Technische Universität München
Ingenieurfakultät Bau Geo Umwelt
Lehrstuhl Geoinformatik
Univ.-Prof. Dr. rer. nat. Thomas H. Kolbe

Erzeugung eines semantischen 3D-Stadtmodells der Stadt New York in CityGML auf der Basis von Open Data

DGM, Gebäude, Flurstücke, Gewässer, Parks und Bäume

Bearbeitet von: B. Sc. Berit Cantzler

Studiengang: Geodäsie und Geoinformation (Master)

Betreuer: Univ.-Prof. Dr. rer. nat. Thomas H. Kolbe

2015

Zusammenfassung

Anwendungsbereiche für 3D-Stadtmodelle werden immer umfangreicher. Sie werden z. B. als Entscheidungshilfe bei der Stadtplanung, für Schattenwurfsimulationen oder Solarpotentialanalysen eingesetzt. Dabei spielt nicht nur die Visualisierung eine Rolle, sondern v.a. der semantische Informationsgehalt. Das in dieser Arbeit generierte semantische 3D-Stadtmodell im LoD 1 beruht auf dem OGC-Standard CityGML, der ein offenes Datenmodell und XML-basiertes Format zur Repräsentation und zum Austausch virtueller 3D-Stadtmodelle darstellt. Für das Modell wird die Stadt New York herangezogen, da sie eine Vielfalt an amtlich erhobenen Open Data bereitstellt, wobei die folgenden 2D-Daten für diese Arbeit ausgewählt werden: Adressen, Bäume, Digitales Geländemodell (DGM), Flurstücke, Gebäude, Gewässer, hydrologische Strukturen, Parks, Straßen und Zonierung. Die größten Herausforderungen stellen die Transformation von 2D nach 3D, die Inhomogenität und der Umfang der Daten dar. Die vorliegende Arbeit beschäftigt sich mit dem DGM, das aufgrund seiner Größe jeweils in kleinen Kacheln generiert wird, die an ihren Grenzen nahtlos ineinander übergehen. Des Weiteren wird auf die Modellierung der oben erwähnten Datensätze (ohne Straßen und Zonierung) eingegangen. Besonderes Augenmerk wird hierbei auf die Verknüpfung verschiedener Datensätze, sowie die Darstellung der Objekte gelegt. Die Datensätze der Straßen und Zonierung werden in der ergänzenden Arbeit von B. Burger behandelt.

Abstract

The variety of applications of 3D city models has been growing steadily within recent years. They are used e.g. to support city planning decisions, shadowing simulations or solar potential analyses. For tasks like these, not only the aspect of visualization is of importance, but also the content of semantic information. The semantic 3D city model generated in this thesis (in LoD 1) is based on the OGC standard CityGML, which is an open data model and XML-based format for the representation and exchange of virtual 3D city models. New York City is chosen for this project since the city administration provides a variety of officially created open data. In the present work, the following 2D data sets are used: addresses, buildings, digital terrain model (DTM), lots, parks, streets, trees, water bodies (and structures) and zoning. The biggest challenges are the transformation from 2D to 3D, the data's inhomogeneity and their volume. This thesis sets out the generation of the DTM which, due to its enormous size, was processed in small tiles with special respect to seamless transitions at their borders. Furthermore, the process of modelling the data sets (without streets and zoning) mentioned above is being demonstrated. Thereby, great emphasis is put on the combination of different data sets, as well as the representation/illustration of objects. The other data sets mentioned above are being discussed in the complementary thesis of B. Burger.

Inhaltsverzeichnis

1 Einleitung **9**

2 Theoretischer Hintergrund **11**
2.1 3D-Stadtmodelle . . . 11
2.2 City Geography Markup Language (CityGML) . . . 15
2.3 Verwendete Hard- und Software: FME, Oracle Spatial, 3DCityDB, 3D-WebClient . . . 20

3 Open Data **23**
3.1 Open Data allgemein . . . 23
3.2 New York City Open Data Portal . . . 25
3.3 Open Data in Deutschland . . . 26
3.4 Vergleich NYC – Deutschland . . . 28

4 Überblick über die verwendeten Daten und Herausforderungen **33**
4.1 Transformationsprobleme . . . 33
4.2 Koordinatenreferenzsysteme . . . 34
4.3 Verwaltung der Datenmengen . . . 36
4.3.1 Überblick über die Daten und ihren Umfang . . . 36
4.3.2 Allgemeiner Ablauf der Datenbearbeitung . . . 40

5 Digitales Geländemodell **43**
5.1 Definition in CityGML . . . 43
5.2 Verwendete Daten . . . 45
5.3 Erzeugung des CityGML-DGMs . . . 46

6 Gebäudemodell inklusive Adressen und Flurstücke **51**
6.1 Definition in CityGML . . . 51
6.1.1 Gebäude und Adressen im Modul Building . . . 51
6.1.2 Flurstücke im Modul LandUse . . . 52
6.2 Verwendete Daten . . . 52
6.2.1 Gebäude . . . 52
6.2.2 Adressen . . . 54
6.2.3 Flurstücke . . . 54
6.3 Umsetzung . . . 55
6.3.1 Gebäude und Adressen . . . 55
6.3.2 Flurstücke . . . 59

7 Bäume **61**
7.1 Definition in CityGML . . . 61
7.2 Verwendete Daten . . . 62
7.3 Umsetzung . . . 63

8 Gewässer und hydrologische Infrastruktur **67**
8.1 Gewässer . . . 67
8.1.1 Gewässer im Modul WaterBody . . . 67

8.1.2 Datensatz der Gewässer . . . 68
8.1.3 Höhenübertragung auf Gewässer . . . 69
8.2 Hydrologische Infrastruktur . . . 73
8.2.1 Hydrologische Infrastruktur im Modul CityFurniture . . . 73
8.2.2 Datensatz der hydrologische Infrastruktur . . . 74
8.2.3 Umsetzung und Höhenübertragung der hydrologischen Infrastruktur . . . 74

9 Parks **75**
9.1 Definition in CityGML . . . 75
9.2 Verwendete Daten . . . 75
9.3 Umsetzung . . . 76

10 Visualisierung der Ergebnisse **79**

11 Fazit und Ausblick **87**

A Herkunft der verwendeten Datensätze im Überblick **93**

B Codelisten **95**
B.1 Übersicht über originalen Attribute der Gebäude [CityGML Notation] . . . 97
B.2 Attribute und Codelisten der Bäume [CityGML Notation] . . . 98
B.3 Übersicht über Attribute der Gewässer [CityGML Notation] . . . 103
B.4 Übersicht über Attribute der hydrologischen Infrastruktur [CityGML Notation] . . . 105
B.5 Codeliste der Parks [CityGML Notation] . . . 106

C FME-Workspaces zum Digitalen Geländemodell **107**

D Geometrie und Attribute der einzelnen Objektarten **109**

Abbildungsverzeichnis **111**

Tabellenverzeichnis **113**

Literaturverzeichnis **115**

Abkürzungsverzeichnis

ADE	Application Domain Extension
BIM	Building Information Modelling
CAD	Computer-Aided Design
CityGML	City Geography Markup Language
CRS	Coordinate Reference System
COLLADA	Collaborative Design Activity
DB	Datenbank
DCP	Department of City Planning
DEM	Digital Elevation Model
DFDTM	Department of Finance's Digital Tax Map
DGM	Digitales Geländemodell
DHM	Digitales Höhenmodell
DoITT	Department of Information Technology & Telecommunications
DOM	Digitales Oberflächenmodell
DPR	Department of Parks and Recreation
DSM	Digital Surface Model
DTM	Digital Terrain Model
GIS	Geoinformationssystem
GML	Geography Markup Language
GRS80	Geodätisches Referenzsystem 1980
IFC	Industry Foundation Classes
KML	Keyhole Markup Language
LiDAR	Light Detection And Ranging
LoD	Level of Detail
NAD83	North American Datum 1983
NYC / NY	New York City
OGC	Open Geospatial Consortium
OSM	OpenStreetMap
SRDBMS	Spatially-enhanced Relational Database Management System
TIN	Triangulated Irregular Network
UML	Unified Modeling Language
VRML	Virtual Reality Modeling Language
WGS84	World Geodetic System 1984
X3D	Extensible 3D
XML	Extensible Markup Language

1 Einleitung

Verfasser: B. Burger und B. Cantzler

Die Anzahl von 3D-Stadtmodellen wächst in den letzten Jahren stetig, da immer mehr Städte und Kommunen, aber auch Unternehmen sie erzeugen und/oder nutzen. Vor allem Großstädte haben flächendeckende Modelle. Der Trend geht von kleinräumigen, projektbezogenen hin zu großräumigen ganzheitlichen Stadtmodellen. Sie werden überwiegend für raumbezogene Analysen und Visualisierungszwecke verwendet und in den verschiedensten Themenbereichen eingesetzt, wie Stadt- und Raumplanung, Umwelt, Energie, Katastrophenmanagement, Tourismus, 3D-Kataster oder Computer Management. Die dafür benötigten Geodaten werden von öffentlichen Verwaltungen, (privaten) Unternehmen oder inzwischen auch von Privatpersonen (z. B. durch OpenStreetMap (OSM)) erfasst, aufbereitet und zur Verfügung gestellt.

In Deutschland existieren z. B. 3D-Stadtmodelle aus Katasterdaten bundesweit im *Level of Detail 1 (LoD 1*–3D-Gebäude als einfache Klötzchen mit Flachdach). *Google Earth* stellt dreidimensionale Stadtmodelle, die aus digitalen Oberflächenmodellen und Luftbildern generiert wurden, zur Verfügung und OpenStreetMap stellt 3D-Modelle in *OSM Buildings* dar. Diese Stadtmodelle weisen Schwächen auf. So sind die Stadtmodelle aus den Katasterdaten kostenpflichtig. Die Modelle von Google Earth enthalten weder einzelne Objekte noch Semantik. Außerdem verwenden Google Earth und OSM keine amtlich erzeugten Daten für ihre Stadtmodelle. Letzterem liegt zudem kein Höhenmodell zugrunde, wodurch alle Objekte auf einer Ebene liegen.
Das Stadtmodell dieser Arbeit unterscheidet sich von den oben genannten Modellen darin, dass es ein vollständiges und semantisches 3D-Stadtmodell auf Basis von amtlichen Daten repräsentiert. Die verwendeten Daten sind sogenannte Open Data, die von der Stadt New York im Internet kostenfrei der Öffentlichkeit zur Nutzung bereitgestellt werden. Des Weiteren besteht es nicht nur aus Gebäuden, wie die meisten bisherigen flächendeckenden Stadtmodelle (im CityGML-Format), sondern aus einer Vielzahl von Datensätzen. In dem Stadtmodell sind ein Digitales Geländemodell (DGM), Gebäude, Adressen, Flurstücke, Gewässer, Parks, Bäume, Straßen und Zonierung mit inbegriffen. Solch vielfältige 3D-Stadtmodelle sind bisher lediglich für kleine Testgebiete erzeugt worden. In dieser Arbeit liegt jedoch der Fokus darauf, ein Modell für die gesamte Stadt New York mit der eben erwähnten Vielfalt an Daten zu erstellen, auch wenn dabei geometrische Feinheiten bei der Generierung von Objekten vernachlässigt werden.

Der Schwerpunkt des OGC-Standards CityGML (City Geography Markup Language), der ein XML-basiertes Format zur Repräsentation und zum Austausch virtueller 3D-Stadt- und Landschaftsmodelle darstellt, liegt ohnehin mehr auf der Semantik als auf der Visualisierung. Dies unterscheidet CityGML sichtlich von Formaten wie X3D (Extensible 3D) oder KML (Keyhole Markup Language). Weiterhin definiert der Standard fünf Detaillierungsstufen, sogenannte *Levels of Detail (LoD)*, wobei in dieser Arbeit das Stadtmodell in LoD 1 generiert wird [Kolbe, 2009].

Die verwendeten Daten stammen von den unterschiedlichen Behörden der Stadt New York. Dies hat zur Folge, dass z. B. verschiedene Koordinatensysteme verwendet werden oder jeder Datensatz eine eigene Struktur der Attribute aufweist. Diese Inhomogenitäten werden in der Arbeit behoben, wodurch ein konsistentes Stadtmodell generiert wird. Die Datensätze liegen im Original zweidimensional vor. Die Höhenkomponente wird den Daten v. a. aus dem erzeugten digitalen Geländemodell sowie mit Hilfe von Attributwerten übertragen, wodurch der Übergang in die dritte Dimension erfolgt. Des Weiteren sind die großen Datenmengen eine Herausforderung. Da die Datensätze – z. B. das Geländemodell – nicht in einem Stück berechnet werden können, werden je nach Datensatz und Datentyp verschiedene Methoden angewendet, um diese in kleinen Portionen, so genannten Kacheln, zu generieren. Ziel der Arbeit ist es, ein vielfältiges dreidimensionales Stadtmodell für die gesamte Stadt New York im CityGML-Format zu erzeugen.

Die übergeordnete Arbeit, *ein semantisches 3D-Stadtmodell der Stadt New York auf Basis von Open Data*, besteht aus zwei Masterarbeiten. Dabei liegt der Schwerpunkt der vorliegenden Arbeit auf dem *DGM*, den *Gebäuden*, *Flurstücken*, *Parkanlagen*, *Bäumen* und *Gewässern*. Die ergänzende Arbeit von B. Burger legt den Fokus ebenfalls auf das *DGM* und weiterhin auf die *Straßen* und die *Zonierung*. Dadurch existieren in beiden Arbeiten Kapitel, die gemeinsam angefertigt oder vom jeweils anderen Verfasser erstellt wurden, die als solche gekennzeichnet sind.
Zunächst wird ein Überblick über 3D-Stadtmodelle, CityGML und die in dieser Arbeit verwendete Software FME, Oracle und die 3DCityDB mit dem Importer/Exporter gegeben (Kapitel 2). Darauf folgt in Kapitel 3 ein Einblick in die Thematik der sogenannten *Open Data* und ein Überblick über den Umgang mit den Daten in New York City und Deutschland. Anschließend werden die Herausforderungen der Arbeiten erläutert, wie die Übertragung von in zwei Dimensionen vorliegenden Daten in die dritte Dimension, die Transformation in ein einheitliches Koordinatensystem und v. a. die Verwaltung der anfallenden Datenmengen (Kapitel 4). In den darauf folgenden Kapiteln wird auf die verwendeten Datensätze eingegangen, wobei die Priorität bei der (semi-)automatischen Generierung der Objekte für die gesamte Stadt liegt. Beim DGM bereitet die Größe der Daten Schwierigkeiten und die damit aufwändige Berechnung mit Hilfe von Kacheln und die dazu benötigte Rechenzeit (Kapitel 5).
In den folgenden Kapiteln werden die verwendeten Datensätze und ihre Verarbeitung in das CityGML-Format dieser Arbeit genauer erläutert. In Kapitel 6 wird die Verarbeitung der Gebäude und Flurstücke ausgeführt. Die Gebäude liegen als zweidimensionale Polygone vor und werden mit Hilfe der Gebäudehöhe extrudiert. Zusätzlich werden die Objekte der Gebäude unter Verwendung eines Adress-Punkte-Datensatzs und ausgewählter Attribute der Flurstücke mit zusätzlichen Attributen angereichert. Die Darstellung der Bäume in Form von so genannten „Lollipops“ erfolgt auf zwei getrennte Arten, zum Einen als implizite Geometrie, zum Anderen in der expliziten Geometrie (Kapitel 7). Die Bearbeitung der Gewässer und ihrer hydrologischen Infrastrukturen erfolgt in Kapitel 8. In Kapitel 9 werden zwei voneinander unabhängige Parkflächendatensätze über eine einheitliche ID kombiniert. Hierbei wird besonderer Wert auf nicht-redundante Daten-Speicherung gelegt. Abschließend werden die Ergebnisse beider Arbeiten, d.h. die Datensätze der vorliegenden und der ergänzenden Arbeit von B. Burger, visualisiert (Kapitel 10) und ein Ausblick über weitere Möglichkeiten (Kapitel 11) gegeben.

2 Theoretischer Hintergrund

Verfasser: B. Burger

Im folgenden Kapitel wird ein allgemeiner Überblick über dreidimensionale Stadtmodelle gegeben. Der Schwerpunkt liegt dabei jedoch in den Fachbereichen Geodäsie und Geoinformation, der Bereich der Computergraphik beispielsweise wird hingegen nur marginal gestreift. Darauf folgt der theoretische Hintergrund zu CityGML, das der Repräsentation und dem Austausch virtueller 3D-Stadt- und Landschaftsmodelle dient. Abgeschlossen wird das Kapitel mit einer Übersicht über die in dieser Arbeit verwendete Hard- und Software wie FME, Oracle Spatial, 3DCityDB, 3D-WebClient.

2.1 3D-Stadtmodelle

Dreidimensionale Stadtmodelle sind keine neue Erfindung. Der Drechslermeister Jakob Sandtner fertigte beispielsweise im Jahr 1570 bereits ein vollständiges und maßstabsgetreues hölzernes Modell der Stadt München an. Dieses Altstadtmodell und weitere Modelle werden laufend aktualisiert, da sie sowohl bei Ausstellungen und zu Stadtmarketing-Zwecken als auch bei Planungsarbeiten und als Entscheidungshilfe zum Einsatz kommen. Bei Wettbewerben oder Präsentationen werden sie jedoch inzwischen von zweidimensionalen Planzeichnungen und 3D-Repräsentationen am Computer unterstützt, um so eine noch bessere räumliche Vorstellung zu erzielen.[1]

Durch die breite Verfügbarkeit digitaler Daten haben sich in den letzten Jahren immer mehr 3D-Stadtmodelle im digitalen Bereich etabliert. Meist werden dafür zweidimensionale digitale Karten (z. B. aus dem Kataster) als Grundlage verwendet, immer seltener traditionell auf Papier gefertigte Karten. Diese 2D-Karten beschreiben die darin vorkommenden Objekte wie Gebäude, Straßen oder Gewässer durch deren jeweilige Umrisse. Im Gegensatz dazu können in einem 3D-Stadtmodell (weitere) Informationen bzw. Objekte intuitiver wahrgenommen werden, z. B. Dachformen von Gebäuden oder Objekte im Straßenbereich, wie Straßenlaternen [van Essen, 2008].

