

RUNDER TISCH GIS E.V.

Projekt 3D Digitales Landschaftsmodell (3D-DLM) am Runden Tisch GIS e.V.

Abschlussbericht (Demonstrationsphase):
*Datenvorverarbeitung, Anwendung des 3Dfiers,
Abbildung auf CityGML-Datenmodell, Bereitstellung
der Ergebnisdaten & Qualitätsbewertung*

Autoren: **Georg Fiutak** (M.O.S.S. Computer Grafik Systeme GmbH)
Caroline Marx (TU München - Lehrstuhl für Geoinformatik)
Philipp Willkomm (M.O.S.S. Computer Grafik Systeme GmbH)
Andreas Donaubauer (TU München - Lehrstuhl für Geoinformatik)

März 2018

Inhaltsverzeichnis

1	<i>Einführung</i>	5
1.1	Projektbeteiligte und Projektphasen	5
1.2	Ziel der Demonstrationsphase	7
1.3	Testgebiet	8
2	<i>Semantisches 3D-Landschaftsmodell – Mehrwert</i>	9
3	<i>Vorverarbeitung der Daten</i>	13
3.1	Herausforderungen	13
3.2	Allgemeines Vorgehen	14
3.3	Betrachtung der einzelnen Objektarten	14
3.3.1	BUILDING.....	14
3.3.2	BRIDGE.....	15
3.3.3	ROAD	16
3.3.4	WATER.....	17
3.3.5	FOREST	17
3.3.6	Sonstige Objektarten.....	19
3.3.7	TERRAIN.....	19
4	<i>Anwenden des 3Dfiers</i>	20
4.1	Allgemeine Informationen	20
4.1.1	Eingangsdaten.....	20
4.1.2	Ergebnisdaten	20
4.2	Umsetzung	21
4.2.1	Konfigurationsdatei.....	21
4.2.1.1	input_polygons	21
4.2.1.2	lifting_options	21
4.2.1.3	input_elevation	22
4.2.1.4	options.....	22
4.2.1.5	output	22
4.2.2	Prozessschritte	23
4.2.2.1	Einlesen der Konfigurationsdatei	23
4.2.2.2	Analyse der Eingangspolygone.....	23
4.2.2.3	Einlesen der Punktwolke	23
4.2.2.4	Erstellung des 3D-Modells	23
4.2.3	Dauer der Prozessierung	24
4.2.3.1	Gesamtumsetzung	25
4.2.3.2	Parallelisierte Umsetzung	25
4.3	Ergebnisbewertung	27
4.3.1	Allgemeine Aussagen.....	27
4.3.2	Analysen / Auffälligkeiten	27
4.3.2.1	Ausdünnung der Punktwolke	27
4.3.2.2	Triangulation reduzieren (simplification)	29
4.3.2.3	Gebietsrand.....	31
4.3.3	Weitere Anmerkungen.....	32
4.3.3.1	Protokollierung (Logging)	32
4.3.3.2	Zeichenkodierung.....	32
4.3.3.3	Bezugssystem	32
5	<i>Abbildung auf CityGML-Modell</i>	33
5.1	Semantisches Mapping	33

5.2	Geometrie	35
5.3	Zusätzliche Transformationen	35
5.3.1	Setzen des korrekten Koordinatensystems.....	35
5.3.2	Einfügen zusätzlicher Attribute.....	35
6	Austausch der mit dem 3Dfier erzeugten Objektarten	36
6.1	Gebäude	36
6.2	Wald	36
6.3	DGM.....	37
6.3.1	Ausgangsdaten.....	37
6.3.1.1	LDBV	37
6.3.1.2	LGL.....	37
6.3.2	Bearbeitung	37
6.3.3	Ergebnis	39
7	Qualitätsbewertung.....	40
7.1	Umgang mit Inkonsistenzen in Eingangsdaten.....	40
7.1.1	Allgemein.....	40
7.1.2	Problematisch.....	40
7.1.2.1	Nicht freigestellte Objekte	40
7.1.2.2	Lücken in Punktwolke	40
7.1.3	Positiv	41
7.1.3.1	Inkonsistenzen in 2D-Ausgangsdaten	41
7.1.3.2	Lücken im 2D-Datenbestand.....	41
7.2	Verwendung von unklassifizierten Punktwolken	41
7.3	Betrachtung der einzelnen Objektarten	42
7.3.1	BUILDING.....	42
7.3.1.1	Grundlage.....	42
7.3.1.2	Lage	43
7.3.1.3	Bodenhöhe	43
7.3.2	BRIDGE	44
7.3.2.1	Einheitliche Höhe	44
7.3.2.2	Triangulationsprobleme.....	45
7.3.2.3	Brücken querenden Brücken.....	47
7.3.2.4	Brücken in dichter Vegetation	47
7.3.2.5	Einzelpolygone vereinigen	48
7.3.2.6	Zusammenfassung und Fazit.....	48
7.3.3	ROAD	49
7.3.3.1	Lücken im Punktdatenbestand.....	49
7.3.3.2	Geringe Punktdichte.....	50
7.3.3.3	Unterschiedliches Gefälle	51
7.3.3.4	Neigungen im Straßenquerschnitt	51
7.3.3.5	Enger Kurvenverlauf	52
7.3.3.6	Unterscheidung der Verkehrswege.....	53
7.3.3.7	Zusammenfassung und Fazit.....	53
7.3.4	WATER.....	54
7.3.4.1	Stehendes Gewässer	54
7.3.4.2	Fließgewässer	55
8	Bereitstellung der Ergebnisdaten.....	57
9	Zusammenfassung & Ausblick.....	59
	Literaturverzeichnis	61

1 Einführung

1.1 Projektbeteiligte und Projektphasen

Das Projekt „3D Digitales Landschaftsmodell“ (3D-DLM), das am Runden Tisch GIS e.V. durchgeführt wird, besteht aus zwei aufeinander aufbauenden Phasen:

- Phase 1: Konzeptionsphase
- Phase 2: Demonstrationsphase

Die Konzeptionsphase fand bereits im Jahr 2016 statt. Diese Projektphase 1 umfasste die folgenden Themenblöcke:

- 1) Identifizierung potentieller Anwendungsfelder für ein 3D-DLM
- 2) Evaluierung der bereitgestellten Daten
- 3) Identifizierung der Anforderungen an die Datenmodellierung & Semantisches Mapping
- 4) Beschreibung von Methoden zur automatisierten 2D→3D-Transformation zur Überführung in ein konsistentes, semantisches 3D-DLM

Der Abschlussbericht zur Konzeptionsphase kann auf der Homepage des Runden Tisch GIS e.V. heruntergeladen werden¹.

Die Demonstrationsphase folgte im Jahr 2017.

Diese 2. Phase des Projekts erfolgte in Kooperation mit den Auftraggebern 'Landesamt für Digitalisierung, Breitband und Vermessung Bayern' (LDBV), 'Landesamt für Geoinformation und Landentwicklung Baden-Württemberg' (LGL) sowie 'Bundesamt für Landestopographie swisstopo' (swisstopo ist dabei lediglich beratend tätig).

Das 'Bundesamt für Eich- und Vermessungswesen' (BEV), das in der Konzeptionsphase als Auftraggeber beteiligt war, war nicht mehr Bestandteil der Demonstrationsphase.

Im Folgenden sind alle 3D-DLM-Projektbeteiligte der 2. Projektphase aufgeführt:

AUFTRAGGEBER		
	Name	Anmerkungen
Landesamt für Digitalisierung, Breitband und Vermessung Bayern	Dr. Robert Roschlaub	
Landesamt für Geoinformation und Landentwicklung Baden-Württemberg	Manfred Güttlinger	
BERATER		
	Name	Anmerkungen
Bundesamt für Landestopographie swisstopo	Dr. André Streilein	

¹ https://rundertischgis.de/images/2_publicationen/stud_berichte/3D-DLM%20-%20Abschlussbericht.pdf

AUFTRAGNEHMER

	Name	Position
Runder Tisch GIS e.V.	Dr. Gabriele Aumann	Geschäftsführerin RT GIS e.V.

PROJEKTBEARBEITER

	Name	Position
M.O.S.S. Computer Grafik Systeme GmbH	Georg Fiutak Philipp Willkomm	2. Vorstand des RT GIS e.V. und Projektleiter
TU München - Lehrstuhl für Geoinformatik	Prof. Thomas H. Kolbe Dr. Andreas Donaubauer Caroline Marx	1. Vorstand des RT GIS e.V.

Ziel des Projekts ist es, die getrennten Modelle *DGM* und (2D) *DLM* in ein einheitliches und konsistentes 3D-DLM zu überführen. Dabei geht es neben der geometrischen 3D-Modellierung insbesondere auch um die Frage der Erweiterung der fachlichen Modellierung um Objektarten, die vertikale Elemente repräsentieren (z.B. Mauern und Wände von Gebäuden, Brücken und weiteren Bauwerken). Der Weg zu einem solchen dreidimensionalen Modell sollte möglichst durch automatisierte Methoden unterstützt werden.

Die Ausgangssituation bei den Projektpartnern zur Thematik 3D-DLM stellte sich vor Beginn der ersten Projektphase wie folgt dar:

- **BEV und österreichische Bundesländer:** Im Land Vorarlberg wurde bereits 2015 in einer Machbarkeitsstudie die Erstellung und Nutzung eines 3D-DLMs untersucht (*VoDLM3D*).
- **LDBV / LGL:** Bayern und Baden-Württemberg haben neben dem DGM und hochaufgelösten DOMs auch 3D-Gebäudemodelle als Produkte. In der Initiative Baden-Württemberg 4D sollen Geoinformationen sowohl um die dritte Dimension als auch um zeitliche Aspekte erweitert werden.
- **Swisstopo:** In der Schweiz wird mit dem *swissTLM^{3D}* bereits ein entsprechendes Produkt angeboten.

Die genannten Aktivitäten werden bisher weitgehend unabhängig voneinander vorangetrieben. Ein wesentliches Ziel des Projekts ist daher auch der Erfahrungsaustausch, bei dem alle Beteiligten von den Erkenntnissen der anderen Partner profitieren können.

1.2 Ziel der Demonstrationsphase

Aufbauend auf der 1. Phase des 3D-DLM-Projekts erfolgt in der Demonstrationsphase der Aufbau eines konsistenten 3D Digitalen Landschaftsmodells; die festgelegten Objektklassen (siehe Phase 1) werden auf das DGM im LoD1 modelliert und um die 3D-Gebäudemodelle im LoD2 ergänzt.

Diese Modellierung soll mittels des Open Source Tools „3Dfier“² umgesetzt werden. Dieses Werkzeug wurde an der TU Delft entwickelt und wird in den Niederlanden bereits in der Praxis eingesetzt. Dabei wird jedes Polygon auf eine aus dem DGM bzw. dem DOM abgeleitete Höhe angehoben. Ein entscheidender Vorteil ist dabei, dass basierend auf der Semantik die einzelnen Objektklassen mittels angepasster Methodik in die dritte Dimension transformiert werden. So werden zum Beispiel Seen als horizontale Polygone, Gebäude als Klötzchenmodelle oder Straßen als glatte Oberflächen modelliert. Nach Anwendung des 3Dfiers liegt jedes Polygon trianguliert vor. Die extrudierten Flächen werden dann so miteinander „vernäht“, dass ein konsistentes Oberflächenmodell ohne Überschneidungen oder Lücken entsteht.

Konkret bedeutet dies für die Durchführung der Demonstrationsphase, dass die folgenden Arbeitspakete notwendig sind:

1) Vorverarbeitung der Daten

Methoden zur Erzeugung eines konsistenten 2D-DLMs: Dabei soll untersucht werden, wie die oft nur linienhaft vorliegenden Daten (z.B. Verkehrswege) in flächenhafte Daten konvertiert werden können, um eine Tessellation (Landkarte im topologischen Sinn) zu erreichen; ein mögliches Vorgehen könnte sein, die linienhaften Elemente zu puffern (siehe Abschlussbericht Phase 1) und den im 2D-DLM nicht klassifizierten Teil der Erdoberfläche aus dem DGM zu übernehmen.

2) Anwendung des 3Dfiers auf die Daten

- Untersuchung, inwieweit das Tool auf die vorhandenen Daten anwendbar ist
- Ziel ist die prototypische Ableitung eines Kompletmodells

3) Abbildung auf das CityGML-Datenmodell

Es ist zu prüfen, welche CityGML-Klassen vom 3Dfier automatisch erzeugt werden; falls erforderlich, wird im Nachgang eine Anpassung entsprechend der Mapping-Tabelle aus Projektphase 1 vorgenommen bzw. generische Objekte werden den entsprechenden thematischen Klassen zugewiesen. Außerdem muss ein entsprechendes Attribut-Mapping vorgenommen werden.

4) Qualitätsbewertung der erzeugten 3D-Objekte

- Detaillierte Untersuchung, welche Probleme mit dem 3Dfier nicht gelöst werden können (und Aufzeigen, was geändert werden müsste) bzw. wo sich der 3Dfier mit zufriedenstellenden Ergebnissen einsetzen lässt.
- Untersuchung folgender Fragestellungen:
 - o Bis zu welchem Grad kann der 3Dfier beispielsweise mit Inkonsistenzen umgehen?
 - o Welche Ergebnisse können mit unklassifizierten Punktwolken (z.B. bDOMs) erreicht werden?
 - o Welche Fehler treten bei der Verschneidung von DGM bzw. DOM mit dem Basis-DLM auf? Ausblick auf Algorithmen bzw. wo muss die Geometrie des Basis-DLM verbessert werden?

5) Bereitstellung der Ergebnisdaten

Download und Web-basierte 3D-Visualisierung (Demonstrator mit 3DCityDB und WebClient)

² <https://github.com/tudelft3d/3dfier>

1.3 Testgebiet

In Absprache zwischen dem LGL und dem LDBV wurde sich auf ein Testgebiet im Grenzbereich zwischen Baden-Württemberg und Bayern festgelegt. Das Gebiet nordöstlich vom Bodensee umfasst 254km²; die genaue räumliche Ausdehnung kann den folgenden Abbildungen entnommen werden.



2 Semantisches 3D-Landschaftsmodell – Mehrwert

3D-Stadt- und Landschaftsmodelle werden häufig immer noch als reine graphische oder geometrische Modelle definiert; dabei werden die semantischen Aspekte oft vernachlässigt³. Dies hat zur Folge, dass diese Modelle fast ausschließlich für Visualisierungszwecke verwendet werden können, nicht jedoch für thematische Abfragen, Analyseaufgaben oder Simulationen. Für viele Anwendungsszenarien sind aber genau diese semantischen Informationen von enormer Wichtigkeit.

Im Folgenden seien einige Anwendungsfelder vorgestellt, die die semantischen Informationen der einzelnen Stadt- und Landschaftsobjekte benötigen:

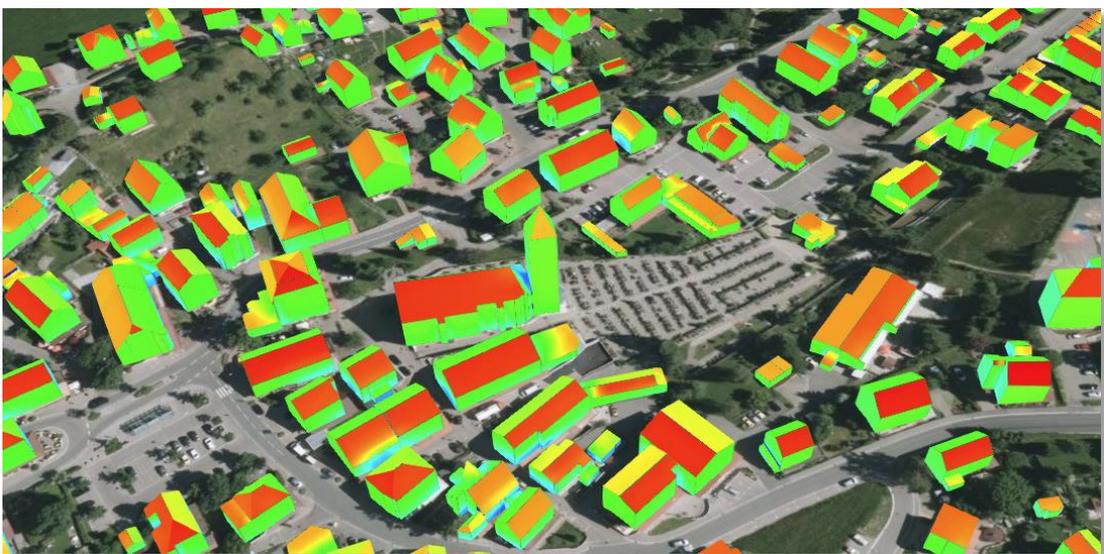
- **Überschwemmung:**

Bei Starkregen und damit einhergehenden Überflutungsereignissen ist es enorm wichtig, schnell und effektiv zu handeln. Hierfür benötigt man Informationen über neuralgische Punkte, wie beispielsweise Objekte der baulichen Infrastruktur (Dämme, Durchlässe, Kanäle, u.v.m.). Müssen Personen evakuiert werden, ist es zudem wichtig zu wissen, bis zu welchem Stockwerk die Evakuierung erfolgen muss, aber beispielsweise auch, welche Straßen überhaupt noch durch Rettungsfahrzeuge befahrbar sind. Hierfür muss es u.a. möglich sein, räumliche Abfragen hinsichtlich der Höhe von Gebäuden oder auch der Höhenlage von betroffenen Straßen zu stellen.

- **Sichtbarkeitsanalysen / Verschattungen / Solarpotentiale:**

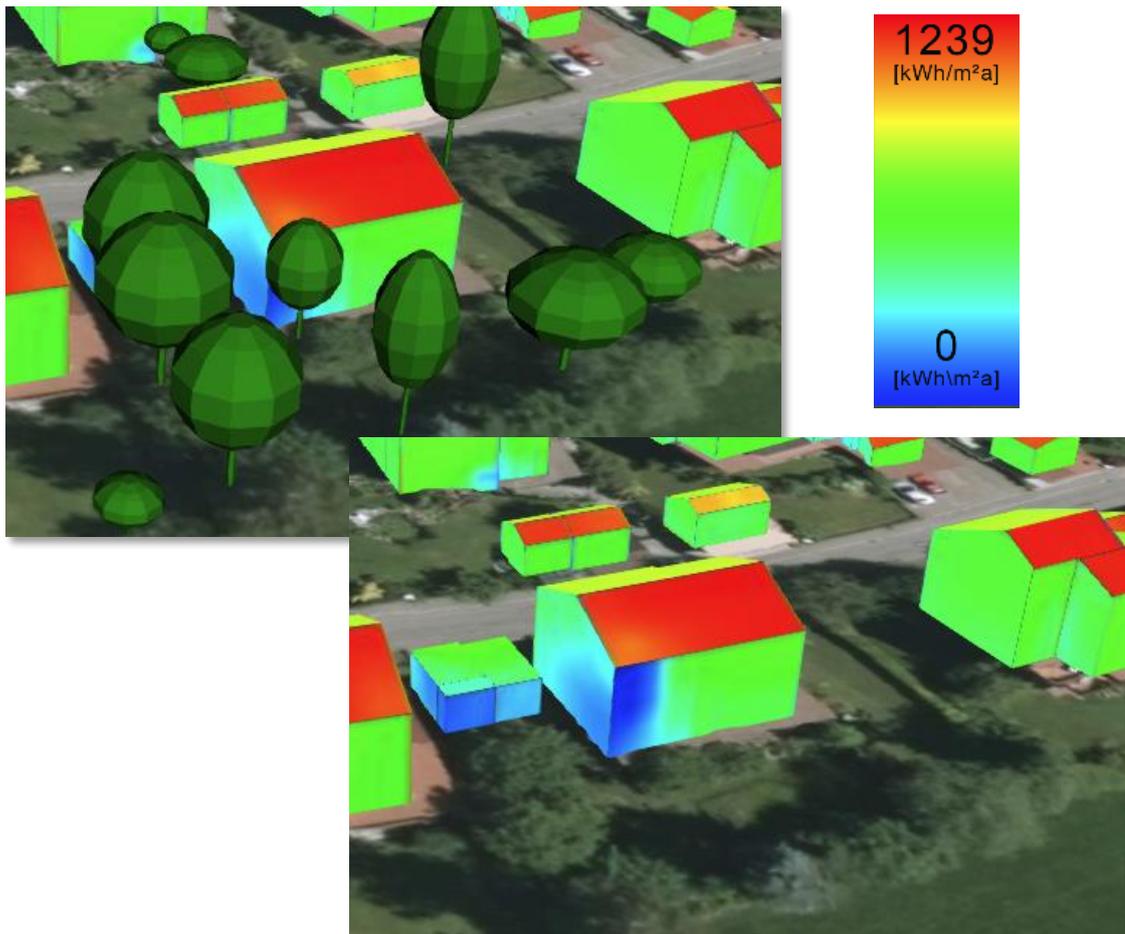
Die Analyse möglicher Sichtverbindungen spielt in diversen Bereichen eine entscheidende Rolle. So kann z.B. ermittelt werden, welche Auswirkungen die in der Landschaft errichteten Objekte, wie z.B. Windkraftanlagen oder Funkmasten, auf das Landschaftsbild haben. Dabei kann nicht nur angegeben werden, welche Gebäude von Verschattungen betroffen sind, sondern auch, um welche spezifische Fläche eines Hauses es sich dabei handelt.

Diese Thematik ist beispielsweise auch für Solarpotentialanalysen relevant. So lässt sich z.B. die jährliche Globalstrahlung für jede Gebäudefläche (Wand- / Dachfläche) separat angeben; damit ist es u.a. möglich, die Dachfläche mit dem größten Potential für eine Photovoltaikanlage zu bestimmen (siehe Abbildung: Solarpotentialanalyse für den Ortskern von Scheidegg).



³ siehe "OGC City Geography Markup Language (CityGML) Encoding Standard"

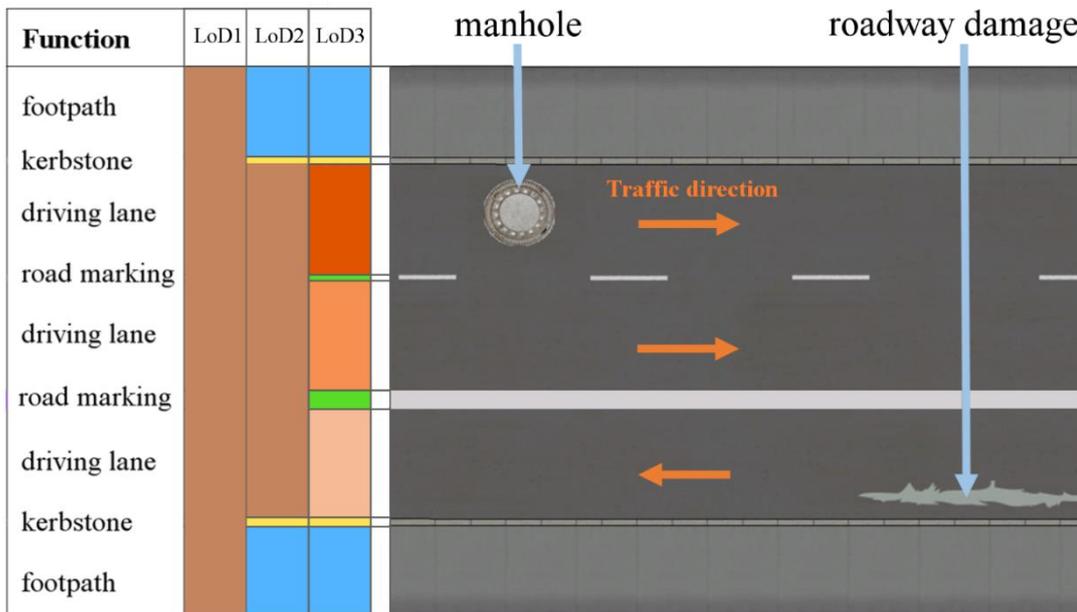
Die dargestellte Sonneneinstrahlungsanalyse wurde mittels eines an der TUM entwickelten Programms durchgeführt, welches für 3D-Stadtmodelle im CityGML-Format verwendet werden kann. Die ermittelten Ergebnisse werden dabei im Ausgangsdatensatz in Form von geeigneten Werteaggregationen angereichert und als Texturen für die LoD2-Gebäudemodelle visualisiert. Dabei wird neben dem Sky View Factor (prozentuales Sichtfeld von einem Beobachtungspunkt auf die Hemisphäre) auch die globale Strahlung (Direkt- und Diffusstrahlung) unter Berücksichtigung der umliegenden Topographie im dreidimensionalen Raum berechnet. Wie in den beiden unteren Abbildungen zu erkennen ist, fließen in die Berechnung auch umliegende Objekte wie Bäume oder angrenzende Gebäude (inkl. Balkone) ein, die u.U. die Hausfassaden bzw. die Dächer abschatten können.



