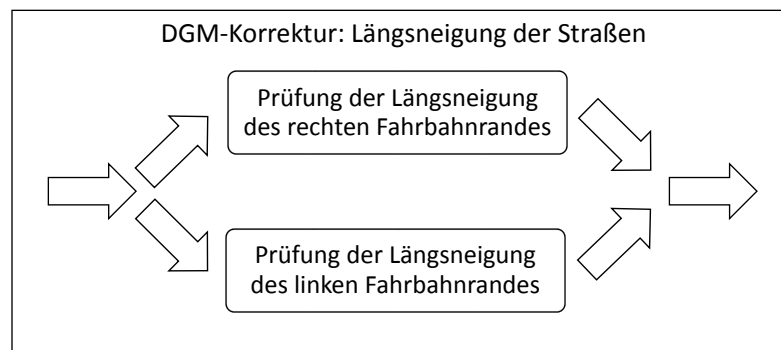


Abb. 8.2 – Unterteilung des Konzeptes der Korrektur der Längsneigung

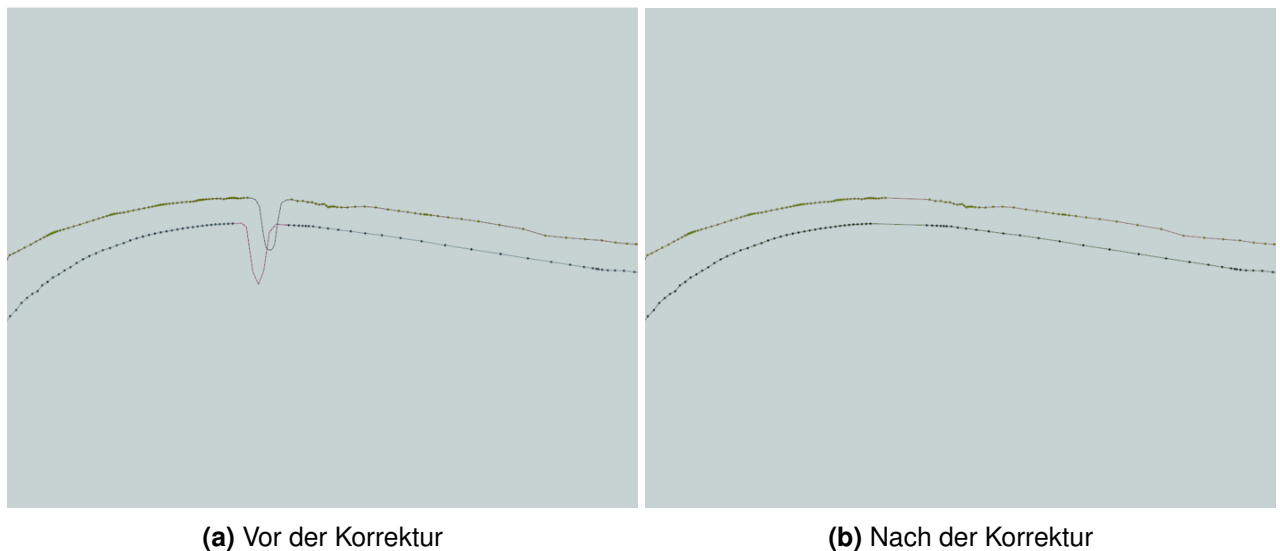
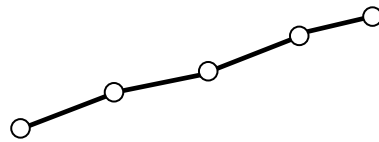


Abb. 8.3 – Beispiel einer DGM-Korrektur der Längsneigung

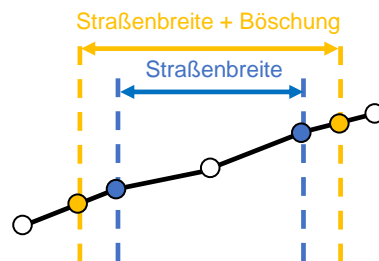
8.1.2 Querneigung

Um die Querneigung der Straßen korrigieren beziehungsweise minimieren zu können, bedarf es einem Vergleich für jeden Knotenpunkt entlang der Straßen. Dieser Vergleich wird durch die Betrachtung der, um die jeweilige Straßenbreite voneinander entfernten, Straßenränder durchgeführt. Diese parallelen Linien liegen zumeist auf einem unterschiedlichen Höhenniveau, was einer positiven oder negativen Querneigung entspricht. Im Konzept zur Korrektur der Querneigung werden die nächstgelegenen gegenüberliegenden Knotenpunkte bezüglich ihres Höhenwertes verglichen. Der höhere Knotenpunkt wird dann dem niedrigeren Knotenpunkt angepasst (vgl. Abbildung 8.4). So wird die Querneigung des DGM entlang der Straßen minimiert. In der konkreten Umsetzung mit

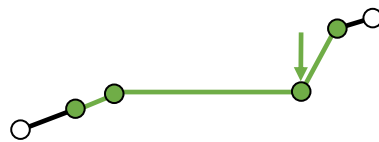
FME, wird die Überprüfung der Querneigung für beide Seitenränder parallel durchgeführt. Ein Beispiel einer Korrektur ist in Abbildung 8.5 dargestellt. Darin ist im linken Bild zu erkennen, dass die Fahrbahn eine starke Querneigung aufweist. Die korrigierte Fahrbahn ist im rechten Bild abgebildet. Wie die Korrektur der Längsneigung, so ist auch die Korrektur der Querneigung Bestandteil des FME Workspace *2_DGM_Korrektur.fmw* (vgl. Abbildung 10.20).



(a) Original DGM Querschnitt einer Straße mit Kanten und Knoten

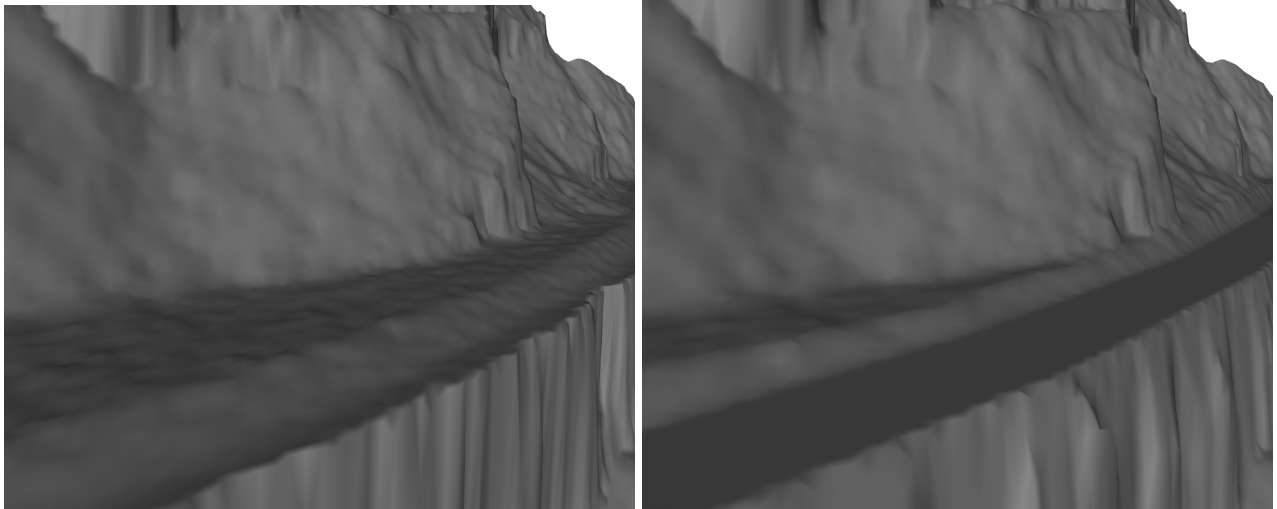


(b) Abbildung der Straßenbreite (blau) und der Straßenbreite plus Böschung (orange)



(c) Knoten werden auf korrekte Höhe angepasst (grün) und durch Kanten verknüpft (grün)

Abb. 8.4 – Konzept der Korrektur der Querneigung



(a) Vor der Korrektur

(b) Nach der Korrektur

Abb. 8.5 – Beispiel einer DGM-Korrektur der Querneigung

Das Konzept zur DGM-Korrektur im Straßenbereich erfüllt zwar seinen Zweck, jedoch erzeugt es ungewollte Nebeneffekte. So werden Kreuzungsbereiche mit großen Höhenunterschieden teilweise fehlerhaft modelliert. Des Weiteren wird durch die Querneigungskorrektur zwar die Neigung der Straßen minimiert, allerdings erzeugt der Ansatz eine wellige Fahrbahn in Fahrtrichtung (vgl. Abbildung 8.6). Diese Fehler konnten im Rahmen dieser Arbeit nicht behoben werden.



(a) Fehlerhafte Korrektur im Kreuzungsbereich

(b) Wellige Fahrbahn in Fahrtrichtung

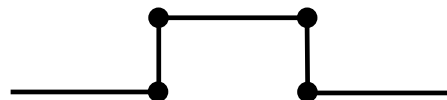
Abb. 8.6 – Fehlerhafte DGM-Korrekturen der Straßen

8.2 DGM-Korrektur: Brückenkörper

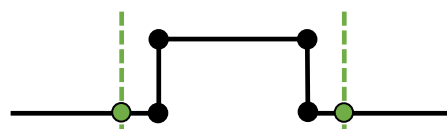
In diesem Abschnitt der Arbeit werden zwei Ansätze zur DGM-Korrektur im Bereich der Brücken vorgestellt und verglichen. Dabei werden die Konzepte mit ihren Unterschieden dargelegt. Die DGM-Korrektur ist nötig, da die Brückenkörper in einem DGM nicht vorgesehen sind. Im Gegensatz zu einem DOM, sollten in einem DGM keine Bauwerke oder Vegetation enthalten sein. Die Korrektur ist ebenfalls notwendig, um Platz für die modellierten Brückenkörper zu schaffen. Ein Grund, warum Brückenkörper im DGM enthalten sind, ist zum Beispiel die Entzerrung von digitalen Orthofotos. Hierbei ist für eine korrekte Entzerrung notwendig, dass die Brückenkörper im DGM enthalten sind [AdV (2012)].

8.2.1 Brückenumfeld ohne Höhenvergleich

Im ersten Korrekturansatz, wird das Brückenumfeld links und rechts der Brückenmittelachsen betrachtet. Dabei ist es das Ziel, die Brückenkörper auf Basis des fehlerfreien Umfeldes zu eliminieren. Konkret werden dazu links und rechts der Brückenmittelachsen zwei Linien extrahiert, welche jedoch genügend Abstand zum jeweiligen Brückenkörper aufweisen müssen. Der Abstand der linken und rechten Linie beträgt in dieser Arbeit 13m. Dieser Abstand ergab in den Berechnungen die besten Ergebnisse. Die Brückenkörper, innerhalb dieser Linien, werden dann entfernt. Um die so entstehenden Lücken im DGM wieder zu schließen, werden die extrahierten Linien, mit Hilfe einer Dreiecksvermaschung, je Brücke verbunden (vgl. Abbildung 8.7).



(a) Original DGM als Brückenquerschnitt mit Kanten und Knoten



(b) Definition der Linien links und rechts der Brückenmittelachse



(c) Entfernter Brückenkörper wird durch verbinden der Linien ersetzt

Abb. 8.7 – Konzept der DGM-Korrektur mit Ansatz 1a

Das vorgestellte Konzept zur DGM-Korrektur im Brückenbereich wird, in dieser Arbeit, mit Hilfe der Software FME umgesetzt. Dabei lässt sich die DGM-Korrektur mit Ansatz 1a in die zwei Blöcke *DGM-Schnitte parallel zur Fahrbahnachse extrahieren* und *Verbinden der Brückenumfeldes* einteilen, welche in Abbildung 8.8 aufgeführt sind. Die DGM-Korrektur mit Ansatz 1a ist Bestandteil des FME Workspace *2_DGM_Korrektur.fmw* (vgl. Abbildung 10.20).

Im ersten Block werden die zwei Linien parallel zur Brückenmittelachse extrahiert, welche die Höheninformationen des DGMs erhalten. Zudem werden die Linien in kleinere Linienstücke unterteilt, um eine bessere Abtastung des Brückenumfeldes zu ermöglichen.

Im zweiten Block erfolgt die eigentliche Korrektur, indem der Brückenkörper zwischen den Linien aus Block 1 zunächst entfernt wird. Danach dienen die Linien als Bruchkanten in einer Delaunay-Triangulation, welche die entstehende Lücke im DGM schließt.

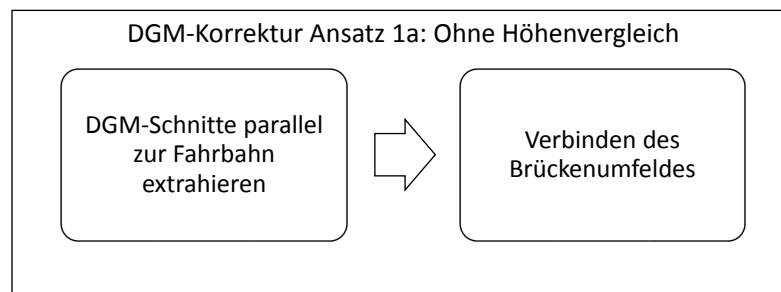
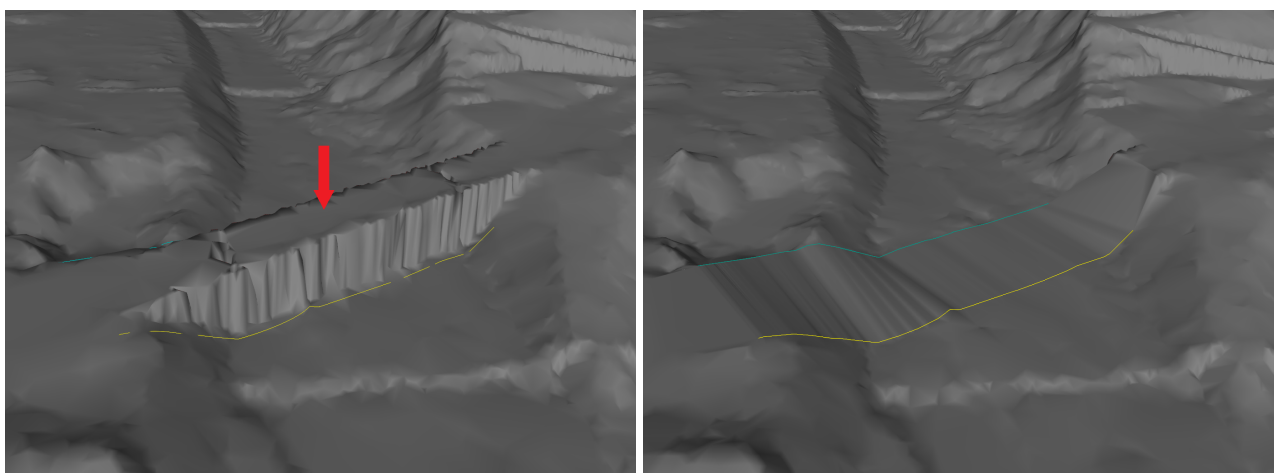


Abb. 8.8 – Unterteilung des Konzepts der DGM-Korrektur mit Ansatz 1a

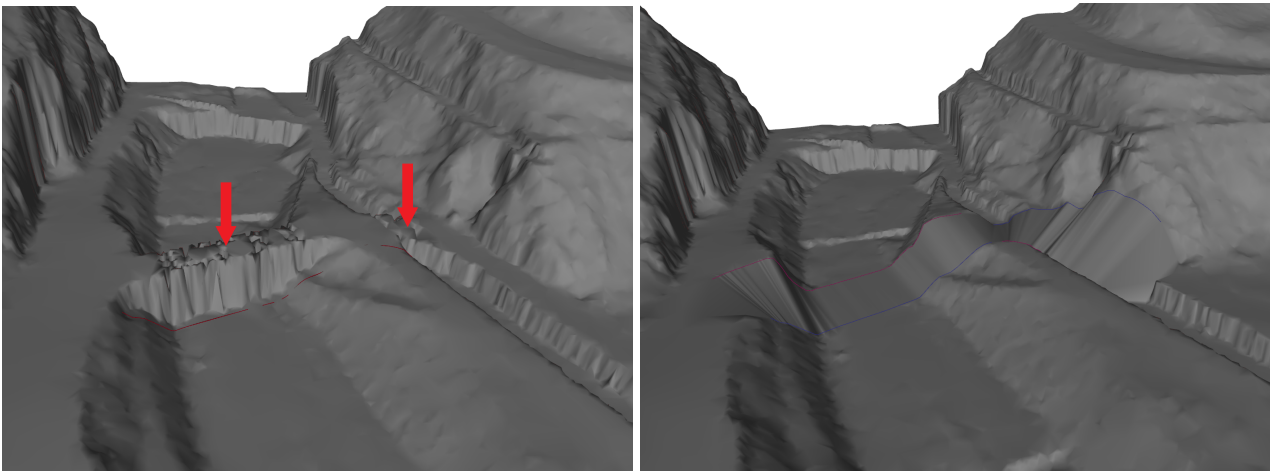
In Abbildung 8.9 ist ein Beispiel einer DGM-Korrektur mit Ansatz 1 dargestellt. Darin sind das original DGM, das korrigierte DGM sowie die erzeugten Seitenlinien zu erkennen.



(a) Original DGM mit Brückenkörper (mit Pfeil markiert) und Seitenlinien **(b)** Korrigiertes DGM ohne Brückenkörper und mit Seitenlinien

Abb. 8.9 – Beispiel der DGM-Korrektur mit Ansatz 1a

In Abbildung 8.10 ist das DGM mit jeweils zwei Brücken vor und nach der Korrektur dargestellt. Darin ist zu erkennen, dass der linke Brückenkörper gut entfernt wird. Allerdings ergeben sich beim rechten Brückenkörper Probleme. Zum einen führt die Brücke in einer Kurve über einen Bach und zum anderen befindet sie sich nahe an einem Hang. Der Verlauf des Baches wird in diesem Ansatz nicht mit einbezogen und führt zur Verfälschung des korrigierten DGM. Da die Seitenlinien der rechten Brücke im Bachbett und auf dem Hang liegen, werden diese verbunden, was wiederum zur Verfälschung des korrigierten DGM führt. Daher ist der Ansatz 1a zur DGM-Korrektur im Brückenbereich nur bei Brücken anwendbar, wenn die Mittelachsen dieser, orthogonal zum Fluss- oder Talverlauf orientiert sind. Des Weiteren sollten sich keine Erhebungen oder Senkungen direkt neben der Brücke befinden.

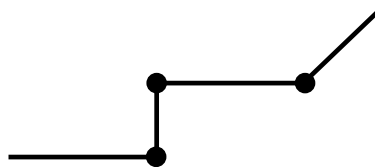


(a) Original DGM mit Brückenkörpern (mit Pfeilen markiert) und Seitenlinien (b) Korrigiertes DGM ohne Brückenkörper und mit Seitenlinien

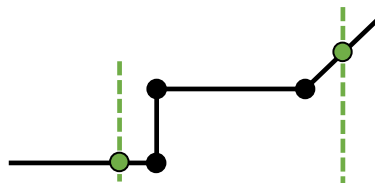
Abb. 8.10 – Probleme der DGM-Korrektur mit Ansatz 1a

8.2.2 Brückenumfeld mit Höhenvergleich

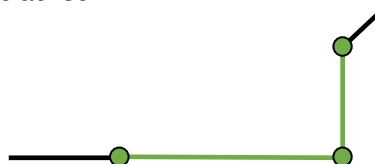
Um die Verfälschungen des ersten Ansatzes, wie in Abbildung 8.10 dargestellt, vermeiden zu können, bietet es sich an, die Seitenlinien bezüglich ihrer Höhenlage zu vergleichen. Konkret werden dazu ebenfalls links und rechts der Brückenmittelachsen zwei Linien extrahiert, welche jedoch genügend Abstand zum jeweiligen Brückenkörper aufweisen müssen. Die Linien werden dann nicht direkt verbunden, um die Brücke zu ersetzen, sondern sie werden verglichen. Dabei werden die Höhenwerte der Linienknotenpunkte überprüft, wobei die höheren Punkte den nächstgelegenen, gegenüberliegenden niedrigeren Punkten angepasst werden. So wird verhindert, dass Erhebungen oder Senkungen im Brückenumfeld zu Verfälschungen der DGM-Korrektur führen. Die angepassten Linien werden dann durch eine Dreiecksvermaschung verbunden, um die Brückenkörper zu ersetzen. Dieses Vorgehen ist in Abbildung 8.11 an einem Brückenquerschnitt skizzenhaft dargestellt.



(a) Original DGM als Brückenquerschnitt im Hang mit Kanten und Knoten



(b) Definition der Linien links und rechts der Brückenmittelachse



(c) Entfernter Brückenkörper wird durch verbinden der angepassten Linien ersetzt

Abb. 8.11 – Konzept der DGM-Korrektur mit Ansatz 1b

Wie der erste Ansatz zur DGM-Korrektur, wird auch dessen Erweiterung 1b mit FME durchgeführt und ist Bestandteil des FME Workspace *2_DGM_Korrektur.fmw* (vgl. Abbildung 10.20). Dabei lässt sich die DGM-Korrektur mit Ansatz 1b in zwei Blöcke einteilen. Diese sind *DGM-Schnitte parallel zur Fahrbahnachse extrahieren* und *Entfernung der Brücken durch Höhenvergleich und Verbinden der Brückenumfeldes links und rechts*. Die Unterteilung in die Blöcke ist in in Abbildung 8.12 dargestellt.

Beim ersten Block werden ebenfalls zwei Linien parallel zur Brückenmittelachse extrahiert und mit den Höheninformationen des DGM angereichert. Die Unterteilung in kleinere Linienstücke dient ebenfalls der besseren Abtastung.

Im zweiten Block wird der Brückenkörper analog zu Ansatz 1a aus dem DGM entfernt. Im Unterschied zu Ansatz 1a, werden die parallelen Linien nun bezüglich ihrer Höhe verglichen. Ziel ist es, dass die nächstgelegenen und gegenüberliegenden Knotenpunkte der Linien auf gleicher Höhe liegen. Konkret werden höhere Punkte den niedrigeren angepasst. Im letzten Schritt dienen die angepassten Linien ebenfalls als Bruchkanten in einer Delauney-Triangulation, welche die Lücke im DGM schließt. In Abbildung 8.13 ist ein Beispiel einer DGM-Korrektur mit Ansatz 1b dargestellt. Darin sind das original DGM, das korrigierte DGM sowie die erzeugten Seitenlinien abgebildet.