Nach Blade und Padgett (2002) ist „ein Stadtmodell ein dreidimensionaler Datensatz, der die Umgebung bzw. das Umfeld anhand von Realwelt- oder abstrakten Objekten beschreibt“ [Jahnke, 2013]. Viele 3D-Stadtmodelle legen ihren Fokus ganz auf grafische oder geometrische Modelle, wodurch die Visualisierung oder fotorealistische Wiedergabe von sichtbaren urbanen Objekten angestrebt wird, um den physischen Objekten so nah wie möglich zu kommen. Dabei bleiben jedoch die semantischen (Bedeutung, Thematik) und topologischen (Beziehungen, Nachbarschaften) Aspekte völlig unberücksichtigt, die entscheidend für thematische Fragen oder Analyseaufgaben sind [Döllner et al., 2006].

[1] http://www.muenchen.de/rathaus/Stadtverwaltung/Referat-fuer-Stadtplanung-und-Bauordnung/Stadt-und-Bebauungsplanung/Stadtmodell.html (05.04.2015)

Anwendungen

Allgemein betrachtet ist wohl die Hauptaufgabe eines dreidimensionalen Stadtmodells die Visualisierung und räumliche Darstellung der einzelnen Objekte und dieser im Gesamten. Dafür genügen bereits die eben erwähnten rein grafisch oder geometrisch erstellten Modelle. Werden jedoch die semantischen und topologischen Gesichtspunkte mit berücksichtigt, wie verschiedene Detaillierungsgrade der Objekte (vgl. „Level of Detail (LoD)“ in Kapitel 2.2), sind die Anwendungsbereiche und -möglichkeiten sehr umfangreich, wodurch wiederum verschiedenste Nutzergruppen angesprochen werden. Im Folgenden einige Beispiele:

Deutschlandweit existieren bereits 3D-Gebäudemodelle, die auf den Gebäudegrundrisse des Liegenschaftskatasters basieren. Bei stadtplanerischen Überlegungen, z. B. zur Erstellung von Bebauungsplänen, erleichtern sie oft die Entscheidung. Das Katastrophenmanagement bezieht neben Routenplanung auch Überflutungs- oder Explosionssimulationen mit ein. Mit Sichtbarkeitsanalysen oder Schattenwurfsimulationen können Standortplanungen für Gebäude oder Windenergieanlagen vereinfacht werden. Mit einem 3D-Gebäudemodell und den nötigen zusätzlichen Informationen können Solarpotentialanalysen oder Wärmebedarfskarten abgeleitet werden. Ebenso können Lärm- und Luftschadstoffprognosen berechnet werden. Des Weiteren werden 3D-Stadtmodelle in der mobilen Telekommunikation, dem Facility Management oder der Navigation (sowohl für Fahrzeuge als auch Fußgänger) genutzt [Coors et al., 2013].

3D-Stadtmodelle könnten ebenfalls als Grundlage von *Smart Cities* in Zukunft eine immer größere Rolle spielen. Darauf und auf weitere Einsatzgebiete, beispielsweise in der Grafik- oder Spielebranche, wird hier jedoch nicht eingegangen.

Daten

Die für ein 3D-Stadtmodell benötigten Geodaten werden von öffentlichen Verwaltungen, privaten Unternehmen (z. B. Google, Microsoft) oder inzwischen auch von Privatpersonen (z. B. in OpenStreetMap (OSM)[2]) erfasst, aufbereitet und zur Verfügung gestellt. Diese Daten werden mit Hilfe von Erfassungsverfahren erstellt, wodurch (semi-)automatisiert eine flächenhafte Aufnahme von Objekten erfolgt. Dazu gehören beispielsweise photogrammetrische Luftbildaufnahmen oder LiDAR-Systeme (Light Detection And Ranging Systems). Konstruktionsverfahren hingegen basieren auf CAD (Computer-Aided Design)-Modellen, die einzelne Geometrien beinhalten, wodurch sie einen hohen Detaillierungsgrad aufweisen können. Des Weiteren werden Daten und Informationen beispielsweise aus dem Kataster, aus digitalen georeferenzierten, terrestrisch aufgenommenen Fotos oder aus terrestrisch erfassten Laser Scanning-Punktwolken eingesetzt [Coors et al., 2013].

Durch den Fortschritt der verfügbaren Technik, der wirtschaftlichen Datenerfassungsmethoden und der daraus resultierenden größeren Verfügbarkeit von digitalen georeferenzierten Daten ist die Generierung von dreidimensionalen Stadtmodellen effektiver und effizienter geworden [Jahnke et al., 2011].

[2] Das Ziel von OpenStreetMap ist es, eine freie Weltkarte zu erschaffen. Es werden weltweit Daten über Straßen, Eisenbahnen, Flüsse, Wälder, Häuser und alles andere, was gemeinhin auf Karten zu sehen ist, gesammelt: http://www.openstreetmap.de/ (06.04.2015)

Datenformate

Neben der oben aufgezeigten Fülle von Anwendungen für dreidimensionale Stadtmodelle existiert ebenso eine Vielzahl verschiedener Formate. Die in dieser Arbeit verwendete City Geography Markup Language (CityGML), ein Open Data Modell und XML-basiertes Format zur Repräsentation und zum Austausch von virtuellen 3D-Stadtmodellen, wird in Kapitel 2.2 ausgeführt. Daneben gibt es weitere 3D-Standards wie IFC (Industry Foundation Classes), X3D (Extensible 3D) oder KML (Keyhole Markup Language), die im Folgenden im Vergleich zu CityGML erläutert werden.

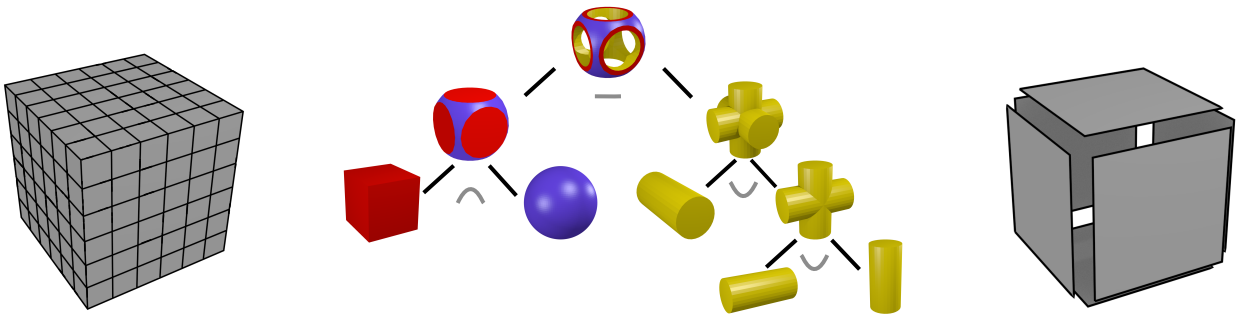
Bauwerksinformationsmodelle (BIM – Building Information Modelling) werden im Bauwesen durch das ISO-standardisierte Austauschformat IFC beschrieben. Dieser von *buildingSmart International* definierte Standard befasst sich im Wesentlichen mit Gebäuden, wobei er sämtliche Aspekte eines Gebäude-Lebenszyklus (Planung, Bau und Betrieb) berücksichtigt, wodurch die Repräsentation eines dreidimensionalen Bauwerks ein sehr detailliertes semantisches Modell darstellt. Vernachlässigt werden dabei jedoch Inhalte eines 3D-Stadtmodells, wie das Geländemodell, die Vegetation oder Gewässer; diese werden als generische Objekte modelliert. Im Gegensatz zu CityGML unterstützt das Geometriemodell der IFC sowohl B-Rep als auch CSG-Repräsentationen (vgl. im Folgenden: „Objektmodellierung“). Es existieren Überlegungen, die Datenmodelle des zukünftigen CityGML 3.0 und des IFC-Formats zu harmonisieren, um Transformationen zwischen beiden zu erleichtern. Es ist allerdings noch nicht geklärt, ob eine vollständige semantische Transformation von IFC nach CityGML 3.0 überhaupt notwendig ist, da IFC-Modelle bereits in verschiedenen LoDs und mit der Mehrheit ihrer semantischen Information nach CityGML konvertiert werden können. In Bezug auf die geometrische Repräsentation besteht die Schwierigkeit, ein CityGML-Modell nach IFC zu überführen, da in IFC Volumenkörper in den meisten Fällen mit Hilfe von CSG und nicht durch ihre Oberflächen repräsentiert werden [Kolbe, 2009],[Löwner et al., 2014].

Im Gegensatz zu X3D, VRML (Virtual Reality Modeling Language), COLLADA (Collaborative Design Activity) oder KML ist CityGML nicht auf Visualisierungszwecke beschränkt, da der Fokus in erster Linie auf dem Austausch der hinterlegten Informationen liegt.

X3D, der Nachfolger von VRML, ist ein ISO-Standard, mit dessen Hilfe 3D-Szenen und -Objekte in XML repräsentiert und übermittelt werden können.[3] Dieses Format unterstützt sowohl georeferenzierte Koordinaten als auch den Austausch semantischer Information. Allerdings standardisiert X3D nicht die Repräsentation semantischer Objekte und ihrer Beziehungen untereinander [Kolbe, 2009].

Der OGC-Standard KML wurde hauptsächlich zur Visualisierung und Annotation graphischer Daten für Internetkarten und digitale Globen wie *Google Earth* entwickelt. Er basiert ebenfalls auf XML, kann allerdings nur geographische Koordinaten des WGS84 (World Geodetic System 1984)-Ellipsoids verwenden. CityGML hingegen kann jedem Geometrieobjekt ein eigenes (3D-)Koordinatensystem zuweisen. Mit KML können ebenfalls semantische Informationen ausgetauscht werden, jedoch ist dies nicht dessen Zweck. Die Stärke liegt eher darin, dass COLLADA-Dateien eingebunden werden können, wodurch beispielsweise fotorealistische Gebäudemodelle dargestellt werden können und der Zugang zu weiteren 3D-Softwaretools (z. B. AutoCAD, Esri City Engine) gegeben ist. Die

[3] http://www.web3d.org/x3d/what-x3d (06.04.2015)



(a) Zellzerlegung (b) CSG (c) B-Rep

Abbildung 2.1: Beispielhafte Skizzen der drei Verfahren zur Generierung von Volumenkörpern: (a) Zellzerlegung, (b) ein CSG-Baum mit den jeweiligen Operationen, (c) B-Rep.

3DCityDB unterstützt bereits die Konvertierung von CityGML nach KML bzw. COLLADA (vgl. „3DCityDB (Importer/Exporter)“ in Kapitel 2.3) [Kolbe, 2009],[Löwner et al., 2013].

Objektmodellierung

Wenn von dreidimensionalen Stadtmodellen die Rede ist, handelt es sich in den meisten Fällen um die Modellierung von Gebäudeobjekten, da diese im Wesentlichen ein solches Modell prägen. Dabei werden die weiteren Bestandteile, wie ein zu Grunde liegendes Geländemodell, Vegetation, Gewässer oder die Infrastruktur eines Stadtmodells vernachlässigt. Die räumlichen Eigenschaften all dieser Objekte werden z. B. in CityGML, durch die geometrischen Primitive Punkt, Kurve, (Ober-)Fläche und Körper repräsentiert [Andrae, 2009].

Um nun auf die Gebäude, die, wie eben erwähnt, am häufigsten modelliert werden, einzugehen, werden im Folgenden drei Verfahren zur Generierung von Volumenkörpern beschrieben (vgl. Abbildung 2.1). Die speicher- und rechenintensive *Zellzerlegung* beruht auf dem zweidimensionalen Prinzip, bei dem Bilder aus Pixeln zusammengesetzt werden. Im dreidimensionalen Raum werden dementsprechend Körper aus würfel- oder quaderförmigen Voxeln erzeugt, deren Stärke bei Mengenoperationen und Volumenberechnungen liegt. Bei der ebenfalls sehr rechenintensiven Methode der *Konstruktiven Festkörpergeometrie (CSG – Constructive Solid Geometry)* werden komplexere Körper aus der Vereinigung, Schnittmenge und Differenz verschiedener geometrischer Grundformen wie Quader, Kugel oder Zylinder gewonnen. Die Modellierung mit der *Randflächendarstellung (B-Rep – Boundary Representation)* beschreibt einen Körper über die Menge seiner Oberflächen, wodurch Volumenberechnungen aufwändig sind, da die Punktmenge innerhalb der Grenzflächen zunächst bestimmt werden muss. Im Gegensatz zu den beiden anderen Methoden sind jedoch verschiedene Ansichten der Objekte schnell generiert [Andrae, 2009].

Die Randflächendarstellung wird z. B. vom *Spatial Schema* von ISO 19107 und somit auch von CityGML verwendet. Das bedeutet beispielsweise, dass jeder Wandfläche eines Gebäudemodells eine eigene Textur zugewiesen werden kann. Das Datenaustauschformat IFC unterstützt hingegen sowohl die B-Rep als auch das CSG-Verfahren.

Die Leistungsfähigkeit von Computern erlaubt heutzutage dreidimensionale Geoinformationssysteme (GIS) und immer detailliertere Ansichten der Umwelt. Schwierigkeiten bei der Generierung von

3D-Stadtmodellen bereitet immer noch die (Semi-)Automation der Abläufe, um nicht nur kleine Ausschnitte einer Stadt, sondern auch großflächige Gebiete modellieren zu können. Des Weiteren kann sich die dafür benötigte Rechenzeit wiederum auf die Aktualität eines Modells auswirken. Abgesehen von der Verfügbarkeit der Daten müssen diese in das Modell integriert werden, weswegen die Aktualisierung eines 3D-Stadtmodells regelmäßig durchgeführt werden sollte. Andernfalls veraltet es und verliert seinen Wert als Informationsträger [Döllner et al., 2006].

2.2 City Geography Markup Language (CityGML)

Der OGC-Standard City Geography Markup Language (CityGML) ist ein offenes Datenmodell und XML-basiertes Format zur Repräsentation und zum Austausch virtueller 3D-Stadt- und Landschaftsmodelle. CityGML wurde entwickelt von der SIG 3D (Special Interest Group 3D – eine Initiative der Geodaten Infrastruktur Nordrhein-Westfalen, Deutschland), die sich aus Mitgliedern aus den Bereichen Industrie, Hochschule und öffentliche Verwaltung zusammensetzt. Das Ziel war es, die Definition und das Verständnis grundlegender Objekte, Attribute und ihrer Beziehungen innerhalb eines 3D-Stadtmodells zu schaffen [Kolbe, 2009].

Bisherige 3D-Stadtmodelle dienen nur der Visualisierung oder der grafischen Darstellung. Allerdings sind für viele Anwendungsbereiche (Beispiele siehe Kapitel 2.1) zusätzliche Informationen über die zu repräsentierenden Objekte in standardisierter Form notwendig. Dies ist mit einem semantischen Modell realisierbar, das u. a. thematische Klassen, Attribute und ihre Beziehungen untereinander umfasst, d.h. Objekte werden aufgrund logischer (und nicht grafischer) Merkmale untergliedert. Des Weiteren zahlt sich die semantische Modellierung aus, wenn qualitativ hochwertige Daten einbezogen werden und somit ein großer Nutzen für eine Vielzahl verschiedener Anwendungen daraus hervorgeht. CityGML dient damit als Bindeglied zwischen unterschiedlichen Anwendungen mit verschiedenartigen Informationen, wodurch Daten wiederverwertbar werden [Kolbe, 2009].

Semantik / Core Modul

Das Datenmodell von CityGML basiert auf der ISO 19100-Familie und ist ein Anwendungsschema der Geography Markup Language (GML 3.1.1 – ebenfalls ein OGC-Standard). Die Repräsentation geographischer Objekte wird darin mit Hilfe der abstrakten Klassen *_Feature* und *_FeatureCollection* definiert. Mit Hilfe von UML-Diagrammen (vgl. folgende Abbildungen) können Objektarten, (räumliche oder nicht-räumliche) Attribute und Datentypen auf standardisierte Weise eindeutig definiert und präsentiert werden.

Das in Abbildung 2.2 dargestellte *Core Modul* von CityGML beinhaltet die grundlegenden Komponenten des CityGML-Datenmodells und besteht sowohl aus abstrakten als auch nicht-abstrakten Klassen. Die abstrakte Klasse *_CityObject* erbt von der bereits oben erwähnten Superklasse *_Feature* die Attribute *gml:id* (verpflichtend für jedes Objekt), *name* und *description*, die sie wiederum an die thematischen Klassen weiter vererbt (vgl. auch Abbildung 2.3).

Thematische Klassen repräsentieren die wichtigsten Objektarten eines 3D-Stadtmodells, wobei jedes Themengebiet ein eigenes CityGML-Modul darstellt: *Appearance, Bridge, Building*, CityFurniture*, CityObjectGroup*, Generics*, LandUse*, Relief*, Transportation*, Tunnel, Vegetation** und *WaterBody**. Die in dieser Arbeit verwendeten Module (mit *) werden in den jeweiligen Kapiteln näher

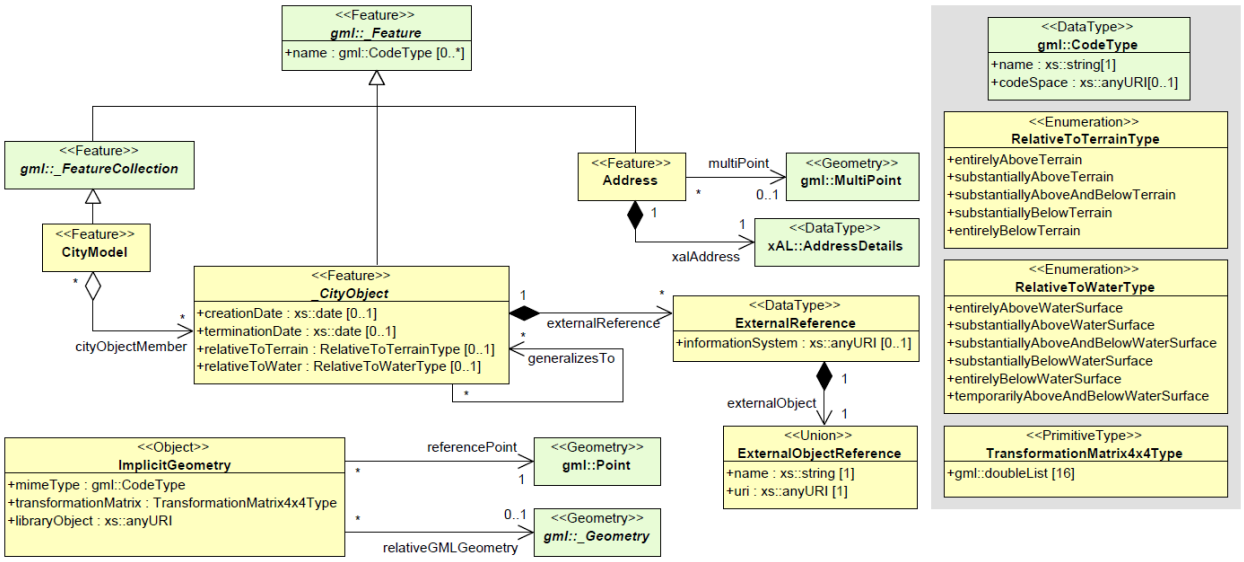


Abbildung 2.2: Das CityGML-UML-Diagramm des *Core Moduls* [Gröger et al., 2012].

erläutert.
Die nicht-abstrakten Klassen im *Core Modul* beschreiben z. B. thematische Klassen, die von mehr als einem Erweiterungsmodul verwendet werden können. Dies wären beispielsweise *External-References*, mit deren Hilfe jedes *_CityObject* mit Objekten aus anderen Datensätzen bzw. externen Datenbanken verknüpft werden kann.