Diese Berechnung lässt sich beispielsweise auch auf den Straßenraum anwenden. So lassen sich u.a. urbane Hitzeinseln identifizieren und visualisieren.

- **Verkehrsplanung:**

Auch hinsichtlich der Verkehrsplanung ist die semantische Anreicherung der einzelnen Objekte von Interesse. So sieht das im Rahmen der Masterarbeit von Christof Beil weiterentwickelte Konzept des Transportation Modells vor, ab LoD2 auch Gehwege flächenhaft zu repräsentieren und ab LoD3 zusätzlich die einzelnen Fahrspuren zu modellieren und dabei beispielsweise auch Straßenschäden oder die Positionen von Gullys zu berücksichtigen (siehe Abbildung). Diese Informationen sind z.B. für Trainingssimulatoren oder auch für Schwerlasttransporte wichtig.



[Beil, 2017]

Ein Ergebnis der detaillierten Repräsentation des Straßenraums ist in folgendem Ausschnitt abgebildet (Columbus Circle in NYC).



[Beil, 2017]

Für die Repräsentation von Semantischen Stadt- und Landschaftsmodellen hat sich der internationale Standard des OGC "**CityGML**" etabliert.

City Geography Markup Language (CityGML) ist ein XML-basiertes Open-Source-Format zur Repräsentation, Datenhaltung und zum Datenaustausch von virtuellen 3D-Stadt- und Landschaftsmodellen mit semantischen, topologischen und geometrischen Informationen, sowie Informationen zur Erscheinung der einzelnen Objekte. Mit der Möglichkeit, in CityGML topologische und semantische Eigenschaften von Objekten zu speichern, werden neben graphischen auch fachgebundene Analysen ermöglicht und somit zusätzliche Anwendungsbereiche abgedeckt. CityGML wird deshalb und auf Grund der Plattformunabhängigkeit als zentrale Informationsschnittstelle zwischen verschiedenen Anwendungsdisziplinen, wie z.B. Tourismus, Katastrophenmanagement oder Navigation, verwendet. Jedes Stadtobjekt kann zudem in fünf verschiedenen Detaillierungsgraden (Levels of Detail, LoD0 – LoD4) dargestellt werden, wobei der graphische Informationsgehalt einzelner Objekte mit zunehmender LoD-Stufe steigt.

Aus den aufgeführten Gründen wird für das 3D-DLM-Projekt *CityGML* als Datenmodell verwendet. Weitere Gründe, die für das Nutzen des CityGML-Standards sprechen, können außerdem dem Abschlussbericht zur 1. Projektphase entnommen werden.

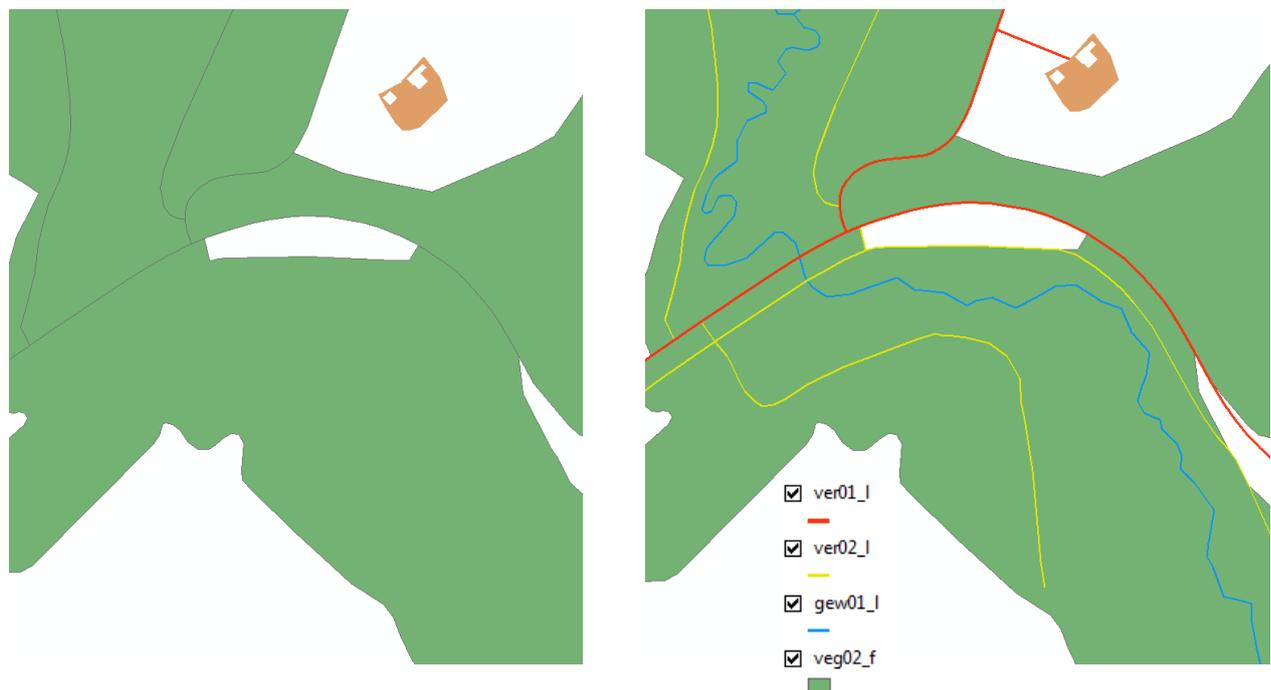
3 Vorverarbeitung der Daten

Für die Anwendung der Software 3Dfier ist es erforderlich, dass die zweidimensionalen Input-Daten lückenlos und überlappungsfrei vorliegen. Eine Ausnahme bilden dabei die Brücken, da sich diese über dem eigentlichen Gelände befinden und dort somit Überlappungen mit den darunter liegenden Objekten auftreten.

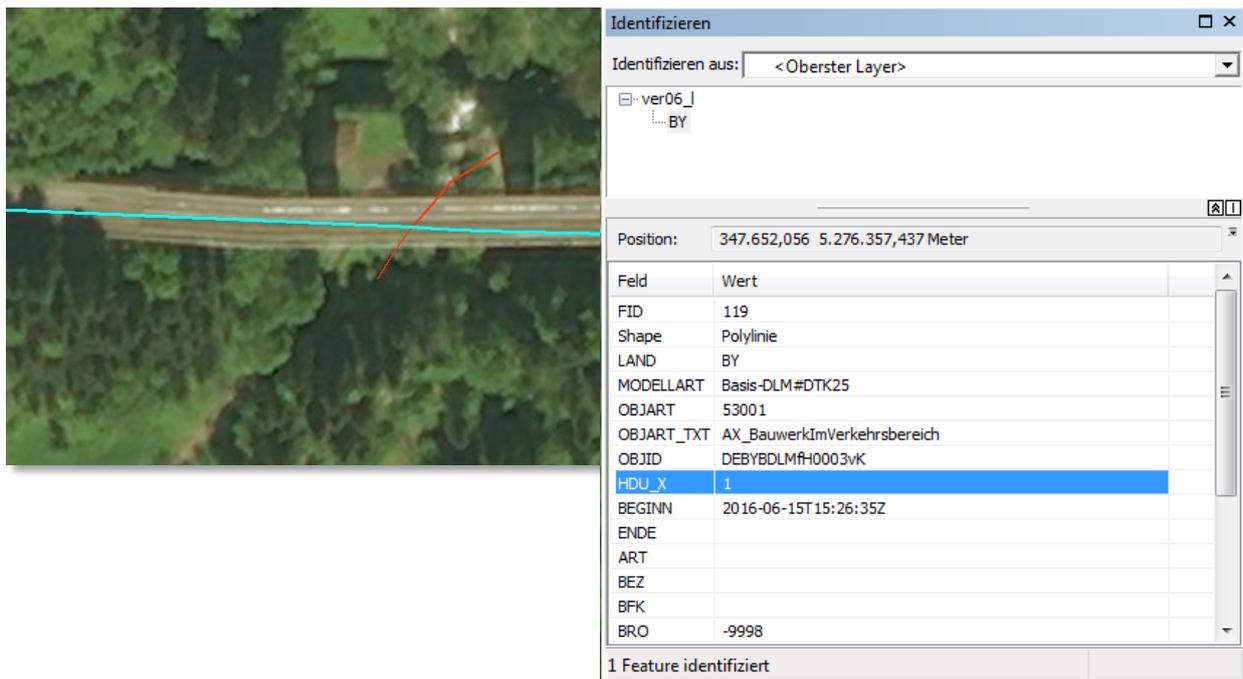
3.1 Herausforderungen

Ausgangsdaten für die Erzeugung des 3D Digitalen Landschaftsmodells sind die ATKIS-Daten der AdV. Viele dieser Daten liegen lediglich linienhaft vor. Dies erfordert eine entsprechende Pufferung um standardisierte bzw. klassifizierte Werte ausgehend von den ATKIS-Attributen.

Eine weitere Herausforderung besteht darin, dass sich die Geometrien der einzelnen Objektarten teilweise gegenseitig überlagern. Dies trifft beispielsweise für den linienhaften Verlauf der Straßen oder von Flüssen innerhalb eines Wald-Polygons zu. Dies verdeutlicht die Gegenüberstellung der beiden unteren Abbildungen. Die Straßen (rot), die Wege (gelb) und die Flüsse (blau) zerschneiden einige der Waldpolygone. Dies macht es einerseits erforderlich, dass die neu entstehenden Teilflächen des ursprünglichen Waldpolygons den Bezug zu diesem nicht verlieren; d.h. nach dem Zerlegen eines Wald-Polygons in mehrere kleine Objekte, müssen diese mittels der ursprünglichen ID zu der Geometrie MultiSurface aggregiert werden. Andererseits müssen für die Pufferung der linienhaft vorliegenden Fluss- und Straßenachsen Teile der Waldflächen entsprechend freigestellt werden.



Für die korrekte Anwendung des 3Dfiers werden für jedes Objekt die Attribute „uniqueID“ und „height_field“ (mit den Werten 0 oder 1) benötigt. Der 3Dfier berücksichtigt zunächst alle Objekte mit dem height_field = 0 (planare Zerlegung); im Anschluss werden die überlappenden Objekte wie z.B. Brücken hinzugefügt. An dieser Stelle sei darauf hingewiesen, dass das Attribut „height_field“ nicht gleichzusetzen ist mit dem Attribut „HDU_X“ (hatDirektUnten) der ATKIS-Daten. Bei den Brückenobjekten der ATKIS-Daten wird mittels HDU_X nicht die Lage-Beziehung zu den Straßen angegeben (für diesen Fall müsste der Wert des HDU_X-Attributs der Brücke stets 1 sein); sondern es gibt die Beziehung von sich kreuzenden Brücken zueinander an. So hat die Brücke, die über einer anderen Brücke verläuft, den Wert '1' (siehe folgende Abbildung).



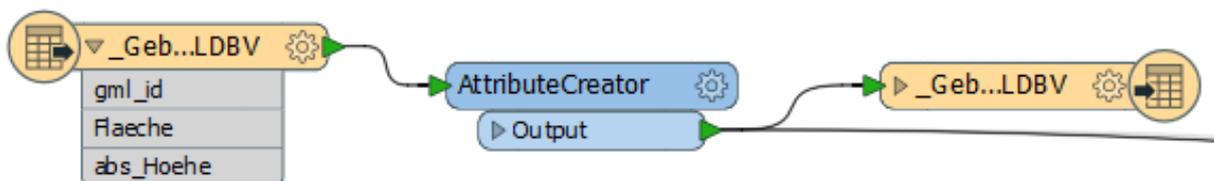
3.2 Allgemeines Vorgehen

Zunächst werden alle relevanten Objektarten einzeln betrachtet und entsprechend aufbereitet. Dies umfasst unter anderem die Filterung, Pufferungen um definierte Werte oder die Erzeugung zusätzlicher Attribute. Da als Resultat der Pufferung von Linienobjekten sich diese neu erzeugten Polygone mit den benachbarten Flächen unter Umständen überlagern, müssen diese in einem nächsten Schritt freigestellt werden. Dadurch liegen in der Ebene ($HDU_X = 0$) alle Objekte überlappungsfrei vor. Schließlich erfolgt die Erzeugung der Terrain-Flächen dadurch, dass die bisher noch vorhandenen Lücken als Terrain-Objekte deklariert werden (das Terrain / Gelände wird nach der Anwendung des 3Dfiers allerdings wieder durch das ursprüngliche DGM des LDBV bzw. des LGL ersetzt; dies wird im Detail im Kapitel 6.3 behandelt).

3.3 Betrachtung der einzelnen Objektarten

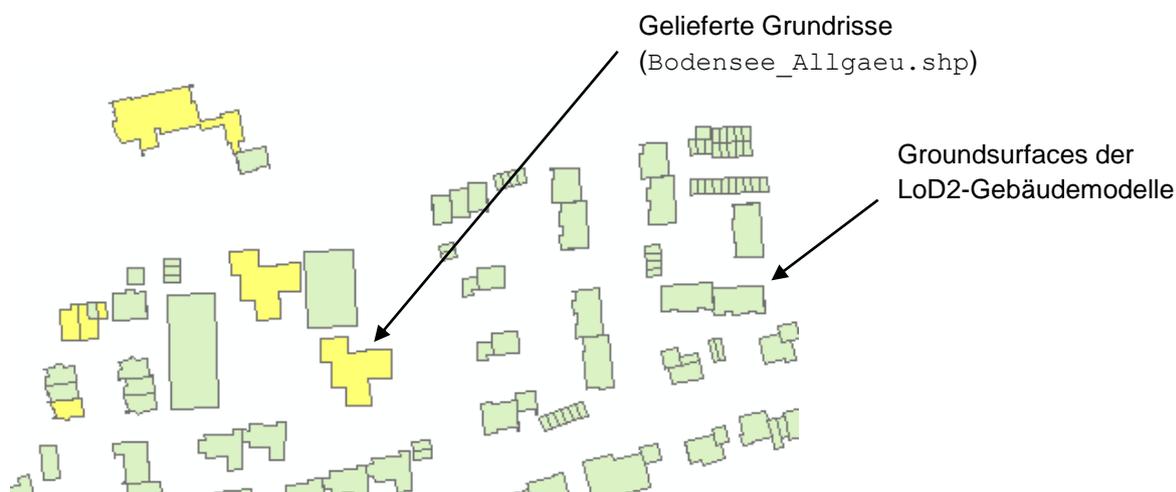
3.3.1 BUILDING

Die vom 3Dfier erzeugten Gebäudemodelle sollen im Nachgang durch die vorhandenen LoD2-Gebäudemodelle ersetzt werden. Für eine korrekte Triangulation der an die Gebäude angrenzenden Polygone müssen dennoch zunächst die entsprechenden Gebäude-Grundflächen in den Workflow eingehen.



Für die Anhebung der LoD2-Gebäude auf die korrekte Höhe wird die absolute Höhe mit übergeben (CoordinateExtractor). Zusätzlich wird das Attribut „height_field“ erzeugt und auf den Wert 0 gesetzt.

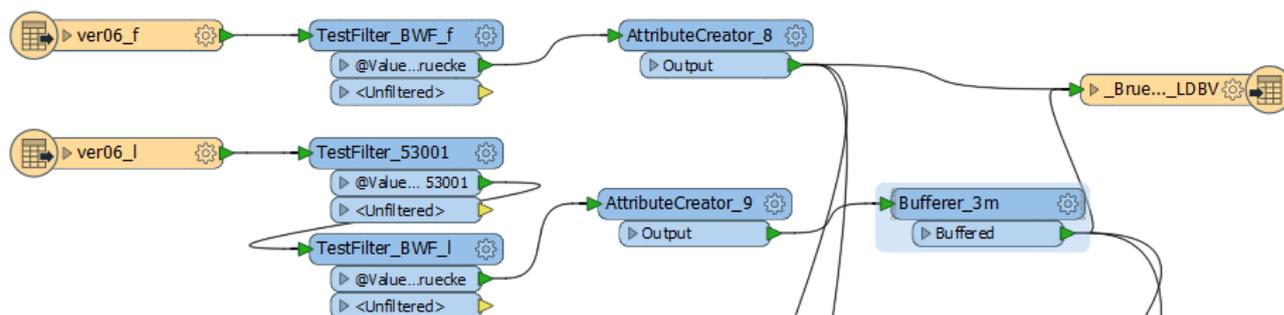
Zu beachten ist außerdem, dass nicht die gelieferten Grundrisse in den Workflow eingehen, sondern tatsächlich die Groundsurfaces der LoD2-Gebäudemodelle. Wie man in der folgenden Abbildung erkennen kann, sind diese Datensätze (Grundrisse / LoD2-Gebäudemodelle) nicht kongruent zueinander und es könnte bei Verwendung der gelieferten Grundrisse und dem im Nachgang stattfindenden Austausch durch die LoD2-Gebäudemodelle zu Lücken im bzw. zu Überlagerungen mit dem Terrain kommen.



3.3.2 BRIDGE

Als Grundlage für die Aufbereitung der Brücken-Objekte dienen die Files „ver06_f“ sowie „ver06_l“. Diese umfassen Bauwerke, Anlagen und Einrichtungen für den Verkehr (detaillierte Informationen bzgl. der Datengrundlage können dem Abschlussbericht zur 1. Phase entnommen werden).

Sowohl die flächenhaft, als auch die linienhaft vorliegenden Bauwerke werden mittels der Bauwerksfunktion *BWF* gefiltert. Dadurch werden Objekte herausgefiltert, die Tunnel (1870), Schutzgalerien (1880), Schleusenkammern (1890) oder Durchfahrten (1900) darstellen. Außerdem wird ein zusätzliches Attribut für das *height_field* ergänzt und auf den konstanten Wert 1 gesetzt.



Die Geometrie der flächenhaften Brückenobjekte kann dabei direkt übernommen werden. Die linienhaften Brückenobjekte werden zusätzlich gepuffert. Da keine Angaben über die reale Breite der Brücken vorhanden sind und außerdem diese der Breite der anschließenden Straßen entsprechen sollen, werden diese Breiten für die Pufferung übernommen (diese liegen oft lediglich klassifiziert vor).

Nach Rücksprache mit den Auftraggebern wurde sich außerdem darauf verständigt, dass die Straßen an den Brückenobjekten nicht unterbrochen werden sollen, sondern diese auch zusätzlich zum Brückenobjekt dargestellt werden sollen. Dadurch ist es beispielsweise leichter möglich, die Gesamtfläche einer Straße zu berechnen.

3.3.3 ROAD

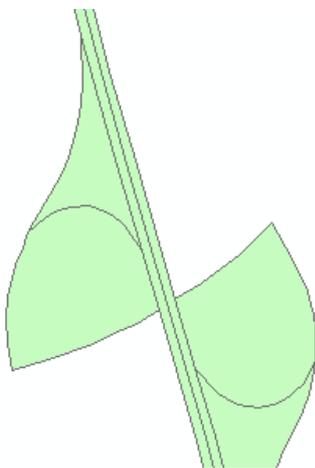
Diese Klasse setzt sich aus

- dem Straßenverkehr (`ver01_f` / `ver01_l`),
- den Wegen (`ver02_l`) und
- den Gleisen (`ver03_f` / `ver03_l`)

zusammen.

Straßenverkehr

Da die flächenhaft vorliegenden Objekte des Straßenverkehrs nicht nur die eigentlichen Fahrbahnen darstellen, sondern zusätzlich beispielsweise auch Verkehrsbegleitflächen (siehe Abbildung), werden diese entsprechend gefiltert und nicht weiter als Straßenobjekte behandelt.



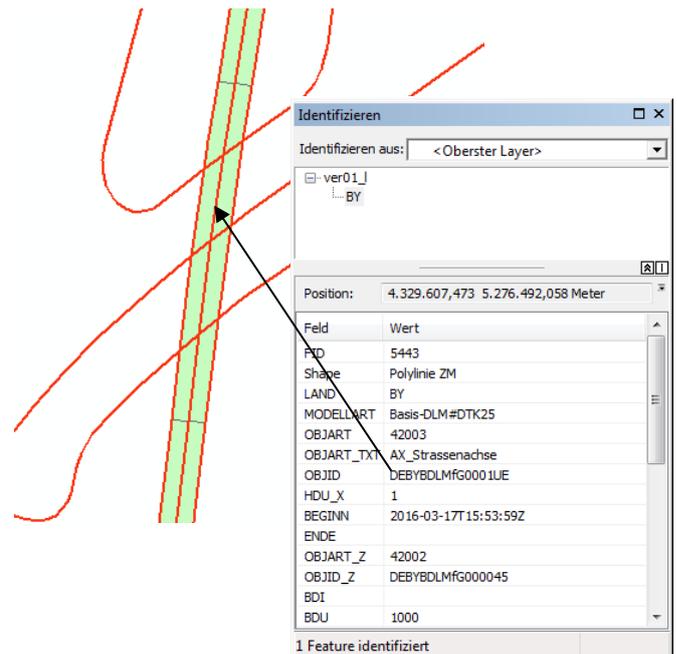
Feld	Wert
FID	44
Shape	Polygon ZM
LAND	BY
MODELLART	Basis-DLM#DTK25
OBJART	42001
OBJART_TXT	AX_Strassenverkehr
OBJID	DEBYBDMgG0002n7
HDU_X	0
BEGINN	2014-02-13T10:44:04Z
ENDE	
FKT	2312
NAM	
STS	

1 Feature identifiziert

Außerdem ist zu beachten, dass sich die flächenhaften Straßen (`ver01_f`) und die Straßenachsen (`ver01_l`) teilweise überlagern. In diesem Fall werden die bereits vorhandenen Straßenflächen übernommen und lediglich die Achsen gepuffert, die keine Überlappung aufweisen.⁴

Die Pufferung erfolgt über das Attribut `BRV` (Breite des Verkehrsweges). Diese sind über Klassenwerte angegeben:

- Klasse 6: 0m – 6m Breite
- Klasse 9: 6m – 9m Breite
- Klasse 12: 9m – 12m Breite
- usw. in Schritten von 3m.