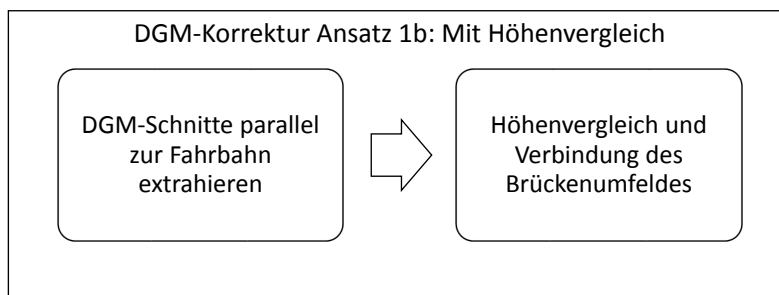
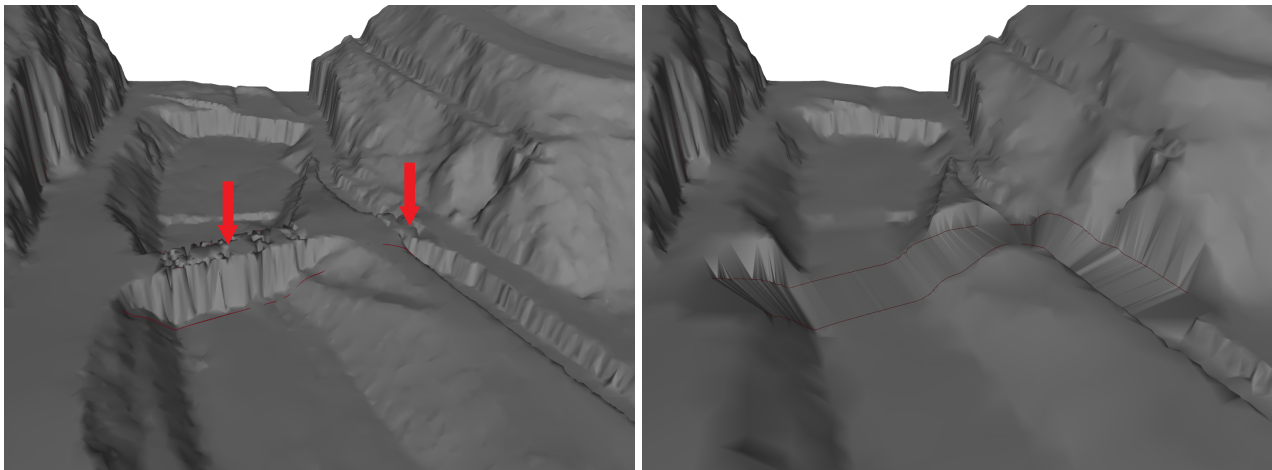


Abb. 8.12 – Unterteilung des Konzepts der DGM-Korrektur mit Ansatz 1b

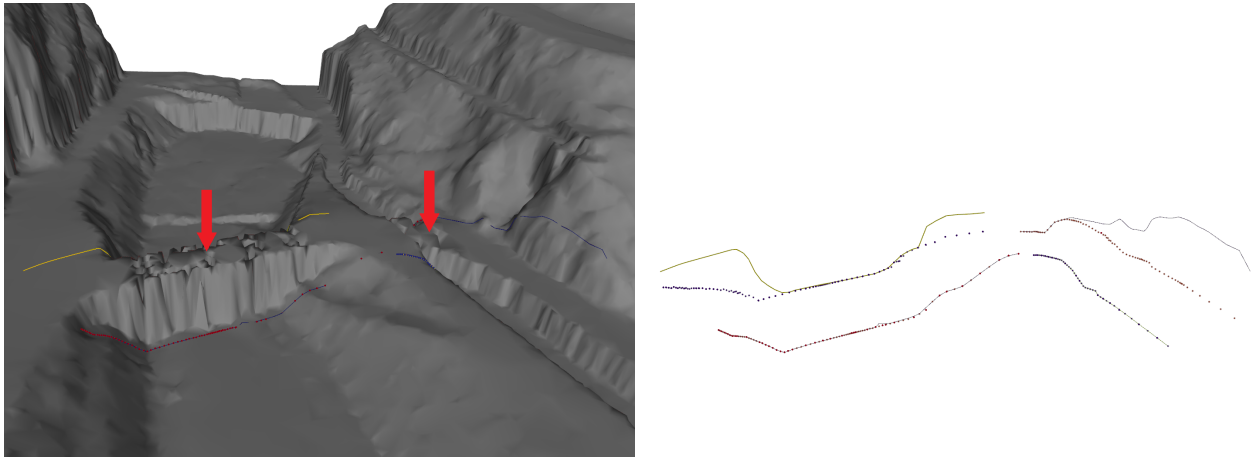


(a) Original DGM mit Brückenkörpern (mit Pfeilen markiert) und Seitenlinien **(b)** Korrigiertes DGM ohne Brückenkörper und mit Seitenlinien

Abb. 8.13 – Beispiel der DGM-Korrektur mit Ansatz 1b

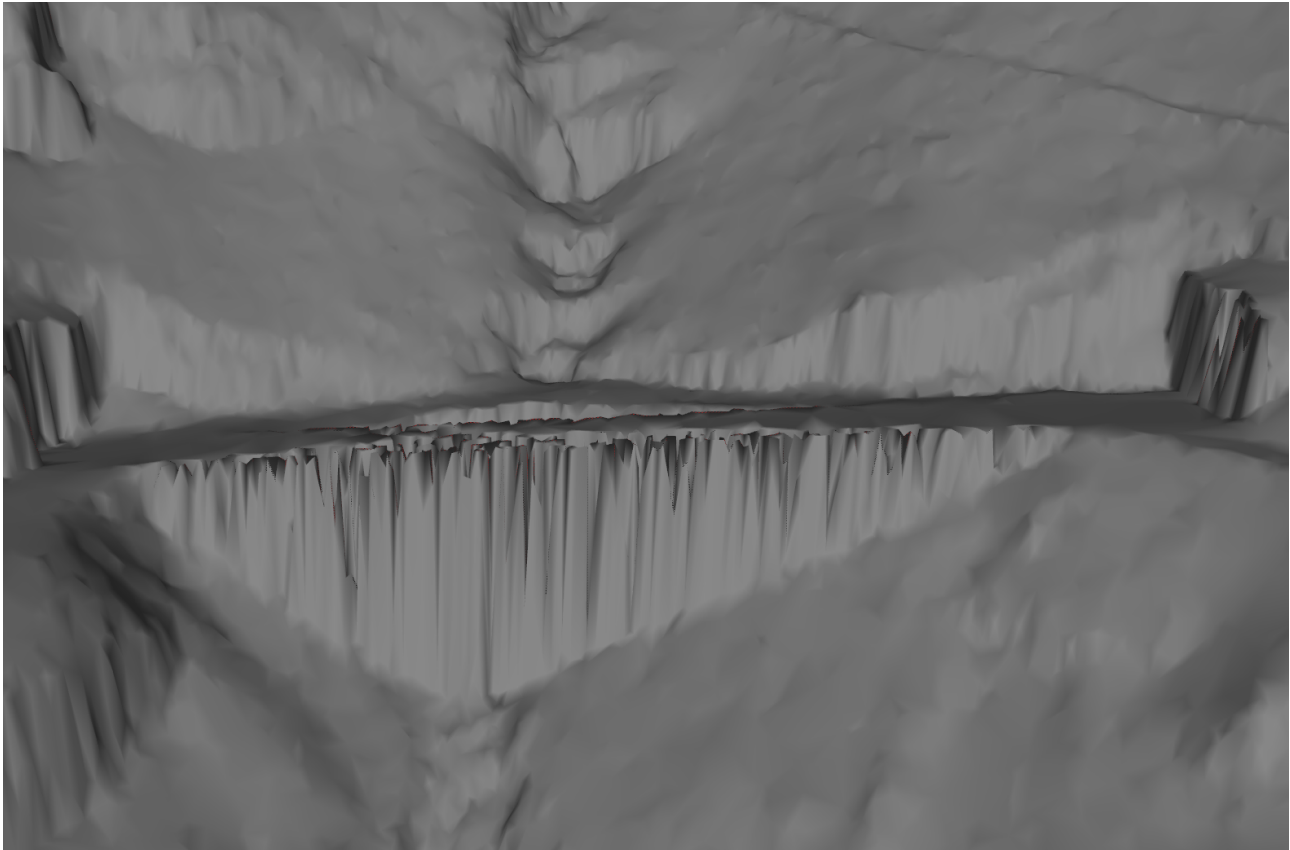
Um die Ansätze 1a und 1b vergleichen zu können, sind in Abbildung 8.14 die neuen Bruchkanten beider Ansätze dargestellt. Dazu ist in der linken Abbildung das originale DGM mit den neuen Bruchkanten visualisiert. Hierbei entsprechen die Linien Ansatz 1a und die Punkte Ansatz 1b. In der rechten Abbildung sind nur die neuen Bruchkanten dargestellt. Es ist zu erkennen, dass bei der lin-

ken Brücke, der linke Böschungsrand mit Ansatz 1a realistischer modelliert wird. Ansatz 1b entfernt hierbei, wegen dem schrägen Verlauf der Brücke über das Flussbett, mehr von der Böschung. Das große Flussbett wird bei beiden Ansätzen wieder hergestellt. Bei der rechten Brücke ist der Vorteil von Ansatz 1b sichtbar. Die Höhen der Seitenpunkte, welche am Hang liegen, werden den Höhen der niedrigeren Punkte angepasst. Dies ermöglicht es, die Brückenkörper fast vollständig zu entfernen. Trotz des erläuterten Nachteils, liefert Ansatz 1b bessere Ergebnisse und wird für die konkrete DGM-Korrektur in dieser Arbeit verwendet. Ein weiteres Beispiel des, für die weitere Arbeit verwendeten, Ansatzes 1b, ist in Abbildung 8.15 dargestellt.

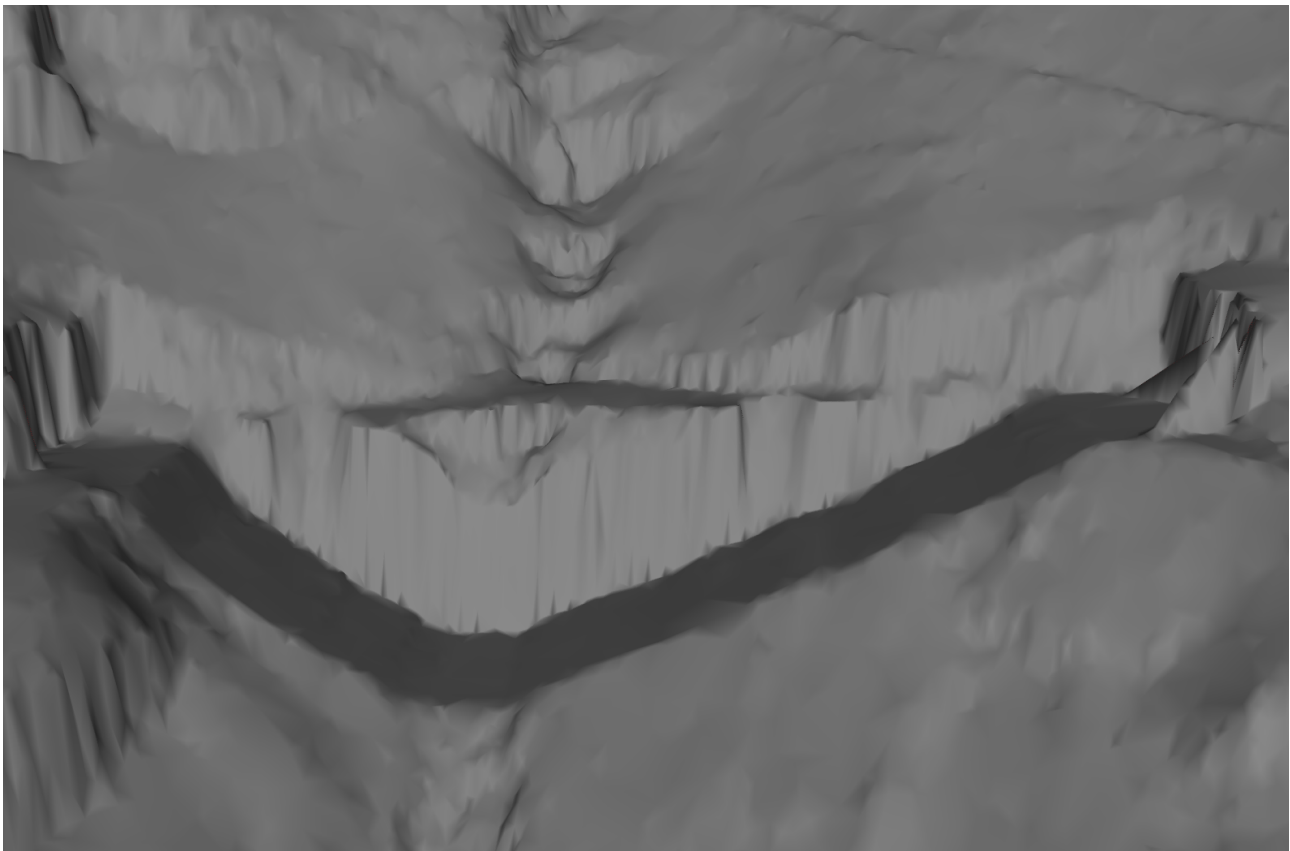


(a) Original DGM mit den neuen Bruchkanten: Linien (b) Neue Bruchkanten: Linien (Ansatz 1a) und Punkte (Ansatz 1a) und Punkte (Ansatz 1b) (Brücken mit (Ansatz 1b) Pfeilen markiert

Abb. 8.14 – Vergleich der DGM-Korrektur mit Ansatz 1a und 1b



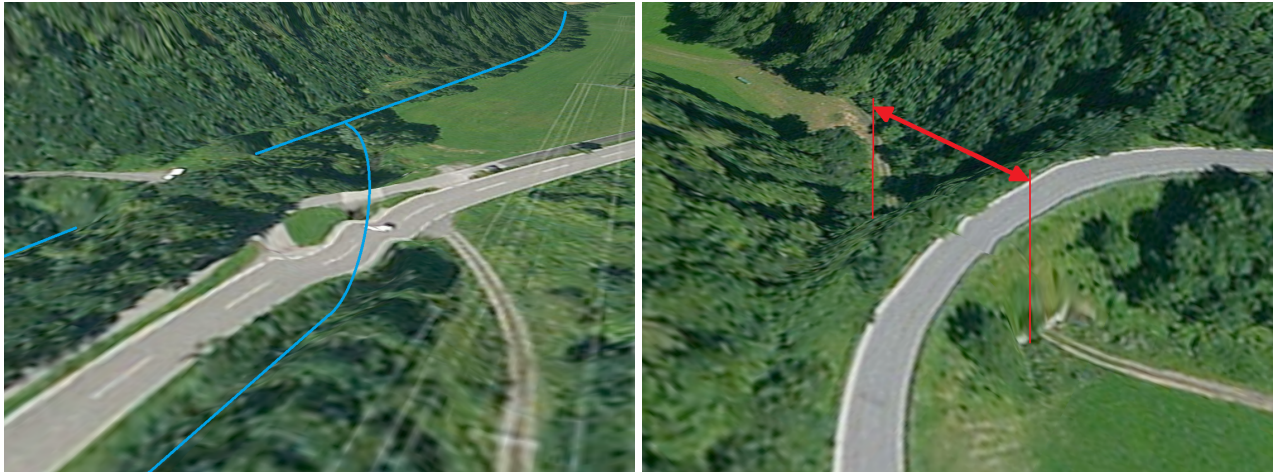
(a) DGM mit dem Brückenkörper



(b) Korrigiertes DGM ohne den Brückenkörper

Abb. 8.15 – Weiteres Beispiel der DGM-Korrektur mit Ansatz 1b

Bei Betrachtung aller DGM-Korrekturen in den Brückenbereichen, lassen sich zwei Nachteile des verwendeten Korrekturansatzes 1b erkennen. Von den 31 Brückenkörpern werden 23 zufriedenstellend und 8 fehlerhaft entfernt. Der größte Teil dieser fehlerhaft entfernten Brücken verläuft schräg über einem Graben beziehungsweise einem Fließgewässer. Bei den restlichen fehlerhaft entfernten Brückenkörpern liegt eine zu geringe Pufferungsbreite der Seitenlinien aus Ansatz 1b vor. Diese Umstände lassen den verwendeten Ansatz versagen und verursachen im verwendeten Testgebiet eine Fehlerquote von circa 26 %. Diese Fehler sind in Abbildung 8.16 anhand zweier Beispiele dargestellt.

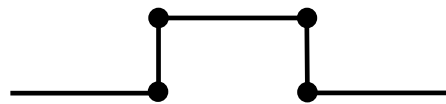


(a) Fehlerhaft korrigiertes DGM mit Markierung des Gewässer-/Grabenverlaufs (Blau) (b) Fehlerhaft korrigiertes DGM mit Markierung der extremen Brückenkörperbreite (Rot)

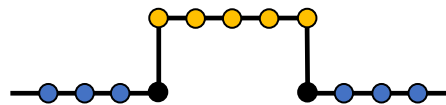
Abb. 8.16 – Beispiele fehlerhafter DGM-Korrekturen im Brückenbereich

8.2.3 Klassifikation der Punktwolke

Der zweite Korrekturansatz hat das Ziel, die Brückenkörper auf Basis der Laser (LAS) Punktwolke zu eliminieren. Konkret wird dazu eine klassifizierte Punktwolke verwendet. Dabei sind speziell die Klassen *Brücken* und *Wasser* für diese Arbeit interessant. Mit Hilfe der Brückenpunkte werden die Brückenkörper detektiert und aus dem original DGM entfernt. Die Gewässerpunkte vor und nach der Brücke werden verbunden und dienen in der folgenden Modellierung als Bruchkanten. Dieses Vorgehen ist in Abbildung 8.17 an einem Brückenquerschnitt skizzenhaft dargestellt. Darin entsprechen die gelben Punkte der Klasse *Brücken* und die blauen Punkte der Klasse *Wasser*. Falls eine oder mehrere Brücken nicht über ein Gewässer führen, kann für diese Brücken Ansatz 1b verwendet werden.



(a) Original DGM als Brückenquerschnitt mit Kanten und Knoten



(b) Brücken- (gelb) und Gewässerpunkte (blau) aus Punktwolke extrahieren



(c) Entfernter Brückenkörper wird durch verbinden der Gewässerpunkte ersetzt

Abb. 8.17 – Konzept der DGM-Korrektur mit Ansatz 2

Der zweite Ansatz zur DGM-Korrektur wird wie Ansatz 1a und 1b mit FME durchgeführt und ist Bestandteil des FME Workspace *2_DGM_Korrektur.fmw* (vgl. Abbildung 10.20). Dabei lässt sich die DGM-Korrektur mit Ansatz 2 in die drei Blöcke *Brücken- und Gewässerpunkte aus Punktwolke extrahieren*, *Wasserpunkte in Brückenumgebung extrahieren* und *Bruchkanten bilden* und *Brückenumfeld aus original DGM löschen und korrigiertes DGM modellieren* einteilen. Diese Blöcke sind in Abbildung 8.18 dargestellt. Im ersten Block wird die klassifizierte Punktwolke in ihre Klassen aufgeteilt. Die für die DGM-Korrektur interessanten Klassen, *Wasser* und *Brücken*, werden als eigene Punktwolken weitergegeben. Ein Beispiel einer Klassifikation des zweiten Ansatzes ist in Abbildung 8.19 dargestellt. Hierbei ist das originale DGM mit den Klassen *Wasser* und *Brücken* visualisiert. Im zweiten Block werden die Gewässerpunkte vor und nach der Brücke extrahiert und zu Linien verbunden.

Zudem werden die Brückenpunkte zu einer Fläche verschmolzen. Auf Basis dieser Fläche werden im dritten Block die Brückenkörper aus dem DGM entfernt. Die zu Linien verbundenen Punkte dienen dann als Bruchkanten in der abschließenden Delauney-Triangulation, welche das DGM wieder verschließt.