Im Allgemeinen enthält jedes Objekt einer thematischen Klasse die Attribute *class, function* und *usage*. Dabei darf ersteres allerdings nur einmal auftreten, im Gegensatz zu den anderen beiden, die mehrfach vorkommen dürfen. Für diese Attribute können mögliche Werte in sogenannten Code-Listen vordefiniert werden. Beispiele dafür sind in Gröger et al. (2012) zu finden; jedoch können sie auch erweitert oder gar benutzerdefiniert neu erstellt werden. Wie für die in dieser Arbeit verwendeten Attribute vorgegangen wurde, wird ebenfalls in den Kapiteln zu den einzelnen Datensätzen ausgeführt [Gröger et al., 2012],[Kolbe, 2009].

Geometrie

Das Geometriemodell von CityGML bezieht sich, entsprechend dem semantischen Modell, auf das von GML3 und auf den ISO-Standard 19107, wobei Objekte durch ihre ID und weitere (geometrische) Unterstrukturen beschrieben werden. Diese können dabei auch mehrere räumliche Eigenschaften, z. B. für die verschiedenen LoDs, erhalten [Kolbe, 2009].

Das Modell besteht aus den Primitiven Punkt (0D), Kurve (1D), (Ober-)Fläche (2D) und Körper (3D), die nicht mehr in weitere Primitive zerlegt werden können, wodurch sich die Geometrie eines Objektes immer aus $1 - n$ geometrischen Primitiven zusammensetzt. Sie können allerdings zu komplexen bzw. Verbund (composite)-Geometrien oder zu Aggregaten zusammengesetzt werden (vgl. Abbildung 2.4).

Ein Aggregat repräsentiert dabei eine lose und ungeordnete Menge an geometrischen Objekten bzw. Primitiven, die sich u. a. überlappen dürfen und unabhängig von ihrer Zugehörigkeit zum Aggregat existieren. Objekte, die einem Komplex angehören, dürfen sich zwar berühren, aber nicht

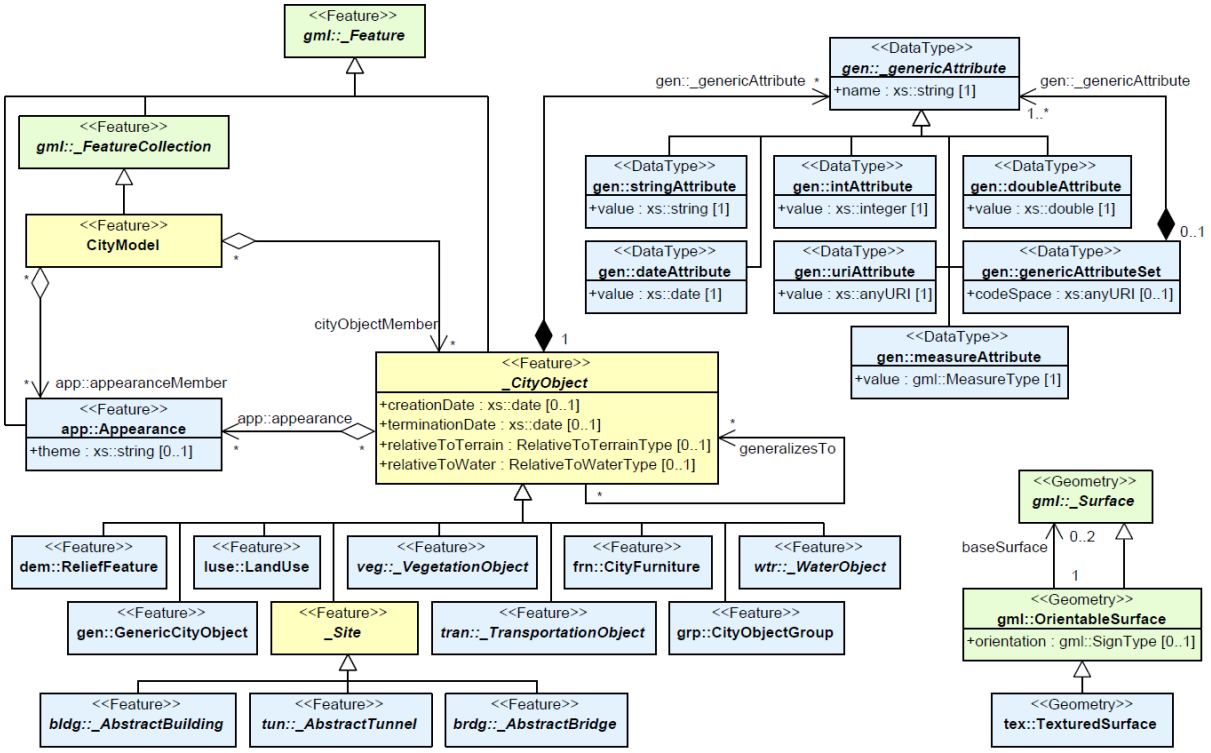


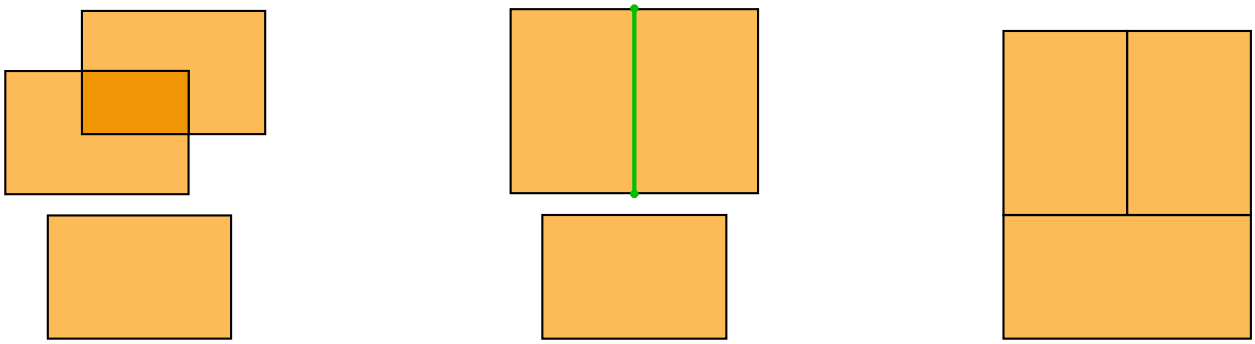
Abbildung 2.3: Das CityGML-UML-Diagramm der Top Level Klassenhierarchie [Gröger et al., 2012].

überschneiden und sollten simpel sein (d.h. sie dürfen sich nicht selbst berühren oder überschneiden). Des Weiteren gehören die Randmengen der betroffenen Primitive und wiederum deren Randmengen ebenfalls zu dem Komplex. Composite-Objekte stellen schließlich eine spezielle Form der komplexen Geometrie dar, da sie sich nur aus gleichartigen Geometrien zusammensetzen dürfen und geometrisch so miteinander verbunden sein müssen, dass der Komplex sich wie ein Primitiv verhalten kann [Andrae, 2009].

Wie bereits in Kapitel 2.1 erwähnt, beruht die Modellierung für Volumenkörper in CityGML auf der Randflächendarstellung (B-Rep), bei der allen Koordinaten ein Referenzsystem (CRS) zugewiesen wird. Im Gegensatz zu absoluten Koordinaten ist es auch möglich, *implizite Geometrien* zu generieren, die auf Geometrieformen in einem lokalen Koordinatensystem verweisen (um wieder verwendbar zu sein) und Transformationsmatrizen und Ankerpunkte im CRS bereitstellen. Dadurch können z. B. Bäume, die die gleiche Geometrie besitzen, mehrfach im Stadtmodell repräsentiert werden; die Geometrie für den jeweiligen Baum muss hingegen nur einmalig in einem lokalen Koordinatensystem erstellt und an den jeweiligen Ankerpunkt in das Welt-CRS transformiert werden [Gröger et al., 2012],[Kolbe, 2009].

Zudem ist die Richtigkeit der Topologie von Objektgeometrien für die Modellierung eines semantischen 3D-Stadtmodells wichtig, da beispielsweise das Volumen eines Gebäudeobjekts nicht berechnet werden kann, wenn dessen umfassende Flächen es nicht vollkommen umschließen.

Das Topologiemodell von CityGML beruht ebenfalls auf ISO 19107 und GML3, welches n-dimensionale topologische Primitive (Knoten, Kanten, Maschen, Körper) in n-1 dimensionale Primitive zerlegt, die wiederum solange zerlegt werden bis sie sich auf der Ebene der Knoten befinden. Dabei wird jedes Primitiv zu einem Objekt mit einer eigenen Identität [Kolbe, 2009].



(a) Aggregat (b) Complex (c) Composite

Abbildung 2.4: Beispielhafte Darstellung eines Aggregats (a), einer komplexen (b) und einer Verbundgeometrie (c) von Flächen.

Level of Detail (LoD)

CityGML unterstützt fünf Detaillierungsstufen (LoD0–LoD4), die aufeinander aufbauen und sowohl die Geometrie als auch die thematische Unterscheidung betreffen. Um die Visualisierung und Analyse in Bezug auf unterschiedliche Auflösungen zu ermöglichen, kann jedes einzelne Objekt gleichzeitig in verschiedenen LoDs repräsentiert werden. Mit Hilfe dieser Abstufungen werden einerseits Datensätze miteinander vergleichbar und andererseits geben sie einen Überblick über die Komplexität und Genauigkeit der Daten. Die Beschreibung der einzelnen Levels of Detail erfolgt am Beispiel der Gebäude (vgl. Abbildung 2.5).

Beim LoD0, der gröbsten Detaillierungsstufe, handelt es sich um ein 2,5-dimensionales digitales Geländemodell (DGM), auf das ein Luftbild projiziert werden kann. Die Gebäude können dabei z. B. mit ihren Grundrissen darauf abgebildet werden. Das sogenannte *Klötzchenmodell* im LoD1 stellt die Gebäude als einfache Blöcke ohne Dachstrukturen, d.h. mit Flachdach, dar. Die Dachstrukturen und größere Anbauten wie Balkone oder Treppen werden schließlich im LoD2 repräsentiert. LoD3 beschreibt ein Architekturmodell sowohl mit detaillierten Wand- und Dachstrukturen als auch mit Fenstern und Türen. Die genaueste Detaillierungsstufe, das LoD4, erweitert das LoD3 um die inneren Strukturen eines Gebäudes, z. B. Räume oder Mobiliar. Die Texturierung, die dem äußeren Erscheinungsbild entspricht, kann dabei in alle Levels of Detail integriert werden [Gröger et al., 2012],[Kolbe, 2009].

Wie sich die verschiedenen LoDs auf weitere Objekte wie Straßen oder Gewässer auswirken, wird in den jeweiligen Kapiteln näher erläutert.

Äußeres Erscheinungsbild (Appearance)

Das äußere Erscheinungsbild von Objektoberflächen ist wie die Semantik oder die räumlichen Eigenschaften fester Bestandteil eines dreidimensionalen Stadtmodells. Es kann nicht nur die reale Erscheinung repräsentiert werden, sondern auch Thematiken wie Lärm- oder Sonnenlichtimmissionen, um sowohl Visualisierungs- als auch Analyseaufgaben zu bewerkstelligen. Dabei können jeder (Ober-)Fläche mehrere Texturen zugewiesen werden. Diese liegen als Bilder oder durch Materialdefinitionen, die aus Formaten wie X3D oder COLLADA abgeleitet werden, vor [Gröger et

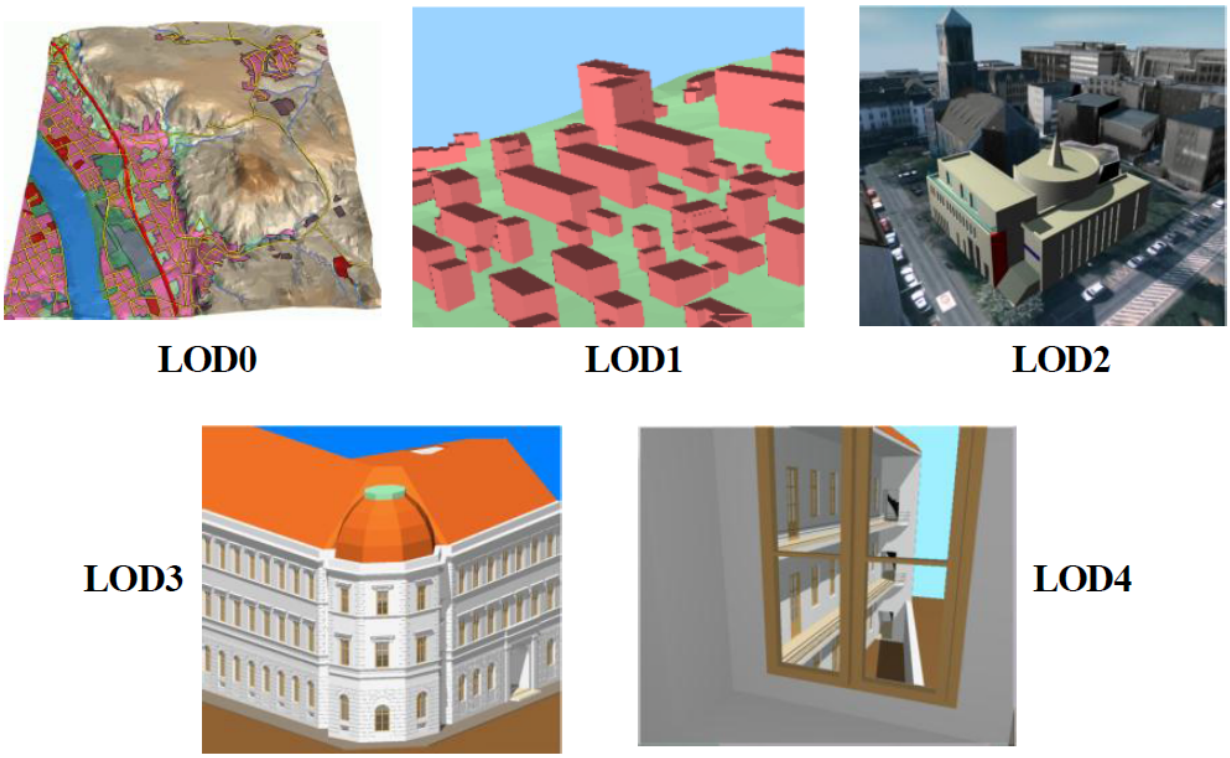


Abbildung 2.5: Die von CityGML definierten Detaillierungsstufen (LoD0–LoD4) [Kolbe, 2009].

al., 2012],[Kolbe, 2009].
Da in dieser Arbeit die *Appearance* vernachlässigt wird, wird nicht weiter darauf eingegangen.

Erweiterungsmöglichkeiten

CityGML ist für eine Vielzahl von Anwendungen ausgelegt. In der Praxis ist oft das Speichern oder der Austausch von zusätzlichen Attributen bzw. ganzen Objekten nötig, die nicht den vordefinierten entsprechen. Für eine solche Erweiterung gibt es zwei Möglichkeiten:

Die erste Option ist das Verwenden von generischen Objekten (*GenericCityObjects*) oder Attributen (*_genericAttributes*), die durch das Modul *generics* definiert sind. Ein *_CityObject* kann dementsprechend beliebig viele *_genericAttributes* enthalten, für die allerdings jeweils der Name, der Typ (vgl. Abbildung 2.3) und der Wert im CityGML-Datensatz gegeben sein muss. Demzufolge kann ein *GenericCityObject*, das von *_CityObject* abgeleitet wird, aus zahlreichen (impliziten) Geometrien bestehen.

Die zweite Option sind sogenannte *Application Domain Extensions (ADE)*, welche im Gegensatz zu den generischen Objekten bzw. Attributen in einer gesonderten XML-Schema-Definitionsdatei mit einem eigenen Namensraum definiert werden, wodurch die Erweiterung formal festgelegt wird. Demnach kann ein Instanzdokument eines erweiterten CityGML-Modells gegenüber eines CityGML bzw. des jeweiligen ADE-Schemas validiert werden. Ein Beispiel wäre die *CityGML Noise ADE*, wobei darüber hinaus mehrere ADEs in einem Modell verwendet werden können [Gröger et al., 2012],[Kolbe, 2009].

2.3 Verwendete Hard- und Software: FME, Oracle Spatial, 3DCityDB, 3D-WebClient

Die Berechnungen in dieser Arbeit werden auf Computern mit unten aufgelisteter (oder ähnlicher) Ausstattung durchgeführt. Die Datenbank läuft ebenfalls auf einem solchen mit zusätzlich 2×4 TB Festplatten. Inwieweit und für welche Arbeitsschritte die weiteren, unten erwähnten Werkzeuge genutzt werden, wird u. a. in Kapitel 4.3 näher erläutert.