⁴ Die Masterarbeit „Detaillierte Repräsentation des Straßenraums in 3D-Stadtmodellen“ von Christof Beil führt diese Thematik weiter aus (<http://mediatum.ub.tum.de/doc/1350734/1350734.pdf>)

Wege

Bei den Wegen (`ver02_1`) erfolgt die Pufferung ebenfalls über die Breite der Fahrbahn und der entsprechenden Klassenkategorie (siehe Straßenverkehr).

Gleise

Um die Gleise zu bearbeiten, erfolgt zunächst eine Filterung über das Attribut `BKT` (Bahnkategorie). Somit lassen sich zum Beispiel Seilbahnen herausfiltern.

Die Pufferung erfolgt für die Gleise mittels des Attributs `GLS` (Anzahl der Streckengleise). Hat dieses Attribut den Wert 1000 (d.h. ist es eingleisig), so wird um den halben Wert der Standard-Spurweite von 1,435m gepuffert. Für zweigleisige Strecken (`GLS = 2000`) wird ein Puffer-Wert von 2m verwendet.

3.3.4 WATER

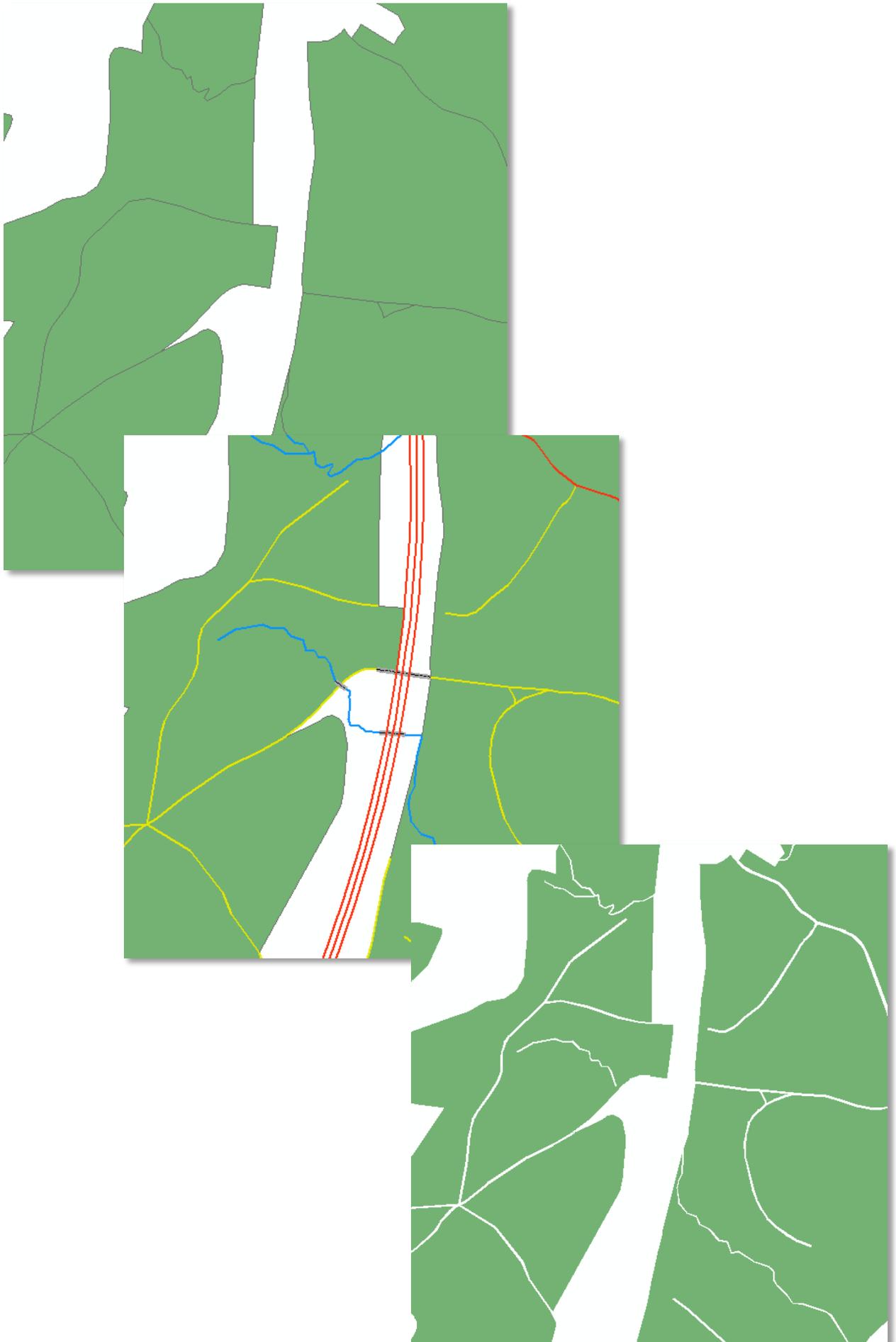
Auch im Bereich der Gewässer-Objekte liegen diese flächenhaft (`gew01_f`) beziehungsweise linienhaft (`gew01_1`) vor. Die flächenhaften Objekte können wiederum direkt übernommen werden. Die linienhaft vorliegenden Gewässerobjekte werden mittels der Breite des Gewässers (`BRG`), die wiederum durch Klassen repräsentiert werden, gepuffert.

- Klasse 3: 0m – 3m Breite des Gewässers → Puffer: 1m
- Klasse 6: 3m – 6m Breite des Gewässers → Puffer: 2,5m
- Klasse 12: 6m – 12m Breite des Gewässers → Puffer: 4,5m

3.3.5 FOREST

Um unerwünschte Überlagerungen zu vermeiden, müssen die folgenden Objektarten in den Wald-Polygonen (`veg02_f`) freigestellt werden:

- Brücken
- Gewässer
- Straßen
- Wege
- Gleise



3.3.6 Sonstige Objektarten

Neben den bereits beschriebenen Objektarten, die sich eindeutig einer Klasse des 3Dfiers zuordnen lassen, gibt es in den ATKIS-Daten noch weitere Objektarten, die bei der Erzeugung des 3D-Landschaftsmodells berücksichtigt werden sollten. Diese werden zunächst einer geeigneten 3Dfier-Klasse zugeordnet und im anschließenden Mapping der passenden CityGML-Klasse zugewiesen. Dies sind die folgenden Objektarten:

Landwirtschaft

veg01_f

Bauwerke oder Anlagen für Sport, Freizeit und Erholung

sie03_f

Bauwerke oder Anlagen für Industrie und Gewerbe

sie03_f

Flugverkehr

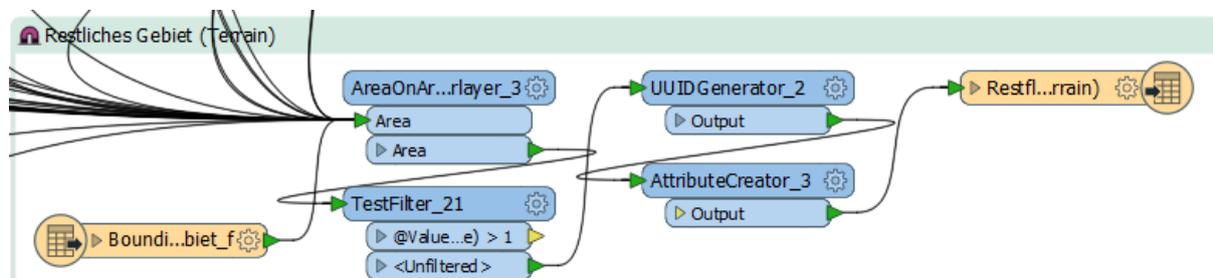
ver04_f

Übrige Objektarten der Vegetation außer Wald und LWS

veg03_f

3.3.7 TERRAIN

Die Klasse „Terrain“ wird nicht auf Grundlage vorhandener ATKIS-Daten erzeugt. Vielmehr werden die nach der Daten-Vorverarbeitung noch vorhandenen ‘Lücken’ der Klasse „Terrain“ zugewiesen. Da unter den Brückenobjekten Gelände dargestellt werden soll, werden die Flächen, die durch die Brücken repräsentiert werden, ebenfalls dem Terrain zugewiesen.



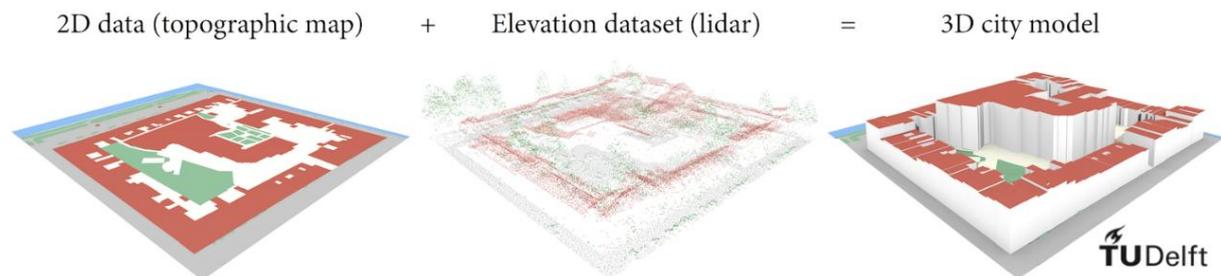
Außerdem müssen für die Anwendung des 3Dfiers die zwei zusätzlichen Attribute `uniqueID` und `height_field` erzeugt werden. Letzteres erhält den konstanten Wert 0.

4 Anwenden des 3Dfiers

4.1 Allgemeine Informationen

Der 3Dfier ist ein Open Source Tool zur Erzeugung von 3D-Objekten aus 2D-DLM und LIDAR Punktwolken, das an der TU Delft in den Niederlanden entwickelt wurde. Detailinformationen zu diesem Werkzeug finden sich unter <https://github.com/tudelft3d/3dfier>.

Das Grundprinzip des 3Dfiers illustriert die folgende Abbildung:



Basierend auf seiner Semantik wird jedes Polygon des zu Grunde liegenden 2D-Datenbestandes mittels angepasster Methoden in die dritte Dimension angehoben. Dabei kommen unterschiedliche Regeln je nach anzuhebender Objektklasse zur Anwendung. Dieses Regelwerk berücksichtigt auch die Semantik der Nachbarschaftsbeziehungen von Polygonen unterschiedlicher Objektklassen.

Ergebnis und Ziel der Anwendung ist ein trianguliertes Digitales Landschaftsmodell in 3D.

Alle hier beschriebenen Tests und Anwendungen wurden mit dem **3dfier Version 0.9.7** durchgeführt.

4.1.1 Eingangsdaten

Der 3Dfier erfordert zwei Arten von Eingangsdaten. Zunächst sind dies lückenlose, flächenhafte 2D-Daten, die die Landschaft beschreiben und auf eine der folgenden Klassen abgebildet werden müssen:

- **Building**
- **Terrain**
- **Road**
- **Water**
- **Forest**
- **Bridge**
- **Separation**

Des Weiteren ist eine entsprechende Punktwolke erforderlich, die mindestens die folgenden Klassen aufweist:

- **0-1:** Created, never classified and / or unclassified
- **2:** Ground
- **3-5:** Vegetation

4.1.2 Ergebnisdaten

Neben CityGML stehen noch die Ausgabeformate OBJ, CSV (nur Gebäude), Shapefile und PostGIS zur Verfügung. Die Tests wurden mit den Ausgabeformaten CityGML (für die semantische Weiterverarbeitung im Projekt) und OBJ (für Testvisualisierungen und visuelle Qualitätsbewertungen) durchgeführt.

4.2 Umsetzung

Die Umsetzung und damit die konkrete Anwendung des 3Dfiers im Rahmen dieses Projektes erfolgte mit den im Vorfeld aufbereiteten ALKIS- und ATKIS-Daten (siehe Kapitel 3) sowie den zur Verfügung gestellten 3D-Punktwolkendaten der Auftraggeber.

Technisch erfolgt der Aufruf des 3Dfiers auf Kommandozeilenbasis mit Übergabe einer Konfigurationsdatei sowie der Ausgabedatei.

Beispielaufruf über CMD-Shell unter dem Betriebssystem Windows:

```
<3dfier.exe> <Konfigurationsdatei> -o <Ausgabedatei>
c:\Tools\3dfier-windows-x64-v0.9.7\config.yml -o output/testarea.obj
```

Ein Tutorial zur Anwendung des 3Dfiers kann unter folgendem Link heruntergeladen werden:

<https://github.com/tudelft3d/3dfier/wiki/General-3dfier-tutorial-to-generate-LOD1-models>

4.2.1 Konfigurationsdatei

Die gesamte Steuerung des 3Dfiers erfolgt über eine Konfigurationsdatei. Diese Konfigurationsdatei hat die Endung `ym1`. Es handelt sich dabei um eine reine ASCII-Datei, die auch einfach über ein Steuerprogramm erzeugt werden kann. Damit eignet sich der 3Dfier sehr gut für die Einbindung in automatisierte rein batchorientierte und auftragsgesteuerte Prozesse, wie sie bei der Verarbeitung von Massendaten in der Praxis zur Anwendung kommen.

Die eigentliche Konfiguration gliedert sich in fünf Teilbereiche, die im Folgenden kurz beschrieben werden:

4.2.1.1 input_polygons

Angabe der Shape-Dateien (2D-Eingangsdaten) und Zuordnung zu den entsprechenden Klassen (siehe Abschnitt 4.1.1) (*lifting*). Für jede der möglichen Klassen wird eine eigene Methode zur Erstellung der 3D-Objekte verwendet.

Beispiel:

```
input_polygons:
- datasets:
  - 2D-Daten_Clip/_Wald_LDBV.shp
  - 2D-Daten_Clip/_Wald_LGL.shp
  uniqueid: OBJID
  lifting: Forest
```

4.2.1.2 lifting_options

Parametrierung der jeweiligen Umsetzungsalgorithmen (*lifting*). Im Wesentlichen handelt es sich dabei um die folgenden Angaben:

- Angabe der zur verwendenden Punkte des DOM als Perzentile zur Ermittlung der Höhe des Objekts. Dies betrifft die Klassen `Water`, `Road` und `Bridge`. Beispiel:

```
Water:
  height: percentile-10
Road:
  height: percentile-50
```

- Angabe zur Ausdünnung der triangulierten Flächen bei den Klassen `Forest` und `Terrain`.
Beispiel:

```
Terrain:
  simplification: 5
Forest:
  simplification: 10
```

Je höher der angegebene Wert ist, desto stärker wird das Ergebnis der Triangulation geglättet und vereinfacht.

4.2.1.3 input_elevation

In diesem Abschnitt werden die zu verwendenden LAS-Dateien angegeben.

Beispiel:

```
input_elevation:
- datasets:
  - g:/RTGIS/3DDLML/LDBV/DOM/4329_5273_all.laz
  - g:/RTGIS/3DDLML/LDBV/DOM/4329_5274_all.laz
  - g:/RTGIS/3DDLML/LDBV/DOM/4330_5273_all.laz
  - g:/RTGIS/3DDLML/LDBV/DOM/4330_5274_all.laz
```

WICHTIG: Es muss darauf geachtet werden, dass auch wirklich *ALLE* LAS-Dateien vorhanden sind. Fehlt eine Datei, so wird der 3Dfier mit einem Fehler beendet. Dies erfolgt aber leider nicht im Sinne einer initialen Prüfung, sondern erst, wenn die entsprechende .laz Datei in der Verarbeitungskette benötigt wird. Dieses Problem wird in einer Produktionsumgebung sinnvollerweise von einem Steuerprogramm abgefangen, in das der 3Dfier eingebettet ist.

Zusätzlich können LAS-Klassen angegeben werden, die nicht ausgewertet werden sollen:

```
omit_LAS_classes:
- 1 # unclassified
- 6 # building
```

Außerdem können die LAS-Punkte bereits beim Einlesen ausgedünnt werden.

Beispiel:

```
thinning: 10
```

Dabei wird die Anzahl der Punkte angegeben, die überlesen werden sollen (im Beispiel wird also nur jeder zehnte Punkt übernommen).

4.2.1.4 options

Angabe von globalen Parametern zur Feinjustierung der Umsetzung. Hier wurden während der Testumsetzung die Default-Parameter verwendet.

4.2.1.5 output

Angabe des zu erzeugenden Ausgabedateityps.

WICHTIG: Der Ausgabedateityp sollte mit der übergebenen Ausgabedatei beim Aufruf übereinstimmen.

4.2.2 Prozessschritte

Die Umsetzung erfolgt in vier Einzelschritten, die im Folgenden näher erklärt werden:

4.2.2.1 Einlesen der Konfigurationsdatei

Als erstes wird die Konfigurationsdatei eingelesen und auf korrekte Syntax geprüft:

```
Reading config file: c:\RTGis\2D-Daten_Clip.yml Config file is valid.
```

4.2.2.2 Analyse der Eingangspolygone

Anschließend werden alle Polygone analysiert und ggf. aufbereitet:

```
Gebiete:
(MultiPolygon split into 2 Polygons)
```

Fehlerhafte Polygone werden dabei so weit wie möglich automatisch korrigiert:

```
Warning 1: Ring Self-intersection at or near point 4330099.090002181
5273423.7502102088 0
```

oder

```
Geometry invalid: 2c471891-alfa-4f6e-84d8-2c7523622a52
```

Am Ende dieses Prozesses werden die Gesamtanzahl der verwendeten Polygone und die Gebietskoordinaten (Extent) ausgegeben.

```
Total # of polygons: 1,281 Constructing the R-tree... done. Spatial extent:
(4,329,317.262, 5,273,000.014) (4,331,000.000, 5,275,000.000)
```

4.2.2.3 Einlesen der Punktwolke

Es folgt nun der zeitintensivste Prozess, das Einlesen der 3D-Punktwolken (LAZ-Dateien):

```
Reading LAS/LAZ file: g:/RTGIS/3DDLML/LDBV/DOM/4330_5274_all.laz (1,559,828
points in the file) (all points used, no skipping)
```

4.2.2.4 Erstellung des 3D-Modells

Als vierter und letzter Teilschritt erfolgt das eigentliche Erzeugen des 3D-Modells:

```
Lifting all input polygons to 3D...
===== /LIFTING =====
===== LIFTING/ =====
===== /ADJACENT FEATURES =====
===== ADJACENT FEATURES/ =====
===== /STITCHING =====
===== STITCHING/ =====
===== /BOWTIES =====
===== BOWTIES/ =====
===== /VERTICAL WALLS =====
===== VERTICAL WALLS/ =====
===== /CDT =====
```

```
===== CDT/ =====  
done with calculations.  
OBJ output
```

4.2.3 Dauer der Prozessierung

Die folgenden Angaben zu Laufzeiten basieren auf der Durchführung auf einem Testrechner mit den folgenden Merkmalen:

```
Operating System:      Microsoft Windows 7 Professional (x64)  
CPU:                   Intel(R) Core(TM) i7-4712MQ CPU @ 2.30GHz  
Number of CPU Cores:   4  
Number of Logical CPUs: 8  
Total Memory Size [MB]: 8192
```

Die Dauer der Umsetzung wird insbesondere von der Größe / Auflösung der Punktwolke bestimmt. Im Folgenden seien die Zeiten für die Umsetzung des Datenbereichs des LDBV angegeben (189 km², 214 LiDAR-Kacheln á 1km² - siehe Abschlussbericht Phase 1):

```
All points read in 5048 seconds || 01:24:08  
Features written in 225254 ms  
Successfully terminated in 5548 seconds || 01:32:28
```

➔ 90,99% der Gesamtzeit wurden für das Einlesen der Punktwolke verwendet.

Dabei verschlingt das Einlesen der Punkte selber kaum Ressourcen, weder an RAM, noch an CPU-Leistung (CPU-Auslastung ~13%).

4.2.3.1 Gesamtumsetzung

Komplettes Testgebiet (254km², 296 LIDAR-Kacheln á 1km² - siehe Abschlussbericht Phase 1).

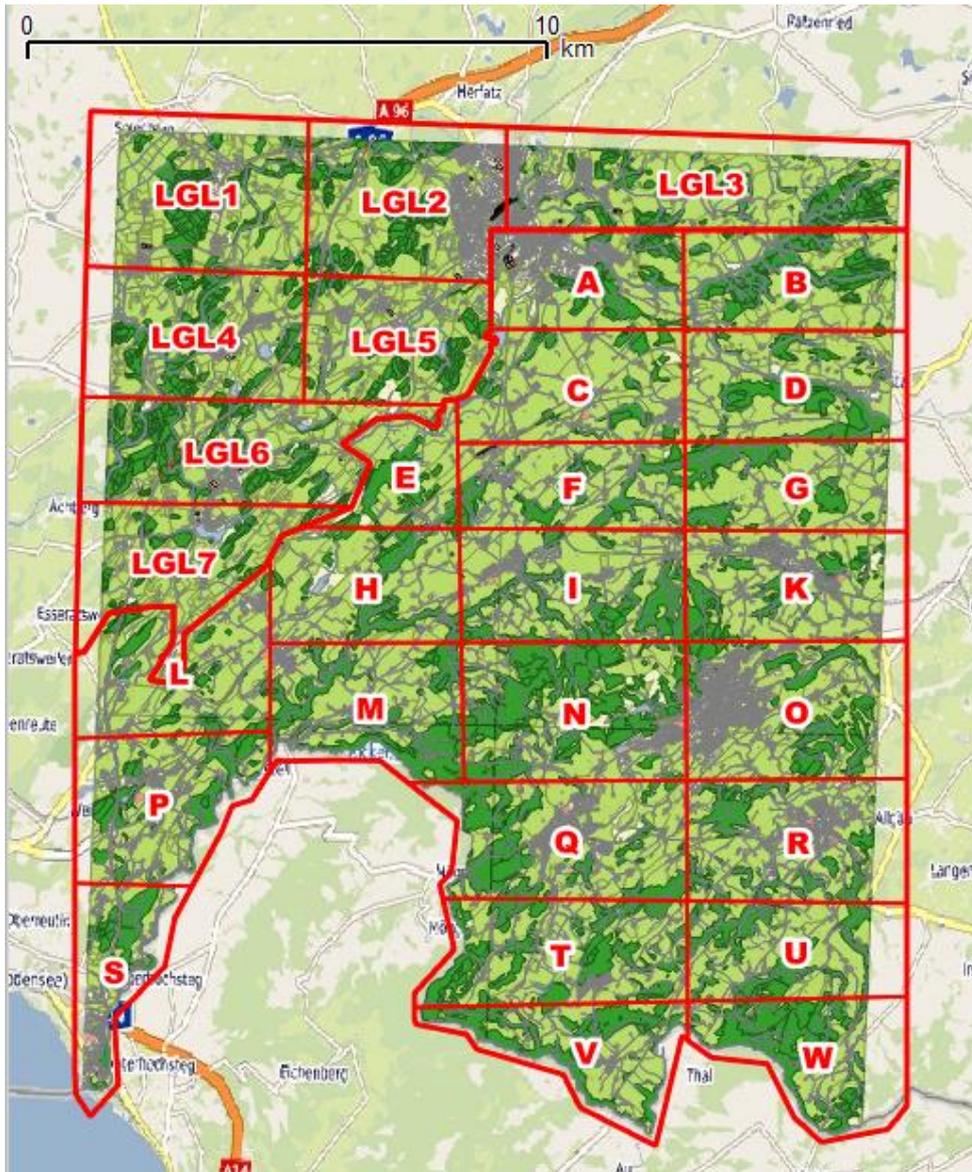


Umsetzung in einem Stück, **Dauer: 1 Stunde und 50 Minuten**

4.2.3.2 Parallelierte Umsetzung

Vor dem Hintergrund einer landesweiten Umsetzung wurde im Rahmen des Projektes untersucht, inwieweit sich die Anwendung des 3Dfiers parallelisieren lässt und welche Einsparungseffekte bzgl. Laufzeit durch eine solche Parallelisierung erreichbar wären.