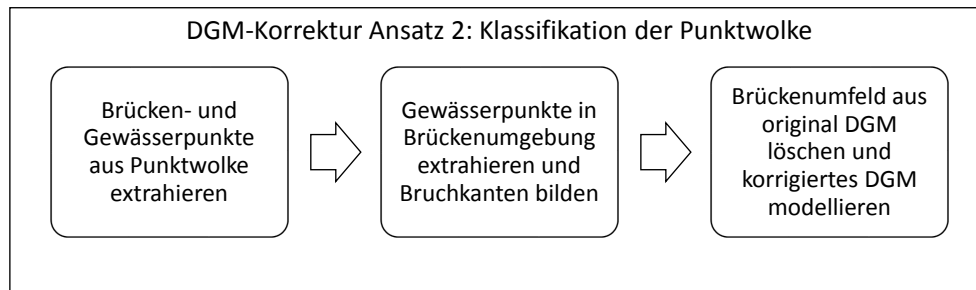


Abb. 8.18 – Unterteilung des Konzepts der DGM-Korrektur mit Ansatz 2

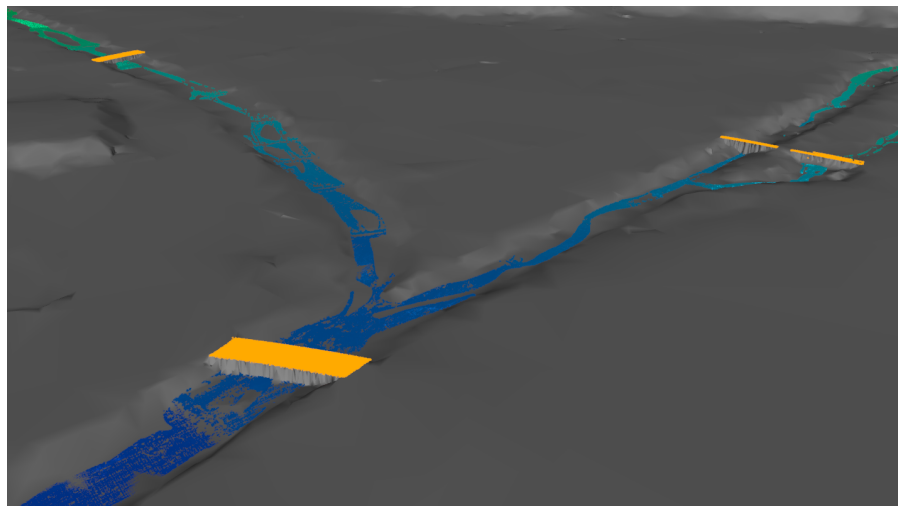
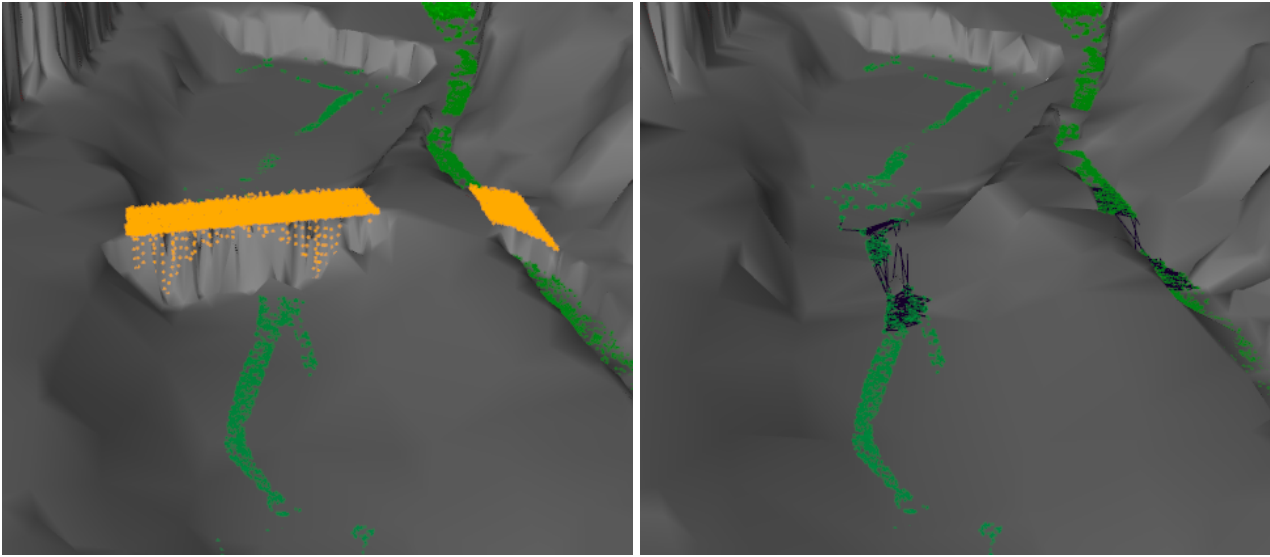


Abb. 8.19 – Beispiel einer Klassifikation mit Ansatz 2. Original DGM mit den Klassen Wasser (Blau-Grün) und Brücken (Orange)

Da in diesem Ansatz die Gewässerverläufe mit einbezogen werden, ist eine realistischere Modellierung als bei Ansatz 1a und 1b möglich. Böschungen von Gewässern werden an den Gewässerverlauf angepasst, unabhängig vom Winkel mit dem die Brücke das Gewässer überspannt. Dazu sind in Abbildung 8.20 zwei Brücken vor und nach der Korrektur dargestellt. Es ist zu erkennen, dass die Böschungen und die Flussbetten des großen sowie des kleinen Gewässers modelliert werden. Im Unterschied zu Ansatz 1a und 1b werden nur Brückenkörper über Flussbetten und Erhebungen entfernt. Ob die in Abbildung 8.20 entfernte rechte Brücke neben dem eigentlichen Flussbett weitergeht, kann mit diesem Ansatz nicht bestimmt werden. Trotz der guten Ergebnisse mit dem zweiten Ansatz, wird dieser nicht für die weiteren Berechnungen dieser Arbeit verwendet. Zum einen versagt dieser Ansatz, wenn unterhalb von Brückenkörpern keine Gewässerpunkte vorhanden sind und zum anderen müssen flächendeckende, hoch genaue und klassifizierte Punktwolken vorliegen. Dies ist

zwar in dieser Arbeit der Fall, jedoch ist dies in anderen Gebieten nicht immer der Fall. Daher wird der erste Ansatz mit der Erweiterung des Höhenvergleiches verwendet.



(a) Original DGM mit den Klassen Wasser (Blau-Grün) und Brücken (Orange) (b) Korrigiertes DGM ohne Brückenkörper und mit der Klasse Wasser (Blau-Grün) sowie den neuen Bruchkanten (Schwarz)

Abb. 8.20 – Beispiel der DGM-Korrektur mit Ansatz 2

8.3 DGM-Korrektur: Tunnelportale

Um eine Darstellung von Tunnelkörpern in 3D-Landschaftsmodellen zu ermöglichen, muss das DGM auch für die Tunnel angepasst werden. Hierbei ist es notwendig, dass Oberflächen, Punkte oder Kanten des DGMs nicht innerhalb der Tunnelkörper liegen und diese auch nicht verschließen. Dies ist gerade im Bereich der Tunnelportale der Fall. In dieser Arbeit wird das DGM bestehend aus TIN-Kanten korrigiert. Dabei werden zunächst alle Kanten ausgeschnitten, welche oberhalb der Tunnelgrundfläche liegen. In einem zweiten Schritt werden die zuvor ausgeschnittenen Kanten weiterverarbeitet. Dabei werden diejenigen Kanten entfernt, welche auf der Höhe des Tunnelkörpers liegen. Die übrigen Kanten werden dann wieder in das DGM eingefügt. Das Ergebnis ist dann das DGM ohne die Durchstoßflächen der Tunnel. Ein Beispiel einer DGM-Korrektur ist in Abbildung 8.21 dargestellt. Konkret wird die Korrektur mit der Software FME durchgeführt. Dabei ist die Korrektur Bestandteil des FME Workspace *CityGML_DGM.fmw* (vgl. Abbildung 10.30).

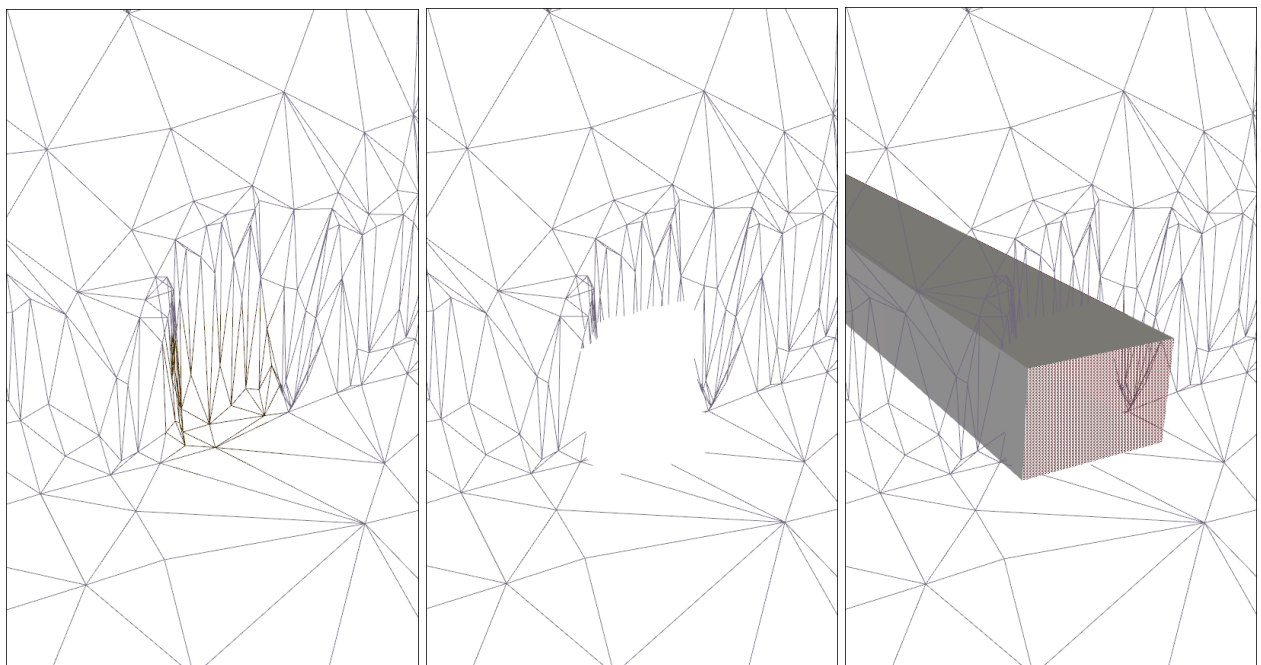


Abb. 8.21 – Entfernung des DGMs im Bereich eines Tunnelportals. Links: DGM vor der Korrektur. Mitte: Korrigiertes DGM. Rechts: Korrigiertes DGM mit Tunnelkörper.

8.4 DGM-Aufbereitung

Nach der DGM-Korrektur für die Bereiche Straßen, Brücken und Tunnel, folgt die DGM-Aufbereitung. Hierbei müssen die einzelnen Kacheln einander angepasst werden. Dies ist notwendig, weil die vorangehenden DGM-Korrekturen in jeweils einer Schleife für die einzelnen Kacheln erfolgt, da sonst der Arbeitsspeicherbedarf extrem ansteigt. Das Problem der Berechnung mit Hilfe von Schleifen ist, dass die Kachelgrenzen ohne Informationen zur benachbarten Kachel generiert werden. Um jedoch ein komplettes DGM ohne Sprünge an den Kachelgrenzen herstellen zu können, müssen die Kachelgrenzen abschließend neu berechnet werden. Was passiert, wenn die Kachelgrenzen nicht neu berechnet werden, ist in Abbildung 8.22 dargestellt. Die Übergangszone zwischen den Kacheln weist Höhengsprünge auf.

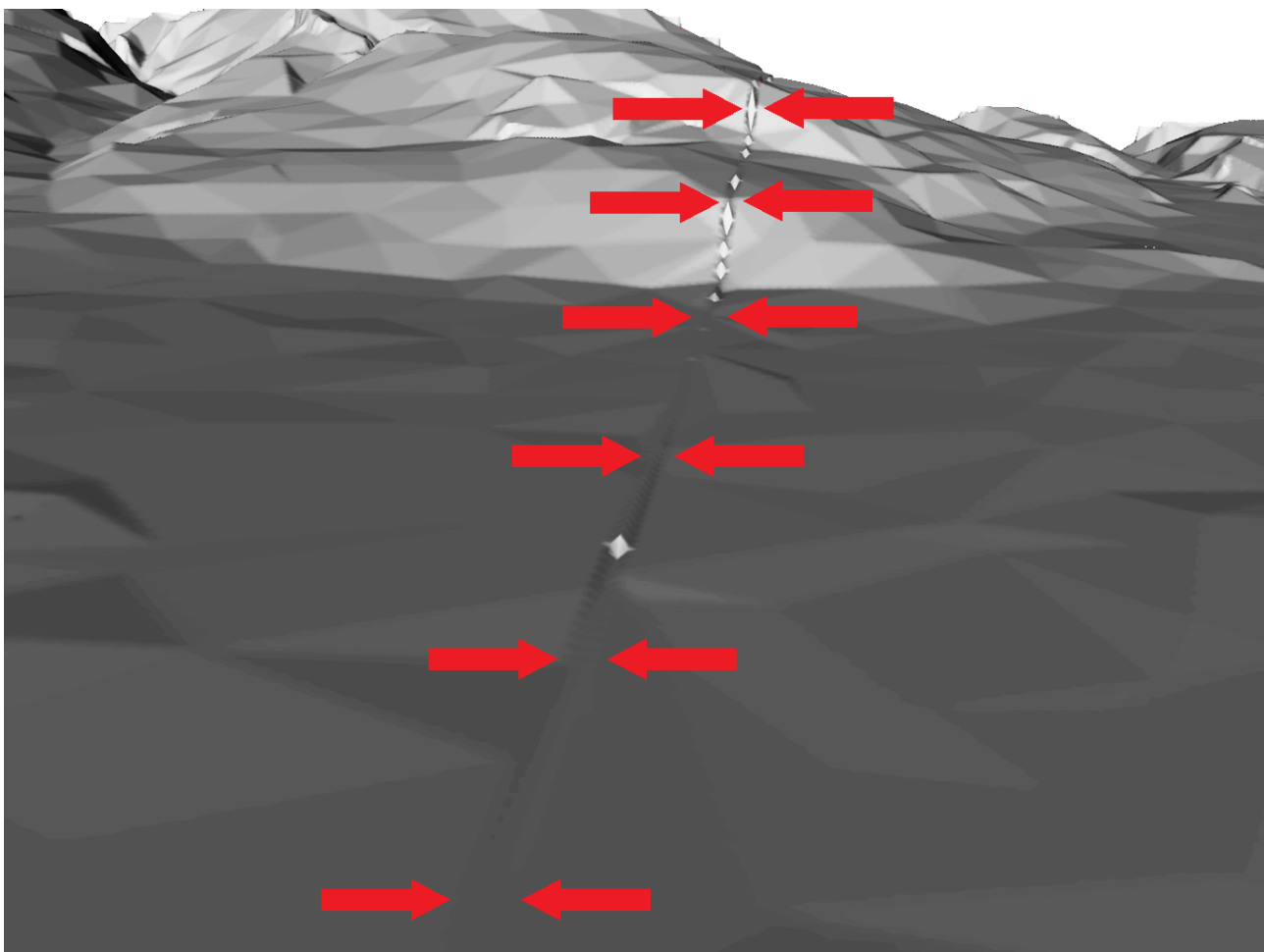
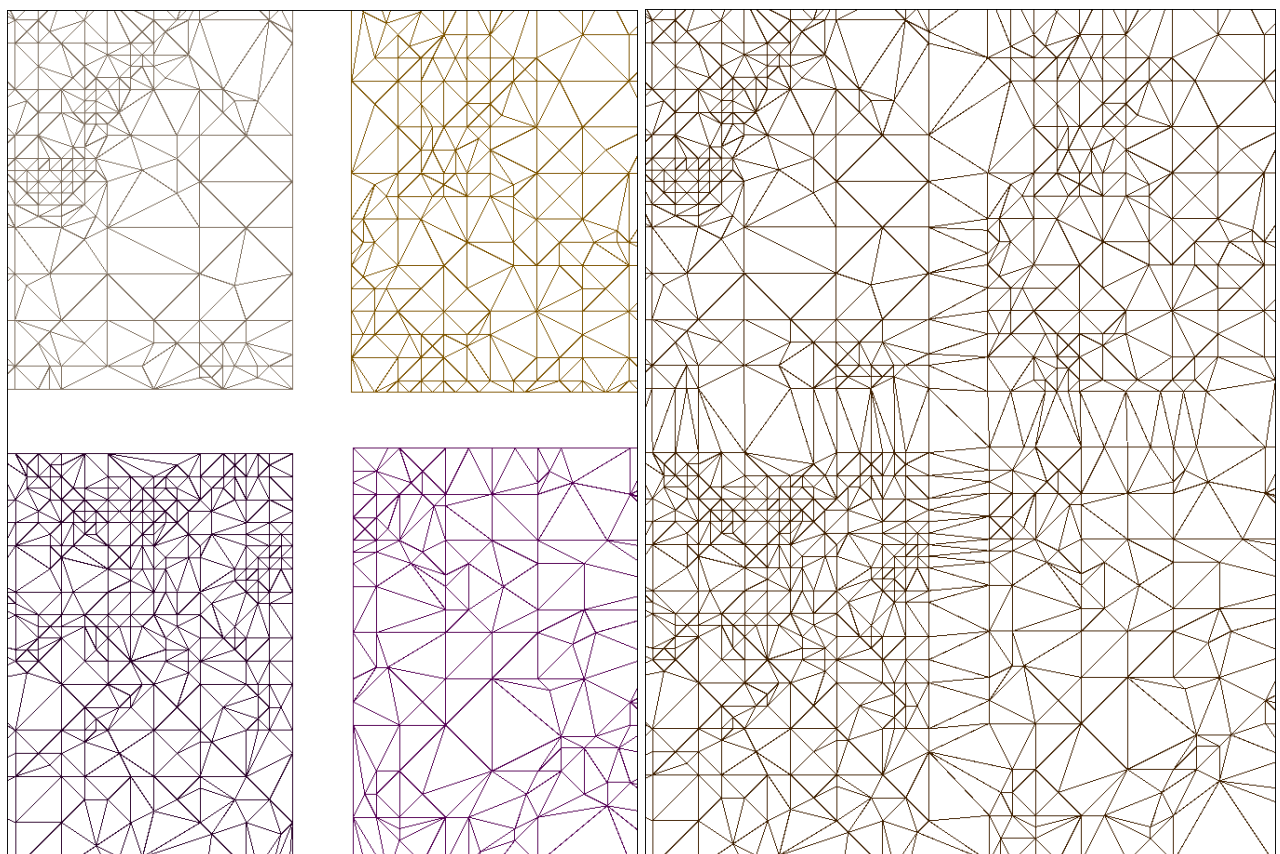


Abb. 8.22 – Ausschnitt eines DGM mit fehlerhaften Kachelgrenzen (Der Verlauf der Kachelgrenze ist mit roten Pfeilen markiert)

Um die Kachelgrenzen einander anzupassen, wird zunächst der äußerste Rand je Kachel entfernt, weil dieser die fehlerhaften Höheninformationen enthält. Hierbei werden an jeder Seite jeweils 50 cm entfernt. In einem nächsten Schritt, werden alle Kacheln zu einem kompletten DGM vereinigt und die Lücken zwischen den Kacheln durch eine Delaunay Triangulation geschlossen (vgl. Abbil-

dung 8.23). Somit liegen die Kacheln mit gemeinsamen Kachelgrenzen vor. Um die Daten für den Übergang in das CityGML-Schema vorzubereiten, wird das DGM wieder in die ursprünglichen Kacheln unterteilt. Der äußerste Rand entlang des kompletten DGM geht bei diesem Schritt verloren, da keine Anschlusshöhen vorliegen. Die Korrektur der Kachelgrenzen basiert auf dem Konzept gemäß Kolbe u. a. (2015a). Im Unterschied dazu, werden die äußeren Ränder der Kacheln in dieser Arbeit vor der Triangulation entfernt. Die konkrete DGM-Aufbereitung erfolgt mit der Software FME und den Workspaces *3_DGM_Kacheln_1.fmw*, *3_DGM_Kacheln_2.fmw*, *3_DGM_Kacheln_3.fmw*, *3_DGM_Kacheln_4.fmw* sowie *3_DGM_Kacheln_5.fmw* (vgl. Abbildung 10.21 - 10.25). Im Workspace *3_DGM_Kacheln_1.fmw* werden die äußeren Ränder entfernt und die Kacheln für den nächsten Workspace vorbereitet. Die um den äußeren Rand bereinigten Kacheln werden im Workspace *3_DGM_Kacheln_2.fmw* an ihren Rändern neu modelliert. Dies dient der Zuschneidung welche im Workspace *3_DGM_Kacheln_3.fmw* stattfindet. Die eigentliche Neuberechnung der Kachelgrenzen erfolgt im Workspace *3_DGM_Kacheln_4.fmw*. In einem letzten Schritt wird der äußerste Rand des kompletten DGM entfernt, weil dort keine Anschlusshöhen bekannt sind. Dies erfolgt mit dem Workspace *3_DGM_Kacheln_5.fmw*.



(a) Einzelne DGM-Kacheln ohne den äußersten Rand

(b) Angepasste Kachelgrenzen

Abb. 8.23 – Ausschnitt einer Anpassung der DGM-Kachelgrenzen

8.5 Übergang in das CityGML-Schema

In diesem Abschnitt wird der Übergang des zuvor aufbereiteten DGMs in das CityGML-Schema dargestellt. CityGML bietet mehrere Möglichkeiten ein Geländemodell abzuspeichern. In dieser Arbeit wird das DGM in Form von Bruchkanten in die Klasse *BreaklineRelief* überführt. Hierbei wird die Geometrie *MultiCurve* verwendet. Die Klasse *BreaklineRelief* ist eine Kindklasse von der abstrakten Klasse *_ReliefComponent* und erbt somit das Attribut *lod*, welches die LOD Stufe beinhaltet. Weitere wichtige Attribute für das CityGML-Schema, sind der *CityGML Lod Name* und die *FeatureRole*. Diese beinhalten für die Klasse *BreaklineRelief* die Werte *breaklines* beziehungsweise *reliefComponent*. Um das CityGML-Schema besser verstehen zu können, ist in Abbildung 8.24 ein Auszug eines CityGML-Dokumentes aufgeführt. Darin ist das DGM als *BreaklineRelief* in einzelnen Bruchkanten gespeichert. Diese Bruchkanten entsprechen den Kanten eines TIN. Da jedoch durch die DGM-Korrektur nicht alle Dreiecke geschlossen sind, erfolgt die Speicherung in den einzelnen Kanten (vgl. Abbildung 8.21). Diese Speicherung ist Bestandteil des FME Workspace *CityGML_DGM* (vgl. Abbildung 10.30).