- 2x CPU Xeon E5-2609, Quadcore, 2.40 GHz
- 32 GB Arbeitsspeicher (RAM)
- SSD-Festplatte 512 GB
- Windows 7 Enterprise

FME (Safe Software, Inc.)

Die von der Stadt New York zur Verfügung gestellten Daten werden mit Hilfe der *FME Desktop Oracle Edition 2014 SP4 (WIN64)* aufbereitet, verarbeitet und in das CityGML-Format überführt. Mit der 64-bit-Version können Daten aus dem Cache-Speicher auf der Festplatte zwischengelagert und große Datenmengen effizient bearbeitet werden.

Mit Hilfe der FME (*Feature Manipulation Engine*) von *Safe Software, Inc.* können Interoperabilitätsschwierigkeiten von Daten behoben werden. Sie besteht aus *FME Desktop, FME Server* und *FME Cloud*, die folgende Einsatzmöglichkeiten bieten: Datenkonvertierung, Datentransformation, Datenintegration, Datenvalidierung, Datenmigration, Echtzeitanwendungen, Automatisierung, Koordinatensystemtransformationen, usw.[4]

Oracle Spatial

Eine Oracle-Datenbank bildet die Grundlage zur Speicherung und zum Management der hier verwendeten Daten. Dabei wird die 64-bit Version von *Oracle Spatial*, einer Zusatzoption der *Oracle Database 12c, Enterprise Edition* verwendet.

Um raumbezogene Daten in Anwendungen zu integrieren oder Kartenfunktionen zu nutzen, genügt der *Oracle Locator*, der bereits in jeder Edition von *Oracle Database* enthalten ist. *Oracle Spatial* steuert dabei sowohl uneingeschränkte 3D-Funktionen als auch Web-Services bei und erlaubt beispielsweise die Verwaltung von Vektor- und Rasterdaten oder topologischen und Netzwerkmodellen.[5]

3DCityDB (Importer/Exporter)

Die *3DCityDB 3.0*[6], entwickelt am *Lehrstuhl für Geoinformatik der TU München* und von den Unternehmen *virtualcitySYSTEMS* und *M.O.S.S.*, stellt ein Open Source-Paket dar, das ein Datenbankschema und Werkzeuge zum Importieren, Verwalten, Analysieren, Visualisieren und Exportieren von 3D-Stadtmodellen (basierend auf dem CityGML-Standard) beinhaltet.

[4] http://www.safe.com/ (13.04.2015)
[5] http://www.oracle.com/de/products/database/options/spatial/index.html (13.04.2015)
[6] http://www.3dcitydb.net/3dcitydb (13.04.2015)

Dabei ergibt sich das Datenbankschema durch die Abbildung des objektorientierten Datenmodells von CityGML 2.0 auf das räumlich erweiterte relationale Datenbankmanagementsystem (SRDBMS), wobei sowohl das kommerzielle SRDBMS von Oracle als auch das Open Source SRDBMS von PostGIS unterstützt wird. Geometrieobjekte werden auf die räumlichen Datentypen, wie sie vom jeweiligen SRDBMS zur Verfügung gestellt werden, abgebildet. Geometrien, die auf Flächen basieren (z. B. Polygon, MultiSurface, Körper) werden in primitive Flächen zerlegt und als individuelle Tupel in einer großen Flächentabelle gespeichert. Mit Hilfe der Prozessierungsfähigkeiten des SRDBMS für räumliche Daten können auch sehr große Modelle verwaltet werden. Ein Beispiel dafür ist die Speicherung und das Management der deutschlandweiten 3D-Stadtmodelle im LoD1 und LoD2 [Kolbe et al., 2015].

Der Datenzugriff auf die 3DCityDB kann auf zwei verschiedene Weisen erfolgen:

Eine Möglichkeit ist der direkte Zugriff auf die Datenbanktabellen (für eine ausführliche Beschreibung der relationalen Strukturen siehe Kolbe et al. (2015)) mit Hilfe von Datenbankschnittstellen für Programmiersprachen wie C++, Java oder Python oder mit Werkzeugen wie der FME.

Die zweite Möglichkeit stellt die performante *Import/Export*-Software dar, auf die im Folgenden eingegangen wird. Mittels dieser können CityGML-Datensätze validiert, importiert und exportiert werden, wobei währenddessen Objekte nach räumlichen Regionen (Bounding Box) oder nach ihrer ID, Objektart oder Namen gefiltert werden können. Neben der Ausgabe im GML-Format können Stadtmodelle direkt von der 3DCityDB in KML- oder COLLADA-Dateien exportiert werden, die sowohl von GIS als auch von virtuellen Globen wie *Google Earth* oder *Cesium WebGL Virtual Globe* gelesen werden können. Des Weiteren können sogenannte *Ballons*, die Informationen über die einzelnen Objekte enthalten, erstellt werden. In Hinsicht auf große Datensätze ist zudem eine Kachelung implementiert. Um zusätzlich zur Geometrie (in KML-/ COLLADA-Dateien) die Attribute eines Objekts zu exportieren, gibt es den *Spreadsheet Generator (SPSHG)* als Plugin, der die thematischen Daten in CSV-Dateien, Excel-Tabellen oder direkt in ein *Google Spreadsheet* in der *Google Docs Cloud* exportiert [Kolbe et al., 2015].

3D-Webclient

Der vom *Lehrstuhl für Geoinformatik, TU München* entwickelte *cloud-basierte 3D-Webclient* dient der kollaborativen Nutzung, Bearbeitung und Analyse semantischer 3D-Stadtmodelle. Er stellt eine Schnittstelle dar, um die Komplexität der Geoinformationsverarbeitung und die flach strukturierte Text- bzw. Tabellenverarbeitung zu verbinden, wodurch Nutzern eine einfachere Entscheidungsunterstützung gegeben wird. Mehrere Nutzer haben gleichzeitig die Möglichkeit, lesend und schreibend auf die Daten eines Stadtmodells zuzugreifen und z. B. Analysen aufgrund der tabellarischen Sachdaten oder geometrischen Daten durchzuführen.

Die Grundlage des 3D-Webclient ist ein 3D-Stadtmodell, gespeichert in der 3DCityDB. Wie oben bereits beschrieben, werden die Geometriedaten als KML-/COLLADA-Datei und die thematischen Sachinformationen über den 3DCityDB-Spreadsheet-Generator exportiert. Ersteres wird nun auf einen Cloud-Dienst, z. B. *Dropbox*, hochgeladen, wodurch die Dateien über eine HTTP-Schnittstelle im Web veröffentlicht und in einem 3D-Viewer wie Google Earth oder Cesium visualisiert werden können. Die Spreadsheets werden, wenn nicht direkt dorthin exportiert, auf einen Cloud-Dienst,

z. B. Google Docs, hochgeladen. Werden die Sachdaten im Webclient bearbeitet, können sie auf demselben Weg wieder in die 3DCityDB zurückgeschrieben werden. Die geometrischen und Sachdaten werden miteinander über einen eindeutigen Schlüssel (ihrer *gml:id*) verknüpft und im 3D-Webclient dargestellt [Yao und Kolbe, 2014].

3 Open Data

Der Rohstoff des 21. Jahrhunderts sind Informationen. Beispielsweise erheben Verwaltungen große Mengen an Daten. Dazu zählen u. a. statistische Daten, Umweltdaten und Geodaten. Diese „Datenschätze“ enthalten ein enormes Potential und könnten von Bürgern, Unternehmen, Wissenschaft und anderen Verwaltungen genutzt werden. Die Voraussetzung dafür ist ein offener und strukturierter Zugang zu den Daten. Im diesem Kapitel erfolgt eine Auseinandersetzung mit offenen Daten, wie sie definiert sind und wo ihre Grenzen liegen. Darauf folgt ein Überblick über das Open Data Portal von New York und Deutschland. Abschließend wird ein Vergleich der verfügbaren Datenbestände zwischen New York und Hamburg angestellt.[1]

3.1 Open Data allgemein

Laut der *Open Definition*[2] bedeutet „offen“ im Bezug auf Daten, dass jeder Mensch frei auf sie zugreifen kann, sie für einen beliebigen Zweck nutzen, verändern und weitergeben darf. Also sind Daten offen, wenn sie keinen technischen, rechtlichen oder sonstigen Kontrollen unterliegen und eine uneingeschränkte Nutzung möglich ist [Dietrich, 2010].

Bei **offenen Daten** (Open Data) handelt es sich um Daten und Rohdaten von Regierungsorganisationen, von Verwaltungen, aus der Politik, der Wissenschaft oder der Wirtschaft, wenn diese aus öffentlichen Mitteln finanziert wurden. Diese sind im Interesse der Allgemeinheit der Gesellschaft ohne Einschränkungen frei zugänglich [Dietrich, 2010], [Groß et al., 2014], [von Lucke & Geiger, 2010]. Unter dem Begriff *Open Government Data* sind lediglich Daten der öffentlichen Verwaltung gemeint [Groß et al., 2014]. Das Spektrum der Open Data ist weitreichend und umfasst unter anderem Geodaten, Kartenmaterial, Material aus Archiven, mathematische oder medizinische Informationen bzw. Statistiken [Groß et al., 2014]. Ausschlaggebend für die öffentliche Bereitstellung der Daten ist dabei, dass keine personenbezogenen Daten, Firmen- oder Betriebsgeheimnisse veröffentlicht oder Sicherheitsbeschränkungen missachtet werden und somit der Datenschutz eingehalten wird [Dietrich, 2010], [Dobusch et al., 2012]. Zusätzlich ist es nötig, die Daten in maschinenlesbaren Formaten zu veröffentlichen, damit sie von Nutzern weiterverarbeitet werden können [Dietrich, 2010]. Sinnvoll für die Werterhaltung der Daten ist es, sie so schnell wie möglich der Öffentlichkeit zur Verfügung zu stellen und sie fortlaufend zu aktualisieren.[3]

[1] https://www.govdata.de/ (03.06.2015)
[2] http://opendefinition.org/od/ (02.12.2014)
[3] http://opendata-network.org/2010/03/8-open-government-data-prinzipien-zeitnah/ (28.05.2015)

Zusammengefasst sind Daten offen, wenn sie:

- für jeden,
- für jeden (nicht-)kommerziellen Zweck,
- in vollem Umfang,
- in maschinenlesbaren Formaten
- über offene Schnittstellen
- zeitnah
- und kostenlos

verfügbar sind [Abel et al., 2014],[Dietrich, 2010].

Weiterhin sind *Linked Open Data* offene Daten mit der zusätzlichen Eigenschaft, dass sie über eine eindeutige URI aufzurufen und verknüpfbar sind auch kann über eine URI auf andere Daten verwiesen werden.[4]

Gegen eine Veröffentlichung der Daten spricht, dass der Datenschutz unter Umständen nicht ausreichend vorhanden ist und über den Vergleich verschiedener Datensätze z. B. Rückschlüsse Personen gezogen werden können. Beispielsweise könnten Daten missbraucht werden, wenn Statistiken von Verbrechen mit Daten über das Bildungsniveau der Bewohner kombiniert und auf andere Gebiete übertragen werden und so Rückschlüsse auf sichere und nicht sichere Wohngegenden gemacht werden. Auch sollte bedacht werden, dass bestehende Geschäftsmodelle durch die freie Verfügbarkeit der Daten gefährdet sein können. Bisher wurden teilweise hohe Gebühren von Verwaltungen und Wirtschaft für die Erhebung und Bereitstellung von z. B. Geodaten oder Statistiken erhoben. Werden diese Daten unentgeltlich bereitgestellt, brechen diese Einnahmen weg [Dietrich, 2010], [von Lucke & Geiger, 2010].

Für die öffentliche Bereitstellung der Daten spricht, dass Bürger oder Dritte sich auf ihrer Grundlage dieser informieren und Entscheidungen treffen können. Zusätzlich kann das Handeln der Regierung und Verwaltung transparenter und nachvollziehbarer gestaltet und eine Beteiligung an Entscheidungen ermöglicht werden. Außerdem können durch Anwendung und Weiterverarbeitung sowie Verbreitung neuer Produkte, Dienstleistungen oder Analysen auch von nicht-staatlicher Seite entwickelt werden, die von der öffentlichen Gemeinschaft genutzt werden können [Dietrich, 2010]. Eines von vielen Beispielen ist der APP:CONTEST[5] in Hamburg: hier sind alle Interessenten aufgefordert eine App-Idee oder eine schon funktionsfähige App zu präsentieren. Voraussetzung ist, dass die App einen direkten Bezug zur Stadt Hamburg hat und von Nutzen für Bürger, Unternehmer oder Touristen ist. Dazu wird explizit auf das Transparenzportal Hamburgs[6] hingewiesen. Der Vorteil eines App-Wettbewerbs liegt darin, den Datenbestand in der Bevölkerung, aber v. a. auch in Entwicklerkreisen bekannt zu machen, dass sinnvolle Apps entwickelt und Ideen gesammelt werden, die umgesetzt werden können [Dobusch et al., 2011].
Es sollte also sorgfältig abgewogen werden, welche Daten der Allgemeinheit nutzen können und welche Daten aus Datenschutzsicht unveröffentlicht bleiben sollten.

[4] http://www.bpb.de/gesellschaft/medien/opendata/64055/was-sind-offene-daten (03.12.2014)
[5] http://www.app-contest-hamburg.de/ (28.05.2015)
[6] http://transparenz.hamburg.de/open-data/ (28.05.2015)

Open Data bzw. Open Government Data ist ein bedeutender Teil von **Open Government**. Als Open Government bezeichnet man die Kommunikation zwischen Bürger und Staat, Staat und Wirtschaft & Industrie sowie innerhalb des Staates [Almarabeh & AbuAli, 2010], [von Lucke, 2010].

Für die Bereitstellung von Behördendaten gibt es acht Prinzipien[7]:

1. Vollständig
2. Primärquelle
3. Zeitnah
4. Zugänglich
5. Maschinenlesbar
6. Nicht diskriminierend[8]
7. Nicht proprietär[9]
8. Lizenzfrei

Die *Sunlightfoundation*[10] nennt zusätzlich in dem Bericht *Ten Principles for Opening Up Government Information* noch *Dauerhaftigkeit* und *niedrige Nutzungskosten* als weitere Prinzipien.

Angefangen hat die Open Government-Bewegung nach dem zweiten Weltkrieg in den USA. Demnach war es die Aufgabe der Presse das staatliche Handeln zu überwachen. Wenn die Handlungen des Staates für Presse, Parlament und Öffentlichkeit transparent waren, galt ein Staat als offen [Herzberg, 2013]. Heute gibt es drei tragende Säulen einer funktionierenden Open Government. Dazu gehören *Transparenz*, *Kollaboration* und *Partizipation* [Groß et al., 2014]. Zusätzlich ist eine Infrastruktur für den Dialog zwischen Regierung und Öffentlichkeit nötig [Almarabeh & AbuAli, 2010].

Transparenz steht für die Information der Bürger über das politische Handeln und die daraus entstehende Nachvollziehbarkeit der Entscheidungen[11] [von Lucke & Geiger, 2010]. *Kollaboration* beschreibt die nötige Zusammenarbeit innerhalb der Regierung, sowie zwischen Staat und Öffentlichkeit. Die Beteiligung der Öffentlichkeit an Entscheidungen wird *Partizipation* genannt [von Lucke & Geiger, 2010]. Ziele der Open Government sind es, Effektivität, Akzeptanz und Effizienz politischer Entscheidungen zu erhöhen. Dies kann durch Einbinden von Erfahrungen und Wissen der Öffentlichkeit erreicht werden, sowie durch die Berücksichtigung der Bedürfnisse und Anforderungen der Öffentlichkeit [von Lucke, 2010]. Durch transparentes Entscheiden kann Korruption und Verschwendung vorgebeugt bzw. aufgedeckt werden und somit das Vertrauen in die Demokratie bzw. die Demokratie selbst gestärkt werden.[11] [Dobusch et al., 2011] Dabei sollte auf Feedback der Öffentlichkeit eingegangen werden, um die Zusammenarbeit zu bewerten und zu verbessern, sowie neue Wege der Zusammenarbeit zu erfassen.[11]

E-Government ist ein Sammelbegriff für die Durchführung von Regierungs- und Verwaltungshandeln, wobei Informations-Kommunikations-Technologien genutzt werden, um einen direkten, beidseitigen Dialog zwischen Staat und Öffentlichkeit herzustellen [Almarabeh & AbuAli, 2010], [Herzberg, 2013].

3.2 New York City Open Data Portal

Die in dieser Arbeit verwendeten Daten stammen hauptsächlich aus dem *NYC Open Data Portal*.[12] Im März 2012 wurde der Grundstein des NYC Open Data Portals gelegt und rechtlich festgehalten.