Um eine parallelierte Produktion zu simulieren, wurde das Testgebiet in 29 (A-W, LGL1-7) annähernd gleichmäßige Gebiete unterteilt (siehe folgende Abbildung (A-W, LGL1-7)). Anschließend erfolgte die Zuweisung der Gebiete zu fünf Parallelprozessen (5*13% → CPU-Auslastung ~ 65%).



Zuweisung:

Prozess 1: A, B, C, D, LGL2
 Prozess 2: E, F, G, H, I, LGL5, LGL6
 Prozess 3: K, L, M, N, LGL1
 Prozess 4: O, P, Q, R, LGL3, LGL7
 Prozess 5: S, T, U, V, W, LGL4

Gesamtdauer der parallelierten Prozessierung: 27 Minuten

Mit der Aufteilung in Gebiete, die hinsichtlich der verfügbaren IT-Infrastruktur (Prozessoren, Hauptspeicher, Plattendurchsatz) optimiert portioniert werden und der daraus resultierenden Parallelisierung der Prozesse, kann ein deutlicher Geschwindigkeitszuwachs erreicht werden.

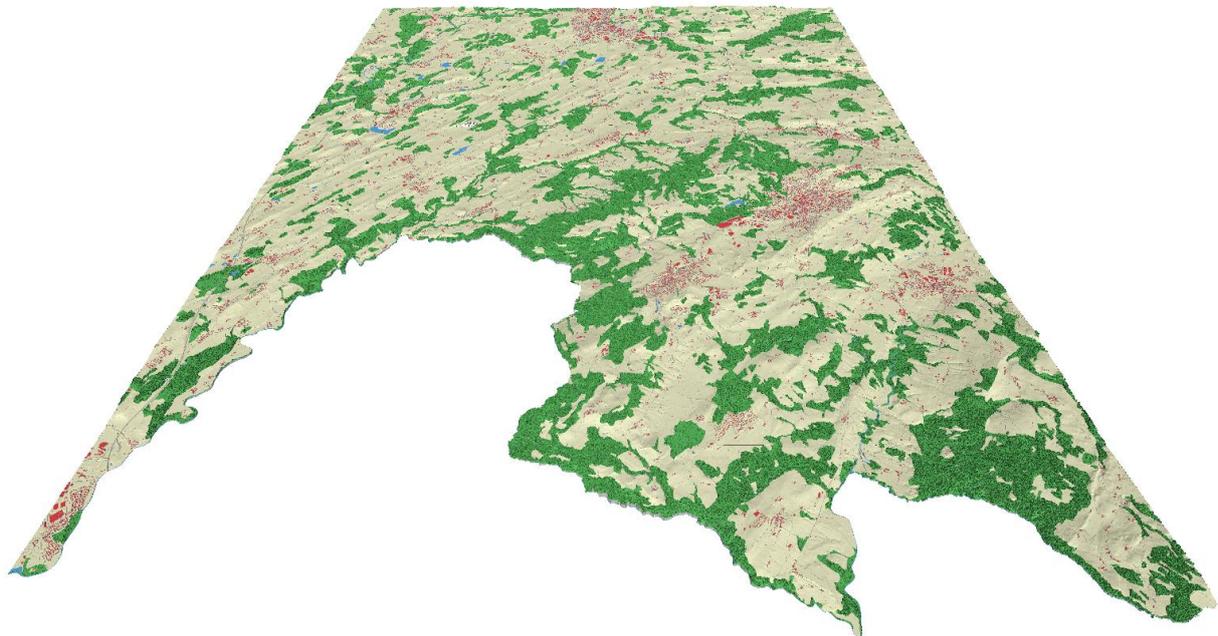
Vergleich Gesamtumsetzung / Parallelisierung:

Komplettumsetzung: 1h 50 Minuten
 Gebiete in 5 Parallelprozessen: 0h 27 Minuten

Die Architektur und Prozesslogik des 3Dfiers lässt eine Parallelprozessierung mit einfachen Mitteln zu. Allerdings sind für die notwendige 'Vernähung' der entstehenden Teilgebiete des 3D-DLMs Vorkehrungen im Pre- und Postprocessing zu treffen. Dies sind zum Beispiel:

- Verwendung eines überlappenden Puffers bei der Portionierung der Teilgebiete, dessen Ausdehnung sich dynamisch so ergibt, dass alle 2D-Objekte, die über den Gebietsrand hinausragen, vollständig im Puffer liegen.
- Nachträgliche Reduzierung der Teilergebnisse auf die Kerngebiete jeder Berechnungsportion
- Vernähung dieser Kerngebiete
- Berücksichtigung des Aspekts, dass jedes 2D-Objekt eindeutig einer Kachel zuzuordnen ist und damit seine 3D-Repräsentation nur einmal im vernähten 3D-DLM erscheint.

4.3 Ergebnisbewertung



4.3.1 Allgemeine Aussagen

Der 3Dfier lieferte mit den beschriebenen Ausgangsdaten im vollautomatischen Prozess ein fachlich sinnvolles Ergebnis, das aber im Detail noch inhaltliche Fehler enthält, auf die im Folgenden näher eingegangen wird. Eine genauere Bewertung der Ergebnisqualität findet sich in Kapitel 7.

Der Hardware-Ressourcenbedarf sowie die Laufzeiten der Prozessierung des Testgebiets erscheinen für eine Anwendung auch landesweiter Datenbestände geeignet.

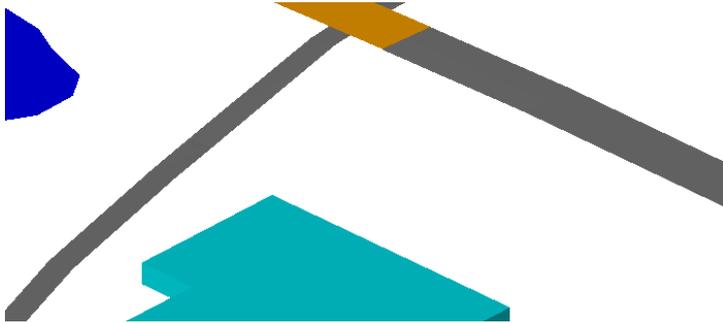
4.3.2 Analysen / Auffälligkeiten

4.3.2.1 Ausdünnung der Punktwolke

Über den Konfigurationsparameter `thinning` (siehe 4.2.1.3) kann die Punktwolke zur Laufzeit ausgedünnt werden. Dabei wird die Anzahl der Punkte angegeben, die überlesen werden sollen. Damit kann die Laufzeit bei besonders hoher Punktdichte erheblich verringert werden.

Eine zu hohe Reduktion kann dann aber in der Folge zu Geometriefehlern der erzeugten Klassen führen.

Punktdichte DOM-Kachel 4329_5274_all.laz: **1,46p/m²**

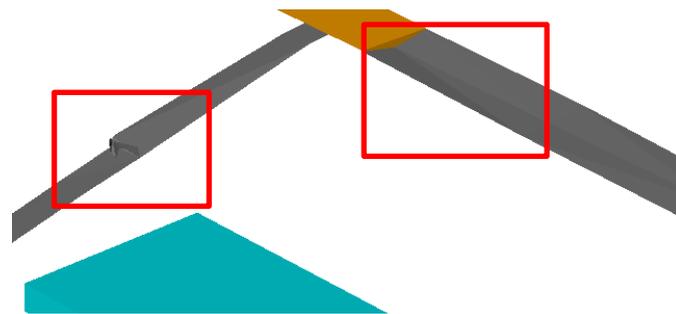


Kein Ausdünnen der Punktwolke: Straßenoberflächen sind glatt

thinning=0

```
Reading LAS/LAZ file: g:/RTGIS/3DDLM/LDBV/DOM/4329_5274_all.laz
(1,464,198 points in the file)
(all points used, no skipping)
```

```
[=====] 100%
All points read in 8 seconds || 00:00:08
```



Ausdünnen (jeder 2. Punkt, entspricht **~0,73p/m²**): Unregelmäßige Straßenfläche

thinning=2

```
Spatial extent: (4,329,000.000, 5,274,000.000) (4,330,000.000,
5,275,000.000)
Reading LAS/LAZ file: g:/RTGIS/3DDLM/LDBV/DOM/4329_5274_all.laz
(1,464,198 points in the file)
(skipping every 2th points, thus 732,099 are used)
```

```
[=====] 100%
All points read in 4 seconds || 00:00:04
```

Erkenntnis

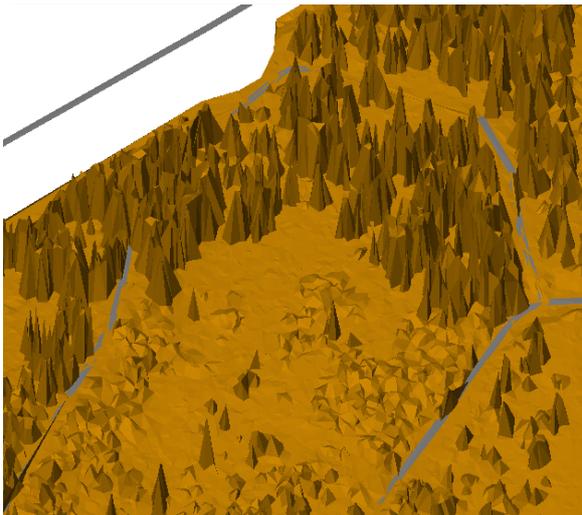
Eine zu geringe Punktdichte führt zu Geometriefehlern bzw. semantisch nicht korrekten Ergebnissen wie unruhigen Straßenoberflächen. Von der Anwendung des Parameters `thinning` ist daher im aktuellen Softwarestand abzuraten.

4.3.2.2 Triangulation reduzieren (simplification)

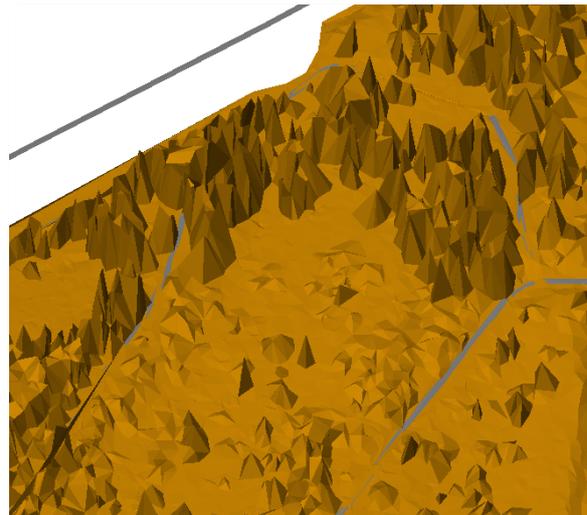
Die Oberflächen der Klassen *Forest* und *Terrain* werden direkt aus der Punktwolke abgeleitet und trianguliert. Die Oberflächen-Triangulation des Ergebnisobjekts kann über den Parameter `simplification` beeinflusst werden (siehe 4.2.1.2). Durch Anwendung dieses Parameters kann zu einem die Dateigröße reduziert werden (gerade bei der Erzeugung von CityGML), zum anderen wird aber natürlich auch das Ergebnis generalisiert.

Beispiel Klasse *Forest*:

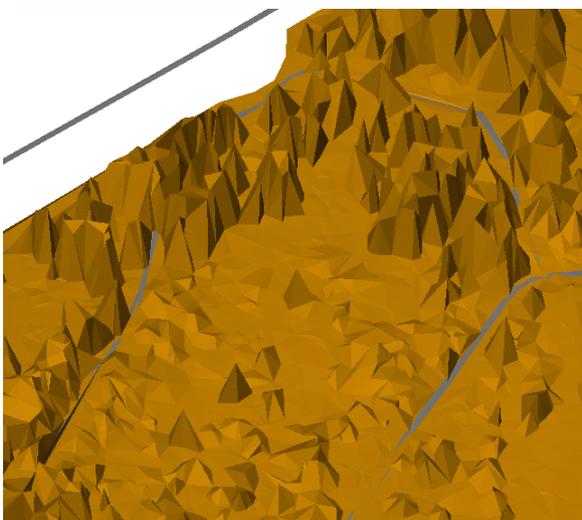
Umsetzung eines Teilbereichs



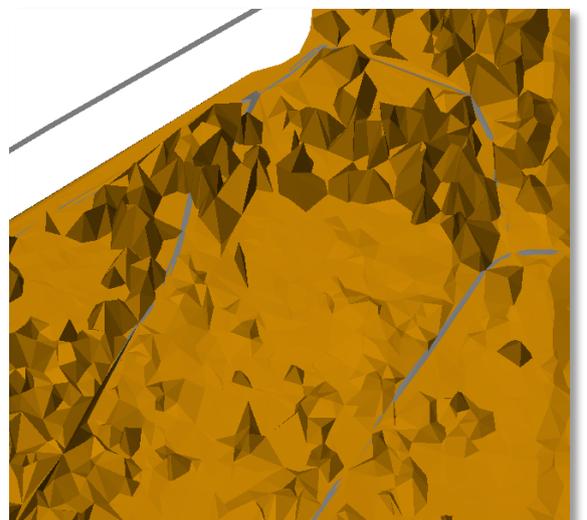
simplification: 10 (default)
Dateigröße: 30 MB



simplification: 30
Dateigröße: 12 MB



simplification: 50
Dateigröße: 8,5 MB

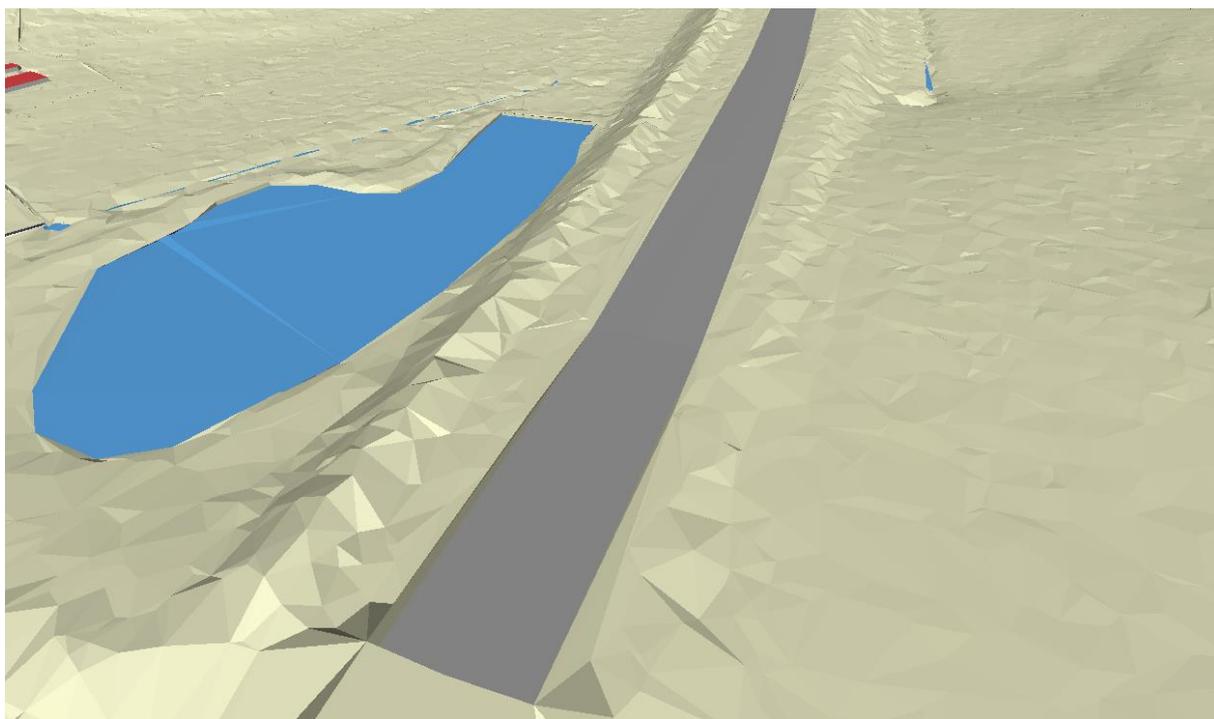


simplification: 90
Dateigröße: 6 MB

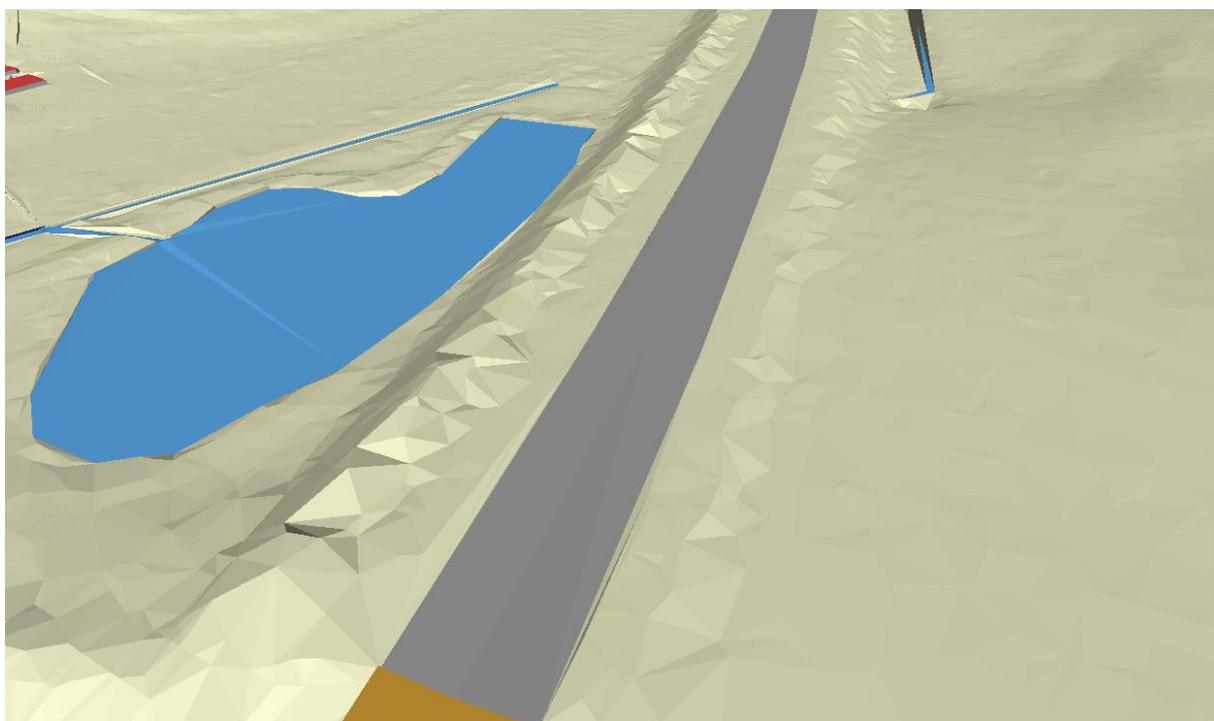
Bei der Klasse *Forest* kann der Detailgrad der Umsetzung je nach Anwendungsfall von niederrangiger Bedeutung sein, dient er doch im Wesentlichen ‚nur‘ zur groben Visualisierung (siehe auch 6.2).

Umso mehr hat dieser Wert aber bei der Visualisierung des DGMs eine Bedeutung, da es hier auf eine möglichst exakte Wiedergabe ankommt, welche dann aber zu einer erheblich größeren Ergebnisdatei führen kann. Dies ist insbesondere dann der Fall, wenn der Wert `simplification` auf 0 gesetzt wird (jeder Punkt wird trianguliert) (siehe auch 6.3).

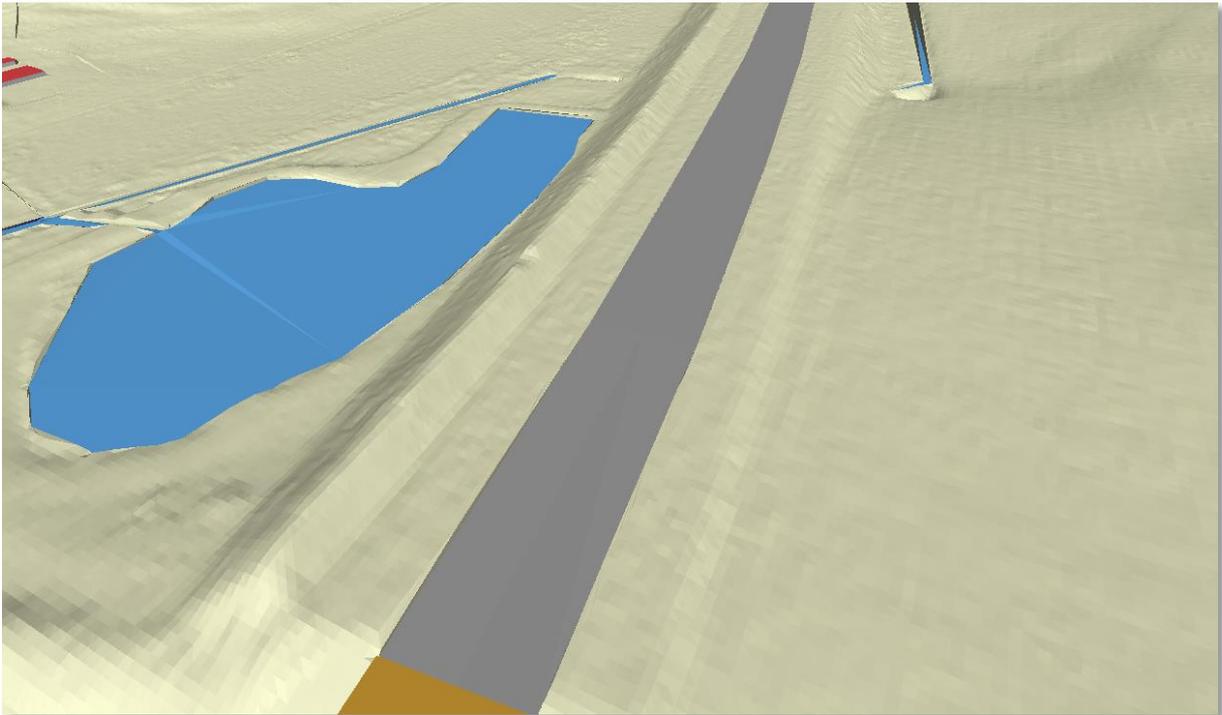
Beispiel *DGM* (Testkachel 2x2km, Ausgabeformat CityGML):



Gelände aus Punktwolke, simplification 10, Dateigröße 329MB



Gelände aus DGM1, simplification 10, Dateigröße 235MB



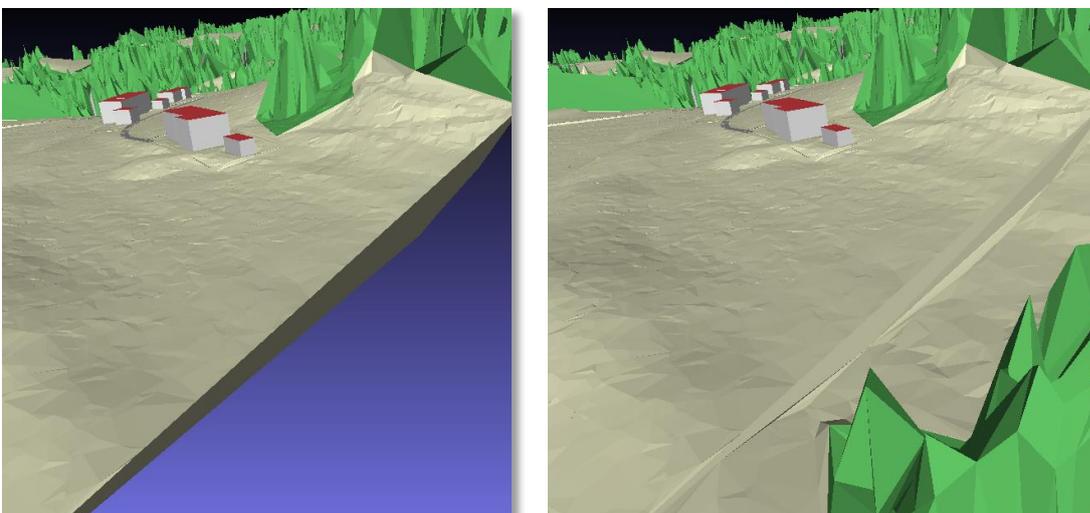
Gelände aus DGM1, simplification 0, Dateigröße 1,46GB

Im Vergleich zum Default-Wert (`simplification 10`) steigt die Dateigröße bei einer Umsetzung (DGM1) ohne Vereinfachung auf das knapp 7-fache an.