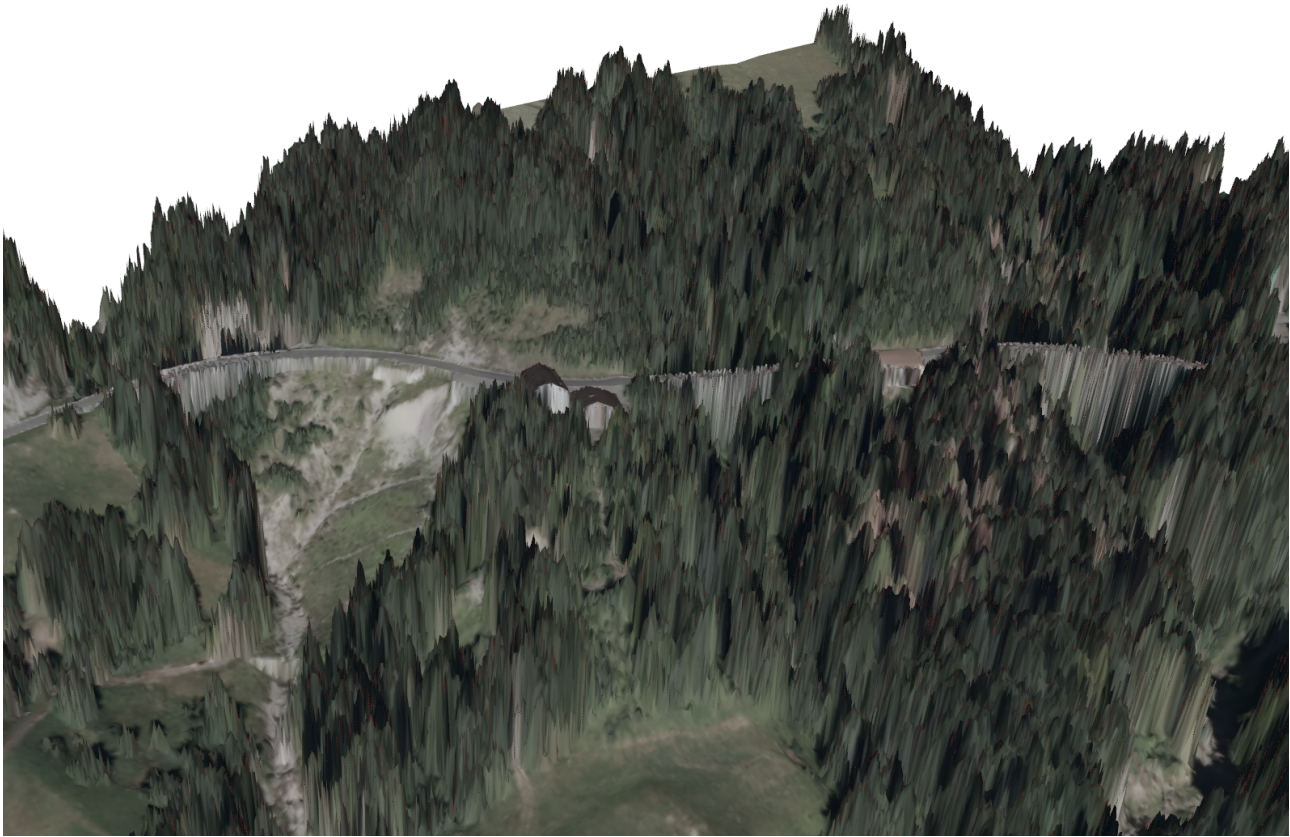
```
<?xml version="1.0" encoding="UTF-8"?>
<core:CityModel xmlns:smil20="http://www.w3.org/2001/SMIL20/" ... >
<gml:name>Vorarlberg</gml:name>
  <gml:boundedBy>
    <gml:Envelope srsName="EPSG:31254" srsDimension="3">
      <gml:lowerCorner>-49975.471813641 237368.526226832 719.734674372282</gml:lowerCorner>
      <gml:upperCorner>-49525.4718136408 237478.526226833 823.253935577571</gml:upperCorner>
    </gml:Envelope>
  </gml:boundedBy>
  <core:cityObjectMember>
    <dem:BreaklineRelief gml:id="bb7e33b5-cdf9-431e-8015-81e83b07b1fd">
      <dem:lod>1</dem:lod>
      <dem:breaklines>
        <gml:MultiCurve srsName="EPSG:31254" srsDimension="3">
          <gml:curveMember>
            <gml:LineString>
              <gml:posList>-49975.4718136409 237368.526226833 ... </gml:posList>
            </gml:LineString>
          </gml:curveMember>
        </gml:MultiCurve>
      </dem:breaklines>
    </dem:BreaklineRelief>
  </core:cityObjectMember>
  ...
</core:CityModel>
```

Abb. 8.24 – Auszug eines CityGML-Dokumentes eines DGM

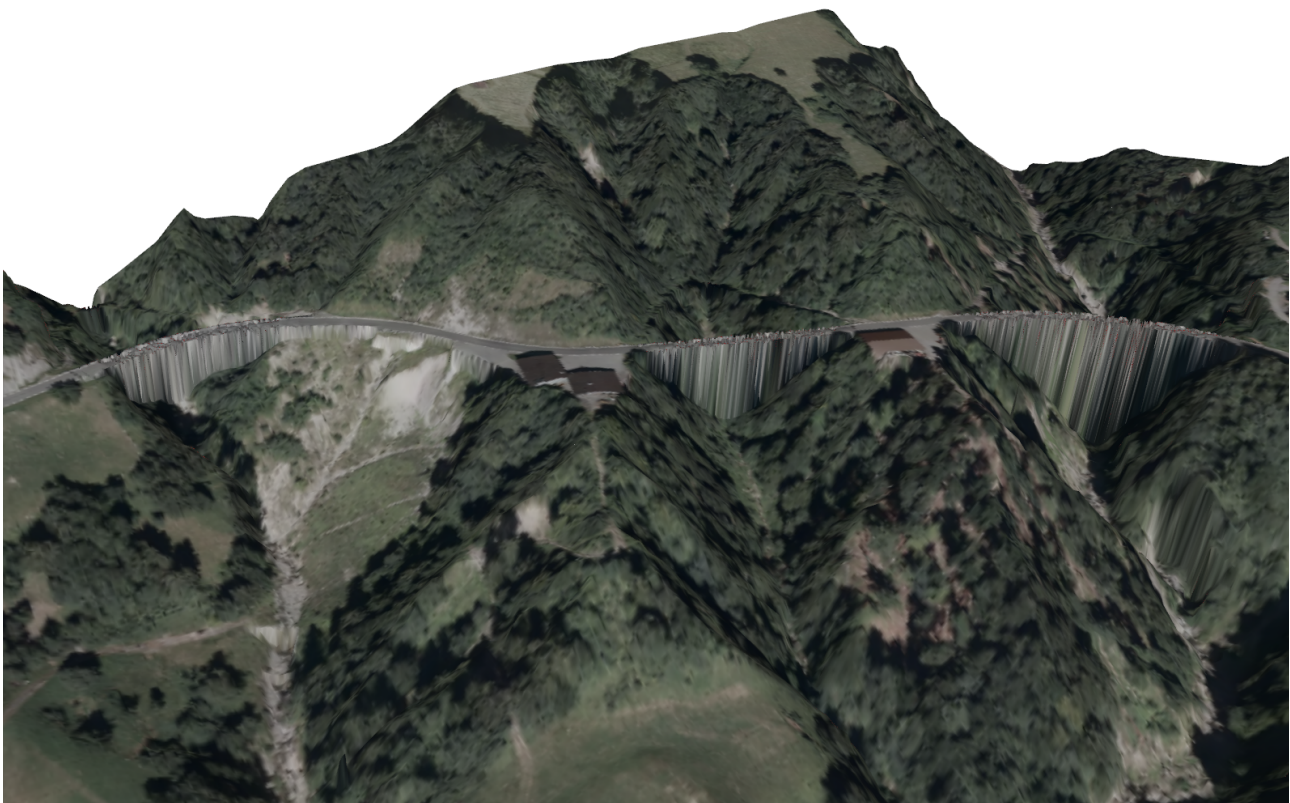
9 Visualisierung und Bewertung der Ergebnisse

9.1 Kombination zum Straßennetzwerk

In diesem Kapitel werden die, in den vorherigen Kapiteln, erzeugten Daten zusammengeführt. Hierbei werden die Straßenflächen im LOD 1, die Brücken und Tunnel im LOD 2 sowie das korrigierte DGM zusammen visualisiert. Dazu ist in den Abbildungen 9.1 und 9.2 ein Ausschnitt mit Straßenflächen sowie Brückenmodellen dargestellt. Dabei ist in Abbildung 9.1 einmal das texturiertes DOM mit Brücken- und Gebäudekörpern sowie Vegetation dargestellt. Zudem ist das texturierte DGM mit den Brückenkörpern abgebildet, welches Grundlage für die DGM-Korrekturen in Kapitel 8 ist. In der zweiten Abbildung 9.2, ist zunächst das korrigierte und texturierte DGM ohne die Brückenkörper visualisiert. Das zweite Bild der Abbildung 9.2 zeigt nochmals das korrigierte und texturierte DGM, jedoch nun mit den Brückenmodellen und den Straßenflächen. Es ist zu erkennen, dass die Brückenkörper an der richtigen Position bezüglich Lage und Höhe platziert sind. Die Straßenflächen befinden sich ebenfalls an der vorgesehenen Position. Allerdings liegen die Straßenflächen sehr nahe über, beziehungsweise unter dem DGM, wodurch die Straßenflächen teilweise nicht erkennbar sind. Dieser Effekt entsteht, weil die Straßenflächen andere Querneigungen und Längsneigungen als das DGM aufweisen. Grund dafür ist, dass die Straßenflächen in *CityEngine* und das DGM mit *FME* unabhängig von einander erzeugt werden. Ein weiterer Ausschnitt ist in den Abbildungen 9.3 und 9.4 aufgeführt. In der Abbildung 9.3 ist das texturierte DGM mit und ohne Brückenkörper dargestellt. In der Abbildung 9.4 sind, zu dem texturierten DGM, auch die Brückenmodelle und die Straßenflächen zu sehen. Hierbei fällt auf, dass die größere der beiden Brücken, eine starke Lageabweichung aufweist. Dies ist durch das verwendete Orthofoto sowie den Vergleich in Kapitel 6.3 erkennbar. Eine Kombination aus Brücken- und Tunnelmodellen sowie Straßenflächen ist in Abbildung 9.5 und 9.6 visualisiert. In Abbildung 9.5 ist wiederum die durchgeführte DGM-Korrektur im Brücken- und Tunnelbereich erkennbar. Die Tunnelkorrektur ist auf Grund von Darstellungsproblemen nicht enthalten. Die konkrete Kombination der Modelle und Flächen ist in Abbildung 9.6 zu sehen. Sowohl die Brücken- und Tunnelmodelle sowie die Straßenflächen befinden sich, bis auf modellierungsbedingte Abweichungen, an der vorgesehenen Position (Lage). Allerdings sind die Höhenverläufe des Straßennetzwerkes in den Bereichen der Übergänge von den Modellen zu den Straßen fehlerhaft beziehungsweise unrealistisch. Dieses Problem wird allein durch den Höhenabgriff der Brückenmittelachse vom korrigierten DGM erzeugt (vgl. Kapitel 5.2). Die Straßenflächen sowie die Erzeugung der Brücken- und Tunnelmodelle basieren auf diesen Höhen. Dieser Höhenabgriff bietet noch Verbesserungspotential.

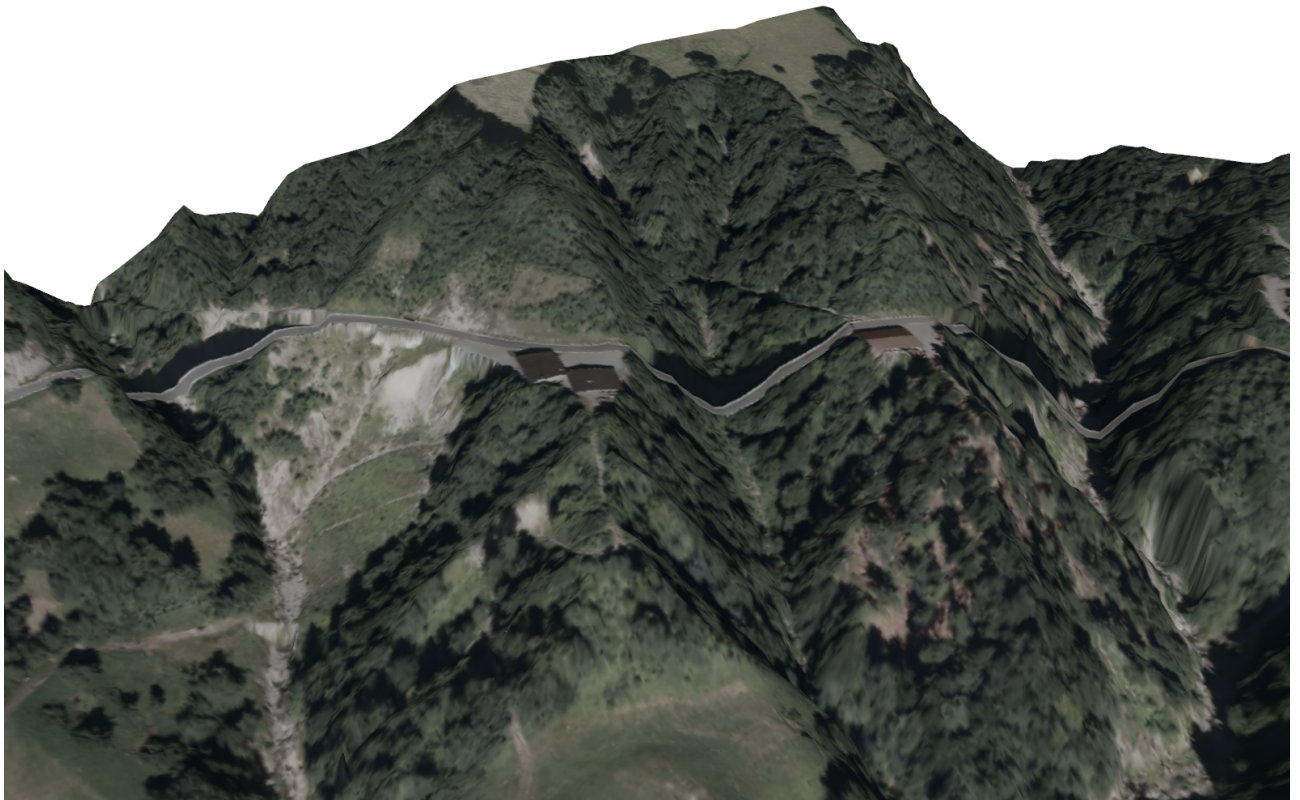


(a) Texturiertes DOM mit Brücken- und Gebäudekörpern sowie Vegetation

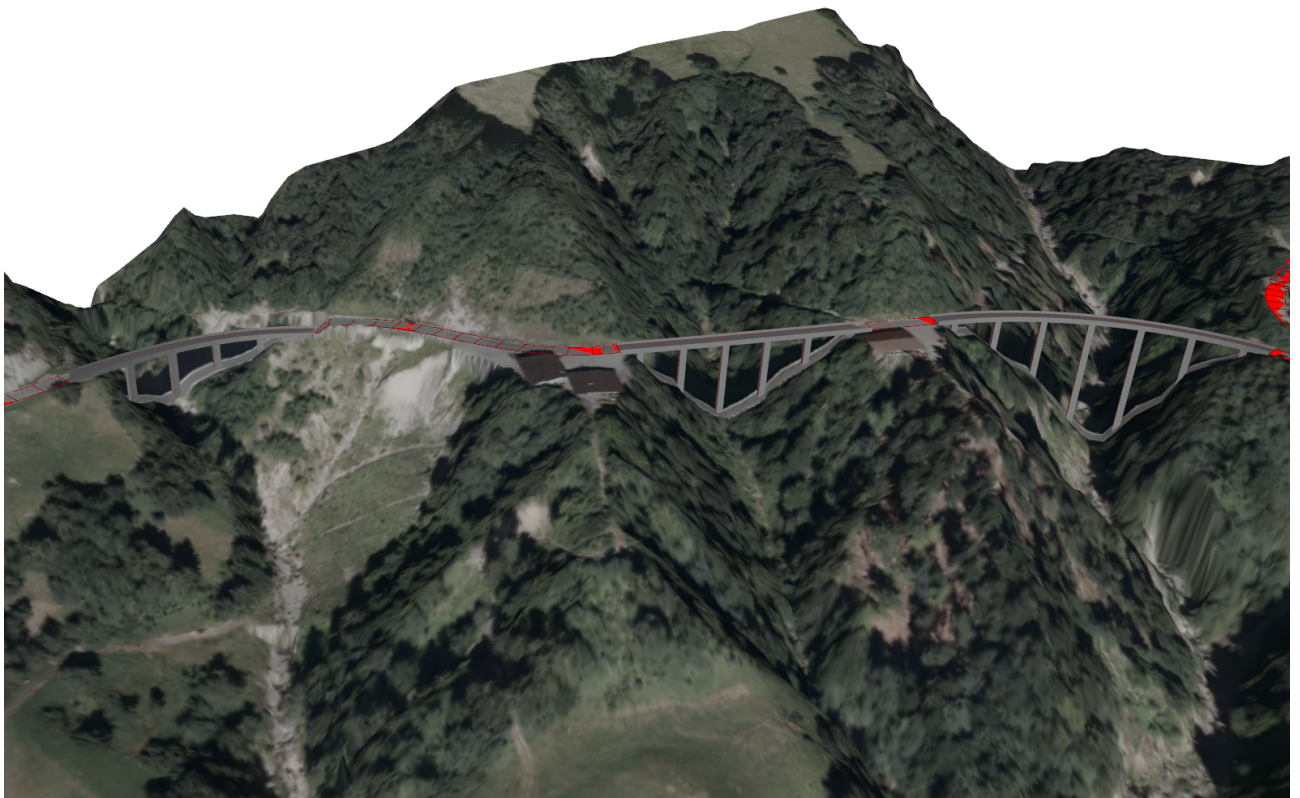


(b) Texturiertes DGM mit den Brückenkörpern

Abb. 9.1 – Ausschnitte des texturierten 3D-Landschaftsmodells (Szene 1a)



(a) Korrigiertes und texturiertes DGM ohne Brückenkörper



(b) Korrigiertes und texturiertes DGM ohne Brückenkörper jedoch mit Brückenmodellen sowie Straßenflächen

Abb. 9.2 – Ausschnitte des texturierten 3D-Landschaftsmodells (Szene 1b)



(a) Texturiertes DGM mit Brückenkörpern

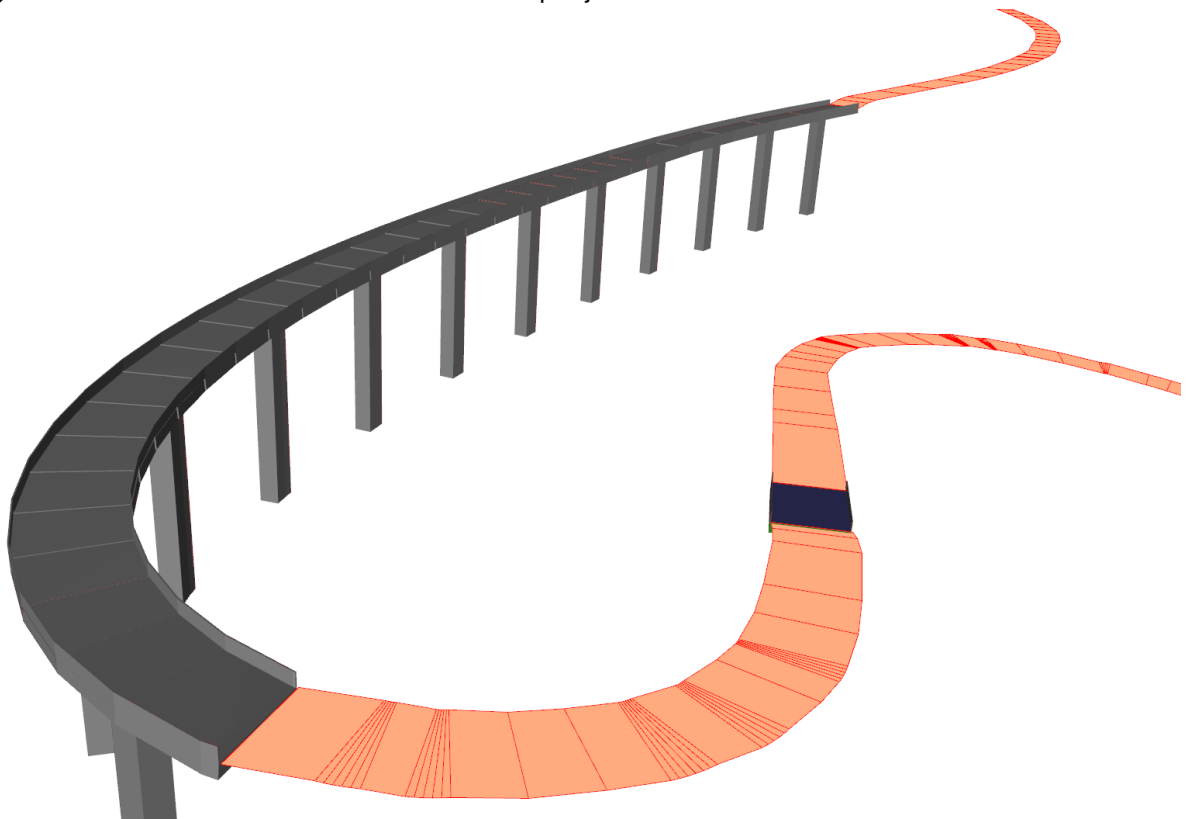


(b) Korrigiertes und texturiertes DGM ohne Brückenkörper

Abb. 9.3 – Ausschnitte des texturierten 3D-Landschaftsmodells (Szene 2a)



(a) Korrigiertes und texturiertes DGM ohne Brückenkörper jedoch mit Brückenmodellen sowie Straßenflächen



(b) Brückenmodelle sowie Straßenflächen

Abb. 9.4 – Ausschnitte des texturierten 3D-Landschaftsmodells (Szene 2b)



(a) Texturiertes DGM mit Brückenkörpern

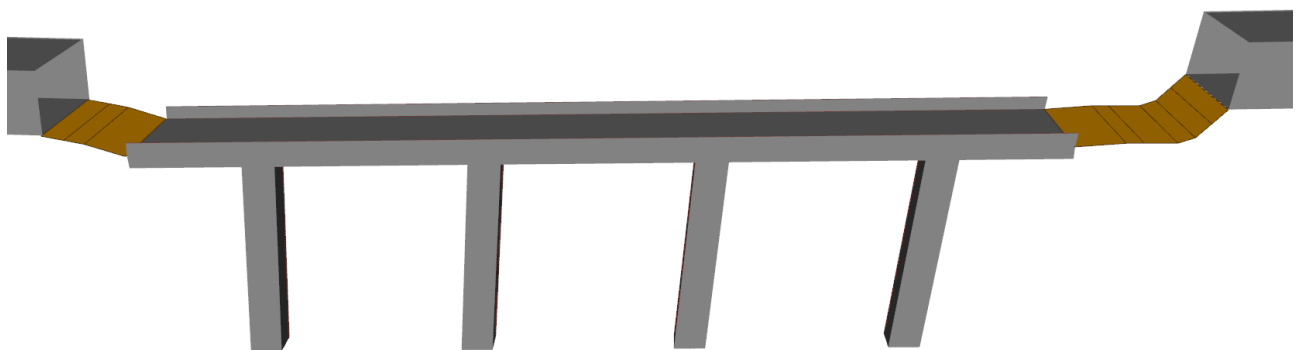


(b) Korrigiertes und texturiertes DGM ohne Brückenkörper

Abb. 9.5 – Ausschnitte des texturierten 3D-Landschaftsmodells (Szene 3a)



(a) Korrigiertes und texturiertes DGM ohne Brückenkörper jedoch mit Brücken- und Tunnelmodellen sowie Straßenflächen



(b) Brücken- und Tunnelmodelle sowie Straßenflächen

Abb. 9.6 – Ausschnitte des texturierten 3D-Landschaftsmodells (Szene 3b)

Weitere Auszüge aus dem Straßennetzwerk sind in Abbildung 9.7 dargestellt.