[7] http://opendata-network.org/2010/03/8-open-government-data-prinzipien-zeitnah/ (28.05.2015)
[8] d. h. ohne Registrierung nutzbar;
[9] d. h. in Standardisierten Formaten bereitgestellt; keine juristische Person hat die alleinige Kontrolle;
[10] http://sunlightfoundation.com/policy/documents/ten-open-data-principles/ (26.05.2015)
[11] https://www.whitehouse.gov/the_press_office/Transparency_and_Open_Government/ (27.05.2015)
[12] www.nycopendata.socrata.com (20.10.2014)

Ein Jahr später wurden die ersten Datensätze bereitgestellt. Bis Ende 2018 sollen alle geeigneten Daten veröffentlicht sein [Bloomberg und Merchant, 2012]. Aktuell sind 1343 Datensätze auf dem NYC Open Data Portal verfügbar (Stand Juni 2015).[12]

Die Datensätze werden von diversen Behörden der Stadt New York und anderen städtischen Organisationen standardmäßig erfasst und für die Öffentlichkeit geeignete Daten werden freigegeben. Sie sollen u. a. dazu dienen, die Erreichbarkeit und Transparenz der Stadtregierung zu verbessern. Die Öffentlichkeit kann die Daten nutzen, weiterverarbeiten und diese auf der Plattform wieder für andere zur Verfügung stellen. Die bereitgestellten Daten sind maschinenlesbar und werden regelmäßig aktualisiert. Zusätzlich sind Metadaten verfügbar, welche die Daten und die Art ihrer Erhebung erläutern. Die Daten liegen vielfältig vor und beinhaltet neben Geodaten, Genehmigungen oder Zulassungen auch Kriminalitätsdaten und Beschwerden. Die originalen Daten werden mit denen anderer Verwaltungen verknüpft um eine möglichst umfassende stadtweite Datenplattform zu erstellen. Ziel ist es, NYC zu einer der weltweit führenden digitalen Städte zu machen, um öffentlich-private Partnerschaften zu unterstützen und so die wirtschaftliche Entwicklung der Stadt zu fördern.[12]

Die Bedingungen zur Nutzung des Portals sind in den *Termes of Use*[12] geregelt. Die Datensätze werden zur Information von Öffentlichkeit und Anwendungsentwicklern auf der Webseite zur Verfügung gestellt. Von Seiten der Stadt bzw. des NYC Open Data Portals wird keine Garantie für Vollständigkeit, Aktualität, Inhalt oder Tauglichkeit der bereitgestellten Daten übernommen. Auch ist sie nicht für die von Dritten veröffentlichten Datensätze verantwortlich. Ein großer Teil der veröffentlichten Daten wird von Stadtverwaltungen bereitgestellt. Für die Qualität und regelmäßige Aktualisierung sowie Korrekturen sind die Autoren selbst zuständig.[12]

Zusätzlich gelten noch die *Terms of Use of NYC*.[13] Diese regeln u. a. den rechtmäßigen Gebrauch, das Verfügbarmachen von Informationen, sowie den Umgang mit geistigem Eigentum. In dem *Open Data Policy and Technical Standards Manual* [Bloomberg und Merchant, 2012] ist festgelegt, dass die jeweiligen Verwaltungen Eigentümer der veröffentlichten Daten bleiben. Die Nutzer erhalten kein Eigentum an den veröffentlichten Daten der Verwaltungen. Diese Daten bleiben eine *public resource* und sind für jeden mit Internetzugang verfügbar. Die öffentliche Nutzung der Daten schließt Entwicklungen von Anwendungen mit ein. In einem solchen Fall ist der Entwickler Eigentümer des geistigen Eigentums der Anwendung, das Eigentum an den Daten verbleibt allerdings bei den Verwaltungen.

Ergänzend ist in dem *Local Law 11 of 2012* der Stadt New York auf rechtlicher Ebene die Bereitstellung offener Daten der Stadt festgelegt [Bloomberg und Merchant, 2012].

3.3 Open Data in Deutschland

Im Februar 2013 lief der Testbetrieb des *GovData* Portals an. Seit 2015 ist der Regelbetrieb aufgenommen und die Betreuung vom Bundesministerium des Inneren an die Stadt Hamburg übergegangen. Die Homepage *GovData*[14] bietet eine einheitliche und zentrale Zugangsstelle zu Verwaltungsdaten aus Bund, Ländern und Kommunen im Sinne des *E-Government-Gesetzes* an.

[13] http://www1.nyc.gov/home/terms-of-use.page (01.06.2015)
[14] https://www.govdata.de/ (03.06.2015)

Es werden keine Daten privater, wissenschaftlicher oder anderer Einrichtungen auf der Plattform geführt. Ziel des Portals ist es, ein deutschlandweites, föderales Datenangebot bereitzustellen. Dazu werden Datenplattformen des Bundes, der Länder und Kommunen verknüpft und der Zugang zu Daten und Informationen für Öffentlichkeit und Verwaltungen verbessert [Klessmann et al., 2012].

Die Bereitstellung erfolgt in dezentraler Datenhaltung. Das bedeutet, die Daten befinden sich weiterhin auf den jeweiligen Portalen der Datenbereitsteller. Über das GovData Portal werden die Datensätze lediglich verlinkt angeboten. Um die Datensätze filtern oder suchen zu können, wird ein einheitliches Schema, der sogenannten Metadatenkatalog[15], benötigt. Für die Nutzungsbedingung bedeutet die dezentrale Datenhaltung, dass eine rechtliche Beziehung nur zwischen Datenbereitsteller und Datennutzer besteht, die individuell von dem Datenbereitsteller geregelt werden muss.

Die Daten werden nach Möglichkeit in ihrer Urform, d. h. unbearbeitet, maschinenlesbar und in offenen Formaten bereitgestellt. Zusätzlich gibt es die Option, Daten in nicht-maschinenlesbaren Formaten herunterzuladen. Bei den Daten handelt es sich nicht nur um Open Data, sondern auch um solche, die nur eingeschränkt nutzbar sind. Um die Nutzungsbestimmungen übersichtlich und einheitlich zu gestalten, wurde eine Standard-Nutzungsbestimmung eingeführt, die sogenannte *Datenlizenz Deutschland*.[14] Das Angebot der Daten ist sehr vielfältig und umfasst von statistischen Daten[16] bis zu Geodaten[17] auch Jahresberichte und Daten aus 14 verschiedenen Kategorien, wie „Gesundheit“, „Umwelt und Klima“ und „Geographie, Geologie und Geobasisdaten“.[14]

Folgende Regionen in Deutschland haben bereits ihre Portale für offene Daten von Verwaltungen mit dem GovData Portal verknüpft:

- Niedersachsen[18]
- Berlin[19]
- Bremen[20]
- Baden-Württemberg[21] (noch Testbetrieb)
- Bayern[22]
- Hamburg[23]
- Rheinland-Pfalz[24]
- Nordrhein-Westfalen[25]

Insgesamt sind auf dem GovData Portal 13.598 maschinenlesbare Datensätze und 785 Dokumente in nicht-maschinenlesbaren Formaten vermerkt (Stand Juni 2015).[14]

Die Internetseite *offeneDaten*[26] bietet neben den Daten der öffentlichen Verwaltung auch Daten aus Wissenschaft bzw. Forschung und privat erzeugte Daten an. Das Portal *offeneDaten* soll nicht als Konkurrenz zu Datenportalen öffentlicher Verwaltungen gesehen werden, sondern als ein zusätzliches Angebot. Dieses wird von der *Open Knowledge Foundation Deutschland* mit Hilfe von Freiwilligen betrieben.[27]

[15] Comprehensive Knowledge Archive Network (CKAN), Standard für Datenkatalogsoftware
[16] https://www-genesis.destatis.de/genesis/online (06.06.2015)
[17] http://www.geoportal.de/DE/Geoportal/geoportal.html?lang=de (06.06.2015)
[18] http://numis.niedersachsen.de/ (04.06.2015)
[19] http://daten.berlin.de/ (04.06.2015)
[20] http://transparenz.bremen.de/sixcms/detail.php?gsid=bremen02.c.730.de (04.06.2015)
[21] https://opendata.service-bw.de/home (04.06.2015)
[22] https://opendata.bayern.de/;jsessionid=E3C4AC093A01D4FADB5AF513956B33CF?0 (04.06.2015)
[23] http://transparenz.hamburg.de/ (28.05.2015)
[24] http://www.daten.rlp.de/legal-notice (04.06.2015)
[25] https://open.nrw/de/startseite (04.06.2015)
[26] offeneDaten.de (03.12.2014)
[27] https://offenedaten.de (03.12.2014)

Die Veröffentlichung von Geodaten inklusive Metadaten wird in Deutschland von der EU-Richtlinie *INSPIRE* geprägt. Diese fordert bis 2019 eine stufenweise Bereitstellung räumlicher Daten in einer gemeinsamen europaweiten Geodateninfrastruktur[28]. Dadurch wird sichergestellt, dass Geodaten grenzübergreifend kompatibel sind und gemeinschaftlich genutzt werden können, um Maßnahmen und politische Konzepte umzusetzen [EU-Richtlinie, 2007]. Die Richtlinie regelt u. a. die Homogenisierung von Koordinatensystemen, Flurstücken, Gebäuden und meteorologischen Daten.[29] Sie gilt für Behörden, die Geodaten erheben und/oder bereitstellen. Es wird aber nicht festgelegt, welche Daten zu erheben sind [EU-Richtlinie, 2007]. Zusätzlich legt die EU-Richtlinie *PSI-RL2013/37/EU* die Weiterverwendung von Informationen aus dem öffentlichen Sektor fest, da es wichtig ist, dass die Mitgliedsstaaten gleichen Regelungen folgen, ob und wie die Weiterverwendung von Daten erlaubt ist [Bundesministerium des Innern, 2014].

3.4 Vergleich NYC – Deutschland

Im Folgenden werden die verfügbaren Geodaten zwischen USA und Deutschland verglichen. Dazu werden stellvertretend die Daten der Städte New York und Hamburg einander gegenübergestellt. Hierbei wird nicht berücksichtigt, wie viele Datensätze jeweils zur Verfügung stehen, sondern ob es möglich ist, vergleichbare 3D-Stadtmodelle für die beiden Städte zu erzeugen. Da diese Arbeit sich mit der Generierung eines Modells für New York beschäftigt, wird im Folgenden v. a. auf Hamburg eingegangen.

Auf Grundlage der ALKIS Datenbank können flächendeckend Daten zu Gebäuden, Adressen, Flurstücken und Verwaltungsgrenzen abgeleitet werden. Der Datensatz *ALKIS Liegenschaftskarte – ausgewählte Daten Hamburg*[30] enthält u. a. Flurstücke mit Flurstücksnummer, tatsächlicher Nutzung und Gebäude. Im Vergleich dazu enthält der entsprechende New Yorker Datensatz der Pluto-Datenbank ähnliche Informationen sowie zusätzliche zu den Flurstücken. Beispielsweise ist hinterlegt, inwieweit es sich um privates, städtisches oder föderales Eigentum handelt. Zusätzlich wird meist der Name des Flurstückeigentümers publiziert. Die Veröffentlichung des Eigentümers eines Flurstückes ist in Deutschland allerdings aus Datenschutzgründen nicht denkbar. Des Weiteren gibt es für New York Angaben zu den geschätzten Werten der Flurstücke.

Beide Datensätze werden regelmäßig erneuert: Hamburg vierteljährlich und New York halbjährlich. Zusätzlich ist es v. a. bei Flurstücken von Interesse, wie genau die Daten sind. Die Daten aus Hamburg besitzen eine Lagegenauigkeit von 0,1 m, die Daten aus New York enthalten bewusst keine Angabe dazu.

Für Hamburg existiert bereits ein dreidimensionales Stadtmodell[31] im LoD1 für Gebäude. Ein weiteres Stadtmodell im LoD2 ist bereits im Aufbau und soll Ende 2016 zur Verfügung stehen. Die Gebäude von New York können auf Grund ihrer Attribute Gebäudehöhe und Höhenlage ebenfalls in 3D generiert werden. Die Information, auf welchem Flurstück Gebäude stehen, ist für Hamburg nicht gegeben, kann aber über eine räumliche Verschneidung mit den Flurstücken ermittelt werden.

[28] http://www.iais.fraunhofer.de/ogd4kommunen.html (03.12.2014)
[29] http://www.geoportal.de/DE/GDI-DE/INSPIRE/Direktive/Data-Specs/data-specs.html?lang=de (06.06.2015)
[30] http://suche.transparenz.hamburg.de/dataset/alkis-liegenschaftskarte-ausgewahlte-daten-hamburg (07.06.2015)
[31] http://suche.transparenz.hamburg.de/dataset/3d-stadtmodell-hamburg (07.06.2015)

Für die Adressdaten gibt es auf dem Hamburger Portal den Datensatz *ALKIS Adressen Hamburg*[32]. Dieser liegt flächendeckend vor und enthält neben der Datensatznummer, Straßennamen, Hausnummer, gegebenenfalls einen Zusatz sowie die Postleitzahl und den Ort. Die Daten entsprechen in diesem Fall den von New York zur Verfügung stehenden Daten zu den Adressen.

Die Zonierung, die in dem New Yorker Stadtmodell vorgenommen wird, kann ähnlich auch in Hamburg umgesetzt werden. Der analoge Datensatz heißt *ALKIS Verwaltungsgrenzen Hamburg*[33] und enthält Landesgrenzen, Bezirke, Stadtteile, Ortsteile, Gemarkungen und Postleitzahlbezirke.

Der Hamburger Datensatz der Parkanlagen[34] enthält den Namen und eine Beschreibung des Parks. Zusätzlich wird für ausgewählte Parkanlagen auf die Homepage *Hamburg.de* verwiesen. Diese enthält eine genauere Beschreibung des jeweiligen Parks inklusive Bild und Karte, sowie weitere Informationen. Die Parkanlagen werden in 2D mit Hilfe von Punkten verortet. Im Gegensatz dazu enthalten die Datensätze von New York die Flächen der Parks. Eine Darstellung der Parkanlagen wäre für Hamburg möglich, allerdings nur als Vermarkung und nicht mit Darstellung einer Fläche.

Für Hamburgs Gewässer gibt es zum Einen Daten über Badegewässer[35], die als Polygone vorliegen, zum Anderen das Basis-Gewässernetz[36]. Letzteres beinhaltet die Mittelachsen der Fließgewässer in Hamburg. Allerdings sind nicht alle Gewässer Hamburgs in diesen beiden Datensätzen enthalten. Für New York werden alle Gewässer in einem einheitlichen Datensatz zusammengefasst und als Polygone repräsentiert. Eine vollständige Umsetzung der Gewässer wie in New York ist in Hamburg mit den beiden genannten Datensätzen demzufolge nicht möglich.

Im Datensatz *Straßenbaumkataster Hamburg*[37] sind flächendeckend die Bäume auf öffentlichen Straßenflurstücken im Stadtgebiet Hamburg erfasst. Die Darstellung der Bäume in einem Stadtmodell von Hamburg wäre semantisch reichhaltiger, da für die Bäume mehr Attribute zur Verfügung stehen, als für die Bäume in New York.

Die Daten über Straßen stammen aus der *Hamburger Straßeninformationsbank TT-SIB*[38]. Hieraus wird eine Teilmenge veröffentlicht, die verschiedene Informationen zur Art der Straßen, Länge, Anzahl der Fahrstreifen und Name, sowie verschiedene administrative Attribute innehat. Eine Darstellung der Straßen in einem 3D-Stadtmodell von Hamburg ist also möglich, allerdings enthalten die Daten im Gegensatz zu New York weder Informationen über die relative Höhe der Straße, noch Hinweise darauf, ob es sich z. B. um eine Brücke oder einen Tunnel handelt.

Die verwendeten Daten stammen in beiden Fällen von Behörden und Verwaltungen, die diese öffentlich auf Webseiten zur Verfügung stellen. Die Daten liegen jeweils flächendeckend für das gesamte Stadtgebiet vor und werden in regelmäßigen Abständen aktualisiert. Die Datensätze sind bei beiden Städten in 2D. Allerdings gibt es für Hamburg schon DGMs[39]. Diese werden in den Auflösungen von 1, 10 und 25 Meter angeboten. Die Daten stammen von Airborne Laserscannings.

[32] http://suche.transparenz.hamburg.de/dataset/alkis-adressen-hamburg (07.06.2015)
[33] http://suche.transparenz.hamburg.de/dataset/alkis-verwaltungsgrenzen-hamburg (07.06.2015)
[34] http://suche.transparenz.hamburg.de/dataset/parkanlagen (07.06.2015)
[35] http://suche.transparenz.hamburg.de/dataset/badegewasser-daten-hamburg (07.06.2015)
[36] http://suche.transparenz.hamburg.de/dataset/basis-gewassernetz-fur-hamburg (07.06.2015)
[37] http://suche.transparenz.hamburg.de/dataset/strassenbaumkataster-hamburg (07.06.2015)
[38] http://suche.transparenz.hamburg.de/dataset/strassen-hh-sib (06.06.2015)
[39] http://suche.transparenz.hamburg.de/dataset/digitales-hohenmodell-hamburg-dgm-1 (07.06.2015)

Für New York liegt ein DGM in Form eines Rasterbilds aus Airborne Laserscanning in der Auflösung von 1 Fuß vor.

Die Datensätze sind in beiden Fällen maschinenlesbar und auch weiterverarbeitbar. Die Daten für New York liegen hauptsächlich im (Esri-/)Shape-Format vor, für Hamburg als GML-Dateien. Im Vergleich zu New York sind in Hamburg die Daten nicht als Open Data im herkömmlichen Sinne verfügbar. Die Daten des *Transparenzportal Hamburg* unterliegen der *Datenlizenz Deutschland-Namensnennung 2.0*. Diese besagt, dass die Daten für kommerzielle Zweck genauso wie für nicht-kommerzielle verwendet werden dürfen, Nutzungsbedingungen aber eingehalten werden müssen. Diese sind unter *https://www.govdata.de/dl-de/by-2-0* nachlesbar. [40,41,42]

Der direkte Vergleich der verfügbaren Daten zu New York und Hamburg zeigt, dass es grundsätzlich möglich wäre, ein annähernd vergleichbares Stadtmodell für Hamburg zu generieren. Alle Datensätze, die für das in dieser Arbeit generierte Stadtmodell von New York gewählt sind, existieren vergleichbar für Hamburg. Allerdings müssen für Hamburg in den meisten Fällen Abstriche im Informationsgehalt der Objekte gemacht werden, wie in den Attributen der Flurstücke, bei der Vollständigkeit der Gewässer und der Geometrie der Parkanlagen. Dies ist begründbar durch einen anderen Umgang bezüglich Transparenz und der Verfügbarkeit von Daten für die Öffentlichkeit in Deutschland als in den USA, die seit 2009 *Open Data* als politische Agenda führen [von Lucke & Geiger, 2010]. Schließlich kann Hamburg nicht stellvertretend für andere Städte in Deutschland gesehen werden, da die Stadt eine Vorreiterrolle in Deutschland hat. In Hamburg wird seit Jahren die Initiative zur Veröffentlichung von behördeninternen Daten unterstützt.[6]

Im internationalen Vergleich der G7 (bzw. der G8) Staaten ist Deutschland mit der Veröffentlichung von Verwaltungsdaten auf dem letzten Platz.[43] Im Jahr 2013 wurde die *Open Data Charta* von den G8 Staaten beschlossen mit dem Ziel, Verwaltungsdaten offenzulegen. Zur Umsetzung der Charta wurden fünf Prinzipien festgelegt. Zu den Prinzipien zählen:

- *Open Data by Default*: öffentliche Veröffentlichung der Daten (mit Ausnahmen) auf einem nationalen Portal
- *Qualität und Quantität*: Bereitstellung möglichst genauer und aktueller Daten mit einheitlichen Metadaten
- *Für alle nutzbar*: kostenlos und in offenen Formaten
- *Veröffentlichung der Daten für besseres Regierungshandeln*: Transparenz bei Datenerhebung und Bereitstellung, internationaler Austausch der Erfahrungen
- *Veröffentlichung der Daten für Innovation*: Bereitstellung von (Roh-)Daten auch für kommerzielle Zwecke

Die Position Deutschlands im internationalen Vergleich kann nicht auf alle Gegenden in Deutschland übertragen werden. Es gibt mittlerweile viele Städte, die ihre Daten und entwickelte Anwendungen öffentlich im Internet zur Verfügung stellen. Wie Hamburg ist auch Berlin ein Vorreiter im Bereich Open Data und hat seit diesem Jahre sein 3D-Stadtmodell als Open Data der Öffentlichkeit zum

[40] https://www.govdata.de/dl-de/by-2-0 (07.06.2015)
[41] https://transparenz.hamburg.de/ (28.05.2015)
[42] https://nycopendata.socrata.com/ (20.10.2014)
[43] https://sunlightfoundation.com/blog/2015/06/08/open-data-without-germany-the-g7-summit-and-the-bumpy-road-to-transparency/ (16.06.2015)

Download bereitgestellt. Dabei sind die Nutzungsbedingungen sehr frei gehalten, z. B. darf das Stadtmodell für beliebige rechtmäßige, auch kommerzielle, Zwecke genutzt werden. Des weiteren ist es erlaubt, eigene Daten zu integrieren, das Modell zu bearbeiten und umzugestalten.[44]

[44] http://www.businesslocationcenter.de/berlin3d-downloadportal/documents/terms.de.html (04.07.2015)

4 Überblick über die verwendeten Daten und Herausforderungen

Verfasser: B. Burger

In diesem Kapitel wird ein Überblick über die verwendeten Daten gegeben und auf Herausforderungen eingegangen, die beide Masterarbeiten betreffen. Schwierigkeiten, die sich auf einzelne Datensätze beziehen, werden in den jeweiligen Kapiteln erläutert. Als die größten Herausforderungen stellten sich die semantischen bzw. geometrischen Transformationen und die Verwaltung der Daten heraus, was v. a. auf ihrer Größe und Vielfalt beruht.