Da aber in der Praxis für die betrachtete Projektregion ein hochgenaues und hochaufgelöstes DGM bereits existiert, ist das durch den 3Dfier erstellte DGM nur als Zwischenergebnis zu betrachten und dieses sollte in der Folge in Form eines Postprozesses durch das erwähnte existierende DGM ersetzt werden.

4.3.2.3 Gebietsrand

Neben den im Abschnitt 4.2.3.2 genannten Aspekten ist bei der Vernäherung von Kacheln darauf zu achten, dass am Gebietsrand angrenzende Flächen in 2-facher Nachbarschaft zur Randberechnung mit einbezogen werden, da es sonst zu Kachelrandfehlern in Form virtueller Wände kommt:



4.3.3 Weitere Anmerkungen

4.3.3.1 Protokollierung (Logging)

Der 3Dfier bietet kein eigenes, konfigurierbares Logging an, so dass die Ausgabe auf der Konsole ‚mitgeschnitten‘ werden muss. Wichtig dabei: Auf `stdout` wird nur ein kleiner Teil der Information über die Laufzeit ausgegeben. Die wesentlichen Informationen werden über `stderr` ausgegeben. Daher sollten beim Logging immer beide Kanäle mitgeschrieben werden:

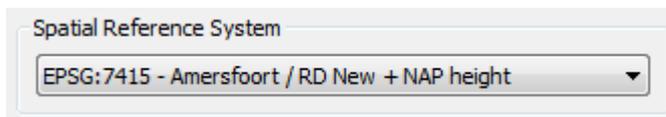
```
call %threedfier% %config% -o %outfile% > LDBV_clip_lasorg_GML.log 2>&1
```

4.3.3.2 Zeichenkodierung

Die erzeugten CityGML-Dateien werden mit UTF-8 kodiert. Daher muss darauf geachtet werden, dass auch die Eingabe-Shape-Dateien UTF-8 kodiert sind.

4.3.3.3 Bezugssystem

CityGML-Daten werden (aktuell) immer im Niederländischen Bezugssystem ausgegeben (EPSG:7415).



Die Koordinaten der Geometrien entsprechen aber dem Bezugssystem der Eingangsdaten (Shape-Dateien).

Daher muss das Bezugssystem in einem Postprozess auf das Bezugssystem der Eingangsdaten korrigiert werden, da das Ausgabebezugssystem aktuell nicht über die Konfigurationsdatei parametrierbar ist.

5 Abbildung auf CityGML-Modell

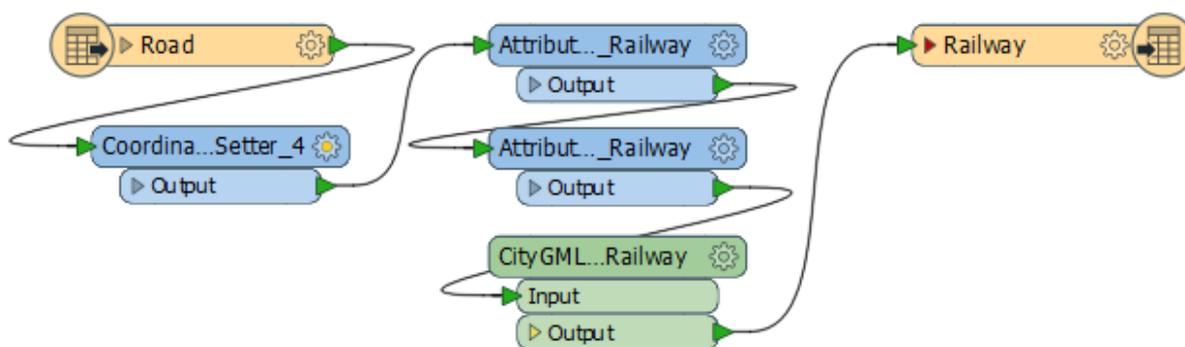
5.1 Semantisches Mapping

Da – wie bereits im Kapitel 3 erwähnt – vom 3Dfiers nur die Klassen `Building`, `Bridge`, `Road`, `Water`, `Forest`, `Terrain` und `Separation` erzeugt werden, ist im Nachgang eine Semantische Transformation bzw. eine Anpassung entsprechend der Mapping-Tabelle aus Projektphase 1 notwendig.

Konkret bedeutet dies:

Brücken:	<code>brg:Bridge</code>	→	<code>bridge:Bridge</code>
Gewässer:	<code>wtr:Waterbody</code>	→	<code>wtr:WaterBody</code>
Gleise:	<code>tran:Road</code>	→	<code>tran:Railway</code>
Straßen:	<code>tran:Road</code>	→	<code>tran:Road</code>
Wege:	<code>tran:Road</code>	→	<code>tran:Road</code>
Sportanlage:	<code>bldg:Building</code>	→	<code>luse:LandUse</code>
LWS:	<code>luse:LandUse</code>	→	<code>luse:LandUse</code>
Wald:	<code>veg:PlantCover</code>	→	<code>veg:PlantCover</code>

So werden beispielsweise die Gleise, die nach Anwendung des 3Dfiers auf die thematische Klasse `Road` abgebildet wurden, auf die Klasse `Railway` gemappt.



Die Sportanlagen, die zunächst als `Building` abgebildet wurden (da die Oberfläche der sportlichen Anlagen als ebene Flächen dargestellt werden sollen), werden im anschließenden Mapping auf die Klasse `LandUse` abgebildet.

Mapping der Attribute

Alle Attribute der ATKIS-Daten werden entsprechend des CityGML-Standards überführt. Das bedeutet beispielhaft:

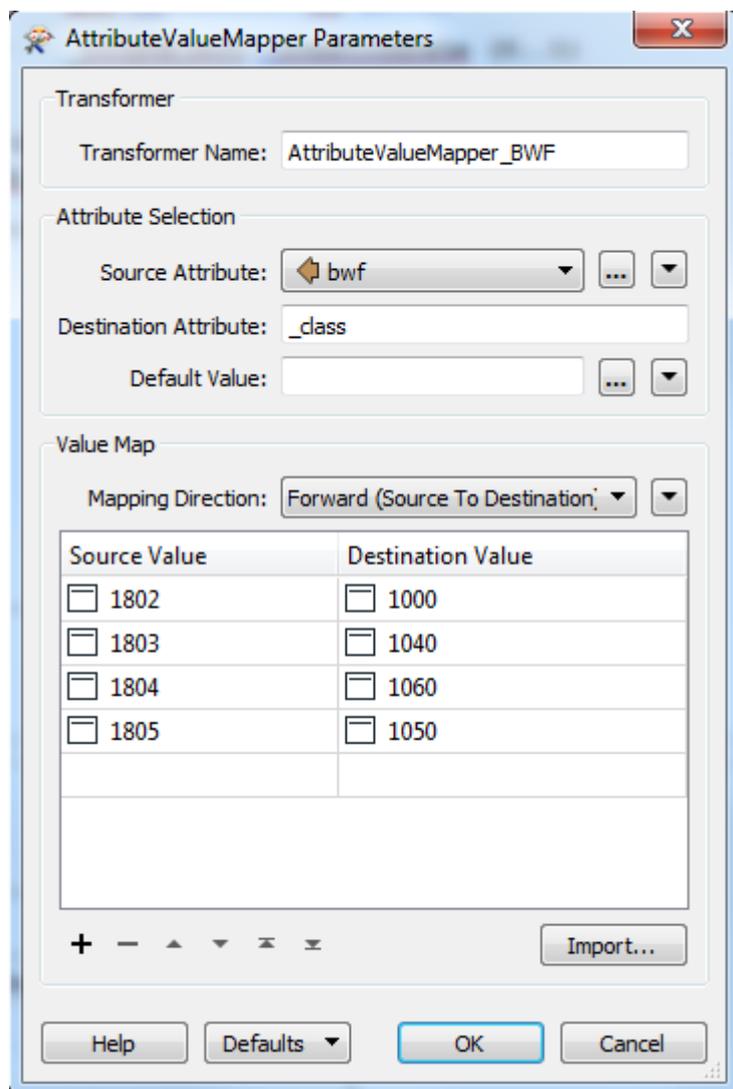
NAM:	Name	→	<code>gml::_Feature::name (0..*)</code>
OBJID:	eindeutiger Objektidentifikator	→	<code>gml::_Feature::id (0..1)</code>
BEGINN:	Lebenszeitintervall beginnt	→	<code>_CityObject::creationDate (0..1)</code>

Für den Fall, dass es für ein Attribut der ATKIS-Daten kein passendes Pendant im CityGML-Standard gibt, werden diese Informationen als generische CityGML-Attribute überführt.

Ein Beispiel für die Klasse der Brücke findet sich in folgender Abbildung:

New Attribute	Attribute Value
Bauwerksfunktion	↗ bwf
Landeskennung	↗ land
Modellartenkennung	↗ modellart
Objektart_Text	↗ objart_txt
Objektart	↗ objart
hatDirektUntenExistiert	↗ hdu_x
Durchfahrtshoehe	↗ dhu
Breite_Objekt	↗ bro

Auch das durchgeführte 'Value Mapping' zwischen der Bauwerksfunktion (BWF) der Brücke und der CityGML-Codelist sei anhand der `Bridge::class` dargestellt:



Konkret bedeutet dies:

1802 (Bogenbrücke)	→	1000 (arced bridge)
1803 (Fachwerkbrücke)	→	1040 (truss bridge)
1804 (Hängebrücke)	→	1060 (suspension bridge)
1805 (Pontonbrücke)	→	1050 (pontoon bridge)

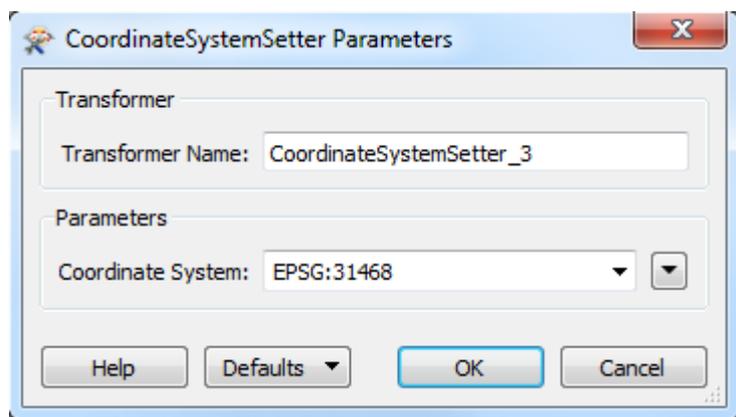
5.2 Geometrie

Alle Klassen wurden als `MultiSurfaces` im `LoD1` modelliert. Lediglich die im Nachgang eingefügten Gebäudemodelle des LDBV und des LGL liegen im `LoD2` vor.

5.3 Zusätzliche Transformationen

5.3.1 Setzen des korrekten Koordinatensystems

Wie in Kapitel 4.3.3.3 bereits erwähnt, werden die CityGML-Daten aktuell im `EPSG:7415` (Niederländisches Bezugssystem) ausgegeben. Dies erfordert es, dass im Nachgang das korrekte Koordinatensystem (`EPSG:31468`) gesetzt wird.



5.3.2 Einfügen zusätzlicher Attribute

Neben den in den ATKIS-Daten vorhandenen Attributen werden zusätzliche Attribute erzeugt. Dies sind zum einen die beiden generischen Attribute `min_Hoehe` und `max_Hoehe`. Das Speichern der minimalen und maximalen absoluten Höhe als Attribut ist deshalb sinnvoll, weil mittels dieser Attribute in einfachen Viewern räumliche Selektionen ausgeführt werden können; eine mögliche Abfrage könnte wie folgt lauten:

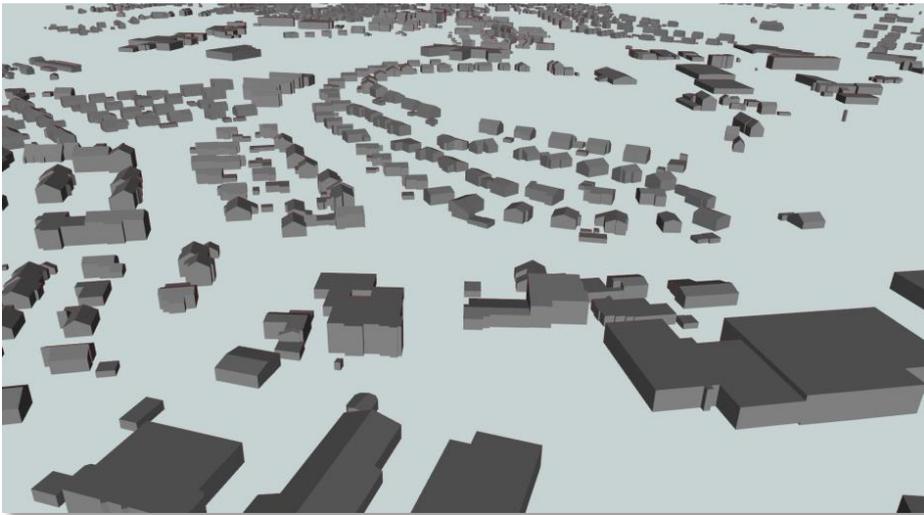
„Selektiere alle Straßen oberhalb von 500 m Höhe“.

Zum anderen wird für jedes Objekt die Fläche berechnet und im Attribut `Flaecheninhalt` abgespeichert. Dies kann dann sinnvoll sein, wenn man beispielsweise für Bauarbeiten an Straßen wissen muss, wie groß die Oberfläche eines Straßenabschnitts ist.

6 Austausch der mit dem 3Dfier erzeugten Objektarten

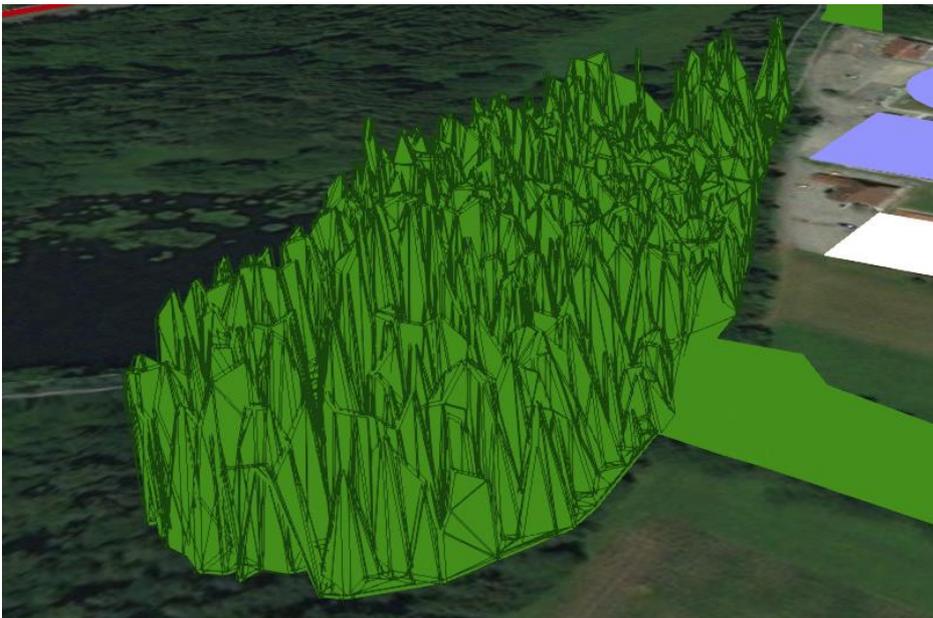
6.1 Gebäude

Für das dreidimensionale Landschaftsmodell sollen die original LoD2-Gebäudemodelle, die vom LDBV und vom LGL bereitgestellt wurden (siehe Projektphase 1), genutzt werden. Da für die Anwendung des 3Dfiers dennoch zunächst eine Tessellation vorliegen muss, müssen die mittels des 3Dfiers erstellten Klötzchenmodelle im Nachgang durch die LoD2-Gebäudemodelle ausgetauscht werden.



6.2 Wald

Nach Absprache mit den Auftraggebern sollte statt der durch den 3Dfier erzeugten Vegetationsobjekte (beispielhaftes Ergebnis siehe untere Abbildung) die Waldfläche besser lediglich über das Gelände gebogen werden. Dies führt zur Reduzierung der Daten und damit zu Performancesteigerungen.



6.3 DGM

Durch den 3Dfier wird ausgehend von den 2D-Polygonen `Terrain` (dies entspricht den vorhandenen Lücken in den aufbereiteten 2D-Daten) ein neues DGM erzeugt. Da sowohl für das LDBV, als auch für das LGL originale DGM-Daten vorliegen, sollte statt des erzeugten Terrains das originale DGM genutzt werden.

6.3.1 Ausgangsdaten

6.3.1.1 LDBV

Die LDBV-Daten liegen im `g01dgm`-Format in einer Kachelgröße von 2x2 km bzw. 0,5x0,5 km vor und haben eine Auflösung von 1m pro Rasterzelle. Das Bezugssystem ist das `DHDN_3_Degree_Gauss_Zone_4` (EPSG:31468) (zum Zeitpunkt der Datenlieferung – 2016 – lagen diese noch nicht in UTM vor).

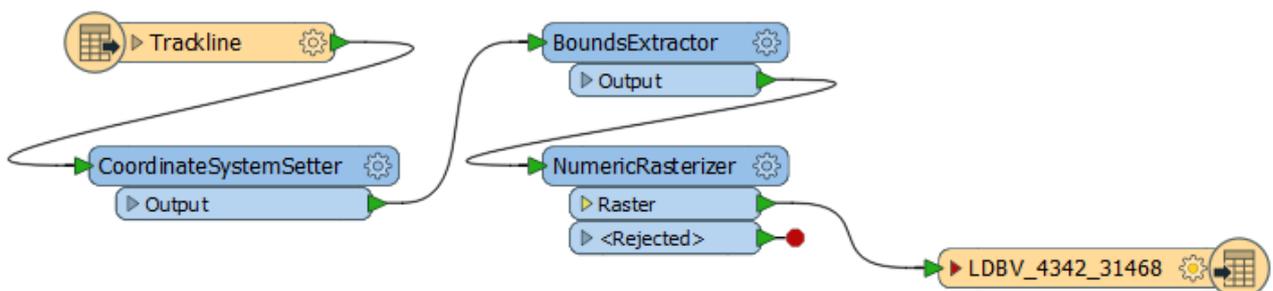
6.3.1.2 LGL

Die LGL-Daten liegen im 1m-Format in einer Kachelgröße von 1x1 km vor und haben eine Auflösung von 1m pro Rasterzelle. Das Bezugssystem ist das `DHDN_3_Degree_Gauss_Zone_3` (EPSG:31467) (vgl. Abschlussbericht zur 1. Phase).

6.3.2 Bearbeitung

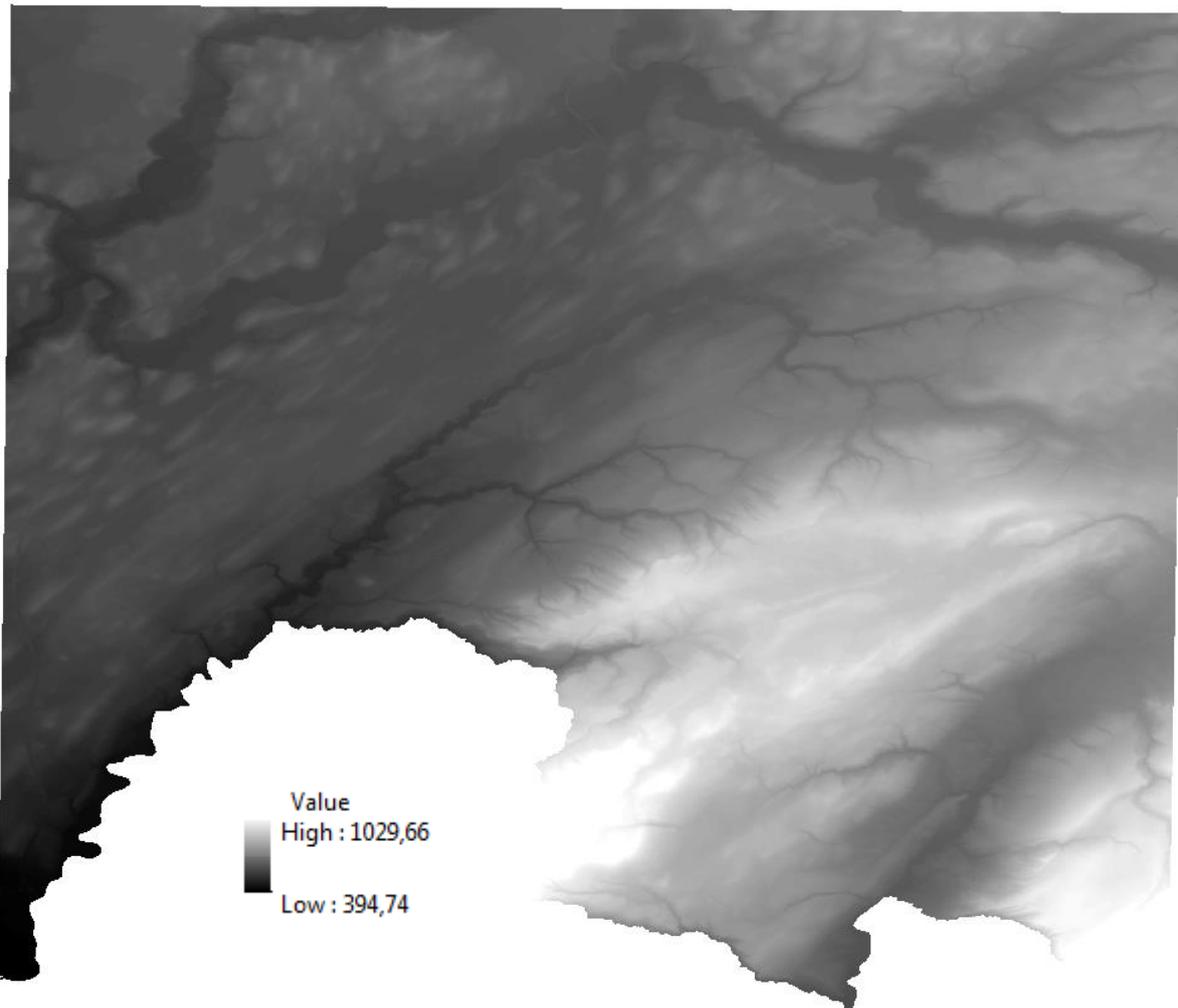
Für die Erzeugung des Terrains, das in Cesium visualisiert wird, wird der sog. `Cesium Terrain Builder` genutzt. Hierfür müssen zunächst einige Vorarbeiten an den Input-Daten geleistet werden:

- 1) Das ursprünglich vorliegende ASCII-XYZ-Format wird zunächst in das ERDAS-IMAGE-Format (.img) konvertiert.