(a) Straßenflächen (Szene 4a)



(b) Straßenflächen (Szene 4b)

Abb. 9.7 – Ausschnitte des texturierten 3D-Landschaftsmodells

Die 3D-Modelle besitzen neben den Attributen der ursprünglichen Datensätze, gemäß Tabellen 10.8 - 10.11, auch neue Attribute, welche zusätzliche räumliche Informationen beinhalten. So zum Beispiel die minimale und maximale Höhe des jeweiligen Straßenabschnittes oder Brücken- beziehungsweise Tunnelmodells. Die Straßenabschnitte besitzen zudem jeweils Informationen zu minimaler und

maximaler Steigung. Die Gesamtheit der Attribute erlaubt es, räumliche und semantische Abfragen durchzuführen. So können die Daten zum Beispiel mit dem 3D-WebClient visualisiert und ausgewertet werden (vgl. Kapitel 3.4). Ein Szenario für eine solche Abfrage, könnte die Suche nach Straßen sein, welche im Winter schwer befahrbar sind. Hierbei könnten Straßen nach den Attributen der maximalen und minimalen Höhenlage sowie der minimalen und maximalen Steigungen gefiltert werden. Weitere Visualisierungen des korrigierten DGM sowie des Straßennetzwerkes sind im Anhang in den Abbildungen 10.2 - 10.17 dargestellt.

Neben den semantischen Informationen eines Straßennetzwerkes und der graphischen Darstellung der Geometrie, ist auch die Nutzung für Routenplaner interessant. Hierbei kann ein digitales Straßennetzwerk Grundlage für die Suche eines kürzesten oder eines optimalen Pfades sein. Dabei wird für die Suche des kürzesten Pfades die reine Länge der Kanten des Netzwerkes betrachtet. Die Suche des optimalen Pfades ist im Vergleich zur Suche des kürzesten Pfades komplexer. Hierbei werden Parameter, wie zum Beispiel Fahrzeit, Fahrstrecke, Energieverbrauch oder maximale Steigungen, mit verschiedenen Gewichtungen zur Suche verwendet [Sanders u. Schultes (2007)]. Ein Beispiel einer Suche nach dem kürzesten Pfad auf Basis des, aus Straßen, Brücken und Tunnel kombinierten Straßennetzwerkes (Liniennetzwerk), ist in Abbildung 9.8 dargestellt. Dabei wird der kürzeste Pfad zwischen einem Anfangs- und Endpunkt gesucht. Hierbei führt der kürzeste Weg auch konkret über eine Brücke (Zwischen Abschnittsnummer 34 und 36). Die für den Pfad verwendeten Straßennetzwerkabschnitte sind als Pfeile mit deren fortlaufenden Nummern abgebildet. Die konkrete Suche wurde mit der Software *FME* durchgeführt, wobei das Straßennetzwerk nur 2D ausgewertet wird. Um eine Suche nach dem optimalen Pfad zu ermöglichen, müssten zunächst Attribute wie Fahrzeit oder Energieverbrauch je Kante berechnet werden. Die Fahrzeit könnte bei diesem Straßennetzwerk durch die Nutzung der Kantenlänge und der Straßenklassifizierung ermittelt werden. So könnten zum Beispiel für Ortsstraßen eine maximale Geschwindigkeit von 50 Kilometer pro Stunde (km/h) und für Landesstraßen eine maximale Geschwindigkeit von 100 km/h angesetzt werden. Der Energieverbrauch je Kante könnte zum Beispiel mit einem Multiplikator vorliegen. So würde eine Steigung den Energieverbrauch erhöhen und ein Gefälle den Energieverbrauch senken. Die Berechnung eines Multiplikators setzt jedoch voraus, dass bekannt ist, ob die Steigung positiv oder negativ ist. Dies ist mit einem abgespeicherten Wert schwierig, weil je nachdem von welcher Seite die Kante „befahren“ wird, das Vorzeichen verändert werden müsste. Das Straßennetzwerk dieser Arbeit verfügt zwar über die Steigungen je Straßenabschnitt, allerdings ist das Vorzeichen nur in der Orientierung der Linienelemente korrekt. Für eine korrekte Ermittlung des optimalen Pfades, müssten diese Vorzeichen je Kante in der für den Pfad relevanten Orientierung vorliegen. Diese Orientierung ist auch relevant, wenn ein Such-Parameter die maximale Steigung ist. Ohne die korrekte Orientierung kann nicht zwischen starken Steigungen oder starken Gefällen unterschieden werden.

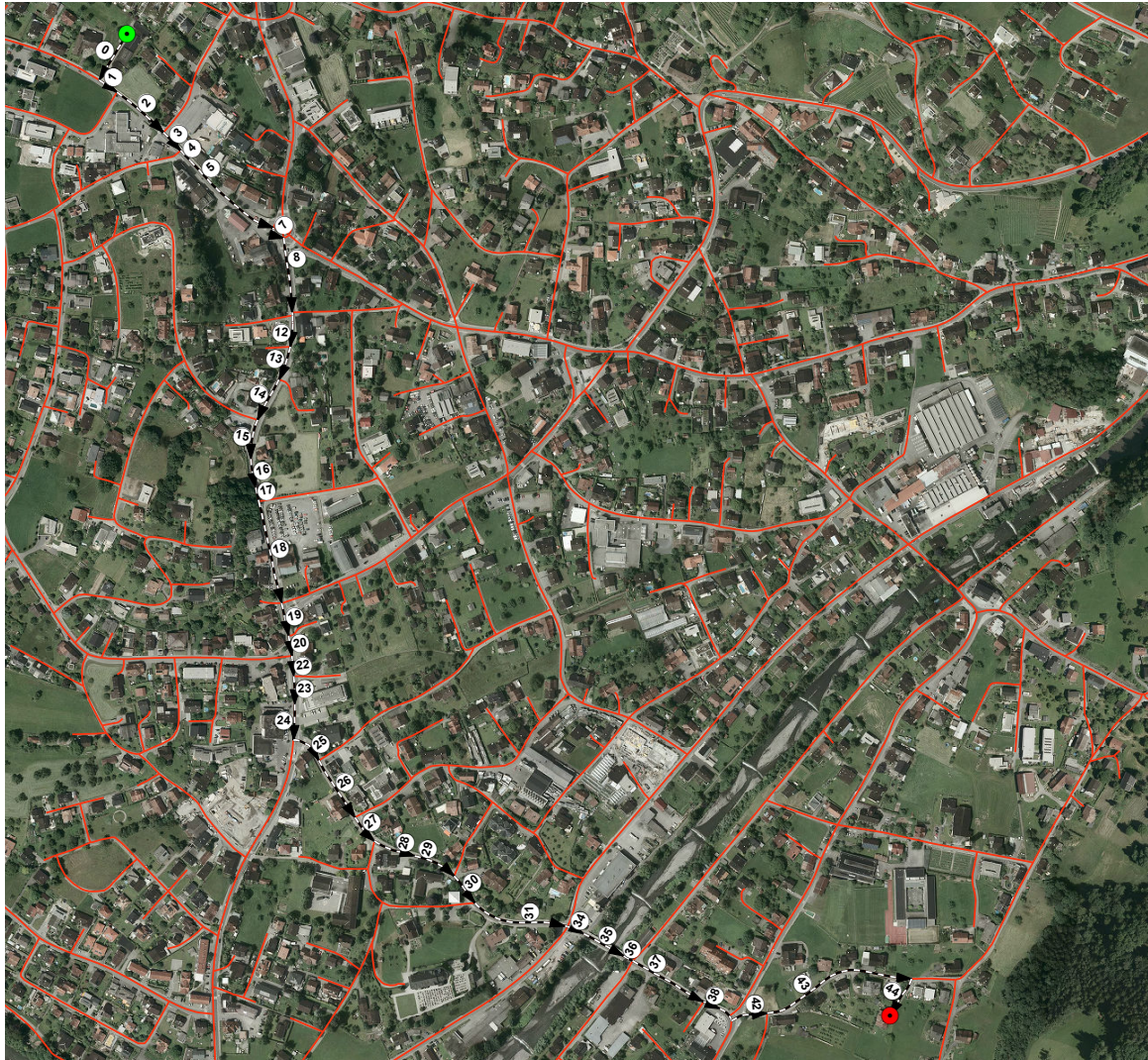


Abb. 9.8 – Beispiel einer Suche nach dem kürzesten Pfad (Pfeile mit Abschnittsnummern) zwischen Anfangs- (Grün) und Endpunkt (Rot)

9.2 Implementierung und Daten

Der größte Teil der praktischen Arbeiten, beziehungsweise die Umsetzung der Konzepte, wird mit der Software FME durchgeführt. Diese Arbeiten lassen sich in fünf Bereiche einteilen.

Zunächst die Extraktion der Straßen-, Brücken- und Tunnelmittelachsen (vgl. Abbildung 10.18).

Im zweiten Bereich erfolgen die DGM-Korrekturen für die Straßen- und Brückenbereiche (vgl. Abbildung 10.20).

Der dritte Bereich, welcher wiederum in fünf Teile unterteilt ist, beinhaltet die Aufbereitung der Kachelgrenzen (vgl. Abbildung 10.21 - 10.25).

Im vierten Bereich erhalten die Straßen-, Brücken- und Tunnelmittelachsen ihre Höheninformationen (vgl. Abbildung 10.26).

Im fünften Bereich erfolgt die abschließende Aufbereitung und Speicherung der Daten gemäß dem

CityGML-Schema (vgl. Abbildung 10.27 - 10.30).

Die konkrete Modellierung sowie die Generierung der Modelle erfolgt mit der Software CityEngine und den CGA-Regeln (vgl. Abbildung 9.9).

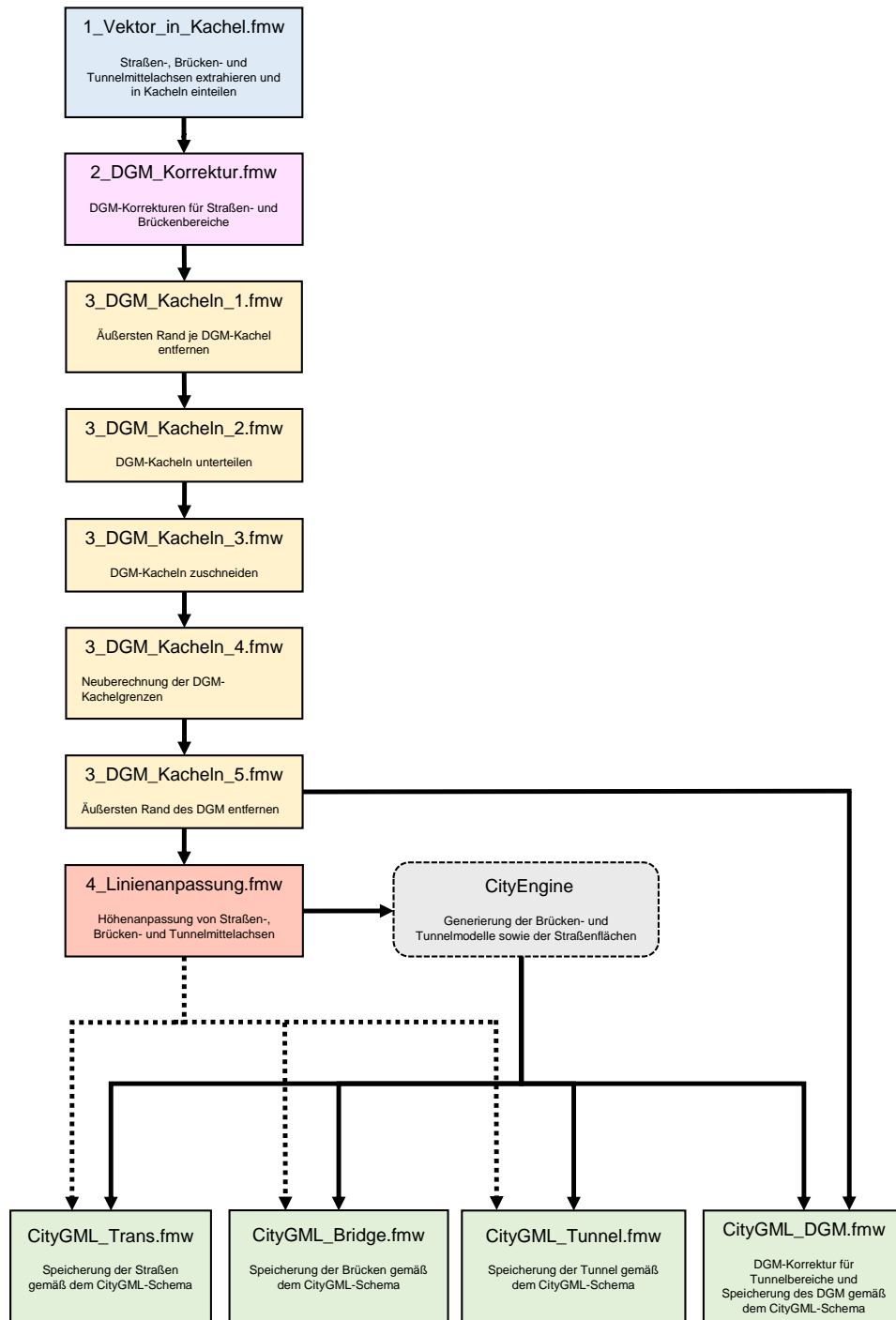


Abb. 9.9 – Überblick über die FME Workspaces und die Nutzung von CityEngine

Die Planung und konkrete Implementierung der verschiedenen Bereiche nimmt unterschiedlich viel Zeit in Anspruch. Den größten Zeitaufwand benötigen, im Rahmen dieser Masterarbeit, die Entwicklung und die Umsetzung der Konzepte der DGM-Korrekturen. Zum einen ist jeder der drei zu korrigierenden Bereiche, Straßen, Brücken und Tunnel, eine komplexe Thematik. Zum anderen ist die Abstimmung der Teilkorrekturen für ein gutes Ergebnis wichtig. Dieser hohe Zeitaufwand lässt sich jedoch mit der entscheidenden Rolle des DGM für die Realisierung eines 3D-Straßennetzwerkes rechtfertigen. Wie in Kapitel 9.1 erläutert, sind die zugrunde liegenden Höheninformationen für eine optisch ansprechende Generierung und Visualisierung der 3D-Straßennetzwerke von großer Bedeutung. Neben der reinen Visualisierung, leidet ebenfalls die Qualität der semantischen Informationen, welche an die 3D-Modelle und Flächen übergeben werden.

Die Eigenschaften der, im Rahmen dieser Masterarbeit, erzeugten Daten sind in Tabelle 9.2 aufgeführt. Neben den konkreten Klassen sind Format, Geometrie, Anzahl der Objekte sowie die Dateigröße je Bereich aufgeführt. Die Ausgangsdaten sind in Tabelle 9.1 aufgelistet.

Datensatz	Datei	Format	Geometrie	Objekte	Dateigröße [MB]
Straßen		Shape			
	landstrasse	Shape	Linie	45	0,174
	ortsstrassen	Shape	Linie	1347	1,48
Brücken		Shape			
	bruecke	Shape	Punkt	31	0,00216
Tunnel		Shape			
	tunnel	Shape	Linie	3	0,0056
DGM		Image			
	gt2011_50cm_*	Image	Raster	28	2550

Tab. 9.1 – Übersicht der verwendeten Ausgangsdaten nach Datei, Format, Geometrie, Anzahl der Objekte und Dateigröße

Datensatz	Klasse	Format	Geometrie	Objekte	Dateigröße [MB]
Straßen		CityGML			12,1/45,3
	<i>TransportationComplex</i>	CityGML	Linie/Fläche	3292/ 3203	
Brücken		CityGML			40,7
	<i>Bridge</i>	CityGML	Körper	31	
	<i>BridgeConstructionElement</i>	CityGML	Körper	429	
	<i>BridgeInstallationElement</i>	CityGML	Körper	130	
	<i>GroundSurface</i>	CityGML	Fläche	93	
	<i>OuterCeilingSurface</i>	CityGML	Fläche	115	
	<i>OuterFloorSurface</i>	CityGML	Fläche	257	
	<i>WallSurface</i>	CityGML	Fläche	4175	
Tunnel		CityGML			5,3
	<i>Tunnel</i>	CityGML	Körper	3	
	<i>GroundSurface</i>	CityGML	Fläche	510	
	<i>RoofSurface</i>	CityGML	Fläche	510	
	<i>ClosureSurface</i>	CityGML	Fläche	6	
	<i>WallSurface</i>	CityGML	Fläche	1020	
DGM		CityGML			(in Berechnung)
	<i>BreaklineRelief</i>	CityGML	Linie	> 200 Mio.	

Tab. 9.2 – Übersicht über die erzeugten Datensätze nach Klasse, Format, Geometrie, Anzahl der Objekte und Dateigröße

9.3 Verbesserungsmöglichkeiten

Eine realitätsnahe Generierung von 3D-Modellen macht nur Sinn, wenn möglichst alle, für die Modellierung benötigten Daten vorliegen. Daher sollten Anforderungen an die Daten möglichst früh kommuniziert und definiert werden. Dies beginnt bereits bei der Datenerhebung und führt über die Datenhaltung bis zur eigentlichen Modellierung. Bei Brücken sollten beispielsweise, neben Informationen wie Lage und Höhe der Straßenmittelachse, Pfeiler und sonstiger elementarer Bauteile, auch Ausrichtungen von Bauteilen vorhanden sein. Bei Tunneln sollten beispielsweise, Informationen des Höhenverlaufs erfasst werden, welcher ohne Kenntnis meist nur interpoliert werden kann. Je nach gewünschtem Detailgrad der Modellierung für die spätere Generierung, müssen mehr Informationen vorhanden sein.

Neben den Vektordaten, welche überwiegend Lageinformationen beinhalten, sind verlässliche Höheninformationen für die Generierung eines 3D-Straßennetzwerkes wichtig. Diese Höheninformationen werden aus dem DGM gewonnen, welche zuvor korrigiert wird. Um die Qualität der Höheninformationen steigern zu können, bedarf es an verfeinerten Konzepten zur DGM-Korrektur.

Im Zusammenhang mit der Modellierung und dem Übergang der generierten Modelle und Flächen in das CityGML-Schema mit der Software CityEngine, ergibt sich ebenfalls Verbesserungspotenzial. Um Straßenflächen realitätsnah, also ohne Generalisierungen und Glättungen zu verwenden, erzeugen zu können, bietet CityEngine begrenzte Möglichkeiten. Die in Kapitel 5.3 erläuterten Probleme an steilen und engen Kurven bedürfen einer verbesserten Modellierung. Ein weiteres Problem sind die initialen Shapes, welche für die Erzeugung der Brückenmodelle erforderlich sind. Gerade bei komplexeren Bauweisen ist eine Vereinigung aller initialen Shapes eines Modells notwendig. Dies ist innerhalb der Software CityEngine nur durch eine Generalisierung möglich. Diesbezüglich wäre es wünschenswert initiale Shapes auf Basis von semantischen Informationen gruppiert bearbeiten zu können. Ebenfalls wäre eine Unterstützung von Spline Geometrien für die Modellierung von linienförmigen Bauwerken mit Steigungs- und Richtungsänderungen praktisch. Um die semantischen Informationen, zum Beispiel Bauteile oder Flächenarten, der Modelle, ohne aufwändige Verarbeitungsschritte, erhalten zu können, wäre auch ein direkter Export gemäß dem CityGML-Schema wünschenswert.

10 Fazit und Ausblick

Ziel dieser Masterarbeit war es, Konzepte zur prozeduralen Generierung von 3D-Straßennetzwerken zu entwickeln und zu bewerten. Neben der Entwicklung und Bewertung der Konzepte wurden diese auch implementiert und erfolgreich auf das Testgebiet angewandt. Das Resultat ist ein semantisch und räumlich auswertbares 3D-Straßennetzwerk, bestehend aus Straßenflächen sowie Brücken- und Tunnelmodellen, welches im CityGML-Schema gespeichert ist.

Die Herausforderung bestand darin, dass die Komponenten des Straßennetzwerkes (Straßen, Brücken und Tunnel) einerseits nur Punkt und Linien Geometrien aufwiesen und zum anderen keinerlei Höheninformationen besaßen. Um diese Herausforderungen meistern und das Ziel erreichen zu können, mussten mehrere Konzepte für vier Teilbereiche entwickelt werden.

Den ersten Teilbereich stellte die Erzeugung der Straßen dar. Diese galt es, ausgehend von einer 2D-Straßenmittelachse, in ein linien- sowie flächenhaftes Straßennetz zu übertragen. Die Erzeugung des linienhaften Straßennetzes verlief ohne größere Hindernisse. Allerdings ergaben sich bei der Erzeugung des flächenhaften Straßennetzes mehrere Probleme, welche im Rahmen dieser Arbeit nicht gelöst werden konnten. Dies waren vor allem die fehlerhaften Flächenerzeugen an Kreuzungen sowie engen und steilen Kurven. Die abschließende Speicherung der Straßen gemäß dem CityGML-Schema funktionierte gut, sodass ein annähernd flächendeckendes Straßennetz erzeugt wurde.

Im zweiten und dritten Teilbereich galt es die Brücken- und Tunnelkörper zu modellieren und zu generieren. Hierbei mussten zunächst Brücken- und Tunnelmittelachsen extrahiert und aufbereitet werden, auf deren Basis die Brücken und Tunnel generiert wurden. Bei der Modellierung ermöglichten die prozeduralen CGA-Regeln einen hierarchischen Aufbau der 3D-Modelle. Dabei wurden sieben verschiedene Brückenbauweisen und eine Tunnelbauweise in CGA-Regeln implementiert. Allerdings wurden bei der Modellierung Grenzen erreicht, weswegen die gewünschten Bauweisen nicht vollständig umgesetzt werden konnten. Hierbei lassen sich zum Beispiel schlecht Gewölbebrücken in Kurven modellieren. Hier besteht bei der prozeduralen Modellierung der Software CityEngine Entwicklungs- und Verbesserungsbedarf. Eine weitere Herausforderung war der Erhalt und der Export der 3D-Modelle inklusive der semantischen Informationen. Um diese semantischen Informationen, zum Beispiel über die Hierarchie der 3D-Modelle, weiterverarbeiten zu können, wurde ein Konzept entwickelt, welches Bauteil- und Flächenzugehörigkeiten über einen Farbschema identifiziert. Anhand dieses Konzeptes wurden die 3D-Modelle gemäß dem CityGML-Schema abgespeichert. Um den Übergang in das CityGML-Schema zu erleichtern und die semantischen Modellinformationen zu bewahren, wäre eine direkte CityGML-Export Funktionalität aus der Software CityEngine sinnvoll.