4.1 Transformationsprobleme

Unter einer Transformation ist hier die Überführung der Daten aus dem Originaldatensatz hin zum Zieldatensatz zu verstehen. Die zweidimensionalen Daten stammen zwar aus einer einzigen Quelle – den Departments von NYC – jedoch bestehen sie aus vielen unabhängigen Datensätzen, die getrennt voneinander erfasst und fortgeführt werden (vgl. Tabelle A.1 im Anhang). Dies führt zu räumlichen und thematischen Inkonsistenzen, wenn die Daten in ein gemeinsames 3D-Stadtmodell im CityGML-Format integriert werden. Allerdings ist das Ziel, ein „geometrisch, topologisch und semantisch konsistentes 3D-Modell zu erzeugen“ [Stadler, 2007], indem möglichst keine semantische oder geometrische Information bei der Transformation verloren geht.

In der CityGML-Modellierung existiert dafür eine duale Hierarchie mit einem Geometriemodell und einem semantischen Modell, innerhalb derer die jeweiligen Objekte durch Beziehungen miteinander verknüpft sind.

Die Semantik repräsentiert reale Objekteinheiten durch Objekte wie Gebäude, Wände, Fenster usw. und die Beschreibung dieser mit Hilfe von Attributen, Beziehungen oder Aggregationshierarchien, so genannte Teil-Ganzes-Beziehungen. Daher sind semantische Informationen unabdingbar, um komplexe Abfragen oder Analysen auszuführen. Die Geometrie hingegen beschreibt Objekte und ihre Beziehungen durch ihre Position und ihr Ausmaß. In Folge dessen ist sie für die Visualisierung erforderlich, um die Objekte so realitätsnah wie möglich zu darzustellen.

Der Vorteil einer Verbindung der semantischen mit den geometrischen Informationen liegt darin, dass die semantischen Objekte wissen, wo sie sich befinden bzw. wie groß sie sind und den Geometrieobjekten bekannt ist, was sie verkörpern. Demzufolge können sowohl thematische und räumliche als auch miteinander kombinierte Abfragen an ein Stadtmodell gestellt werden [Gröger et al., 2012],[Stadler, 2007].

Die Schwierigkeit bezüglich der von der Stadt New York zur Verfügung gestellten Daten liegt darin, dass die Daten weder die Struktur von CityGML aufweisen noch einheitlich strukturiert

sind. Daraus folgt, dass für jeden Datensatz und dessen Attribute einzeln abgewogen wird, ob und wie er ins CityGML-Format übernommen werden kann oder ob generische Objekte bzw. Attribute neu angelegt werden müssen. Selbst innerhalb eines Datensatzes kann es zu Unstimmigkeiten kommen, weshalb einzelne Objekte bzw. Objektgruppen für sich betrachtet werden. Daneben können Zwiespältigkeiten entstehen, wenn Attribute nicht ausreichend oder überhaupt nicht erläutert werden oder die Attributwerte nicht realistisch sind. Solche semantische Inhalte vergleichbar zu machen bzw. 2D-Daten unterschiedlicher Herkunft automatisch zusammenzuführen, stellt Sester (2007), neben strukturellen und geometrischen Problemen, als die größte Herausforderung dar.

Weiterhin gibt es die Möglichkeit, Objekte semantisch anzureichern, indem ihnen neben den bereits vorhandenen Attributen weitere hinzugefügt werden. Dies können z. B. Analyseergebnisse sein, die während der Generierung erzeugt werden, wie die Flächen- oder Volumenberechnung von Objekten. Neben bekannten Ansätzen, bei denen 3D-Objekten mit Hilfe von 2D-Stadtplänen oder zweidimensionalen topographischen Karten Informationen hinzugefügt werden, geben Smart et al. (2011) ein weiteres Beispiel: dabei werden dreidimensionalen Objekten automatisch und georeferenziert Informationen aus z. B. Wikipedia-Artikeln oder OSM-Karten hinzugefügt .

Die Originaldaten der Stadt NY sind durch die verschiedenen Geometrietypen Punkt, Linie und Fläche zweidimensional repräsentiert (die Rasterdaten werden an dieser Stelle nicht erwähnt, da sie zur Bearbeitung ebenfalls in Punkte transformiert werden). Diese entsprechen zwar den im Geometriemodell verwendeten Primitiven, trotzdem gibt es Abweichungen, wodurch die Datensätze an das CityGML-Modell angepasst werden müssen. Beispielsweise sind einige Straßenmittelachsen geometrisch redundant, was die LoD0–Geometrie in CityGML jedoch nicht unterstützt, da sie dort als geometrische Komplexe gespeichert werden.

Des Weiteren erhalten die zweidimensionalen Originaldaten eine dreidimensionale Geometrie. Dabei können allerdings weder alle Datensätze noch gleiche Geometrietypen auf dieselbe Art und Weise behandelt werden. Es werden zwar allen Daten die Höheninformationen mit Hilfe des DGMs übertragen, jedoch gibt es für manche Datensätze weitere geometrische Transformationen durchzuführen: z. B. werden die Gebäudepolygone zu Körpern extrudiert, die Punktgeometrien der Bäume werden zu 3D-Körpern in Form von „Lollipops“ umgewandelt oder die Höhen von Brücken- bzw. Tunnelstraßen werden so festgelegt, dass sie über bzw. unter dem DGM verlaufen.

Weitere geometrische Inkonsistenzen können durch unterschiedliche Genauigkeiten der Datensätze und verschiedene Referenzsysteme entstehen (siehe unten). Wie schließlich die semantische und geometrische Transformation für die einzelnen Datensätze exakt durchgeführt wird, wird in den jeweiligen Kapiteln beschrieben.

4.2 Koordinatenreferenzsysteme

Die Datensätze des NYC Open Data Portals sollen laut dem *Open Data Policy and Technical Standards Manual* im Web Mercator Koordinatenreferenzsystem WGS84/EPSG:3857 veröffentlicht werden, um die Datennutzung mit Online Mapping Systemen zu erleichtern. Allerdings handelt es sich bei dem *World Geodetic System 1984* um keine Projektion, wodurch Distanz- und Flächenmessungen schwieriger sind als in projizierten Koordinatensystemen. Aus diesem Grund dürfen die zu veröffentlichenden Daten auch im *New York State Plane, Long Island Zone*–Koordinatenreferenzsystem

Datensätze	Koordinatenreferenzsysteme
Adressen Bäume DEM Hydrographie Landbedeckung LION (Straßen) Parks Zonierung	**Projected coordinate system** NAD_1983_StatePlane_New_York_Long_Island_FIPS_3104_Feet **Geographic coordinate system** North_American_1983 Unit: Feet
Gebäude Hydrostrukturen	**Projected coordinate system** WGS_1984_Web_Mercator_Auxiliary_Sphere **Geographic coordinate system** GCS_WGS_1984 Unit: Degree **Projection** Mercator_Auxiliary_Sphere Unit: Meter

Tabelle 4.1: Die ursprünglichen Koordinatenreferenzsysteme der verwendeten Datensätze.[2]

(FIPS Zone 3104/EPSG:2263) zur Verfügung gestellt werden. Das dazugehörige Bezugssystem ist das kontinentale nordamerikanische Datum *North American Datum 1983* (NAD83) [Bloomberg und Merchant, 2012].

Dementsprechend sind die in der vorliegenden Arbeit verwendeten Datensätze in diesen beiden Koordinatenreferenzsystemen gegeben (vgl. Tabelle 4.1). Da sich das resultierende 3D-Stadtmodell nur auf die Stadt New York bezieht, ist kein globales System wie das WGS84 notwendig. Durch die hier gegebene Pseudo-Mercator-Projektion handelt es sich zwar um ein metrisches, jedoch nicht um ein anerkanntes geodätisches System. Folglich ist dieses Referenzsystem nur für gewisse Visualisierungsanwendungen im Internet zu nutzen, allerdings nicht für die vorliegende Arbeit.[1]

Aus diesem Grund wird für diese Arbeit das Nordamerikanische Koordinatenreferenzsystem NAD83, beruhend auf dem GRS80, gewählt.[3] Die Projektion *New York State Planes, Long Island* wird übernommen, wobei erwähnt sei, dass der Stadtteil Staten Island nicht mehr in ihrem Einsatzgebiet liegt.[4] Darauf wird nicht weiter eingegangen, da dies die von der Stadt New York selbst vorgegebene Projektion ist. Wenngleich diese Projektion die vor Ort gängige Maßeinheit Fuß beinhaltet, wird sie nicht beibehalten, sondern in die Einheit Meter überführt. Die Erklärung dafür ist das Fehlen eines passenden Höhenreferenzsystems in Fuß, das in dem zuvor erwähnten Koordinatensystem noch nicht enthalten ist. NAVD88 height basiert auf dem *North American Vertical Datum 1988* und liegt in der Einheit Meter vor, weswegen das planimetrische Referenzsystem daran angepasst wird.[5]

In CityGML müssen alle Koordinaten einem Koordinatenreferenzsystem (CRS – coordinate reference system) zugeordnet werden. Es kann jedem Geometrieobjekt ein eigenes (3D-)Koordinatensystem

[1] http://georepository.com/crs_3857/WGS-84-Pseudo-Mercator.html (05.03.2015)
[2] https://nycopendata.socrata.com (20.10.2014)
[3] http://georepository.com/crs_32118/NAD83-New-York-Long-Island.html (05.03.2015)
[4] http://georepository.com/crs_2263/NAD83-New-York-Long-Island-ftUS.html (05.03.2015)
[5] http://georepository.com/crs_5703/NAVD88-height.html (05.03.2015)

CRS	Name	Beschreibung	Datum	Ellipsoid	Einheit
2D	EPSG: 32118	NAD83 New York State Planes, Long Island, Meter [NY83-LI]	NAD83	GRS1980	Meter
1D	EPSG: 5703	NAVD88 height	NAVD88		Meter

Tabelle 4.2: In dieser Arbeit verwendete Koordinatenreferenzsysteme: Das planare System NAD83 [Notation der Software FME] und das vertikale System NAVD88[5].

zugewiesen werden. Dabei unterstützt GML3 sowohl echte 3D CRS als auch 2D+1D CRS mit verschiedenen Systemen für planare (x, y) und vertikale Koordinaten (z). Um letzteres handelt es sich in diesem Fall (vgl. Tabelle 4.2) [Kolbe, 2009].

Um zu gewährleisten, dass die am Ende dieser Arbeit erzeugten Daten schließlich ein einheitliches Koordinatensystem und dieselben Maßeinheiten aufweisen, werden die verwendeten Datensätze in das planimetrische Koordinatensystem reprojiziert und die betroffenen Attributwerte von Fuß in Meter umgerechnet. Die Höhen werden dementsprechend an das Höhenreferenzsystem angepasst.

4.3 Verwaltung der Datenmengen

Im Folgenden wird zunächst auf die Daten eingegangen, aus welcher Vielfalt und in welchem Umfang sie bestehen. Anschließend wird allgemein erläutert, wie diese Datenmengen bearbeitet werden, ohne auf einen bestimmten Datensatz einzugehen; dies erfolgt in den jeweiligen Kapiteln.

4.3.1 Überblick über die Daten und ihren Umfang

Abbildung 4.1 gibt zunächst einen Überblick über die flächenhafte Ausbreitung der Stadt New York. Dabei repräsentiert das graue Rechteck die Ausdehnung des Rasterbilds des Digitalen Höhenmodells, das eine Fläche von 2.292,801 km^2 einschließt. Es ist allerdings zu erwähnen, dass die Mehrheit der Pixel des sichtbaren grauen Bereichs einen Wert von Null, d.h. einen *no data value* und nicht die realen Höhenwerte aufweist. Im Großen und Ganzen stellen somit die blauen Gewässerflächen inklusive der grünen Landbereiche das in dieser Arbeit behandelte Gebiet mit einer Fläche von 1.079,373 km^2 dar. Die Landflächen für sich betragen 783,187 km^2.

Um im Folgenden zu veranschaulichen, um welche Vielfalt und Mengen an Daten es sich handelt, wird in Tabelle 4.3 eine Übersicht über die Formate, Geometrie, Anzahl der enthaltenen Objekte und Attribute und das Volumen der in dieser Arbeit verwendeten Datensätze, gegeben. Dabei entspricht der obere Teil einer Zeile jeweils den Eigenschaften der Originaldaten, wie sie von NYC zur Verfügung gestellt werden, der untere Teil denen der CityGML-Dateien; bei letzterem ist in Bezug auf die Dateigröße diejenige gemeint, die sich vor dem 3DCityDB-Import ergibt.

Diese Gegenüberstellung gibt lediglich ein Gefühl für die Größenordnungen, jedoch müssen einige Faktoren berücksichtigt werden. Neben der Tatsache, dass XML sehr speicherintensiv ist, da es Informationen als Klartext schreibt, liegt u. a. das DGM nicht als eine einzige große CityGML-Datei vor sondern in Form von 35.153 Kacheln, die jeweils eine eigene CityGML-Datei darstellen, wodurch nochmals mehr Speicherplatz benötigt wird. Einen weiteren Aspekt bildet das Rasterbild

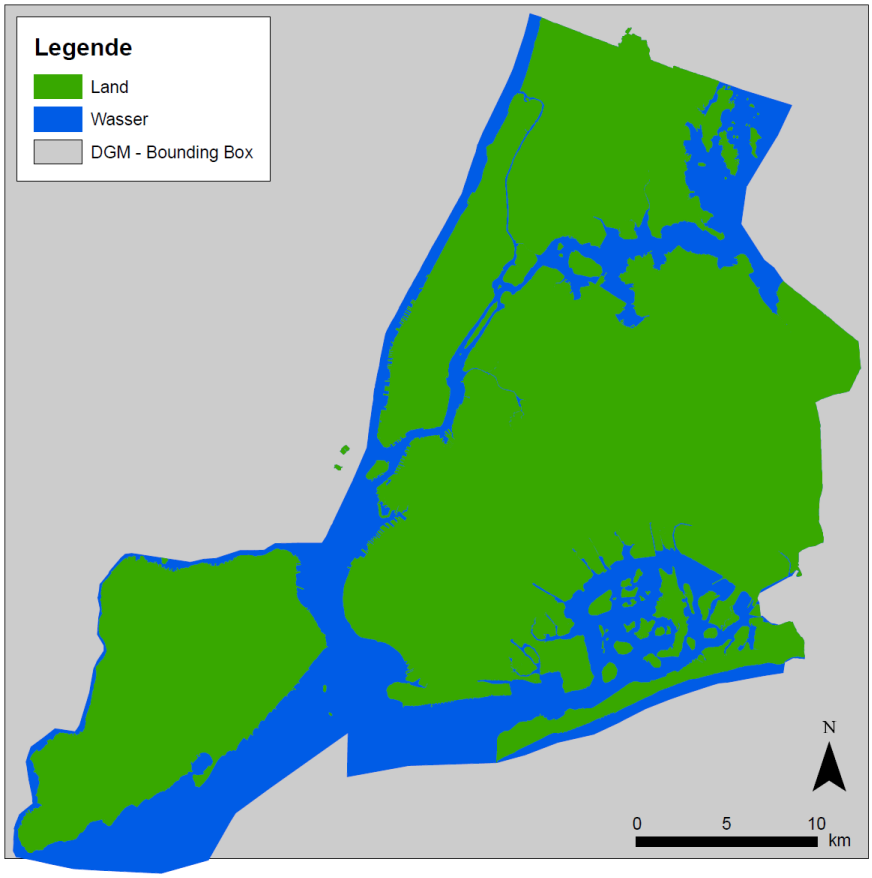


Abbildung 4.1: Flächenhafte Ausdehnung der Stadt New York.