- 2) Die erzeugten Raster-Kacheln werden anschließend mittels des Ansatzes „National Transformation Version 2“ (NTv2) vom DHDN nach ETRS89 transformiert. Durch diese eindeutige Auswahl der Transformationsmethode wird gewährleistet, dass die Daten korrekt transformiert werden.
- 3) Anschließend werden die beiden Rasterdateien des LDBV und des LGL mittels eines Raster-Mosaickers zu einer gemeinsamen Datei zusammengeführt und in ArcGIS mittels einer Maske (in diesem Fall das festgelegte Testgebiet) extrahiert.

Als Ergebnis erhält man die unten dargestellte img-Datei mit Höhen zwischen 395m und 1030m.



Als Software wird für die Berechnung des Geländes auf die sog. Geospatial Data Abstraction Library (GDAL) zurückgegriffen und die folgenden Werkzeuge benutzt:

- `gdalbuildvrt` → dies wird auf den Input-Datensatz angewendet
- `ctb-tile` → dies wird auf die erzeugte vrt-Datei angewendet

Ausgangsdatei ist demnach die `.img`-Datei. Im Kommandofenster (`cmd`) erfolgt nun die eigentliche Mosaikerzeugung. Hierfür ist es zunächst erforderlich, eine Textdatei anzulegen (z.B. `'List.txt'`).

Anschließend wird die VRT (Virtual Raster Tile) -Datei mittels des folgenden `cmd`-Befehls erstellt:

```
"gdalbuildvrt terrain_DSM.vrt -input_file_list List.txt"
```

Nun kann die eigentliche Kachelung der einzelnen Zoom-Stufen erfolgen. In diesem Fall werden 19 verschiedene Zoom-Stufen berücksichtigt, wobei das Level '18' am meisten Details des Geländes wiedergibt, '0' dagegen teilt die Erde lediglich in zwei Teile. Der entsprechende Befehl (für das Zoom-Level '18') sieht wie folgt aus:

```
"ctb-tile -s 18 -e 18 -c 4 terrain_DSM.vrt"
```

- s:** "start-zoom" → Zoom-Level, bei dem gestartet wird (größer als End-Zoom-Level)
e: "end-zoom" → Zoom-Level, bei dem geendet wird (geringer als Start-Zoom-Level und ≥ 0)
c: "thread count" → Anzahl der CPU-Threads, die für die Kachel-Erzeugung gebraucht werden

Nun liegen die Daten für alle Zoomstufen vor. Mittels des Open-Source-Webservers „Apache Tomcat“ und dem entsprechenden Verweis auf die Daten im Webbrowser kann das erzeugte DGM im WebClient angezeigt werden.

6.3.3 Ergebnis

Screenshots aus dem 3D-WebClient zeigen anhand beispielhafter Ausschnitte, wie das Digitale Geländemodell in Cesium dargestellt wird.



7 Qualitätsbewertung

7.1 Umgang mit Inkonsistenzen in Eingangsdaten

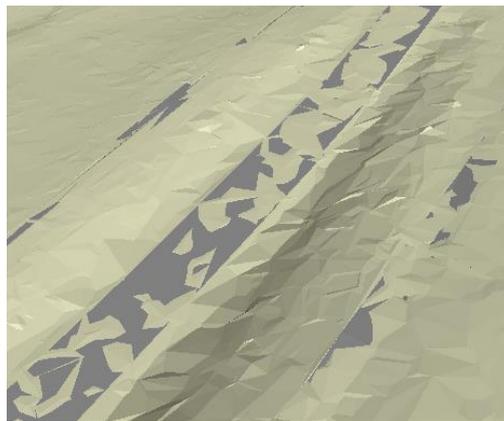
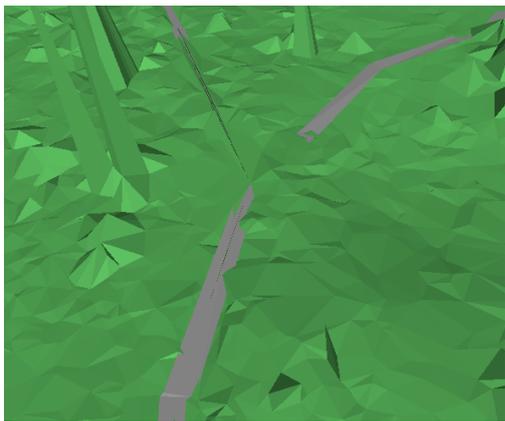
7.1.1 Allgemein

Generell ist der 3dfier tolerant gegenüber inkonsistenten Eingangsdaten. Das Ergebnis ist dann gegebenenfalls unbefriedigend. Das Programm ist daher gut geeignet für ein iteratives Vorgehen auf Basis von nur teilweise korrekten Eingangsdaten.

7.1.2 Problematisch

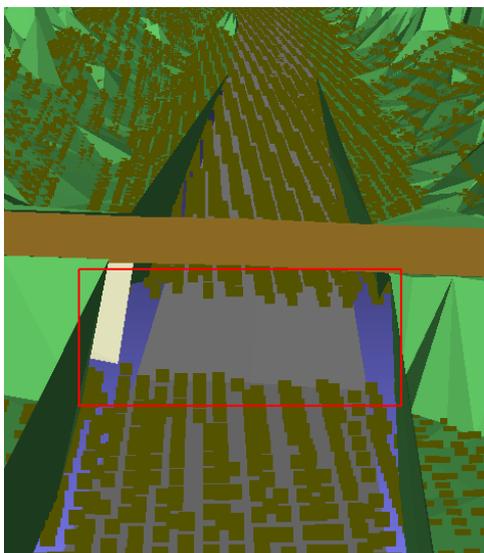
7.1.2.1 Nicht freigestellte Objekte

Nicht freigestellte Objekte in 2D (z.B. Straße in Wald) führt zu Überlappungen, im schlimmsten Fall zu Triangulierungsfehlern. Die korrekte Aufbereitung der 2D-Ausgangsdaten im Sinne der vom 3Dfier vorgegebenen Randbedingungen ist daher essentiell für die Ergebnisqualität.



7.1.2.2 Lücken in Punktwolke

Lücken in der Punktwolke führen zu Fehlern im 3D-DLM.



Die Lücke in der Punktwolke unterhalb der Brücke führt zu einem Knick in der Straße.

Der 3Dfier erscheint sehr sensibel gegenüber Datenlücken, wie sie in der Praxis naturgemäß in 3D-Punktwolken vorkommen. Bei Anwendung der aktuellen Version des 3Dfiers ist daher ein Präprozess sinnvoll, der solche Datenlücken über einfache Interpolationsprozesse oder durch Nutzung des vorhandenen DGMs auffüllt. Dabei können aber naturgemäß nur konkrete Lücken, wie z.B. unter Brücken behandelt werden.

7.1.3 Positiv

7.1.3.1 Inkonsistenzen in 2D-Ausgangsdaten

Inkonsistenzen in den 2D-Ausgangsdaten (Polygone) werden vom 3Dfier soweit möglich korrigiert, z.B. Multipolygone oder Selbstüberschneidungen (siehe 4.2.2.2).

7.1.3.2 Lücken im 2D-Datenbestand

Der 3Dfier toleriert bei der 3D-DLM-Erzeugung Lücken im 2D-Datenbestand; es kommt nicht zum Abbruch. Das Ergebnis beinhaltet aber naturgemäß entsprechende Lücken im 3D-DLM.



7.2 Verwendung von unklassifizierten Punktwolken

Für das Tool '3Dfier' ist die Verwendung eines (unklassifizierten) bDOMs zwar technisch möglich, aber faktisch nicht sinnvoll. Hierzu ein Auszug aus README zum 3Dfier⁵:

We expect the LAS/LAZ to be classified according to the ASPRS Standard LIDAR Point Classes v1.4 (Table 4.9 of this PDF), and at a minimum these should be defined:

- 0-1: Created, never classified and/or unclassified
- 2: Ground
- 3-5: Vegetation

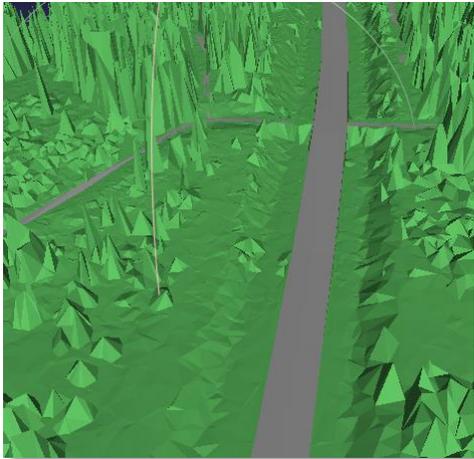
If the vegetation is not classified or not filtered out, then buildings might be taller and there might be artefacts in the terrain.

Das heißt vereinfacht: Bodenpunkte **MÜSSEN** vorhanden sein, Vegetation **SOLLTE** vorhanden sein.

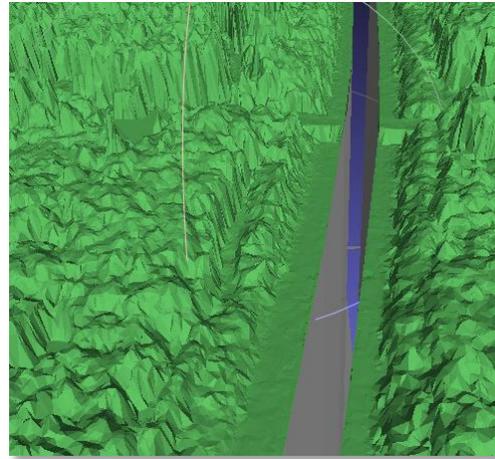
⁵ <https://github.com/tudelft3d/3dfier/blob/master/README.md>

So zieht zum Beispiel die Klasse *Road* die Höheninformation nur aus den als Bodenpunkt (Class 2: Ground) klassifizierten Punkten. Wurden keine Bodenpunkte klassifiziert, werden die Objekte dieser Klasse mit der Höhe 0 versehen:

Umsetzung mit LIDAR-Punktwolke



Umsetzung mit bDOM-Punktwolke



7.3 Betrachtung der einzelnen Objektarten

Die im Folgenden aufgeführten Untersuchungen hinsichtlich der Qualität der mit dem 3Dfier erzeugten Objekte wurden auf unterschiedliche Weise vorgenommen:

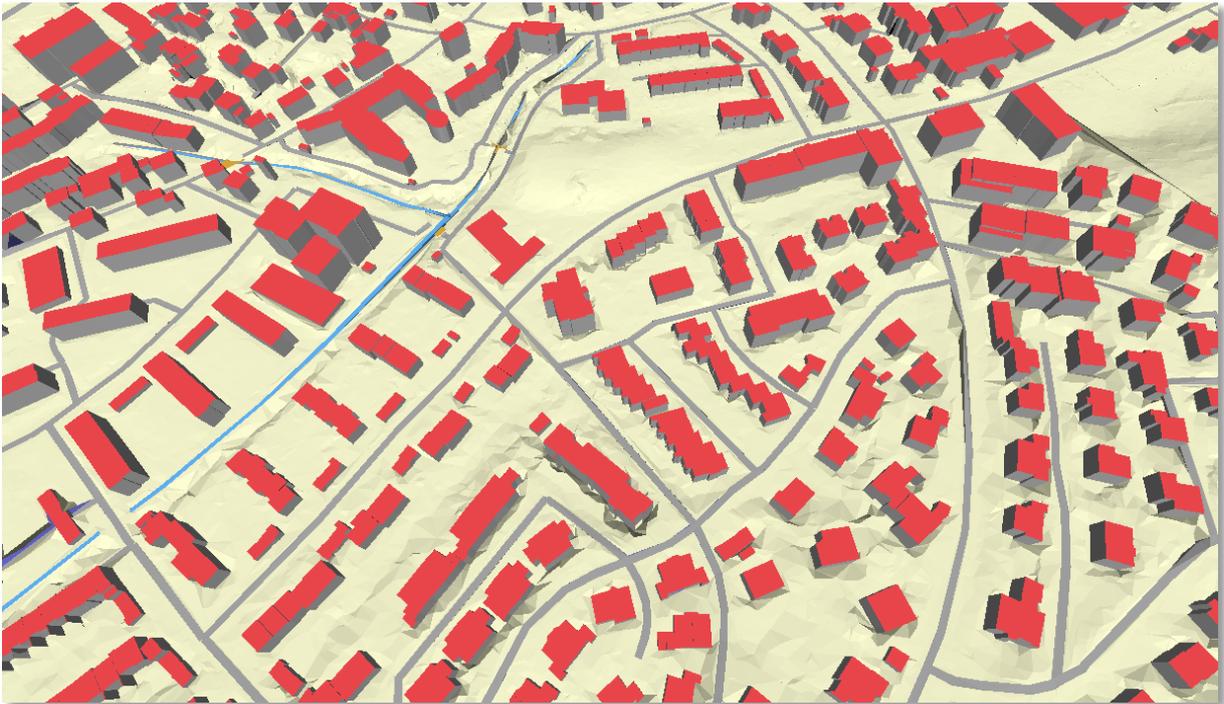
- Zum einen erfolgte eine **manuelle Sichtung der Daten**. Dabei wurden die Objekte sowohl einzeln, als auch in Interaktion mit anderen benachbarten Objektarten betrachtet und hinsichtlich Auffälligkeiten untersucht.
 - Diese Art der Betrachtung wurde für die Objektarten BUILDING, BRIDGE sowie ROAD vorgenommen (siehe 7.3.1, 7.3.2 und 7.3.3).
- Zum anderen erfolgte eine **statistische Auswertung**. Dabei wurden die Normalenvektoren der erzeugten Flächen betrachtet.
 - Diese Art der Betrachtung wurde für die Objektart WATER vorgenommen (siehe 7.3.4).

Detaillierte Beschreibungen sowie die Resultate dieser Untersuchungen sind in den folgenden Kapiteln aufgeführt.

7.3.1 BUILDING

7.3.1.1 Grundlage

Grundlage für die Umsetzung ist die für die amtliche LoD-Erzeugung verwendeten Grundrisse (ALKIS) (siehe auch 3.3.1 BUILDING). Die Erzeugung im Rahmen des 3Dfiers ist wegen Nachbarschaftsbeziehungen notwendig und erfolgt immer in LoD1.



Im Postprozess werden dann die erzeugten LoD1-Modelle durch amtliche LoD2-Modelle ersetzt (siehe 6.1 Gebäude).

7.3.1.2 Lage

Um Lücken im 3D-DLM-Modell zu vermeiden, muss die Lage der verwendeten amtlichen 3D-Gebäude exakt den für den 3Dfier-Lauf verwendeten Grundrissen entsprechen. Kritisch können die unterschiedlichen Zeitpunkte zwischen Modellerzeugung und 3Dfier-Lauf sein.

7.3.1.3 Bodenhöhe

Für die nahtlose Integration des amtlichen 3D-Modells in das umgebende Gelände ist die Verwendung des amtlichen DGMs Voraussetzung (siehe 6.3 DGM).



7.3.2 BRIDGE

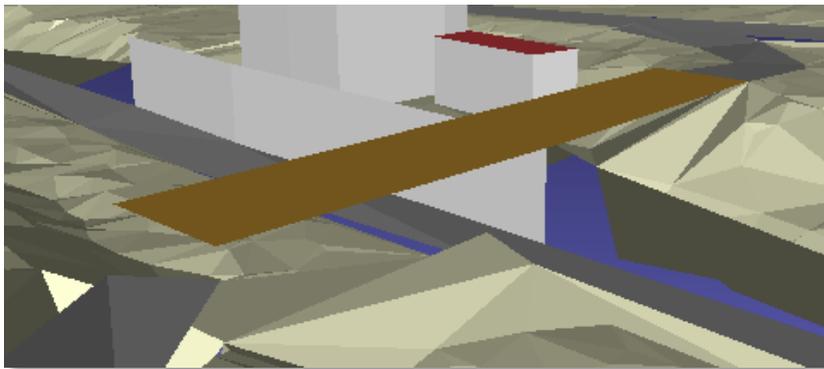
Im Folgenden werden beispielhaft bei der Umsetzung aufgetretene Probleme beschrieben. Wo möglich und sinnvoll, werden Vorschläge zur Vermeidung dieser Probleme aufgeführt.

An dieser Stelle sei darauf hingewiesen, dass es sich bei den 2D-ATKIS-Daten generell um generalisierte Daten handelt, die von der tatsächlichen Lage unter Umständen abweichen können und somit nicht mit dem genauen DGM konsistent sind.

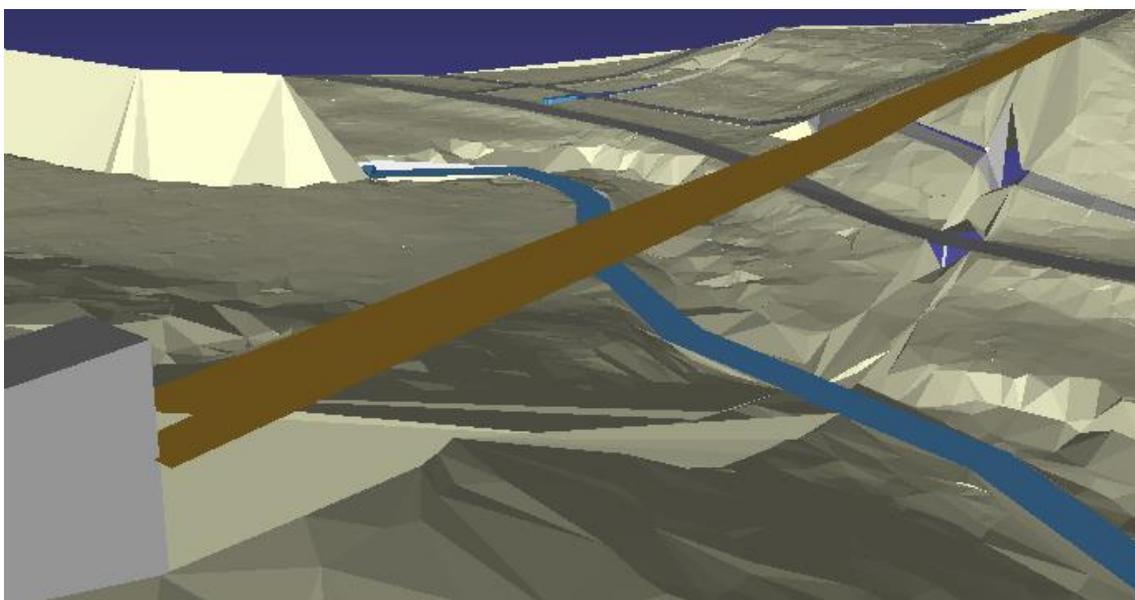
Diese Tatsache führt zu einigen der im Folgenden aufgeführten Probleme. So müssten u.a. Widerlager der Brücken auf das DGM angepasst werden, aber beispielsweise auch die Straßenachse nachgebessert werden (z.B. mittels True Orthophoto oder DGM) (vgl. Kap. 7.3.3.4). Auch die Umringe der ATKIS-Daten (beispielsweise für die Repräsentation von Seen) resultieren aus Generalisierungen und passen somit nicht exakt zum DGM (vgl. Kap. 7.3.4).

7.3.2.1 Einheitliche Höhe

Brücken werden immer auf einer einheitlichen Höhe modelliert (waagrecht). Dies entspricht häufig nicht der Realität und führt zu einem Höhenversatz beim Anschluss an die angrenzenden Verkehrsobjekte.



Offensichtlich wird bei der Objekterzeugung nicht die mittlere Höhe der über dem 2D-Brückenobjekt liegenden Punkte verwendet, sondern die Höhe einer Seite angehalten. Sinnvoller wäre hier zumindest eine lineare Interpolation entlang der Brückenachse unter Verwendung der ermittelten Höhen an beiden Brückenseiten.

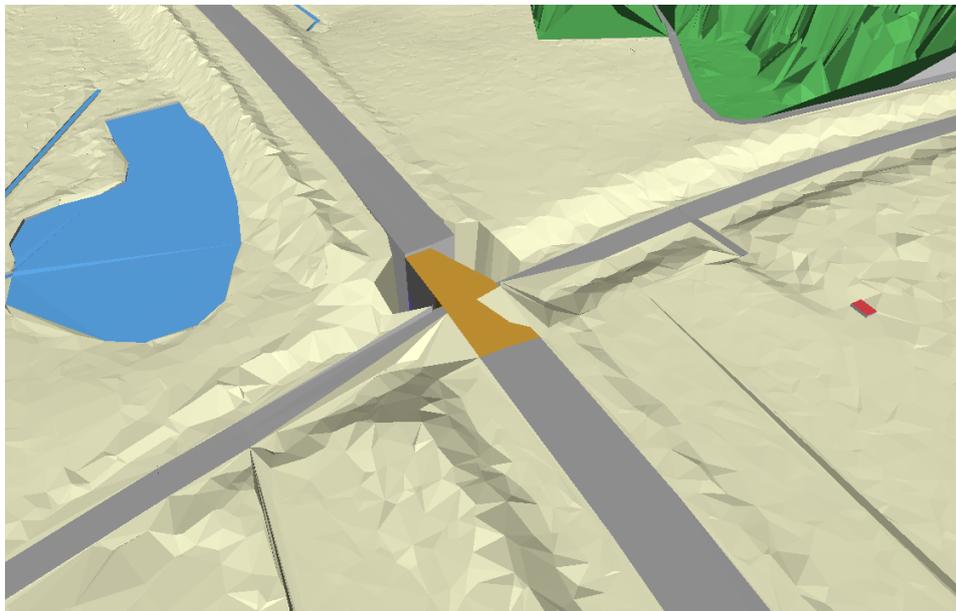


7.3.2.2 Triangulationsprobleme

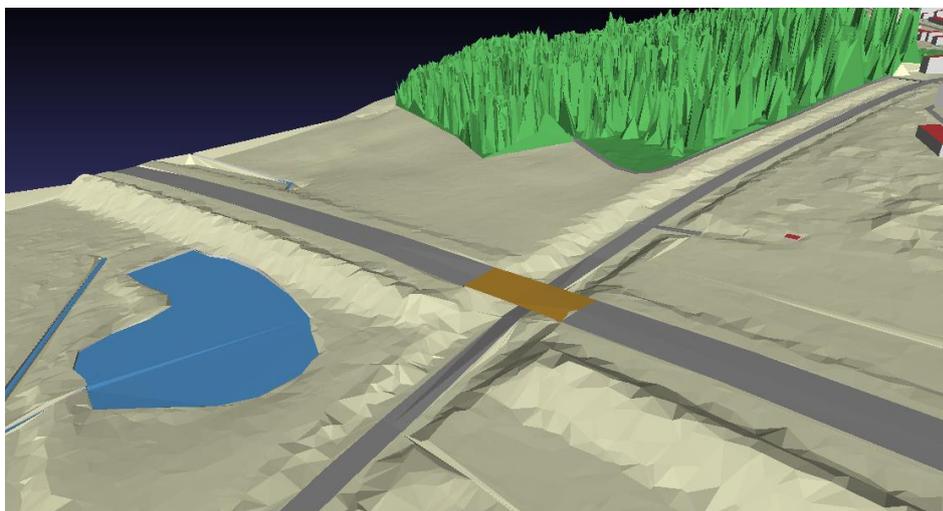
7.3.2.2.1 DGM statt Bodenpunkte

Generell können Triangulationsprobleme durch die Verwendung des DGMs statt der Bodenpunkte aus der Punktwolke minimiert werden.