Der letzte Teilbereich umfasste die Korrektur und Aufbereitung des DGM. Hierbei galt es mehrere Probleme zu lösen. Zum einen wurde ein Konzept erarbeitet, welches das DGM im Bereich der Straßen auf Längs- und Querneigung überprüft und falls notwendig korrigiert. Zum anderen wurden zwei Konzepte entwickelt um Brückenkörper aus dem DGM zu entfernen. Um die Probleme im Bereich von Tunnelportalen zu bereinigen, wurde ebenfalls ein Korrekturkonzept vorgestellt. Hierbei ist zu erwähnen, dass das DGM bei der Generierung eines 3D-Straßennetzwerkes eine entscheidende Rolle spielt, da sämtliche Höheninformationen davon abgeleitet werden. Die Entwicklung und Implementierung der Konzepte der DGM-Korrekturen erforderte viel Zeit und bietet weiterhin Verbesserungspotential und Forschungsarbeit.

Abschließend kann festgestellt werden, dass das Ziel dieser Masterarbeit erreicht wurde. Die vorgestellten Konzepte liefern ein nutzbares semantisches 3D-Straßennetzwerk, ohne über direkte Höheninformationen der Ausgangsdaten zu verfügen. In Zukunft können die Konzepte auch für andere Datensätze implementiert, werden um die Verbreitung und Herstellung flächendeckender semantischer 3D-Straßennetzwerke zu ermöglichen.

Literaturverzeichnis

- [AdV 2012] ADV: Produktstandard für digitale Geländemodelle (ATKIS - DGM) Version 2.0. (2012). <http://www.adv-online.de/icc/extdeu/binarywriterservlet?imgUid=bc6703a1-9cc0-0731-0577-06e172e13d63&uBasVariant=11111111-1111-1111-1111-111111111111>. – Stand: 17.12.2015
- [Chen u. a. 2008] CHEN, Guoning ; ESCH, Gregory ; WONKA, Peter ; MÜLLER, Pascal ; ZHANG, Eugene: Interactive Procedural Street Modeling. In: *ACM Trans. Graph.* 27 (2008), August, Nr. 3, 103:1–103:10. <http://dx.doi.org/10.1145/1360612.1360702>. – DOI 10.1145/1360612.1360702. – ISSN 0730–0301
- [ESRI 2015a] ESRI: *Grundlegendes zu FME Workbench @ONLINE*. <http://help.arcgis.com/de/arcgisdesktop/10.0/help/index.html#/004m0000000t000000>. Version: April 2015
- [ESRI 2015b] ESRI: *Unterstützte Dateiformate für Raster-Datasets @ONLINE*. <http://help.arcgis.com/de/arcgisdesktop/10.0/help/index.html#/009t0000000q000000>. Version: April 2015
- [FII 2010] FII, ForestryInnovationInvestment.: *WOOD IN TRANSPORTATION*. <http://www.naturallywood.com/sites/default/files/Wood-in-Transportation.pdf>. Version: 2010. – Stand: 10.12.2015
- [Galín u. a. 2010] GALIN, E. ; PEYTAVIE, A. ; MARÉCHAL, N. ; GUÉRIN, E.: Procedural Generation of Roads. In: *Computer Graphics Forum* 29 (2010), Nr. 2, 429–438. <http://dx.doi.org/10.1111/j.1467-8659.2009.01612.x>. – DOI 10.1111/j.1467-8659.2009.01612.x. – ISSN 1467–8659
- [Gang u. Guangshun 2010] GANG, Li ; GUANGSHUN, Shi: Procedural modeling of urban road network. 1 (2010), S. 75–79
- [Geißler u. Geißler 2014] GEISSLER, Karsten ; GEISSLER, Karsten: *Brückenentwurf*. (2014), 1–165. <http://dx.doi.org/10.1002/9783433603437.ch1>. – DOI 10.1002/9783433603437.ch1. ISBN 9783433603437
- [Lehrstuhl für Geoinformatik TU München 2015] GEOINFORMATIK TU MÜNCHEN, Deutschland Lehrstuhl f.: *3DCityDB Webclient*. (2015). – Stand: 26.11.2015
- [Gips 1999] GIPS, James: Computer implementation of shape grammars. 55 (1999), S. 56

- [Gröger u. a. 2012] GRÖGER, Gerhard ; KOLBE, Thomas H. ; NAGEL, Claus ; HÄFELE, Karl-Heinz: OGC City Geography Markup Language *CityGML* Encoding Standard. In: *Open Geospatial Consortium* (2012), Nr. OGC 12-019
- [Haack u. Schaefer 2015] HAACK, A ; SCHAEFER, M: Geschichtliche und neuzeitliche Entwicklung des Verkehrstunnelbaus in Deutschland. In: *Bauingenieur* 90 (2015), Nr. 7/8
- [Hofmann u. Heller 2013] HOFMANN, Dominik ; HELLER, Armin: Prozedurale 3D-Stadtmodellierung mit der ESRI CityEngine. (2013)
- [Jahnke u. a. 2011] JAHNKE, Dipl-Ing M. ; BERGER, Thomas ; KRISP, Jukka: Nicht fotorealistische Darstellung von 3D-Stadtmodellen. In: *HMD Praxis der Wirtschaftsinformatik* 48 (2011), Nr. 3, S. 101–112
- [Kelly u. McCabe 2006] KELLY, George ; MCCABE, Hugh: A survey of procedural techniques for city generation. In: *Institute of Technology Blanchardstown Journal* 14 (2006), S. 87–130
- [Koch 2006] KOCH, Andreas: Semantische Integration von zweidimensionalen GIS-Daten und Digitalen Geländemodellen. Dissertation. Universität Hannover. (2006)
- [Kolbe u. a. 2015a] KOLBE, Thomas H. ; BURGER, Barbara ; CANTZLER, Berit: CityGML goes to Broadway. In: FRITSCH, D. (Hrsg.) ; Institute for Photogrammetry, University of Stuttgart (Veranst.): *Photogrammetric Week '15*. Stuttgart : Wichmann, 2015. – ISBN 978–3–87907–587–4, 343-356. – Project page (incl. data downloads): <http://www.gis.bgu.tum.de/projekte/new-york-city-3d/> Book: <https://www.vde-verlag.de/buecher/537587/photogrammetric-week-15.html>
- [Kolbe u. a. 2015b] KOLBE, Thomas H. ; YAO, Zhihang ; NAGEL, Klaus ; KUNDE, Felix ; WILLKOMM, Philipp ; HUDRA, György ; MÜFTÜOĞLU, Arda ; HERREUELA, Javier: 3D City Database for CityGML - Version 3.0.0 - Documentation. (2015)
- [Müller u. a. 2006] MÜLLER, Pascal ; WONKA, Peter ; HAEGLER, Simon ; ULMER, Andreas ; VAN GOOL, Luc: Procedural modeling of buildings. In: *Acm Transactions On Graphics (Tog)* 25 (2006), Nr. 3, S. 614–623
- [Oude Elberink 2010] OUDE ELBERINK, Sander J.: *Acquisition of 3D topography: automated 3D road and building reconstruction using airborne laser scanner data and topographic maps*. University of Twente, 2010
- [Parish u. Müller 2001] PARISH, Yoav I. ; MÜLLER, Pascal: Procedural modeling of cities. (2001), S. 301–308
- [Päffgen 2015] PÄFFGEN: Material beim Brückenbau. (2015). http://www.brueckenbau.de/informationen/material_beim_brueckenbau/. – Stand: 10.12.2015
- [Reinhardt 2015] REINHARDT, Wolfgang: Digitale Geländemodelle. (2015). http://www.unibw.de/inf4/professuren/geoinformatik/lehre/skripten/skripte/skripten_ft_09/gis2_dgm.pdf. – Stand: 15.12.2015

- [Sanders u. Schultes 2007] In: SANDERS, Peter ; SCHULTES, Dominik: *Engineering fast route planning algorithms*. Springer, 2007, S. 23–36
- [Schilling 2014] SCHILLING, Arne: 3D Spatial Data Infrastructures for Web-Based Visualization. Dissertation. (2014). <http://nbn-resolving.de/urn/resolver.pl?urn=urn:nbn:de:bsz:16-heidok-171969>. – URN urn:nbn:de:bsz:16-heidok-171969
- [Smelik u. a. 2014] SMELIK, Ruben M. ; TUTENEL, Tim ; BIDARRA, Rafael ; BENES, Bedrich: A survey on procedural modelling for virtual worlds. 33 (2014), Nr. 6, S. 31–50
- [Vorarlberg 2014] VORARLBERG, a.: Brückenliste VIIb. (2014). – Nicht veröffentlichte Projekttabellen. Stand: 12.12.2014
- [Yao u. Kolbe 2014] YAO, Zhihang ; KOLBE, Thomas H.: Cloud-basierter Webclient für kollaboratives Arbeiten auf semantischen 3D-Stadtmodellen. (2014). <http://www.dgpf.de/neu/Proc2014/index.html>;

Abbildungsverzeichnis

3.1	<i>scope</i> eines Shapes mit Position P und dem Größen-Vektor S [Müller u. a. (2006)]	6
3.2	Beispielhafte CGA-Regel eines 3D-Modells (Links), Hierarchie (Mitte) sowie resultierendes 3D-Modell (Rechts)	6
3.3	LOD Überblick [Gröger u. a. (2012)]	7
3.4	UML-Diagramm des CityGML Kernmodells mit den Unterklassen von <i>_CityObject</i> [Gröger u. a. (2012)]	8
3.5	UML-Diagramm des Brückenmodells [Gröger u. a. (2012)]	9
3.6	Verschiedene <i>BridgeConstructionElements</i> einer Hängeseilbrücke [Gröger u. a. (2012)]	10
3.7	Verschiedene <i>_BoundarySurfaces</i> einer Brücke [Gröger u. a. (2012)]	10
3.8	UML-Diagramm des Tunnelmodells [Gröger u. a. (2012)]	11
3.9	Verschiedene <i>_BoundarySurfaces</i> von unterschiedlichen Tunnelquerschnitten [Gröger u. a. (2012)]	12
3.10	UML-Diagramm des Transportmodells [Gröger u. a. (2012)]	13
3.11	UML-Diagramm des Digitalen Geländemodells [Gröger u. a. (2012)]	13
3.12	<i>TerrainIntersectionCurve</i> eines Gebäudes und eines Tunnels (links in schwarz, rechts in rot) und eine <i>ClosureSurface</i> eines Tunnels (rechts in grün) [IGG Uni Bonn]	14
3.13	Ausschnitt aus dem 3D-WebClient. Bildquelle: Lehrstuhl für Geoinformatik TU München (2015)	15
3.14	Beispielhafter FME Prozessablauf mit <i>Reader landstrasse</i> , <i>Transformer Bufferer</i> und <i>Writer landstrasse_neu</i>	16
3.15	Zufällig erzeugtes Stadtmodell vom Typ <i>organic</i> (links) und <i>raster</i> (rechts)	17
5.1	Beispiel einer Überlagerung von Straßenmittelachsen	21
5.2	Beispiel einer Höhenanpassung der Straßenmittelachsen. Straßenmittelachsen mit einheitlichen Höhen (schwarz) und angepasste Straßenmittelachsen (orange)	22
5.3	Beispiel einer Erzeugung der Straßenflächen	23
5.4	Verschiedene Kreuzungstypen bei der Straßenflächenmodellierung	24
5.5	Ausschnitt eines Straßennetzwerkes aus Straßenflächen	24
5.6	Ausschnitt eines fehlerhaften Straßennetzwerkes aus Straßenflächen	25
5.7	Ausschnitt einer Straße vor und nach einer Korrektur	26
5.8	Ausschnitt einer fehlerhaft erzeugten Kreuzung	27
5.9	Auszug eines CityGML Dokumentes einer Straße	28

6.1	Konzept der Extraktion von Brückenmittelachsen	29
6.2	Unterteilung des Konzeptes der Extraktion von Brückenmittelachsen	30
6.3	Beispiel einer Extraktion von Brückenmittelachsen	30
6.4	Konzept der Höhenanpassung der Brückenkörper	31
6.5	Unterteilung des Konzepts der Höhenanpassung der Brückenkörper	32
6.6	Beispiel einer Höhenanpassung eines Brückenkörper	32
6.7	Konzept der Höhenanpassung der Brückenpfeiler	33
6.8	UML-Diagramm der Brückenbauweisen mit Unterkategorien nach Projektvorgaben Vorarlberg (2014)	34
6.9	Ausschnitt einer CGA-Regel für Balkentragwerkbrücken (1 von 3)	35
6.10	Ausschnitt einer CGA-Regel für Balkentragwerkbrücken (2 von 3)	36
6.11	Ausschnitt einer CGA-Regel für Balkentragwerkbrücken (3 von 3)	37
6.12	Hierarchie der CGA-Regel <i>Balkentragwerkbrücke</i> ausgehend vom initialen Shape	37
6.13	Abweichungen der Brückenmittelachsen durch die Generalisierung. Abszissenachse mit Bauwerksnummern. Ordinatenachse mit maximalen Abweichungen und mittleren Abweichungen je Brücke in Meter.	38
6.14	Brücken mit geringsten (L073 001, unten) und größten (L073 002, oben) Abweichungen. Verlauf der original Brückenmittelachse (Punkte und Flächen), Verlauf der generalisierten Brückenmittelachsen (grüne Linie) sowie Abweichungsvektoren (blaue Linien).	39
6.15	Eingefärbtes 3D-Modell der CGA-Regel <i>Balkentragwerkbrücke</i>	39
6.16	Eingefärbtes 3D-Modell der CGA-Regel <i>Stahltragwerkbrücke</i>	40
6.17	Eingefärbtes 3D-Modell der CGA-Regel <i>Holztragwerkbrücke</i>	40
6.18	3D-Objekt eines Bogentragwerkbrücken-Unterbaus aus <i>Google SketchUp</i>	41
6.19	Eingefärbtes 3D-Modell der CGA-Regel <i>Bogentragwerkbrücke</i>	42
6.20	Eingefärbtes 3D-Modell der CGA-Regel <i>Gewölbebrücke</i>	42
6.21	Eingefärbtes 3D-Modell der CGA-Regel <i>Rahmentragwerkbrücke</i>	43
6.22	Eingefärbtes 3D-Modell der CGA-Regel <i>Wellblechdurchlass</i>	43
6.23	Fehlerhafte 3D-Brückenmodelle in Kurven	44
6.24	Anwendung der CGA-Regel <i>Gewoebebraeuecke</i> auf eine nicht-generalisierte Brückenmittelachse	44
6.25	Unterteilungsmöglichkeiten einer Brücke gemäß dem CityGML-Schema (vgl. Kapitel 3.2.2): Begrenzungsflächen <i>BoundarySurface</i> (schwarz), Konstruktionselemente <i>BridgeConstructionElement</i> (grün), Installationen <i>BridgeInstallation</i> (blau)	45
6.26	Hierarchischer Aufbau des Farbschemas zur semantischen Identifikation von Brücken	46
6.27	Unterteilung des Konzepts des Übergangs von Brücken in das CityGML-Schema	47
6.28	Auszug eines CityGML Dokumentes einer Brücke	50
7.1	Gegebene Tunnelmittelachsen (Rot) mit Orthofoto	51
7.2	Konzept der Höhenanpassung von Tunneln	52
7.3	Hierarchie der CGA-Regel <i>Tunnelkasten</i> ausgehend vom initialen Shape	53

7.4	Ausschnitt eines eingefärbten 3D-Modells der CGA-Regel <i>Tunnelkasten</i>	54
7.5	Ausschnitt eines eingefärbten 3D-Modells der CGA-Regel <i>Tunnelkasten</i> ohne Abschlussflächen	54
7.6	Hierarchischer Aufbau des Farbschemas zur semantischen Identifikation von Tunneln	55
7.7	Unterteilung des Konzepts des Übergangs von Tunneln in das CityGML-Schema . .	56
7.8	Auszug eines CityGML-Dokumentes eines Tunnels	58
8.1	Konzept der Korrektur der Längsneigung	60
8.2	Unterteilung des Konzeptes der Korrektur der Längsneigung	61
8.3	Beispiel einer DGM-Korrektur der Längsneigung	61
8.4	Konzept der Korrektur der Querneigung	62
8.5	Beispiel einer DGM-Korrektur der Querneigung	63
8.6	Fehlerhafte DGM-Korrekturen der Straßen	63
8.7	Konzept der DGM-Korrektur mit Ansatz 1a	64
8.8	Unterteilung des Konzepts der DGM-Korrektur mit Ansatz 1a	65
8.9	Beispiel der DGM-Korrektur mit Ansatz 1a	65
8.10	Probleme der DGM-Korrektur mit Ansatz 1a	66
8.11	Konzept der DGM-Korrektur mit Ansatz 1b	67
8.12	Unterteilung des Konzepts der DGM-Korrektur mit Ansatz 1b	68
8.13	Beispiel der DGM-Korrektur mit Ansatz 1b	68
8.14	Vergleich der DGM-Korrektur mit Ansatz 1a und 1b	69
8.15	Weiteres Beispiel der DGM-Korrektur mit Ansatz 1b	70
8.16	Beispiele fehlerhafter DGM-Korrekturen im Brückenbereich	71
8.17	Konzept der DGM-Korrektur mit Ansatz 2	72
8.18	Unterteilung des Konzepts der DGM-Korrektur mit Ansatz 2	73
8.19	Beispiel einer Klassifikation mit Ansatz 2. Original DGM mit den Klassen Wasser (Blau-Grün) und Brücken (Orange)	73
8.20	Beispiel der DGM-Korrektur mit Ansatz 2	74
8.21	Entfernung des DGMs im Bereich eines Tunnelportals. Links: DGM vor der Korrektur. Mitte: Korrigiertes DGM. Rechts: Korrigiertes DGM mit Tunnelkörper.	75
8.22	Ausschnitt eines DGM mit fehlerhaften Kachelgrenzen (Der Verlauf der Kachelgrenze ist mit roten Pfeilen markiert)	76
8.23	Ausschnitt einer Anpassung der DGM-Kachelgrenzen	77
8.24	Auszug eines CityGML-Dokumentes eines DGM	78
9.1	Ausschnitte des texturierten 3D-Landschaftsmodells (Szene 1a)	80
9.2	Ausschnitte des texturierten 3D-Landschaftsmodells (Szene 1b)	81
9.3	Ausschnitte des texturierten 3D-Landschaftsmodells (Szene 2a)	82
9.4	Ausschnitte des texturierten 3D-Landschaftsmodells (Szene 2b)	83
9.5	Ausschnitte des texturierten 3D-Landschaftsmodells (Szene 3a)	84
9.6	Ausschnitte des texturierten 3D-Landschaftsmodells (Szene 3b)	85

9.7	Ausschnitte des texturierten 3D-Landschaftsmodells	86
9.8	Beispiel einer Suche nach dem kürzesten Pfad (Pfeile mit Abschnittsnummern) zwischen Anfangs- (Grün) und Endpunkt (Rot)	88
9.9	Überblick über die FME Workspaces und die Nutzung von CityEngine	89
10.1	Überblick über das verwendete Testgebiet mit Kachelgrenzen (Schwarz), Landestraßen (Rot), Ortsstraßen (Gelb), Tunneln (Magenta) und Brücken (Grün)	115
10.2	Caesium Visualisierung des korrigierten und texturierten DGM (1)	116
10.3	Caesium Visualisierung des korrigierten und texturierten DGM (2)	117
10.4	Caesium Visualisierung des korrigierten und texturierten DGM (3)	118
10.5	Caesium Visualisierung des korrigierten und texturierten DGM (4)	119
10.6	Caesium Visualisierung des korrigierten und texturierten DGM (5)	120
10.7	Caesium Visualisierung des korrigierten und texturierten DGM (6)	121
10.8	Caesium Visualisierung des korrigierten und texturierten DGM (7)	122
10.9	Visualisierung des Straßennetzwerkes (1)	123
10.10	Visualisierung des Straßennetzwerkes (2)	123
10.11	Visualisierung des Straßennetzwerkes (3)	123
10.12	Visualisierung des Straßennetzwerkes (4)	124
10.13	Visualisierung des Straßennetzwerkes mit den vorkommenden Brückenbauweisen (Balkentragwerkbrücke)	124
10.14	Visualisierung des Straßennetzwerkes mit den vorkommenden Brückenbauweisen (Gewölbebrücke)	124
10.15	Visualisierung des Straßennetzwerkes mit den vorkommenden Brückenbauweisen (Rahmentragwerk)	125
10.16	Visualisierung des Straßennetzwerkes mit den vorkommenden Brückenbauweisen (Wellblechdurchlass)	125
10.17	Visualisierung des Straßennetzwerkes mit den vorkommenden Brückenbauweisen (Stahltragwerkbrücke)	125
10.18	FME Workspace <i>1_Vektor_in_Kachel.fmw</i>	126
10.19	FME Custom Transformer <i>Extraktion der Brückenmittelachsen</i>	127
10.20	FME Workspace <i>2_DGM_Korrektur</i>	128
10.21	FME Workspace <i>3_DGM_Kacheln_1</i>	129
10.22	FME Workspace <i>3_DGM_Kacheln_2</i>	130
10.23	FME Workspace <i>3_DGM_Kacheln_3</i>	131
10.24	FME Workspace <i>3_DGM_Kacheln_4</i>	132
10.25	FME Workspace <i>3_DGM_Kacheln_5</i>	133
10.26	FME Workspace <i>4_Linienanpassung</i>	134
10.27	FME Workspace <i>CityGML_Trans</i>	135
10.28	FME Workspace <i>CityGML_Bridge</i>	136
10.29	FME Workspace <i>CityGML_Tunnel</i>	137
10.30	FME Workspace <i>CityGML_DGM</i>	138

Tabellenverzeichnis

6.1	CityGML-Attribute der exportierten 3D-Brückenmodelle	48
7.1	CityGML-Attribute der exportierten 3D-Tunnelmodelle	56
9.1	Übersicht der verwendeten Ausgangsdaten nach Datei, Format, Geometrie, Anzahl der Objekte und Dateigröße	90
9.2	Übersicht über die erzeugten Datensätze nach Klasse, Format, Geometrie, Anzahl der Objekte und Dateigröße	91
10.1	<i>codelist</i> des Attributs <i>class</i> der Klasse <i>_AbstractBridge</i>	107
10.2	<i>codelist</i> des Attributs <i>function</i> und <i>usage</i> der Klasse <i>_AbstractBridge</i>	107
10.3	<i>codelist</i> des Attributs <i>class</i> der Klasse <i>_AbstractTunnel</i>	107
10.4	<i>codelist</i> des Attributs <i>function</i> und <i>usage</i> der Klasse <i>_AbstractTunnel</i>	108
10.5	<i>codelist</i> des Attributs <i>class</i> der Klasse <i>TransportationComplex</i>	108
10.6	Attribute der Brückenpunkte „bruecke“	108
10.7	<i>codelist</i> des Attributs <i>function</i> und <i>usage</i> der Klasse <i>TransportationComplex</i>	109
10.8	Attribute der Landesstraßenlinien „landstrasse“	110
10.9	Attribute der Ortsstraßenlinien „ortsstrassen“	110
10.10	Attribute der zusätzlichen Brückeninformationen	111
10.11	Attribute der Tunnellinien „tunnel“	112
10.12	Beispiel einer Höhenanpassung der Brückenpfeiler	112
10.13	Abweichungen der Brückenmittelachsen durch die Generalisierung. Abszissenachse mit Bauwerksnummer. Ordinatenachse mit maximaler Abweichung und mittlerer Abweichung. Gewölbebrücken sind nicht enthalten, da diese nicht generalisiert sind.	113