Datensatz	Format		Geometrie	Anzahl Objekte	Anzahl Attribute	Dateigröße [GB]
Bäume	Shape	2D	Punkt	623 920		0,206
	CityGML	3D	Körper	277 108	16	113,000
DGM	Raster	2D	Grid	1		121,000
	CityGML	3D	TIN	35 153	—	1 450,000
Flurstücke	Shape	2D	Fläche	857 851		0,867
	CityGML	3D	Fläche	866 853	75	8,021
Gebäude/Adressen	Shape	2D	Fläche/Punkt	2 023 531		0,931
	CityGML	3D	Körper/Punkt	2 020 523	20 – 55	11,085
Gewässer	Shape	2D	Fläche	1 976		0,010
	CityGML	3D	Fläche	9 542	5	0,026
Gewässerstrukturen	Shape	2D	Fläche	2 464		0,003
	CityGML	3D	Fläche	2 464	3	0,006
Landbedeckung	Raster	2D		1		0,200
	CityGML	3D	—	—	—	—
Parks	Shape	2D	Fläche	14 674		0,025
	CityGML	3D	Fläche	16 159	10	0,054
Straßen	File Geodatabase	2D	Linie	212 890		0,128
	CityGML	3D	Linie & Fläche	149 292	31	0,482
Straßenkreuzungspunkte	File Geodatabase	2D	Punkt	125 118		0,128
	CityGML	3D	Punkt	104 754	1	0,055
Zonierung	Shape	2D	Fläche	2 436		0,005
	CityGML	3D	Fläche	2 436	23	$\star$
Gesamt	Original	2D		3 864 864		123,375
	CityGML	3D		3 484 284	184	$\star$ + 1 582,729

Tabelle 4.3: Überblick der Datensätze in Bezug auf ihr Format, ihre Geometrie, die Anzahl der enthaltenen Objekte und Attribute und die Dateigröße der Originaldatei (oberer Teil einer Zeile) und der CityGML-Datei, bevor diese in die 3DCityDB gespeichert wird (unterer Teil einer Zeile). Der $\star$ ersetzt den Wert für die Dateigröße der Zonierung (in CityGML), da diese bedauerlicherweise bis zum Abgabedatum der vorliegenden Arbeit nicht vollständig berechnet werden konnte; diese wird voraussichtlich 1 GB nicht überschreiten. Ein Überblick über die Herkunft der verwendeten Datensätze befindet sich im Anhang A.1.

der Landbedeckung. Dieses beeinflusst zwar die Originalgesamtdatenmenge, ist jedoch nicht mehr im CityGML-Gesamtvolumen enthalten, da es nur zur Bestimmung der Straßenbreite verwendet wird und demzufolge nicht ins CityGML-Format abgespeichert wird. Jedoch sollte erwähnt werden, dass in dieser Arbeit insgesamt ca. 3,5 Mio. Objekte mit knapp 1,6 TB im CityGML-Format erzeugt werden.

Solche Datenmengen benötigen entsprechende Rechenzeiten für die Bearbeitung und Speicherung, die in Tabelle 4.4 aufgelistet sind. Selbstverständlich hängen diese u. a. von der Leistung der verwendeten Computer oder der Auslastung des DB-Servers ab, wodurch die folgenden Werte nicht absolut betrachtet, sondern nur als Richtwerte gesehen werden sollen. Dabei wird für jeden Datensatz (Spalte 1) dokumentiert, wie viel Zeit benötigt wird, um diesen zunächst in eine/mehrere CityGML-Datei/-en zu schreiben (Spalte 2) und jene anschließend in die 3DCityDB zu importieren (Spalte 3). Das DGM sticht dabei mit nahezu 10 Tagen Berechnungsdauer und ca. 20 Tagen für den Import in die 3DCityDB besonders hervor. Bei letzterem sei angemerkt, dass die Aktivierung der räumlichen und normalen Indizes noch nicht mit einberechnet wurde, da diese bis zum Abgabetag der vorliegenden Arbeit nach 28 Tagen noch nicht vollständig war. Die Generierung der Bäume betrug ebenfalls mehrere Tage. Diese wurden jedoch parallel für jeden Stadtbezirk erzeugt, wodurch die angegebene Dauer eine Summe darstellt. Im Gegensatz dazu wurden z. B. die Parks oder Gewässer (-strukturen) in weniger als 5 min erstellt und gespeichert. Insgesamt ergibt sich ein Zeitraum von ca. 40 Tagen für die Bearbeitung und Speicherung der Daten. Dieser enthält allerdings noch nicht die Zeiten für die Zonierung und die Aktivierung der Indizes (für das DGM). Die Erstellung der Indizes für alle Daten – bis auf das DGM – betrug zwei Tage.

Datensatz	**in CityGML**	**in 3DCityDB**
Bäume	7d 21h 28min	19h 02min
DGM	9d 16h 06min	$\star$ + 20d 2h 27min
Flurstücke	1h	53min
Gebäude/Adressen	9h 46min	2h 57min
Gewässer	2min	30sek
Gewässerstrukturen	5sek	5sek
Parks	1min	1min
Straßen	22h 53min	13min
Zonierung	$\star$	$\star$
Gesamt	18d 23h 16min	21d 01h 34min

Tabelle 4.4: Übersicht über die Rechenzeiten der einzelnen Datensätze. Die zweite Spalte gibt die Berechnungszeit zur Generierung der CityGML-Datei und die dritte Spalte die Importzeit in die 3DCityDB an. Der $\star$ ersetzt sowohl den Wert für die Aktivierung der Indizes des DGMs als auch denjenigen für die Rechenzeiten der Zonierung, da diese bis zum Abgabedatum der vorliegenden Arbeit nicht vollständig berechnet werden konnten.

4.3.2 Allgemeiner Ablauf der Datenbearbeitung

Die Software FME und die 3DCityDB, mit der die oben genannten Datenmengen bearbeitet bzw. gespeichert und verwaltet werden, wurden bereits in Kapitel 2.3 beschrieben. Im Folgenden wird der für alle Datensätze gültige Ablauf erläutert (vgl. Abbildung 4.2).

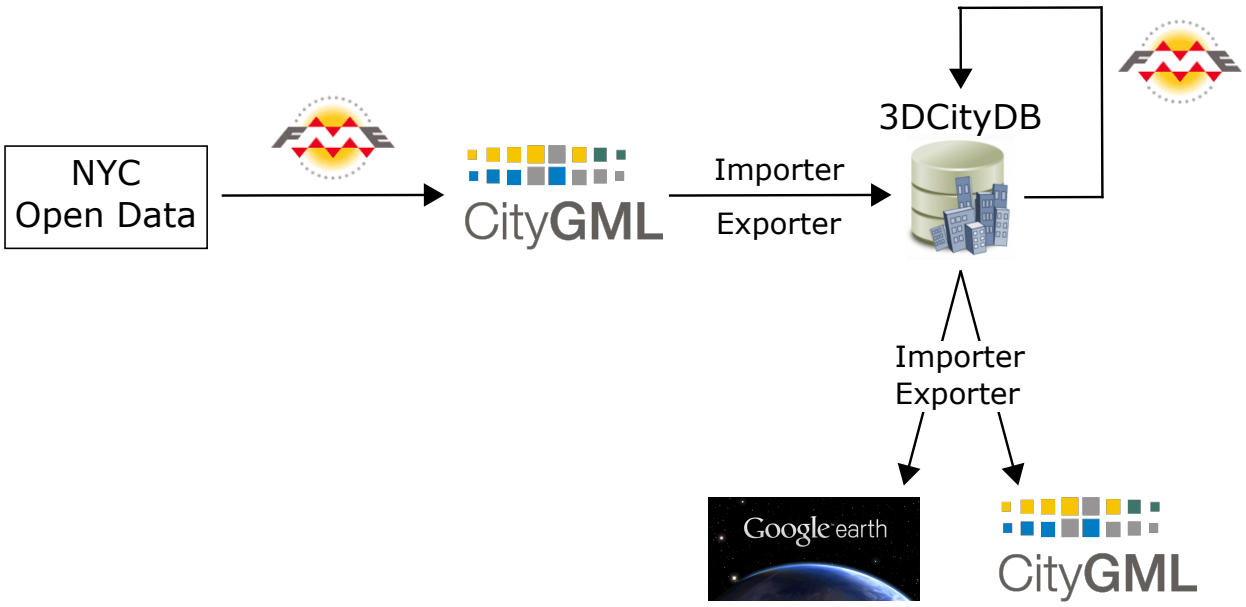


Abbildung 4.2: Allgemeines Ablaufdiagramm der Datenbearbeitung.

Die Daten werden von der Stadt New York auf dem *NYC Open Data Portal* bereit gestellt. Die für diese Arbeite verwendeten Datensätze stehen in verschiedenen Formaten zur Verfügung (vgl. Tabelle 4.3) und werden im ersten Schritt jeweils in das CityGML-Format transformiert. Diese Anpassung sowohl der Geometrie als auch der Semantik, v. a. der Attribute, erfolgt mit Hilfe der Software FME.

Dabei wird u. a. jedem Datensatz eine *External Reference* übergeben, um die Herkunft der einzelnen Objekte zu dokumentieren. Dementsprechend verweist das Informationssystem auf die Internetadresse des Originaldatensatzes und der Name setzt sich wie folgt zusammen: Originaldatensatzname_[Downloaddatum]_ID-Originaldaten_ID-Wert.

Je nach Datensatzgröße und Rechenaufwand kann die zuvor genannte Überführung vollständig oder nur auf gekachelte Weise erfolgen. Letzteres bedeutet, dass der Datensatz in sogenannte Kacheln, d.h. kleinere Einheiten, zerlegt wird, die anschließend jeweils einzeln die Transformation durchlaufen, wodurch v. a. der Arbeitsspeicher des Computers entlastet wird. Dadurch ergibt sich schließlich eine einzelne oder mehrere gekachelte CityGML-Datei/-en für jeden Datensatz.

Diese validen CityGML-Dateien werden anschließend unter Verwendung des Importer/Exporter in die 3DCityDB importiert. Vorteile der Datenhaltung in der DB liegen darin, die große Datenmenge effizient zu speichern und zu verwalten, um somit einen schnellen Zugriff auf einzelne Daten zu ermöglichen [Yao und Kolbe, 2014].

Da die Daten schließlich noch ihre Höheninformation vom Geländemodell übertragen bekommen und letzteres definitiv zu groß ist, um im Ganzen bearbeitet zu werden, wird mit Hilfe von FME

auf die Daten in der 3DCityDB zugegriffen. Dadurch können einzelne Ausschnitte nacheinander aus der DB abgefragt, verarbeitet und wiederum zurück in die DB gespeichert werden. Dabei handelt es sich überwiegend um Geometrieinformationen.

Liegen letztendlich alle Datensätze vollständig und endgültig in der 3DCityDB vor, können die Daten unter erneuter Verwendung des Importer/Exporter aus der DB exportiert werden. Dabei stehen zwei verschiedene Formate zur Verfügung, CityGML und KML.

Mit dem zuerst genannten könnte eine einzige große CityGML-Datei mit allen Daten für die gesamte Stadt NY erzeugt werden. Da bisher jedoch keine Software solch große Datenmengen bewältigen kann, wird zur besseren Handhabbarkeit lediglich für jeden thematischen Datensatz einzeln eine CityGML-Datei für die gesamte Stadt exportiert. Diese dienen v. a. zum Austausch oder zur Repräsentation des 3D-Stadtmodells.

Weiterhin werden KML-Dateien, welche die geometrischen Informationen enthalten, und dazugehörige *Spreadsheets*, die die Sachinformationen speichern, zur Visualisierung der Daten in einem virtuellen Globus wie Google Earth oder zur Nutzung des 3D-Webclients exportiert. Mehr dazu im Kapitel *Visualisierung der Ergebnisse*.

5 Digitales Geländemodell

In dieser Masterarbeit wird ein digitales Geländemodell von der gesamten Stadt New York erzeugt. Für die digitale Repräsentation des Geländes gibt es verschiedene Modelle, u. a. digitale Höhenmodelle, digitale Geländemodelle oder digitale Oberflächenmodelle. Alle drei stellen die Topographie dar.

Unter einem *digitalen Höhenmodell* (DHM) bzw. *digital elevation model* (DEM) ist ein Modell zu verstehen, das die Höhe z. B. der Erdoberfläche beschreibt. Dazu werden die Höhenwerte als Funktion der Lage der Rasterpunkte gespeichert [Kahmen, 2006], [Pfeifer, 2003]. Für New York liegt ein DGM in Form eines *Erdas Bildes* vor. Aus diesem wird ein digitales Geländemodell in Form eines TIN (Triangulated Irregular Network) berechnet.

Das *digitale Geländemodell* (DGM) bzw. *digital terrain model* (DTM) beschreibt unter Verwendung eines DHM die Höhen der Erdoberfläche. Für das DGM ist im Vergleich zum DHM explizit festgelegt, dass Geländehöhen modelliert werden. Zusätzlich kann das DGM Geländekanten und Strukturlinien enthalten [Pfeifer, 2003].

Das *digitale Oberflächenmodell* (DOM) bzw. *digital surface model* (DSM) unterscheidet sich vom DGM darin, dass nicht nur das Gelände dargestellt wird, sondern auch Objekte die über der Erdoberfläche liegen, wie Gebäude oder Vegetation. Über „offenem“ Gelände entspricht das DOM dem DGM [Pfeifer, 2003].

Das Stadtmodell für New York wird im dreidimensionalen Raum generiert. Die hierfür verwendeten Daten enthalten lediglich Lagekoordinaten, aber keine Höhenangaben. Die Höheninformationen für die Objekte werden aus dem, von New York zur Verfügung gestellten, digitalen Geländemodell[1] entnommen. Aus letzterem wird zunächst ein TIN generiert, um anschließend die Höhenkomponente der jeweiligen Daten durch Überlagerung mit dem DGM zu bestimmen.

Im Folgenden wird zunächst ein Überblick über das Modul *Relief* gegeben. Dieses beschreibt die Generierung des Geländes nach den Ansprüchen des CityGML-Standards. Anschließend werden die verwendeten Daten und die Umsetzung des Geländemodells mit der Software FME erläutert.

5.1 Definition in CityGML

Das Gelände wird in CityGML durch das Modul *Relief* beschrieben (vgl. Abbildung 5.1) und durch die Klasse *ReliefFeature* repräsentiert. Diese besteht aus einem oder mehreren Objekten der Klasse *_ReliefComponent*. Die Klassen *ReliefFeature* und *_ReliefComponent* sind Kindelemente der *_CityObject*–Klasse und bekommen von dieser Attribute und Relationen vererbt (siehe Kapitel 2.2).

[1] https://data.cityofnewyork.us/City-Government/1-foot-Digital-Elevation-Model-DEM-/dpc8-z3jc (15.10.2014)

In CityGML besteht die Möglichkeit aus Massenpunkten, Bruchkanten, regelmäßigen Gittern oder aus triangulierten Dreiecken (TIN) Geländemodelle zu erzeugen. Die einzelnen Arten können beliebig miteinander kombiniert werden. Durch die Kombination von Geländemodellen verschiedener LoDs, die verschiedene Auflösungen oder Genauigkeiten ausdrücken, können Gebiete detaillierter dargestellt werden. Ebenso können verschiedene Arten von Geländemodellen benachbarte oder die selben Regionen beschreiben. Bei Letzterem muss die Geometrie der verschiedenen Geländemodelle zusammenpassen, z. B. müssen die Bruchkanten des *BreaklineRelief* mit den Kanten der Dreiecke eines TINs übereinstimmen. Damit eine Kombination verschiedener Geländemodelle möglich ist, wird für jedes ein *extent of validity* Polygon angelegt. Dieses gibt die räumliche Ausdehnung der einzelnen Geländemodelle an. Des Weiteren besteht die Möglichkeit das Geländemodell passgenauer zu gestalten, indem es an Objekte wie Gebäude oder Brücken angepasst wird. Dazu werden *TerrainIntersectionCurves* verwendet, die als Bruchkanten fungieren und die Schnittlinien zwischen Gelände und Objekten darstellen [Gröger et al., 2012].

In dieser Arbeit wird das Gelände mit Hilfe von unregelmäßigen Dreiecksnetzen erzeugt. Hierzu gibt es die Möglichkeiten das Gelände explizit als Kollektion triangulierter Dreiecke in der Geometrie *TriangulatedSurfaces* zu speichern oder die 3D-Punkte der Dreiecke als *Tin*. Die Triangulation kann dann über eine *Delaunay triangulation* rekonstruiert werden.
In diesem Fall wird das erzeugte DGM von New York in der Geometrie *TriangulatedSurface* erstellt [Gröger et al., 2012].