Umsetzung mit LIDAR-Bodenpunkten:



Umsetzung nach Austausch der Bodenpunkte durch die entsprechenden Punkte aus dem DGM:



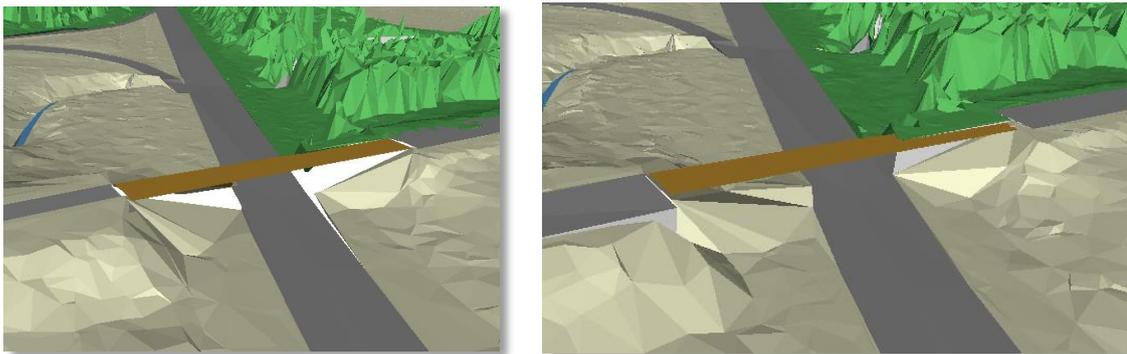
7.3.2.2 Lückenlose 2D-Informationen

Neben den geschlossenen 3D-Informationen aus Punktwolke und DGM sind ebenso die lückenlosen und überlappungsfreien 2D-Informationen entscheidend für eine saubere Umsetzung von Brücken. Andernfalls entstehen auch hier Triangulationsfehler, wie z.B. Lücken im 3D-DLM. Die folgenden Abbildungen illustrieren diesen Sachverhalt beispielhaft:

1. Lückenhafte 2D-Daten ergeben 'Löcher' im Gelände



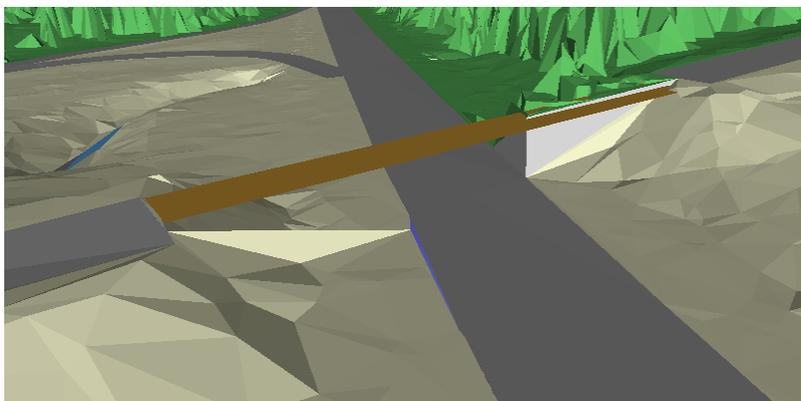
Nach Korrektur sind diese Löcher beseitigt:



Vorher (fehlerhafte Triangulation als weiße Flächen erkennbar)

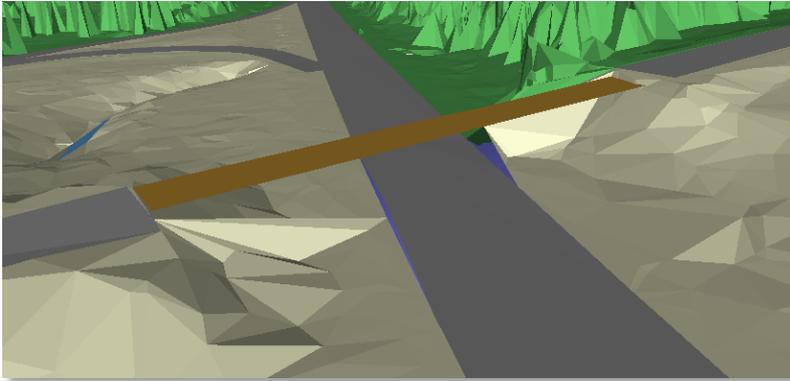
Korrigiert

2. Werden die Bodenpunkte durch Punkte aus dem DGM ersetzt (vgl. 7.3.2.2.1), verschwindet der Straßenknick unter der Brücke:



Aber:

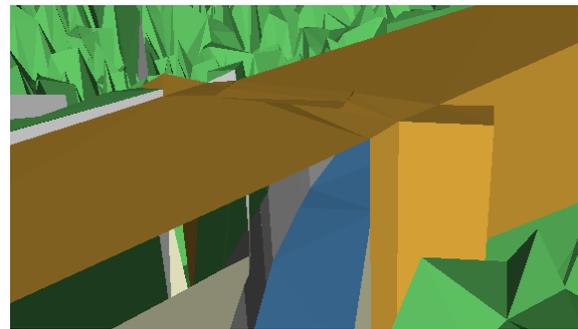
Bei sauberer Bereinigung der 2D-Daten von Wald und Gelände (keine Überlappung von Wald im Straßenbereich / Brücke) entstehen neue Triangulierungsprobleme (erneut Löcher im 3D-DLM).



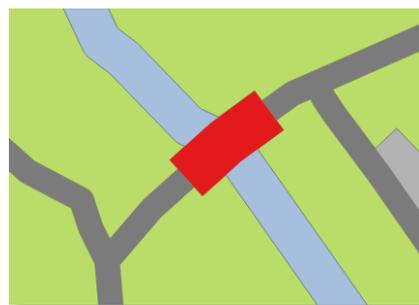
Hierfür ist die Ursache noch unklar.

7.3.2.3 Brücken queren Brücken

Liegen Brücken über Brücken, so hat der 3Dfrier Probleme bei der Erkennung der unterschiedlichen Höhen.

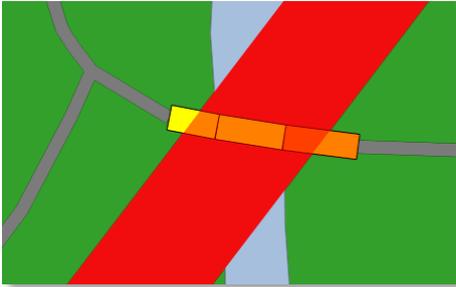
2D-Eingabedaten**3D-DLM****7.3.2.4 Brücken in dichter Vegetation**

Trotz sauber freigestellter 2D-Daten überlappt das Terrainobjekt in 3D das Verkehrsobjekt. Dieser Effekt tritt insbesondere bei Vegetation in unmittelbarer Nähe von Verkehrsobjekten auf (dies liegt u.a. daran, dass die Lidar-Punktwolke u.U. in den Straßenraum ragt).



7.3.2.5 Einzelpolygone vereinigen

Polygone gleicher Objekt-ID sollten vor der Übergabe an den 3Dfrier vereinigt werden, um Fehlerkennungen zu vermeiden.



7.3.2.6 Zusammenfassung und Fazit

Vor allem und gerade im Bereich von Brücken besteht eine hohe Sensibilität in Bezug auf die Ausgangsdaten. In vielen Fällen ist das vollautomatische Ergebnis aber bereits zufriedenstellend.



Die Lage der Brücken passt im Testgebiet zu anschließenden Verkehrsobjekten, so dass keine Lücken in X und Y zu den anschließenden Straßen entstehen.



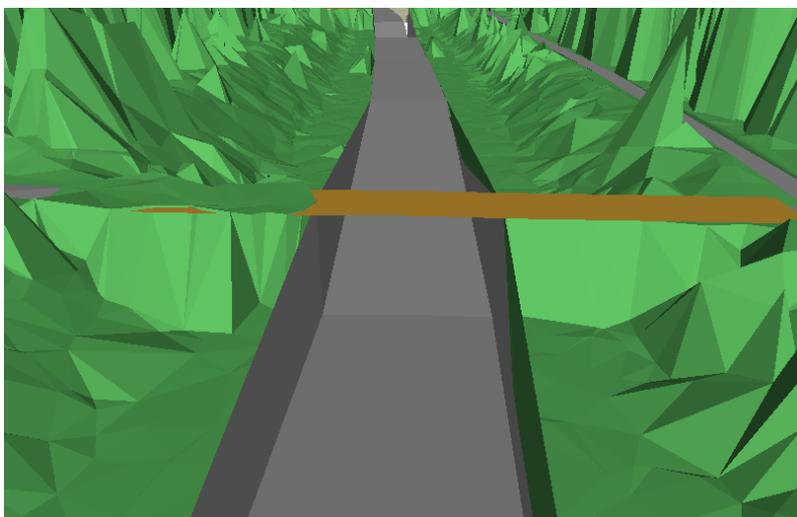
Die Höhe der Brücken passt dagegen häufig nicht exakt zu den angrenzenden Straßenobjekten (siehe 7.3.2.1).

7.3.3 ROAD

Im Folgenden werden beispielhaft bei der Umsetzung aufgetretene Probleme beschrieben. Wo möglich und sinnvoll werden Vorschläge zur Vermeidung dieser Probleme aufgeführt.

7.3.3.1 Lücken im Punktdatenbestand

Liegen Lücken im Punktdatenbestand vor (z.B. unter Brücken) erfolgt eine unsaubere Interpolation der Höhe, welche zu in der Realität nicht vorhandenen Tälern und Kuppen im Straßenverlauf führen.



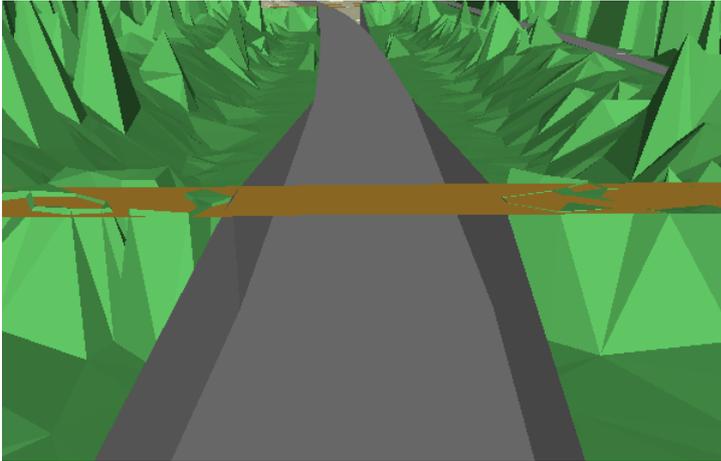
7.3.3.1.1 Lösungsvorschläge

3Dfier-Algorithmus

Interpolation ausschließlich auf Basis vorhandener Punkte. Dabei Berücksichtigung der Semantik des Straßenobjekts und damit Vermeidung von starken Gefällesprüngen oder Unstetigkeitsstellen.

Eingangsdaten

Automatische Lückenfüllung z.B. durch Austausch der Bodenpunkte durch Punkte aus dem amtlichen DGM.



Ebener Straßenverlauf nach Austausch der Bodenpunkte durch das DGM

7.3.3.2 Geringe Punktdichte

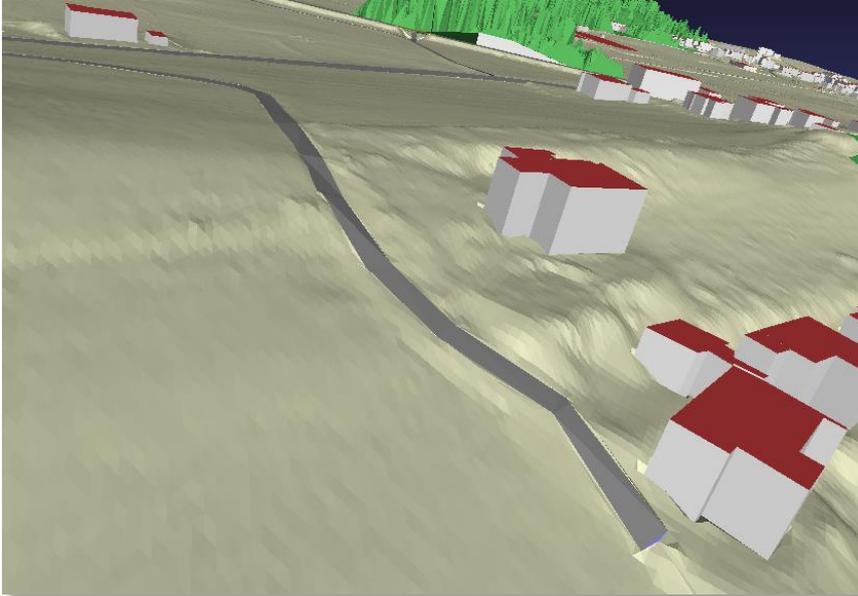
Bei geringer Punktdichte kommt es zu einer unruhigen Straßenoberfläche. Das semantische Wissen um eine ebene Straßenoberfläche kommt hier offensichtlich nicht oder nicht ausreichend gewichtet zum Tragen (siehe 4.3.1).



Unsteter Straßenlauf bei verringerter Punktdichte (*thinning 5*)

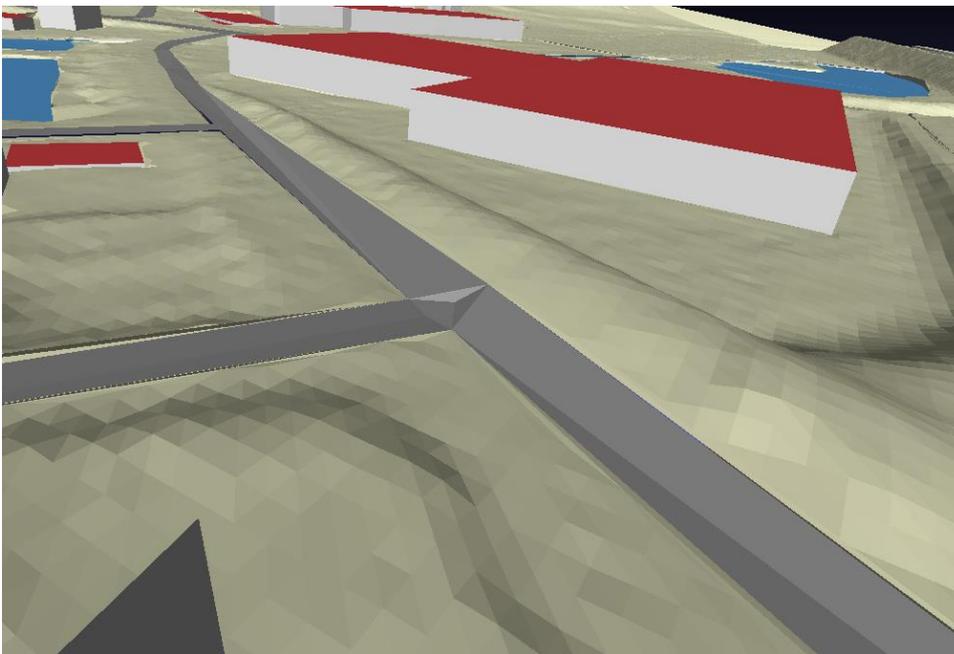
7.3.3 Unterschiedliches Gefälle

Bei sich änderndem Gefälle entstehen Unstetigkeiten im Übergang zwischen den unterschiedlichen Steigungen. Bei geneigtem Straßenverlauf kommt es dann häufig zu einem unebenen Straßenverlauf durch die fehlerhafte Triangulation.

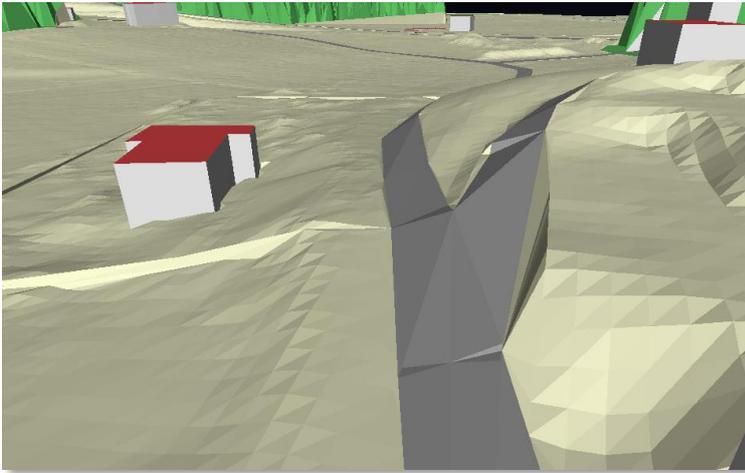


7.3.3.4 Neigungen im Straßenquerschnitt

Bei z.B. durch Lageversatz verursachten uneinheitlichen Höhen der auf der Straße liegenden Punkte kommt es fälschlicherweise zu Neigungen im Straßenquerschnitt. Straßen liegen dann beispielsweise geneigt an einer Böschung.



Außerdem ist für einen ebenen Straßenverlauf eine Pufferung der linienhaften 2D-Straßenelemente mit einer der Realität entsprechenden Breite elementar. Im unten abgebildeten Beispiel führen sowohl ein Lageversatz, als auch eine falsch angenommene Straßenbreite zu Folgeproblemen im 3D-DLM.



7.3.3.5 Enger Kurvenverlauf

Ein enger Kurvenverlauf führt ebenfalls zu Unstetigkeiten im Verlauf der Straßenoberfläche. Außerdem werden die Kurven aktuell zu grob wiedergegeben. Hier ist der Algorithmus zur Pufferung der 2D-Linien zu verfeinern (vgl. 3.3.3).

2D-Daten



3D-DLM



7.3.3.6 Unterscheidung der Verkehrswege

Der 3Dfrier bietet in der aktuellen Version keine Möglichkeit zur Unterscheidung der unterschiedlichen Verkehrswege (z.B. Straßen und Gleise). Aus Sicht der 3D-Modellierung im Anwendungsfall dieses Projektes erscheint dies auf Grund des Generalisierungsgrades unkritisch.



2D-Daten (Gleis in Schwarz)



3D-DLM – Keine visuelle Unterscheidung

Für die Unterscheidung in der Visualisierung erfolgt im Rahmen eines Postprozesses eine Trennung der Verkehrstypen.

7.3.3.7 Zusammenfassung und Fazit

Zusammengefasst ergeben sich folgende Kriterien für eine Qualitätsverbesserung des resultierenden 3D-DLMs:

- Bodenpunkte der Punktwolke durch Informationen aus amtlichem DGM (z.B. DGM1) ersetzen
- Möglichst hohe Übereinstimmung in der Lage zwischen DLM und DGM
- Realistische Breiteninformation pro Abschnitt eines Verkehrsweges

Zufriedenstellende Ergebnisse werden aktuell erreicht bei:

- Ebenem oder leicht welligem Gelände
- Geradem oder nur leicht kurvigem Achsenverlauf
- Daten entsprechen möglichst weitgehend oben genannten Kriterien zur Qualitätsverbesserung
- Realistische Darstellung bei Übersichtsbetrachtung

Grobe Fehler beschränken sich im Wesentlichen auf Kreuzungsbereiche und Über- / Unterführungen.

7.3.4 WATER

Im Folgenden werden beispielhaft bei der Umsetzung aufgetretene Probleme beschrieben. Hierbei wird zwischen den Stehenden Gewässern und den Fließgewässern unterschieden.

7.3.4.1 Stehendes Gewässer

Es gilt die Annahme, dass die Neigung der stehenden Gewässer 0 sein muss. Überprüfen lässt sich dies auch dadurch, dass die Attribute `min_Hoehe` und `max_Hoehe` den gleichen Wert haben müssen.

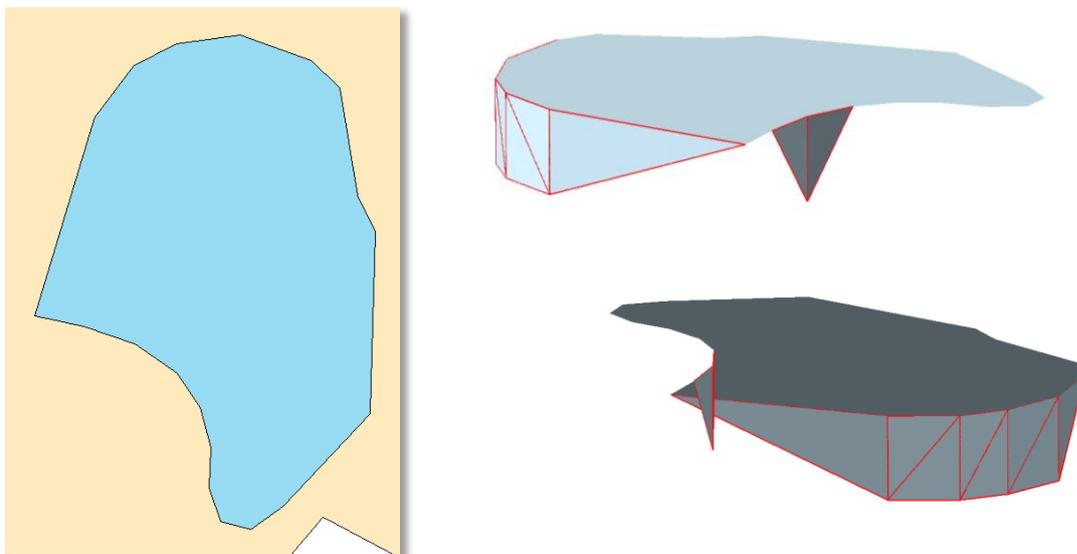
Das folgende selektierte Gewässer-Objekt (siehe Abbildung) ist demnach eben und die Geometrie ist korrekt.



Betrachtet man alle stehenden Gewässer als fehlerhaft, die keine horizontale Fläche repräsentieren, so ergibt sich für die Daten des *LDBV* ein Prozentanteil von **19,8% der stehenden Gewässer**, die **fehlerhaft** sind; für die *LGL*-Daten sind dies **21,2%**.

7.3.4.1.1 Fehlerhafte Stehende Gewässer

Im Folgenden sei beispielhaft ein fehlerhaftes stehendes Gewässer dargestellt.



In der linken Abbildung sieht man das Gewässerpolygon, das komplett von der landwirtschaftlichen Fläche umgeben ist (es gibt keinerlei Überschneidungen zwischen den beiden Polygonen). Auf der rechten Seite erkennt man die rot umrandeten nicht-horizontalen Geometrien des Gewässer-Objekts. Diese Fehler führen auch zu einem falschen Wert des Attributs `min_Hoehe`.