Abkürzungsverzeichnis

2D	zweidimensional
3D	dreidimensional
3DCityDB	3D City Database
CGA	Computer Generated Architecture
CityGML	City Geography Markup Language
COLLADA	Collaborative Design Activity
DGM	Digitales Geländemodell
DOM	Digitales Oberflächenmodell
ESRI Inc.	Environmental Systems Research Institute
FME	Feature Manipulation Engine
GML	Geography Markup Language
km/h	Kilometer pro Stunde
LAS	Laser
LOD	Level of Detail
L-System	Lindenmayer-System
OGC	Open Geospatial Consortium
SRDBMS	Spatially-enhanced Relational Database Management System
TIN	Triangulated Irregular Network
VCS	Virtual City Systems

Anhang

10.1 Tabellen

Code	Bedeutung
1000	arced bridge
1010	cable-stayed bridge
1020	deck bridge
1030	cable-stayed overpass
1040	truss bridge
1050	pontoon bridge
1060	suspension bridge

Tab. 10.1 – *codelist* des Attributs *class* der Klasse *_AbstractBridge*

Code	Bedeutung
1000	railway bridge
1010	roadway bridge
1030	cable link
1040	canal bridge
1050	aqueduct
1060	foot bridge

Tab. 10.2 – *codelist* des Attributs *function* und *usage* der Klasse *_AbstractBridge*

Code	Bedeutung
1000	traffic
1010	supply
1020	historical
1030	others

Tab. 10.3 – *codelist* des Attributs *class* der Klasse *_AbstractTunnel*

Code	Bedeutung
1000	railway tunnel
1010	roadway tunnel
1020	canal tunnel
1030	pedestrian tunnel

Tab. 10.4 – *codelist* des Attributs *function* und *usage* der Klasse *_AbstractTunnel*

Code	Bedeutung
1000	private
1010	common
1020	civil
1030	military
1040	road traffic
1050	air traffic
1060	rail traffic
1070	waterway
1080	subway
1090	others

Tab. 10.5 – *codelist* des Attributs *class* der Klasse *TransportationComplex*

Attribut	Bedeutung	Einheit
StrRVS	Straßenschlüssel	—
StrKey	Straßenschlüssel	—
StrBez	Straßenbezeichnung	—
Richtung	Trennung der Richtungsfahrbahnen	—
bei_km	Position des Brückenpunktes	km
von_km	Anfang der Brücke	km
bis_km	Ende der Brücke	km
Laenge	Brückenlänge	km
BW_Bez	Bauwerksbezeichnung	—
BW_Nr	Bauwerksnummer	—
Baujahr	Baujahr/Erweiterungen	Jahr
x	Hochwert in Gauß-Krüger Koordinaten	m
y	Rechtswert in Gauß-Krüger Koordinaten	m

Tab. 10.6 – Attribute der Brückenpunkte „bruecke“

Code	Bedeutung	Code	Bedeutung
1000	road	1855	railway track
1010	freeway/motorway	1860	magnetic levitation train
1020	highway/national primary road	1900	railway station
1030	land road	1910	stop
1040	district road	1920	station
1050	municipal road	2000	power-wheel
1060	main through-road	2100	airport
1100	freeway interchange/ highway junction	2110	international airport
1110	junction	2120	regional airport
1200	road	2130	landing place
1210	driveway	2140	heliport
1220	footpath/footway	2150	landing place
1230	hiking trail	2160	gliding airfield
1240	bikeway/cycle-path	2170	taxiway
1250	bridleway/bridlepath	2180	apron
1260	main agricultural road	2190	runway
1270	agricultural road	2200	canal
1280	bikeway/footway	2300	harbor
1290	access road	2310	pleasure craft harbour
1300	dead-end road	2400	ferry
1400	lane	2410	car ferry
1410	lane, one direction	2420	train ferry
1420	lane, both direction	2430	ferry
1500	pedestrian zone	2500	landing stage
1600	place	2600	waterway I order
1610	parking area	2610	navigable river
1620	marketplace	2620	inland navigation waterway 0
1700	service area	2621	inland navigation waterway 0
1800	rail transport	2622	inland navigation waterway I
1805	rail	2623	inland navigation waterway II
1810	urban/city train	2624	inland navigation waterway III
1815	city railway	2625	inland navigation waterway IV
1820	tram	2626	inland navigation waterway V
1825	subway	2627	inland navigation waterway VI
1830	funicular/mountain railway	2628	inland navigation waterway VII
1835	mountain railway	2630	maritime navigation
1840	chairlift	2640	navigable lake
1845	ski-lift/ski tow lift	2700	others
1850	suspension railway		

Tab. 10.7 – codelist des Attributs *function* und *usage* der Klasse *TransportationComplex*

Attribut	Bedeutung	Einheit
VLBG_STR	Straßennummer	—
VLBG_STR_I	Straßennummer	—
STR_KEY_RV	Straßenschlüssel	—
STR_KEY	Straßenschlüssel	—
STR_KEY1	Straßenschlüssel	—
STR_NR	Straßennummer	—
STR_NAME	Straßenname	—
STR_BEZ	Straßenbezeichnung	—
STR_TYP_BE	Straßentyp	—
STR_VON_KM	Anfang der Straße	km
STR_BIS_KM	Ende der Straße	km
STR_LAENGE	Straßenlänge	km
STR_ERG_SC	Straßenschlüssel	—
STR_KATEGO	Straßenkategorie	—
STR_TYP	Straßentyp	—

Tab. 10.8 – Attribute der Landesstraßenlinien „landstrasse“

Attribut	Bedeutung	Einheit
PGEM_NAME	Gemeindename	—
STRLANG	Straßenname lang	—
STRKURZ	Straßenname kurz	—
SKZ	Schulkennzahl	—
STR_KATEGO	Straßenkategorie	—
SKZ_INT	Schulkennzahl	—
STR_TYP	Straßentyp	—
BEARBEITER	Bearbeiter	—
Label_Name	Straßenname	—
ALIAS Name	Straßenname	—
DATUM_ERFA	Erfassungsdatum	Jahr-Monat-Tag
STR_LAENGE	Straßenlänge	km

Tab. 10.9 – Attribute der Ortsstraßenlinien „ortsstrassen“

Attribut	Bedeutung	Einheit
RVS_Key Straßenschlüssel	—	
Spur	Spur	—
Strassenbezeichnung	Straßenbezeichnung	—
km	Position des Brückenpunktes	km
km_von	Anfang der Brücke	km
km_bis	Ende der Brücke	km
x_Koordinate	Hochwert im Gauß-Krüger Koordinaten	m
y_Koordinate	Rechtswert im Gauß-Krüger Koordinaten	m
Brückennummer	Brückennummer	—
Brückenbezeichnung	Brückenbezeichnung	—
KG	Gemeindename	—
Bauart (freies Textfeld)	Bauart	—
Bauweise (Auswahlfeld)	Bauweise	—
Baujahr	Baujahr	Jahr
Anhang Baujahr	Zusätzliche Baujahrs-Informationen	—
Winkel	Winkel	°
Feldanzahl	Feldanzahl	—
Stützweiten schief	Stützweiten schief	m
Stützweiten normal	Stützweiten normal	m
Summe Stützweiten schief	Summe Stützweiten schief	m
größte Einzelstützweite	größte Einzelstützweite	m
gemittelte Fahrbahnbreite	gemittelte Fahrbahnbreite	m
Randbalken links	Randbalken links	m
Randbalken rechts	Randbalken rechts	m
Gesamtbreite	Gesamtbreite	m
Fläche	Fläche	m ²
Denkmalschutz	Denkmalschutz	—
elementargefährdet	Gefährdung durch Natureinfluss	—

Tab. 10.10 – Attribute der zusätzlichen Brückeninformationen

Attribut	Bedeutung	Einheit
StrRVS	Straßenschlüssel	—
StrKey	Straßenschlüssel	—
StrBez	Straßenbezeichnung	—
Richtung	Trennung der Richtungsfahrbahnen	—
von_km	Anfang des Tunnels	km
bis_km	Ende des Tunnels	km
Laenge	Brückenlänge	km
BW_Bez	Bauwerksbezeichnung	—
BW_Nr	Bauwerksnummer	—
Baujahr	Baujahr	Jahr

Tab. 10.11 – Attribute der Tunnellinien „tunnel“

Bauwerksnr.	DGM Höhe [m]	Brückenhöhe [m]	Differenz [m]	Max. Differenz [m]
L051 009	934,83	933,99	0,82	2,00
L051 009	934,75	933,71	1,03	2,00
L051 009	934,73	933,64	1,08	2,00
L051 009	934,61	933,18	1,43	2,00
L051 009	934,55	932,92	1,62	2,00
L051 009	934,41	932,41	2,00	2,00
L051 009	934,38	932,52	1,85	2,00
L051 009	934,28	932,85	1,43	2,00
L051 009	934,23	933,00	1,23	2,00
L051 009	934,20	933,09	1,10	2,00
L051 009	934,04	933,58	0,45	2,00

Tab. 10.12 – Beispiel einer Höhenanpassung der Brückenpfeiler

Bauwerksnummer	minimale Abweichung [m]	maximale Abweichung [m]	mittlere Abweichung [m]
L050 002	0,002	0,002	0,002
L050 003	0,011	0,021	0,016
L050 004	0,024	0,099	0,055
L050 006	0,052	0,157	0,096
L051 001	0,021	0,383	0,227
L051 003	0,010	0,010	0,010
L051 005	0,002	0,002	0,002
L051 008	0,045	0,305	0,172
L051 009	0,005	0,005	0,005
L051 011	0,051	0,112	0,080
L051 013	0,015	0,752	0,296
L051 014	0,065	0,271	0,136
L051 015	0,027	0,681	0,270
L051 016	0,050	0,083	0,069
L051 018	0,088	0,186	0,154
L051 019	0,021	0,223	0,133
L051 020	0,029	0,029	0,029
L051 021	0,051	0,064	0,058
L051 022	0,017	0,146	0,061
L069 001	0,065	0,232	0,125
L069 003	0,095	0,145	0,122
L071 001	0,015	0,015	0,015
L072 001	0,011	0,145	0,078
L073 001	0,001	0,001	0,001
L073 002	0,051	5,925	2,375

Tab. 10.13 – Abweichungen der Brückenmittelachsen durch die Generalisierung. Abszissenachse mit Bauwerksnummer. Ordinatenachse mit maximaler Abweichung und mittlerer Abweichung. Gewölbebrücken sind nicht enthalten, da diese nicht generalisiert sind.

10.2 Visualisierungen



Abb. 10.1 – Überblick über das verwendete Testgebiet mit Kachelgrenzen (Schwarz), Landesstraßen (Rot), Ortsstraßen (Gelb), Tunneln (Magenta) und Brücken (Grün)



Abb. 10.2 – Caesium Visualisierung des korrigierten und texturierten DGM (1)



Abb. 10.3 – Caesium Visualisierung des korrigierten und texturierten DGM (2)



Abb. 10.4 – Caesium Visualisierung des korrigierten und texturierten DGM (3)



Abb. 10.5 – Caesium Visualisierung des korrigierten und texturierten DGM (4)



Abb. 10.6 – Caesium Visualisierung des korrigierten und texturierten DGM (5)



Abb. 10.7 – Caesium Visualisierung des korrigierten und texturierten DGM (6)



Abb. 10.8 – Caesium Visualisierung des korrigierten und texturierten DGM (7)

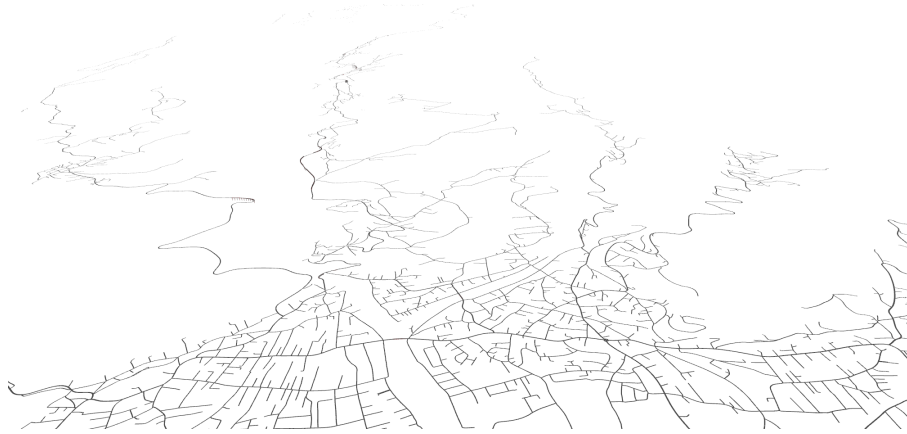


Abb. 10.9 – Visualisierung des Straßennetzwerkes (1)

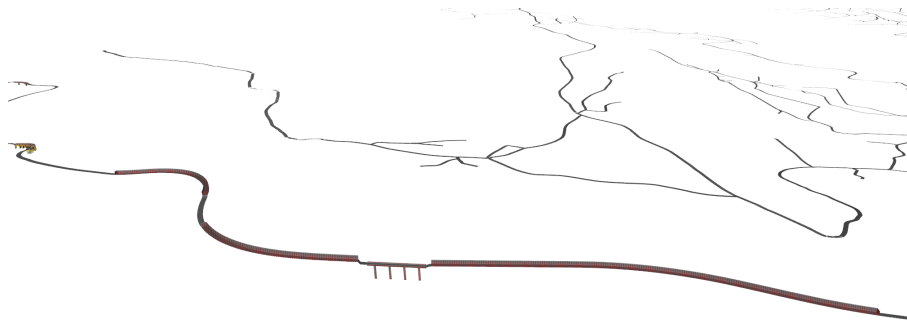


Abb. 10.10 – Visualisierung des Straßennetzwerkes (2)

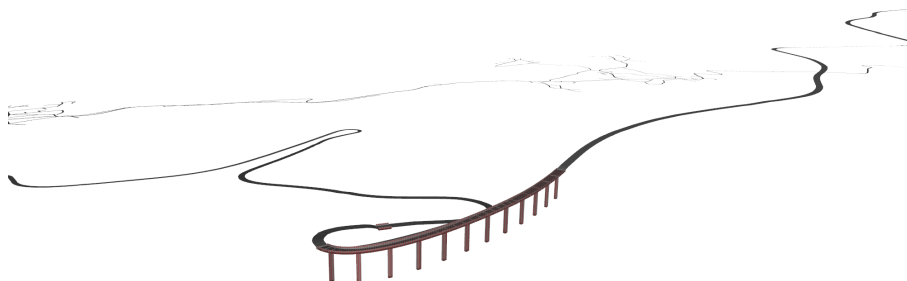


Abb. 10.11 – Visualisierung des Straßennetzwerkes (3)

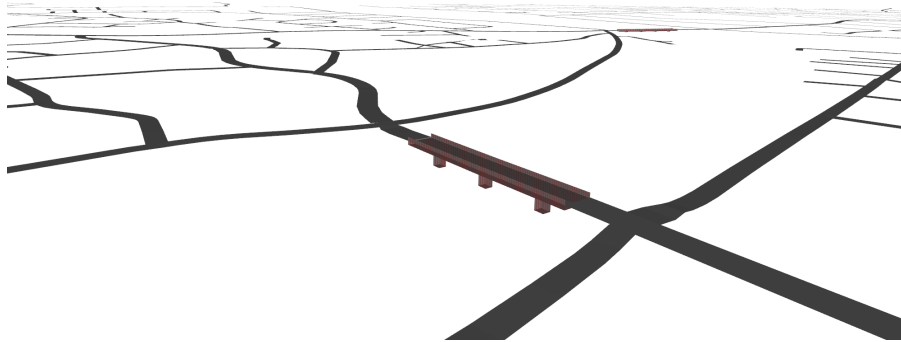


Abb. 10.12 – Visualisierung des Straßennetzwerkes (4)



Abb. 10.13 – Visualisierung des Straßennetzwerkes mit den vorkommenden Brückenbauweisen (Balken-tragwerkbrücke)

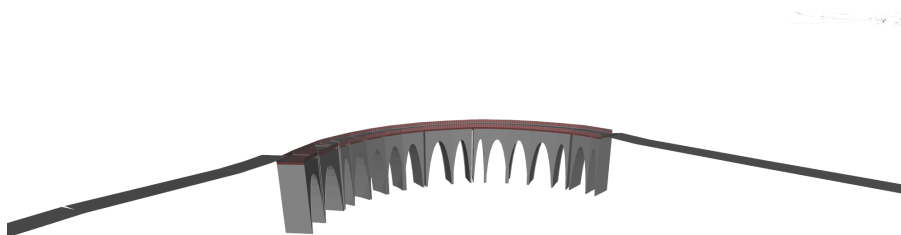


Abb. 10.14 – Visualisierung des Straßennetzwerkes mit den vorkommenden Brückenbauweisen (Gewöl-bebrücke)

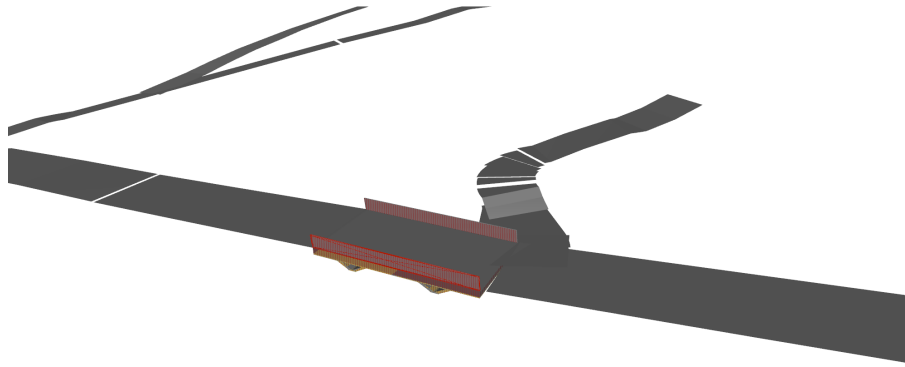


Abb. 10.15 – Visualisierung des Straßennetzwerkes mit den vorkommenden Brückenbauweisen (Rahmentragwerk)

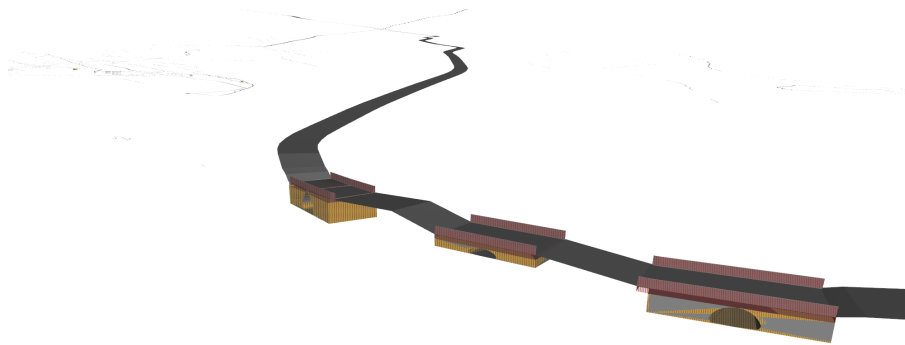


Abb. 10.16 – Visualisierung des Straßennetzwerkes mit den vorkommenden Brückenbauweisen (Wellblechdurchlass)

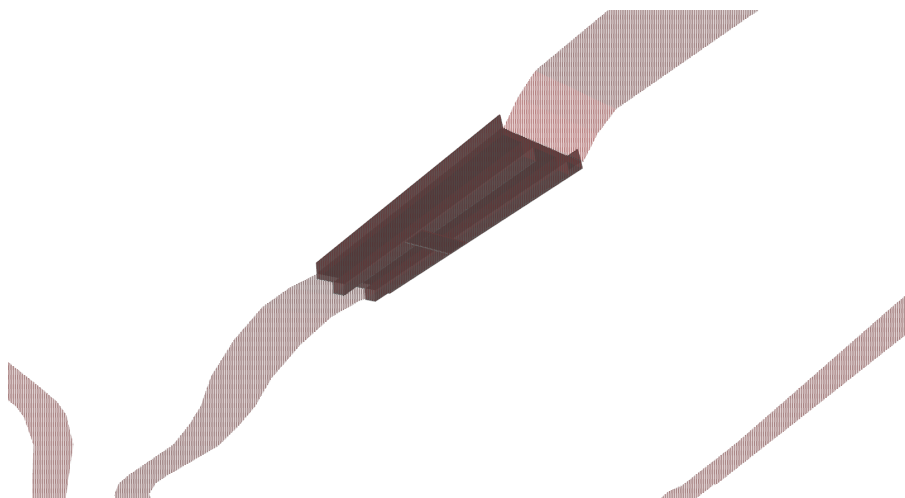


Abb. 10.17 – Visualisierung des Straßennetzwerkes mit den vorkommenden Brückenbauweisen (Stahltragwerkbrücke)