GMLID	DEBYBDLMgH0003kZ
Landeskennung	BY
Modellartenkennung	Basis-DLM#DTK25
class	1080
Objektart_Text	AX_StehendesGewaesser
Objektart	44006
Widmung	1340
hatDirektUntenExistiert	0
min_Hoehe	768.65
max_Hoehe	780.47
Flaecheninhalt	3050.2

7.3.4.2 Fließgewässer

Für die Qualitätsbewertung der Fließgewässer gilt die Annahme, dass die Flächennormalen angrenzender Objekte in etwa (auf 5 Grad genau) in die gleiche Richtung orientiert sein müssen. Ist dies nicht der Fall, werden die Objekte als fehlerhaft bewertet.

Mit der oben aufgestellten Annahme ergibt sich somit für die Daten des *LDBV* ein Prozentanteil von **58% der Fließgewässer**, die **fehlerhaft** sind. Für die *LGL*-Daten sind es **64,3%**.

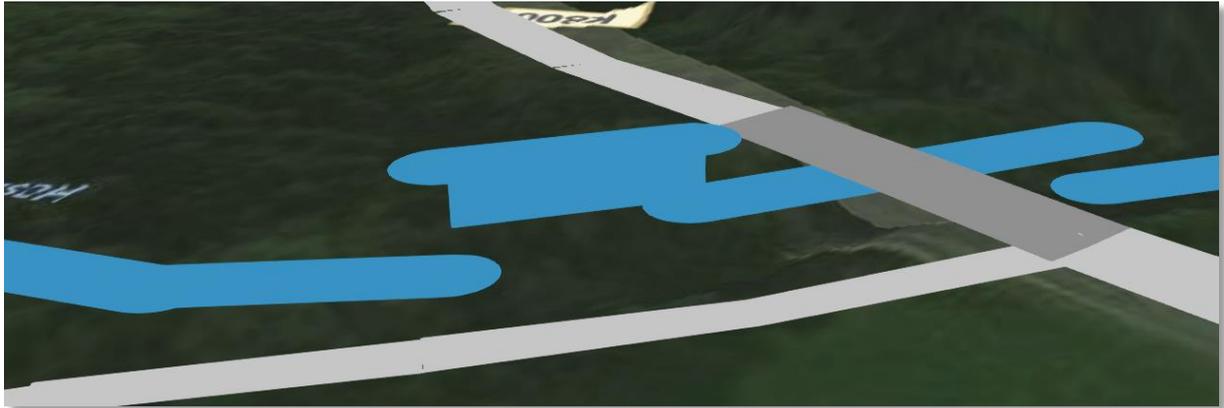
Der Grund hierfür kann oft sehr ähnlich zu den auftretenden Fehlern bei den Straßen sein.

7.3.4.2.1 Neigungen im Gewässerquerschnitt

Wie bereits bei den Straßen erläutert, erfolgt die Pufferung der linienhaft vorliegenden Gewässerachsen lediglich um Standardwerte (es liegen keine realen Flussbreiten vor). Zusätzlich kann es z.B. durch Lageversatz dazu führen, dass die Gewässerobjekte geneigt an einer Böschung liegen.

7.3.4.2.2 Falsche Höhenzuweisung

Vor allem im Datenbestand des LGL sind die ursprünglichen Gewässerachsen durch sehr viele kleine Abschnitte repräsentiert. Dadurch kann es zwischen den einzelnen Objekten zu starken Höhendifferenzen kommen. Dies sollte bereits in der Vorverarbeitung der Daten berücksichtigt und verbessert werden.



7.3.4.2.3 Positive Beispiele

Ein positives Beispiel für die Interaktion zwischen flächenhaft vorliegender Fließgewässer und Brücken zeigen die folgenden Abbildungen.



8 Bereitstellung der Ergebnisdaten

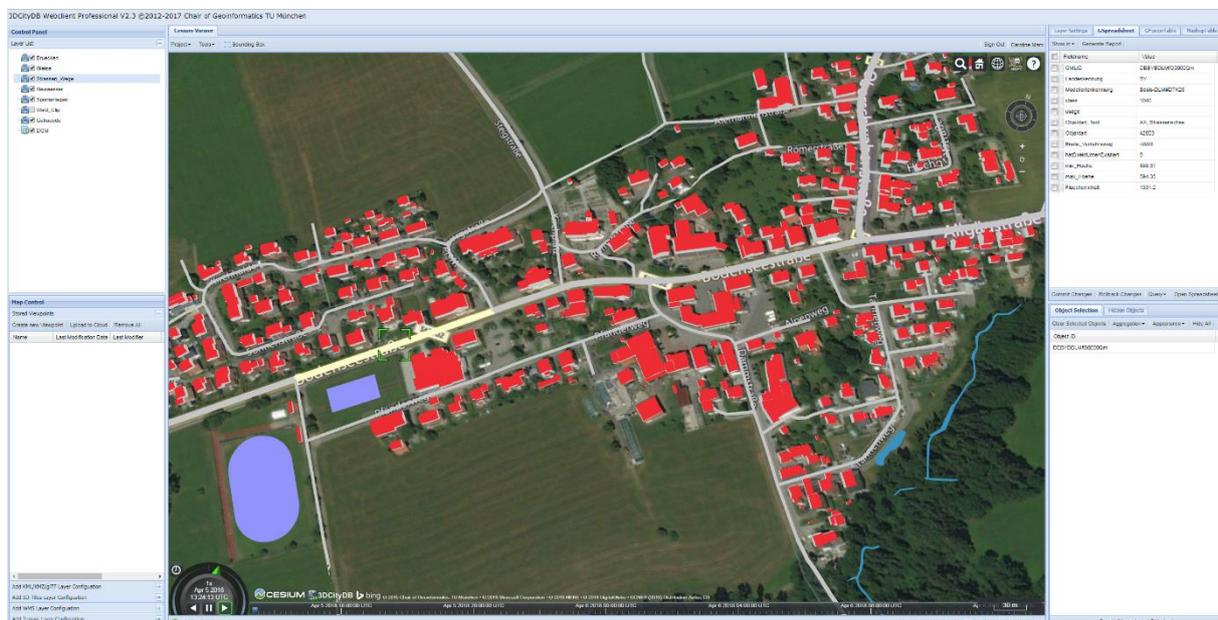
Abgesehen vom DGM wurden alle Daten als CityGML-Instanzdokumente erzeugt und mittels des CityGML-Importers in eine 3D City Database auf einem PostgreSQL DBMS (mit PostGIS) importiert. Nach Festlegen der BoundingBox und Anpassung diverser Einstellungen (z.B. Füllfarbe der Objekte) werden die Daten als COLLADA/gITF exportiert. Dieses Datenformat ist notwendig, um die Inhalte der Datenbank für eine Visualisierung in Cesium aufzubereiten.

Eine Anreicherung der Objekte um thematische Attribute erfolgt mittels des Spreadsheet-Exporters.

Column's title	Column's content
Landeskennung	CITYOBJECT_GENERICATTRIB/STRVAL[ATTRNAME='Landeskennung']
Modellartenkennung	CITYOBJECT_GENERICATTRIB/STRVAL[ATTRNAME='Modellartenkennung']
Objektart_Text	CITYOBJECT_GENERICATTRIB/STRVAL[ATTRNAME='Objektart_Text']
Objektart	CITYOBJECT_GENERICATTRIB/INTVAL[ATTRNAME='Objektart']
Bauwerksfunktion	CITYOBJECT_GENERICATTRIB/INTVAL[ATTRNAME='Bauwerksfunktion']
Breite_Objekt	CITYOBJECT_GENERICATTRIB/REALVAL[ATTRNAME='Breite_Objekt']
Durchfahrtshoehe	CITYOBJECT_GENERICATTRIB/REALVAL[ATTRNAME='Durchfahrtshoehe']

Die Visualisierung des 3D-Landschaftsmodells wird über einen Web-Browser realisiert. Dies geschieht ohne Installieren oder Aktivieren zusätzlicher Erweiterungen (Plug-Ins) für den Browser.

Lädt man den WebClient⁶, erhält man folgende Ansicht:



⁶ Link zum WebClient:

<https://www.3dcitydb.org/3dcitydb-web-map-tum/2.3/index.html?&con-fig=https%3A%2F%2Fdocs.google.com%2Fspreadsheets%2Fd%2F1c9eDZ4jW2BMt7DPqIQC9uPlw4zSAK-XwhPu-XmQSZTc%2Fedit%3Fusp%3Ddrivesdk&gclid=DEBYBDLMgG0002k8&shadows=false&terrainShadows=NaN&latitude=47.61919281940638&longitude=9.832996145765712&height=1130.905755747691&heading=5.050817302078285&pitch=-34.4088944040537&roll=0.02050171162506003>

... mit den folgenden thematischen Attributen für ein beliebig ausgewähltes Straßenobjekt:

Layer Settings		GSpreadsheet		GFusionTable		MashupTable	
Show in ▾	Generate Report						
<input type="checkbox"/>	Fieldname	Value					
<input type="checkbox"/>	GMLID	DEBYBDLMfG0000Qm					
<input type="checkbox"/>	Landeskennung	BY					
<input type="checkbox"/>	Modellartenkennung	Basis-DLM#DTK25					
<input type="checkbox"/>	class	1040					
<input type="checkbox"/>	usage						
<input type="checkbox"/>	Objektart_Text	AX_Strassenachse					
<input type="checkbox"/>	Objektart	42003					
<input type="checkbox"/>	Breite_Verkehrsweg	-9998					
<input type="checkbox"/>	hatDirektUntenExistiert	0					
<input type="checkbox"/>	min_Hoehe	588.37					
<input type="checkbox"/>	max_Hoehe	594.35					
<input type="checkbox"/>	Flaecheninhalt	1331.2					

Mittels dieser Attribute lassen sich nun Abfragen und Analysen durchführen.

Beispielsweise sind in unterer Abbildung alle Gebäude farblich hervorgehoben, die als Dachform ein Flachdach haben (roofType = 1000).



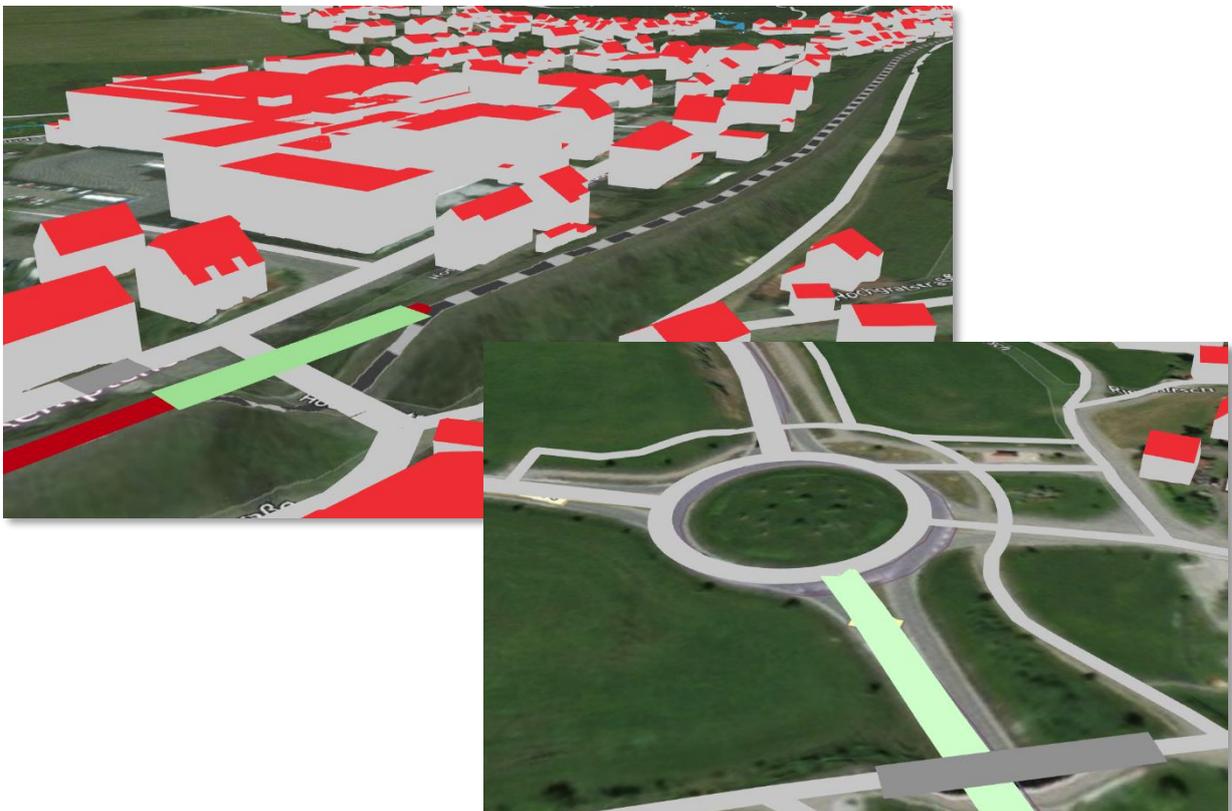
9 Zusammenfassung & Ausblick

3D-Stadt- und Landschaftsmodelle werden häufig als reine graphische oder geometrische Modelle definiert; dabei werden die semantischen Aspekte oft vernachlässigt. Dies hat zur Folge, dass diese Modelle fast ausschließlich für Visualisierungszwecke verwendet werden können, nicht jedoch für thematische Abfragen, Analyseaufgaben oder Simulationen. Für viele Anwendungsszenarien sind aber genau diese semantischen Informationen von enormer Wichtigkeit.

Aus diesem Grund wird im Projekt "3D-DLM" jedes erzeugte Objekt des digitalen Landschaftsmodells semantisch angereichert. Diese sind dadurch sowohl thematisch, als auch räumlich selektierbar und abfragbar (siehe erzeugter 3D-WebClient (Link: S. 57 ⁶) / siehe Kap. 2 'Semantisches 3D-Landschaftsmodell').

Im Rahmen des Projekts „3D Digitales Landschaftsmodell“ (3D-DLM) folgte nach der Konzeptionsphase (2016) in einem zweiten Schritt die Demonstrationsphase (2017), in der für ein definiertes Testgebiet im Grenzbereich von Bayern und Baden-Württemberg der Aufbau eines konsistenten 3D Digitalen Landschaftsmodells erfolgte. Hierfür wurde auf das Open Source Tool „3Dfier“ zurückgegriffen, das an der TU Delft entwickelt und in den Niederlanden bereits in der Praxis eingesetzt wird.

In der Demonstrationsphase konnte die grundsätzliche Machbarkeit einer hochgradig automatisierten Ableitung eines flächendeckenden 3D-DLM aus den vorliegenden ALKIS- und ATKIS-Daten nachgewiesen werden. Das verwendete Werkzeug 3Dfier lieferte mit den beschriebenen Ausgangsdaten im vollautomatischen Prozess ein fachlich sinnvolles Ergebnis, das aber im Detail noch inhaltliche Fehler enthält. Der Hardware-Ressourcenbedarf sowie die Laufzeiten der Prozessierung des Testgebiets erscheinen für eine Anwendung auch landesweiter Datenbestände geeignet.



Für ein konsistentes Ergebnis ist eine entsprechende Aufbereitung der Ausgangsdaten notwendig. Das betrifft sowohl die Vektordaten, als auch die verwendeten Oberflächenpunkte (Austausch der Bodenpunkte durch hochauflösendes DGM). Auch hat die Verwendung von realen Breiten anstelle von Klassen bei Straßen- und Gewässerobjekten elementare Bedeutung. Die notwendige Vorverarbeitung der Daten, genauso wie das abschließende Mapping auf ein CityGML-Zielmodell lässt sich gut in einen automatisierten Gesamtprozessablauf integrieren. Somit bietet der untersuchte Ansatz eine praktikable Möglichkeit, um 2D-Daten automatisiert und konsistent in die dritte Dimension anzuheben und dabei alle gewünschten fachlichen Informationen zu übernehmen. Weitere vertiefte Untersuchungen könnten z.B. die Einbeziehung weiterer Software-Werkzeuge sowie die Verkettung der Teilschritte zu einem Gesamtprozess beinhalten.

Neben den bereits erwähnten Empfehlungen hinsichtlich der Ausgangsdaten, die bei Anwendung des 3Dfiers zu inhaltlichen Fehlern im dreidimensionalen Landschaftsmodell führen können, bestehen hinsichtlich der Erzeugung eines konsistenten semantischen 3D-DLMs weitere offene Forschungsfragen, auf die im Folgenden näher eingegangen werden soll:

- Zentrales Thema ist die Inkonsistenz zwischen dem DLM und dem Höhenmodell (DGM / DOM). Aufgrund der unterschiedlichen Erfassungsmethoden bestehen Genauigkeitsunterschiede der Datengrundlagen, welche unter Umständen zu einem horizontalen Versatz zueinander und damit zu Fehlern im 3D-DLM führen können. Dies führt zu der Forderung, dass einerseits die Geometrie der 2D-Daten an das Geländemodell angepasst werden muss, andererseits aber unter Umständen auch das DGM verändert werden muss. Zudem ist eine passende Methodik zu erforschen, die die ursprünglich linienhaft vorliegenden ATKIS-Daten möglichst automatisiert in der zum DGM passenden Breite flächenhaft erzeugt. Ziel ist somit eine automatisierte Lage- und Breitenanpassung.
- Als erster Schritt zur automatisierten Lageanpassung sollten Methoden zur automatischen Aufdeckung von Inkonsistenzen zwischen 2D-DLM und DGM/DOM auf Grundlage des 3D-DLM entwickelt werden. Die in diesem Projekt zur Qualitätsbewertung der Gewässerobjekte entwickelten Methoden weisen bereits in die Richtung, müssen jedoch für weitere Objektarten angepasst werden. Je nach Fortführungszyklus von DGM/DOM und ATKIS könnten diese Methoden auch zur Ermittlung von Fortführungsfällen im 2D-DLM genutzt werden.
- Für die Erzeugung eines konsistenten 3D-DLMs muss das Geländemodell dahingehend angepasst werden, dass darin enthaltene 3D-Objekte aus diesem entfernt werden müssen. Dies betrifft z.B. Brückenobjekte, die im DGM aktuell oft noch enthalten sind. Gerade im Hinblick auf den künftigen Ansatz der AdV, Brückenkörper bereits in 3D zu erfassen, spielt diese Thematik eine wichtige Rolle.
- Auch die Interaktion der erzeugten 3D-Objekte untereinander muss künftig näher untersucht werden. So muss beispielsweise gewährleistet sein, dass Straßen ohne Unterbrechung an die modellierten Brücken anschließen. Für die korrekte Ableitung von 3D-Modellen für Brückenbauwerke muss das Werkzeug 3dfier zudem erweitert werden, um auch die Längsneigung von Brücken berücksichtigen zu können. Nur so kann eine korrekte Topologie sichergestellt werden, was beispielweise für Routingaufgaben essentiell ist.
- Des Weiteren muss die Problematik der koplanaren Flächen detailliert betrachtet werden. Aktuell besteht bei der Überlagerung der 2D-Geometrien auf das Geländemodell das Problem des sog. "z-fightings", was bei der Visualisierung zu Problemen führen kann. Ein möglicher Lösungsansatz hierbei könnte es sein, das DGM zu triangulieren, alle DLM-Objekte davon zu 'subtrahieren' und somit ein Geländemodell als 'zerstückelte' Restfläche vorliegen zu haben. Damit wäre eine überlappungsfreie Repräsentation der Erdoberfläche möglich.
- Die im Projekt angewandte Methode zur Repräsentation flächenhafter Objekte (z.B. Wald, Landwirtschaftliche Flächen) als aus Dreiecken zusammengesetzte Multisurfaces führt aufgrund der Hochaufgelösten Punktwolken zu sehr großen Datenmengen. Hier sollten alternative Repräsentationsformen erforscht werden.
- Um das 3D-DLM gemäß den Anforderungen der Anwender weiterzuentwickeln sollten nicht zuletzt Tests in konkreten Anwendungsszenarien durchgeführt werden.

Literaturverzeichnis

- BEIL, C.** (2017), Detaillierte Repräsentation des Straßenraums in 3D-Stadtmodellen. Masterarbeit, Lehrstuhl für Geoinformatik, Technische Universität München.
- BILJECKI, F., STOTER, J., LEDOUX, H., ZLATANOVA, S., & CÖLTEKIN, A.** (2015), Applications of 3D City Models: State of the Art Review. In: ISPRS International Journal of Geo-Information; Nr.4; S. 2843
- COMMANDEUR, T. & LEDOUX, H.** (2018): 3Dfier: <https://github.com/tudelft3d/3dfier>
- COORS, V., HUBER, U., JOOS, G., REINELT, A., TEEGE, G. & VOERKELIUS, U.** (2016): Stellungnahme des Beitrags zum Projektvorhaben Digitales Landschaftsmodell (3D-DLM). unveröffentlicht.
- DONAUBAUER, A., ESCH, R. & KOLBE, T. H.** (2016), Prozedurale Verfahren zur Generierung von 3D-Infrastrukturobjekten. In: Kolbe, T. H.; Bill, R.; Donaubaue, A. (Hrsg.): Geoinformationssysteme 2016. Wichmann.
- ESCH, R.** (2016), Prozedurale Generierung von 3D-Straßennetzwerken unter Berücksichtigung von Brücken und Tunneln am Beispiel Vorarlberg. Masterarbeit, Lehrstuhl für Geoinformatik, Technische Universität München.
- FIUTAK, G., MARX, C., WILLKOMM, P., DONAUBAUER, A. & KOLBE, T. H.** (2018), Automatisierte Generierung eines digitalen Landschaftsmodells in 3D. In: Kersten, T. P.; Gülch, E.; Schiewe, J.; Kolbe, T. H.; Stilla, U. (Hrsg.): 38. Wissenschaftlich-Technische Jahrestagung der DGPF und PFGK18 Tagung; Band 27
- GRÖGER, G., KOLBE, T. H., NAGEL, C., & HÄFELE, K.-H.** (2012), OpenGIS City Geography Markup Language (CityGML) Encoding Standard, Version 2.0.0. Open Geospatial Consortium.
- HEIPKE, C.** (2016), Herausforderungen bei der Erstellung von 3D-DLMs. In: Kolbe, T. H., Bill, R., Donaubaue, A. (Hrsg.): Geoinformationssysteme 2016. Wichmann.
- KOCH, A.** (2006), Semantische Integration von zweidimensionalen GIS-Daten und Digitalen Geländemodellen. Dissertation. Universität Hannover.
- KUNDE, F. & MARX C.** (2015), Abschlussbericht zum Projekt VoDLM3D. unveröffentlicht.
- MARX, C., DONAUBAUER, A., FIUTAK, G. & KOLBE, T. H.** (2017), Digitales Landschaftsmodell in 3D. In: Kolbe, T. H.; Bill, R.; Donaubaue, A. (Hrsg.): Geoinformationssysteme 2017. Wichmann.
- MARX, C., FIUTAK, G., WILLKOMM, P. & DONAUBAUER, A.** (2017), Projekt 3D Digitales Landschaftsmodell (3D-DLM) am Runden Tisch GIS e.V, Abschlussbericht zur 1. Phase.: https://rundertisch-gis.de/images/2_publicationen/stud_berichte/3D-DLM%20-%20Abschlussbericht.pdf
- OUDE ELBERINK, S., STOTER, J., LEDOUX, H. & COMMANDEUR, T.** (2013), Generation and Dissemination of a National Virtual 3D City and Landscape Model for the Netherlands. In: Photogrammetric Engineering & Remote Sensing, 79 (2), 147-158.