10.3 FME Workspaces

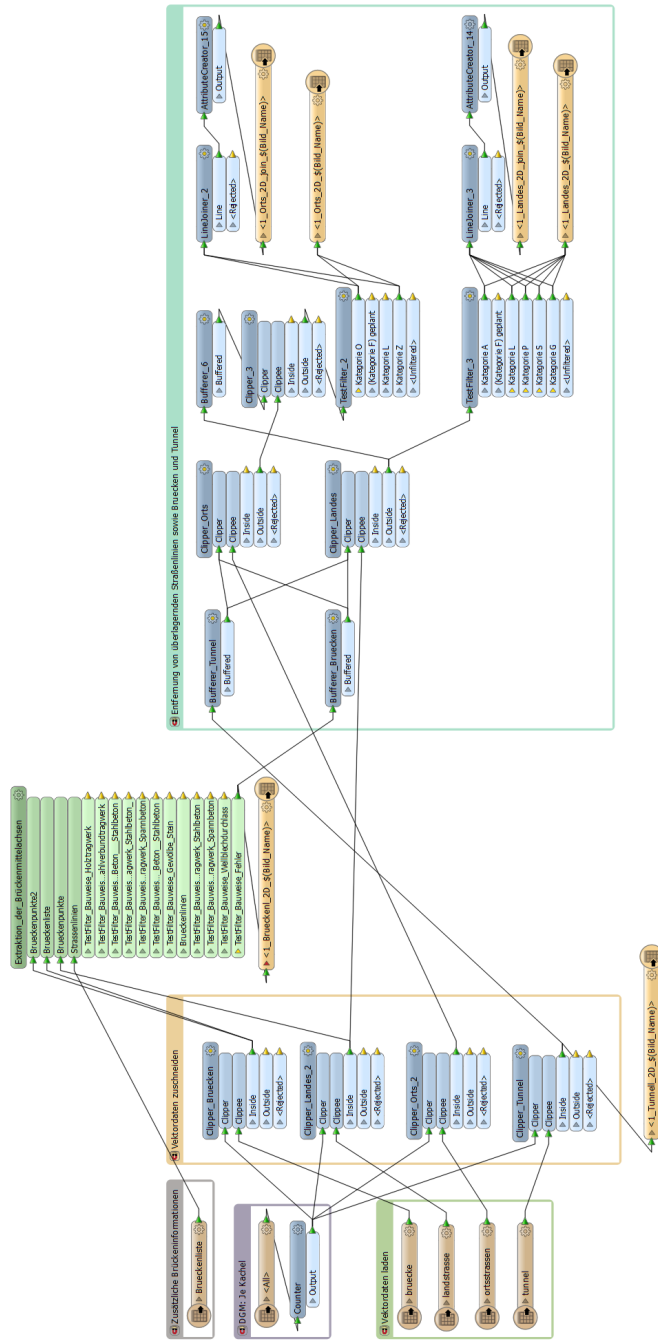


Abb. 10.18 – FME Workspace 1_Vektor_in_Kachel.fmw

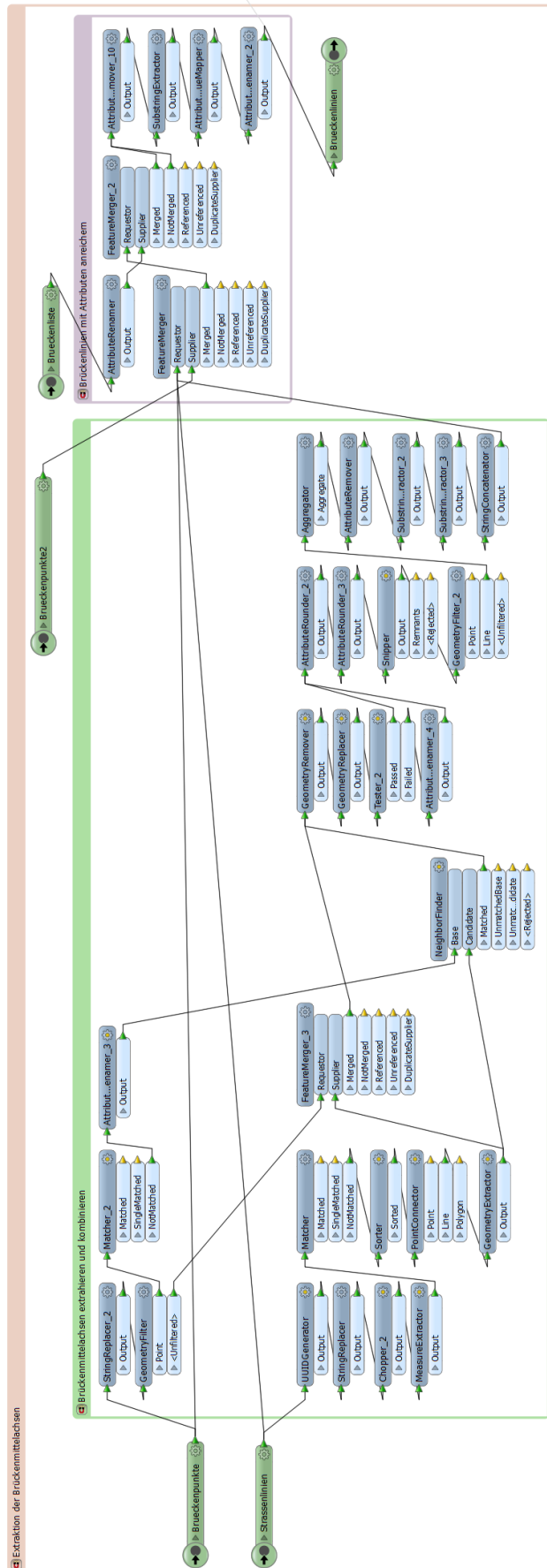


Abb. 10.19 – FME Custom Transformer *Extraktion der Brückenmittelachsen*

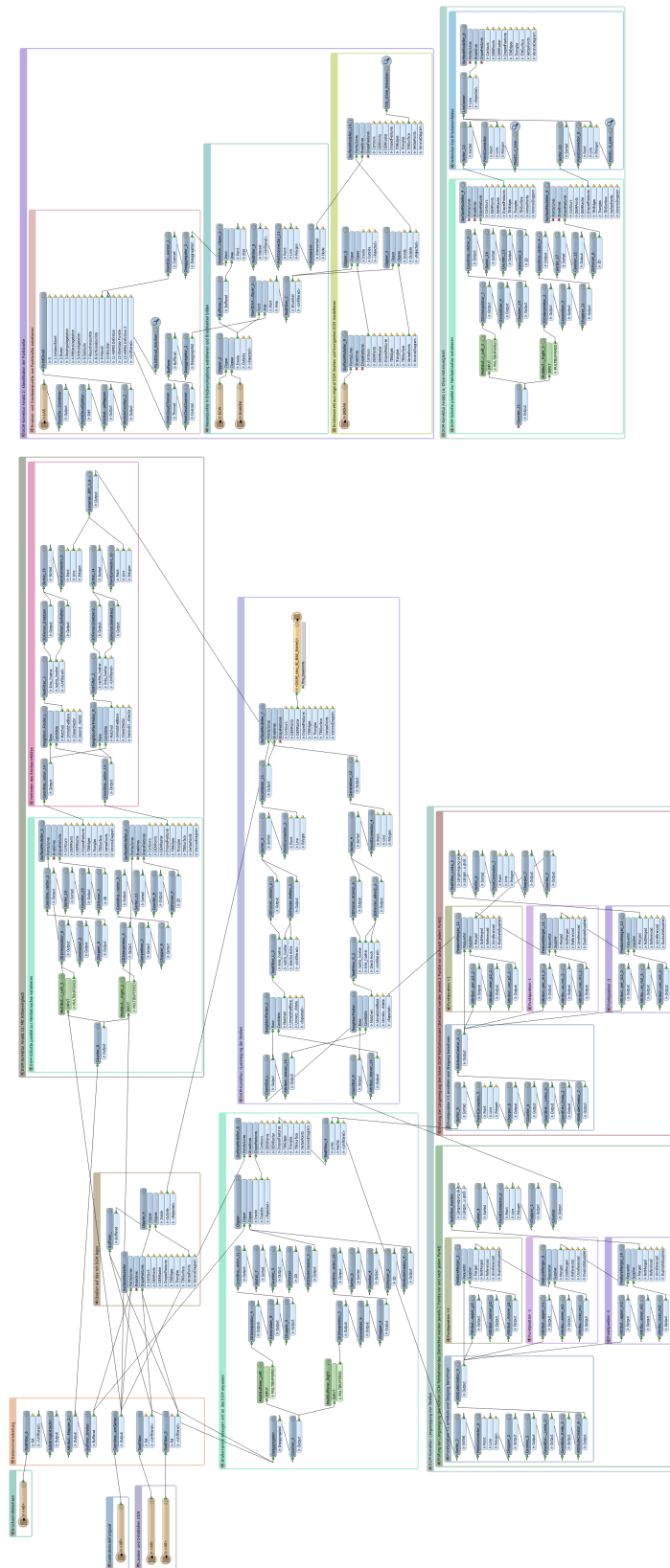


Abb. 10.20 – FME Workspace 2_DGM_Korrektur

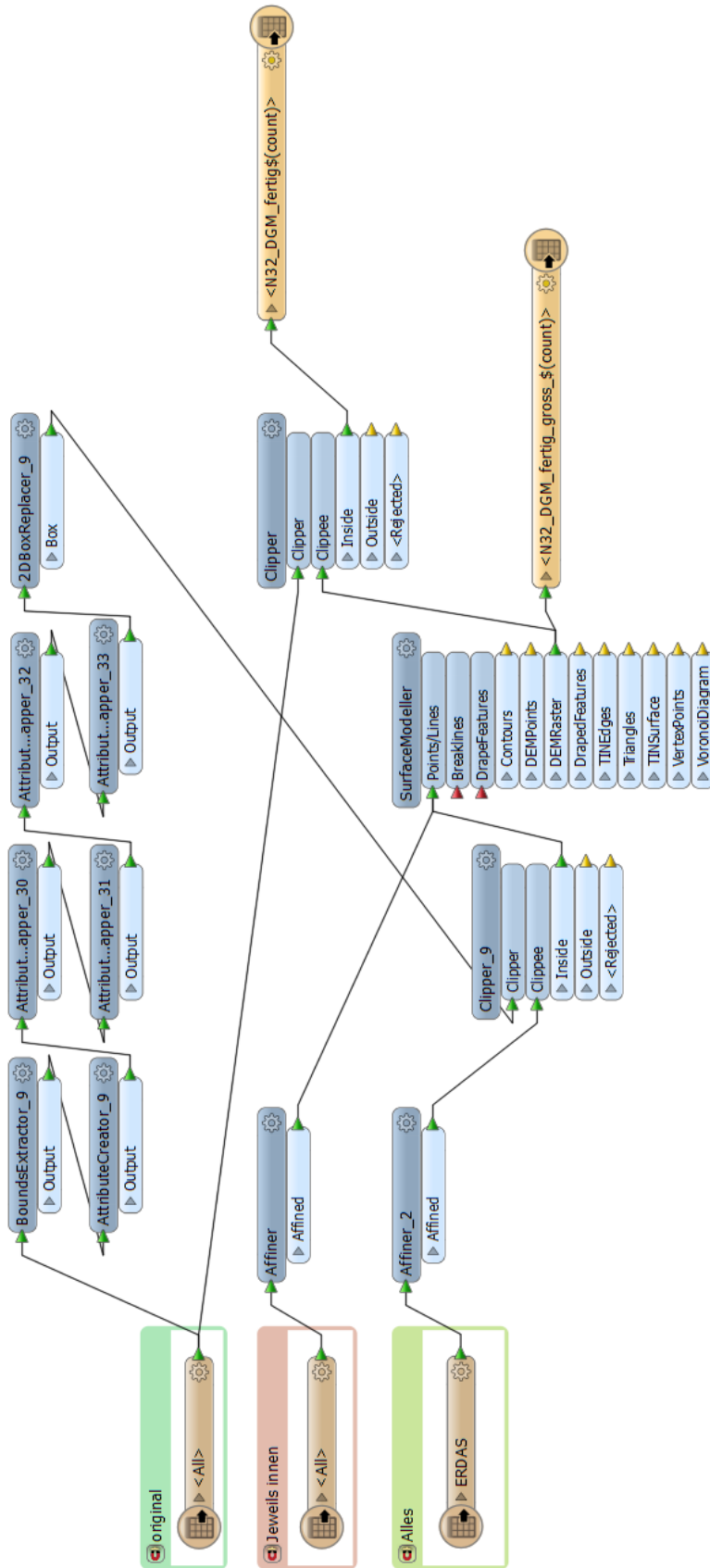


Abb. 10.22 – FME Workspace 3_DGM_Kacheln_2

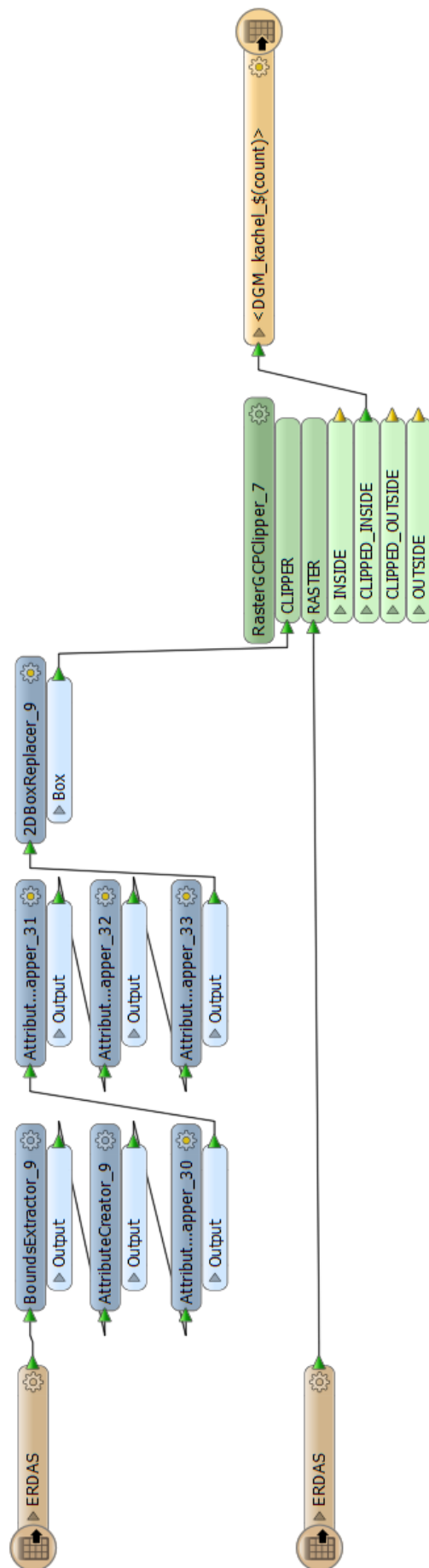


Abb. 10.23 – FME Workspace 3_DGM_Kacheln_3

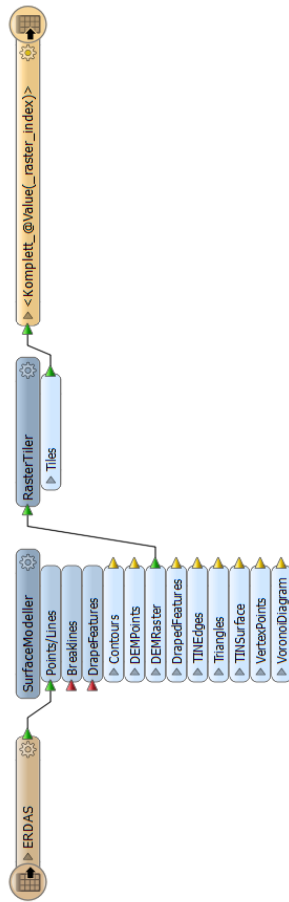


Abb. 10.24 – FME Workspace 3_DGM_Kacheln_4

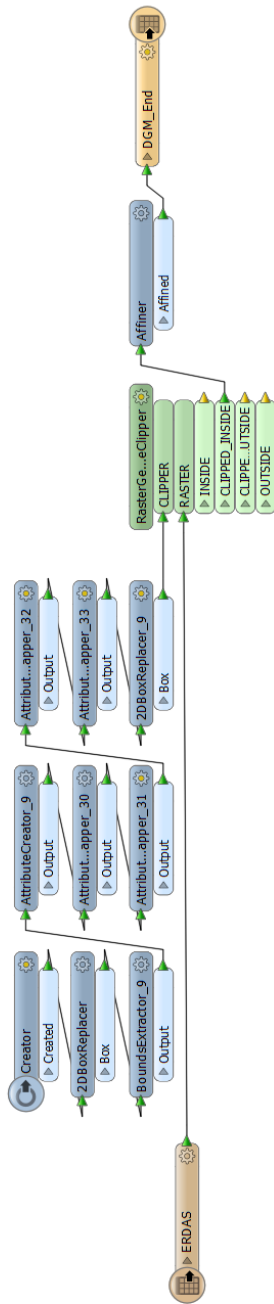


Abb. 10.25 – FME Workspace 3_DGM_Kacheln_5

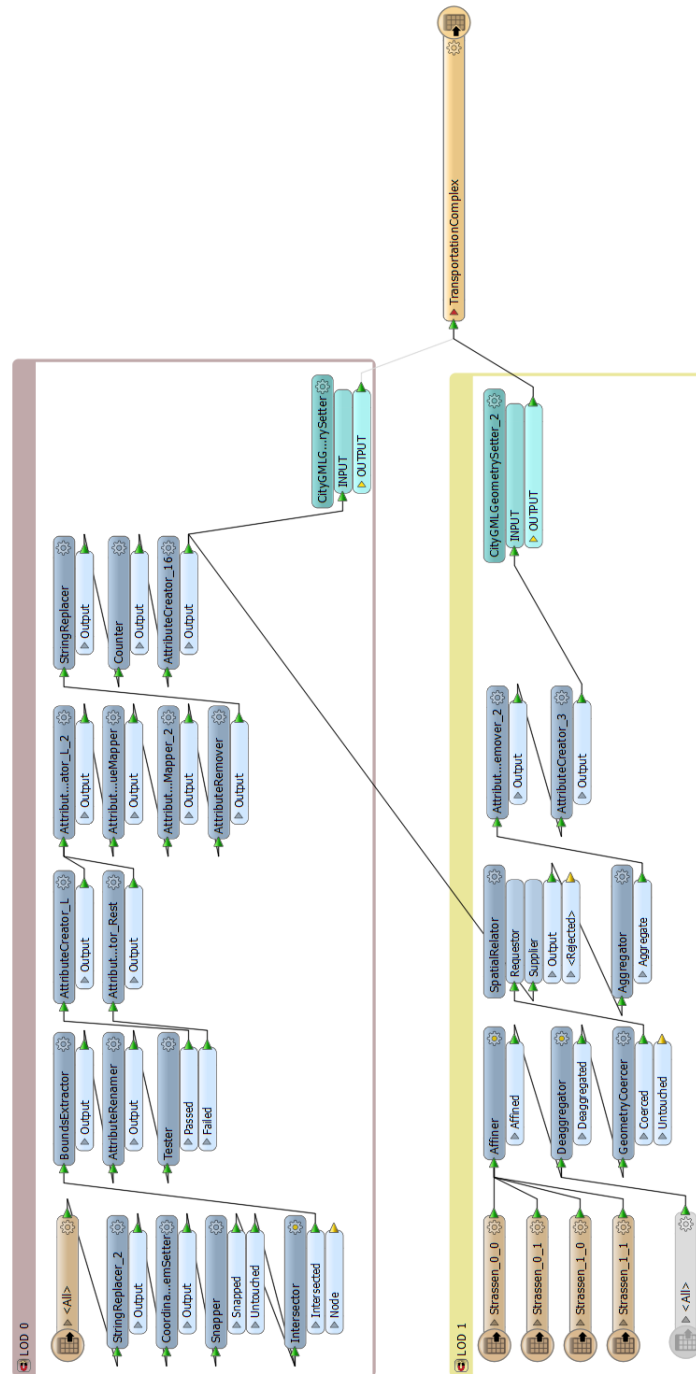


Abb. 10.27 – FME Workspace *CityGML_Trans*

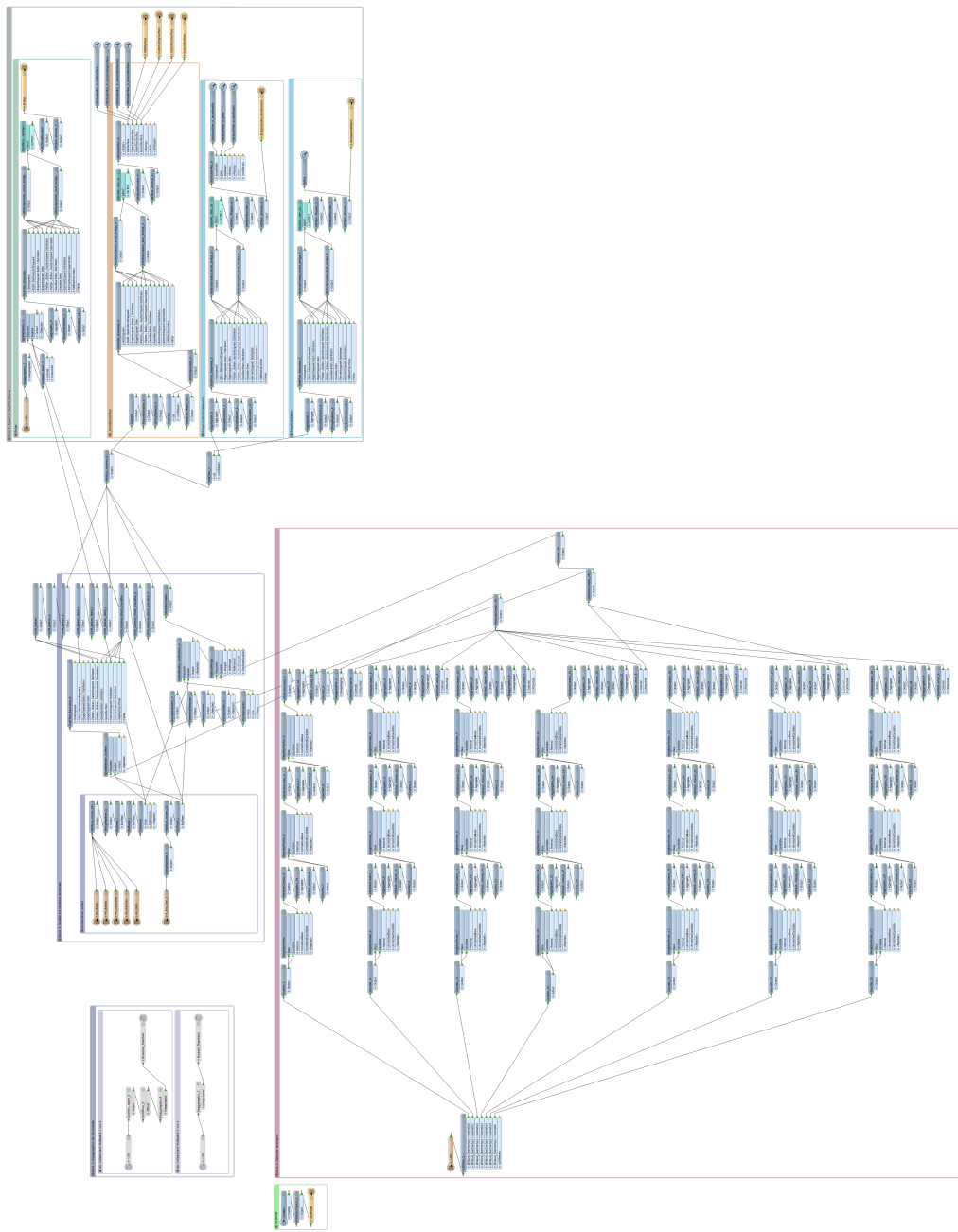


Abb. 10.28 – FME Workspace *CityGML_Bridge*