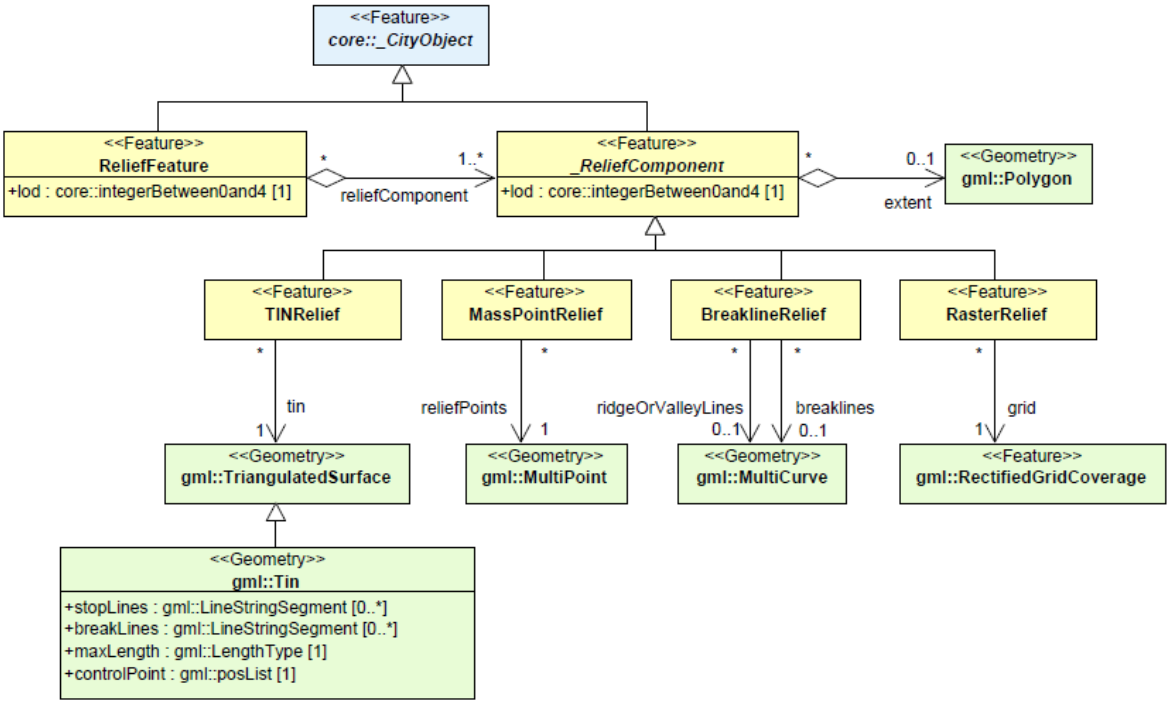
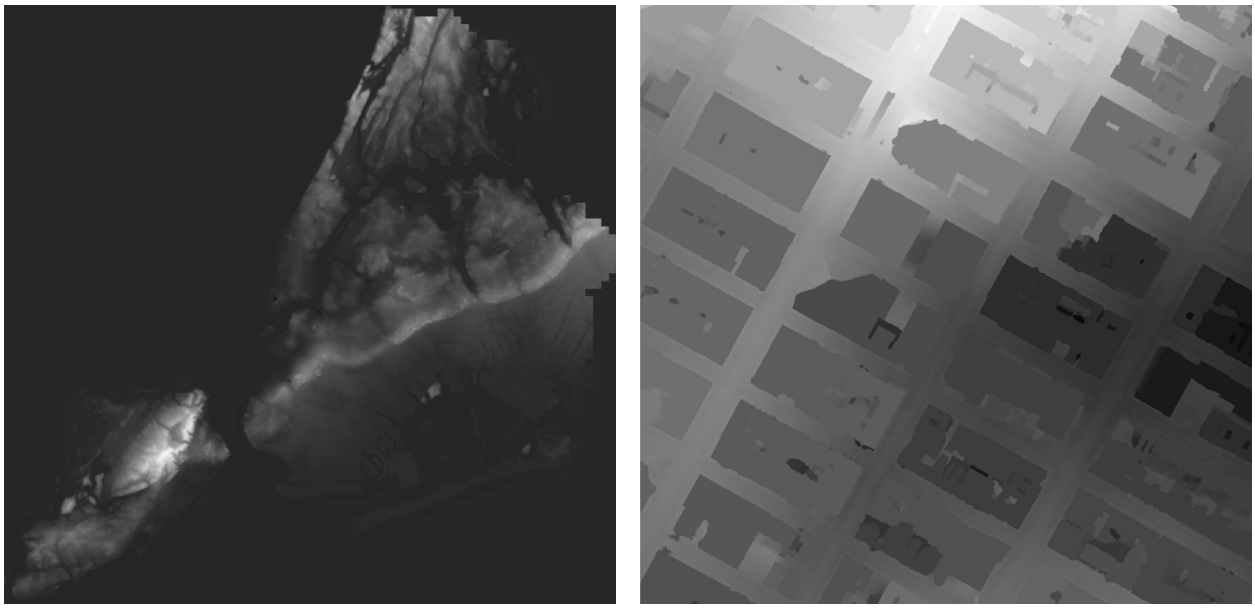


Abbildung 5.1: CityGML-UML-Diagramm des Moduls *Relief* [Gröger et al., 2012].

5.2 Verwendete Daten

Als Grundlage für das DGM wird ein bereits bearbeitetes LiDAR-Bild[1] verwendet. Dieses wird von dem *Department of Environmental Protection* (DEP) und dem *Department of Information Technology and Telecommunications* (DoITT) zur Verfügung gestellt. Die Daten stammen aus einer Laserscan-Befliegung über New York aus dem Jahr 2010.

Die Funktionsweise von LiDAR (Light Detection and Ranging) beruht auf dem Aussenden von Lichtimpulsen und dem Empfangen der Reflexionen. Über die Laufzeit des Lichtimpulses wird die Entfernung zwischen Sensor und Oberfläche bestimmt. Zusätzlich wird noch die Position der Sende- und Empfangseinheit mittels GPS und IMU (= Inertiale Messeinheit) ermittelt [de Lange, 2013]. Das Ergebnis eines Laserscannings ist eine 3D-Punktwolke, die einem DOM entspricht, da diese zusätzlich zur Geländeoberfläche Objekte enthält, die über der Geländeoberfläche liegen. Solche Objekte, die nicht dem Gelände entsprechen, können mit Filterungen eliminiert werden. Des Weiteren kommt es zu Messausfällen und Fehlern über Wasserflächen. Trifft ein Laserstrahl auf die Wasseroberfläche, ist die Absorbtion hoch und zusätzlich ist die Rauhigkeit der Wasseroberfläche im Vergleich zu der Wellenlänge des verwendeten Lasers gering, wodurch eine gerichtete Reflexion auftritt. Dies hat zur Folge, dass die geringe Strahlung die reflektiert wird, nicht in Richtung des Empfängers zurückstrahlt und somit nicht aufgezeichnet werden kann [Antonarakis et al., 2008].



(a) gesamt DGM (b) Ausschnitt des DGMs

Abbildung 5.2: Digitales Geländemodell von New York City.

Das DGM für New York liegt als 120 GB großes Erdas Bild (vgl. Abbildung 5.2 (a)) vor und umfasst ca. eine Fläche von 2 293 km^2. In Abbildung 5.2 (b) ist ein Ausschnitt des Rasterbildes dargestellt, in dem die Flächen der Straßen deutlich erkennbar sind. Insgesamt besteht das Rasterbild aus 158 100 × 156 100 Pixel, wobei jedes Pixel einer Bodenfläche von 1×1 Fuß entspricht und den entsprechenden Höhenwert beinhaltet. Um eine durchgehende Auflösung von einem Fuß zu erhalten, wurden die Bodenpunkte der LiDAR-Messung interpoliert. Die Höhenwerte der Bodenpunkte sind in Fuß über der Meeresfläche gegeben, wobei der Pegel von *Father's Point, Rimouski, Quebec*

als Bezugspunkt für die Höhenwerte des NAVD88 dient.[2] Der Meeresspiegel um New York City schwankt laut den LiDAR-Aufnahmen um mehrere Fuß. Die Genauigkeit der nicht interpolierten LiDAR-Werte liegt in der Lage bei 33,08 cm und in der Höhe bei 9,50 cm. Durch die Interpolation können diese Werte variieren.[1]

5.3 Erzeugung des CityGML-DGMs

Im Folgenden wird die Generierung des CityGML-Geländemodells für die gesamte Stadt New York in vier Schritten erläutert. Abbildung 5.3 gibt einen Überblick über den Ablauf der TIN-Berechnung. Dieser ist anhand der FME-Workspaces dargestellt. Die Datenbank spielt bei der Berechnung eine zentrale Rolle.

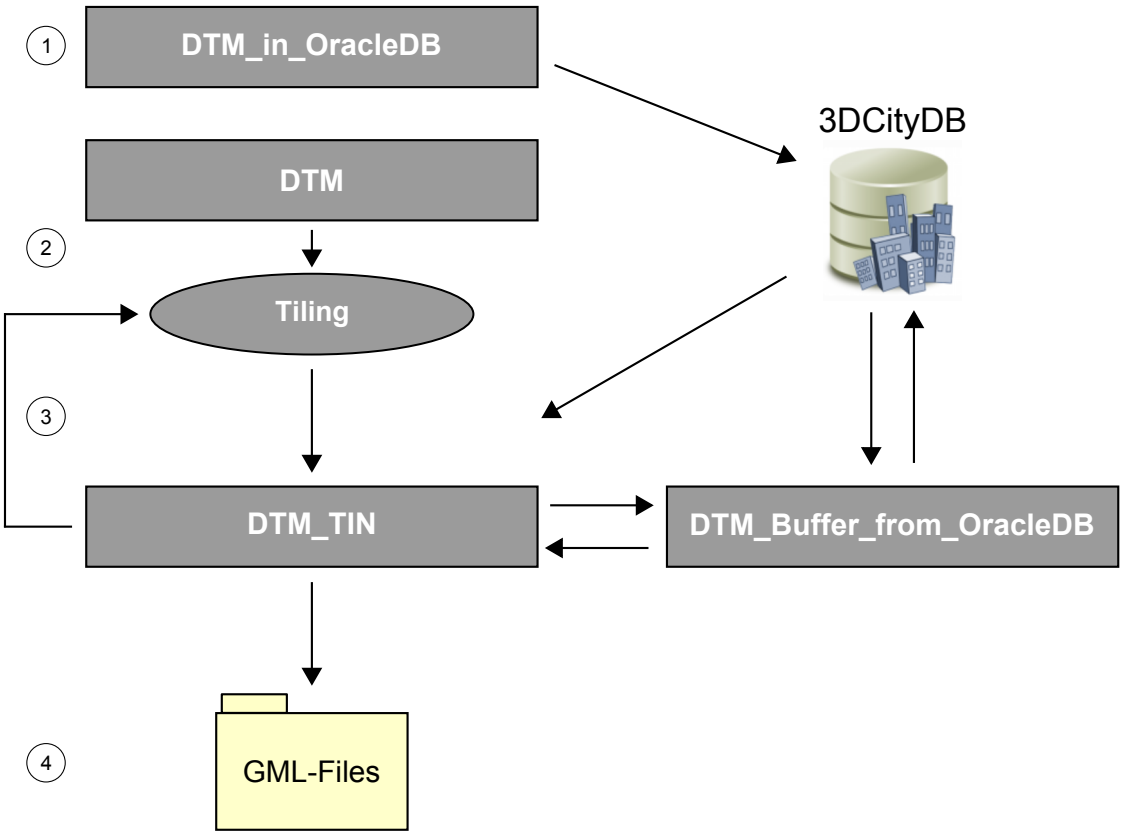


Abbildung 5.3: Übersicht des Ablaufs zur Generierung des DGMs: Die Rechtecke stellen die einzelnen FME-Workspaces und die Ellipsen einen Custom Transformer dar. Die Zahlen geben die Reihenfolge des Ablaufs an.

Schritt 1: Speicherung des Rasterbilds in DB

Das Importer/Exporter-Tool der 3D-CityDB unterstützt keinen Import von rasterbasierten DHMs in die Datenbank [Kolbe et al., 2015]. Aus diesem Grund wird folgendes mit Hilfe der Software FME durchgeführt. In dem Workspace *DTM_in_OracleDB* wird das in Kapitel 5.2 beschriebene Rasterbild in die Oracle-Datenbank gespeichert. Dazu wird für das Raster zunächst eine Tabelle u. a. mit einer Spalte vom Typ *SDO_GEORASTER* angelegt. Zusätzlich wird die benötigte *Raster*

[2] http://georepository.com/crs_5703/NAVD88-height.html [05.03.2015]

Data Table (RDT) erstellt. Diese speichert die Informationen blockweise für jedes *GeoRaster Object* als BLOB (binary large object).[3] Für die Generierung des Geländemodells wird jeweils eine solche Tabellenkombination benötigt um das gesamte Rasterbild, das gekachelte Rasterbild und um Zwischenergebnisse der Berechnung zu speichern.

Sind die Tabellen angelegt, wird das Rasterbild in die Datenbank importiert. Vorher werden die Rasterbilder noch in das verwendete Koordinatensystem transformiert (siehe Kapitel 4.2) und die Auflösung von einem Fuß auf einen Meter reduziert. Die Reduktion der Auflösung verringert den Speicherbedarf wie auch spätere Rechenzeiten. Da das DGM nicht als gesamtes für New York erstellt werden kann, wird es in 35 153 Kacheln (250×250 Meter) aufgeteilt.

Somit liegen die Rasterbilder, als gesamtes und gekachelt, in der DB in den neu angelegten Tabellen vor.

Schritt 2: Aufruf der eigentlichen Berechnungen mit Hilfe einer Schleife

In dem nächsten Schritt ermittelt der Workspace *DTM* die Anzahl der Kacheln des Rasterbilds in der Datenbank. Über eine Schleife wird jeweils für eine Kachel die Berechnung für die insgesamt 35 153 DHM-Kacheln mit Hilfe eines Workspacerunners nacheinander aufgerufen. Die Generierung der DGM-Kacheln läuft in einem weiteren Workspace ab.

Schritt 3: Berechnung des TINs aus dem DHM

Die eigentliche Berechnung des Geländemodells erfolgt in dem Workspace *DTM_TIN*. Dazu wird die jeweilig zu berechnende Kachel aus der Datenbank ausgelesen. Um das DGM aus dem DHM zu berechnen, werden die Pixel in Punkte umgewandelt und zusätzlich die Höhenwerte von Fuß in Meter umgerechnet, da das Koordinatensystem des gesamten Modells in Metern vorliegen soll (vgl. Kapitel 4.2).

Ziel ist es, dass alle Kacheln ohne einen Höhenversatz aneinander anschließen. Dafür wurden verschiedene Ansätze überlegt und der Folgende letztendlich implementiert. Damit die Kacheln an den Rändern homogen in einander übergehen, fließen die Ränder als Bruchkanten bei der TIN-Erzeugung ein. In Abbildung 5.4 ist ein Schema für den Randbereich einer DHM-Kachel dargestellt. Die verschieden grau eingefärbten Rechtecke entsprechen den Bodenpixel an der Kachelgrenze (schwarzes Rechteck). Die Pixel, die die Kachelgrenze von innen her berühren (blaue Punkte), sowie die Pixel, die an die äußere Grenze der Kachel anliegen (grüne Punkte), werden bestimmt. Zusätzlich werden neue Punkte (rot) erzeugt, die exakt in der Mitte der „inneren Pixel“ und der „umliegenden Pixel“ und somit auf der Kachelgrenze liegen. Diese werden als Bruchkanten angeführt, wodurch sie bei der Interpolation nicht verändert werden. Der Vorteil dieser neu generierten Grenzpunkte ist, dass diese für alle Kacheln identisch sind.

Zunächst wird die äußerste Pixelreihe einer Kachel extrahiert (blaue Punkte). Um die „umliegende Pixelreihe“ einer Kachel zu generieren (grüne Punkte), wird ein Ausschnitt des gesamten DHMs mit Hilfe einer Boundingbox aus der Datenbank ausgelesen, die genau ein Pixel größer ist, als die zu berechnende Kachel. Dies geschieht mit einem weiteren Workspacerunner, der den Workspace *DTM_Buffer_from_OracleDB* aufruft. Diese gepufferte Kachel wird in der DB als Zwischenergebnis

[3] http://docs.oracle.com/cd/B28359_01/appdev.111/b28398/geor_operations.htm#CIHCHDFF (23.06.2015)

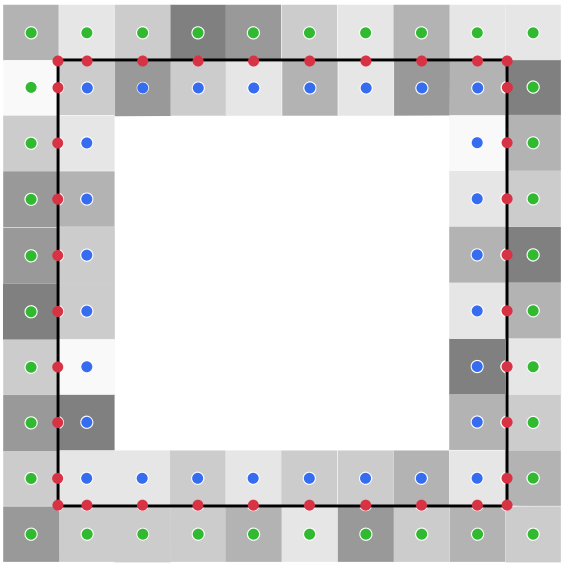


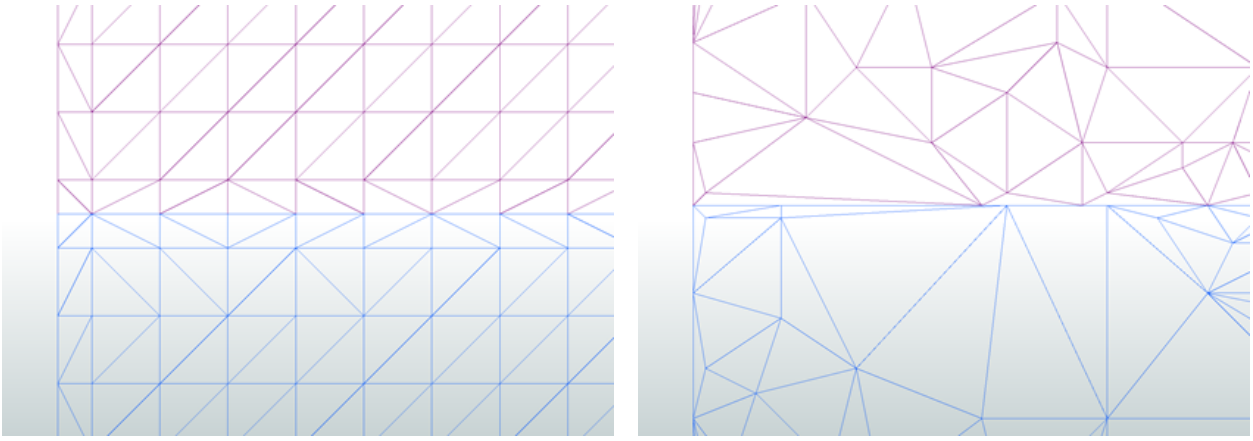
Abbildung 5.4: Schematische Darstellung der Kachelgrenzen; *blau* markiert die äußersten Pixel/Punkte der zu berechnenden Kachel; *grün* markiert die umliegenden Pixel/Punkte der zu berechnenden Kachel; *rot* stellt die neu erzeugten Punkte der Kachelgrenze (schwarzer Rahmen) dar

gespeichert und durch die gepufferte Kachel der nachfolgenden Kachel überschrieben, damit nicht erneut alle Kacheln abgespeichert werden. Dies ist nötig, da Ergebnisse zwischen verschiedenen Workspaces in FME nur genutzt werden können, wenn diese lokal oder in einer DB gespeichert sind und dadurch wieder aufgerufen werden können. Mit der Verschneidung der gepufferten Kachel mit der zu berechnenden Kachel werden die „umliegenden Pixel“ bestimmt.

Die Pixel des inneren und des äußeren Rands werden wie die Kachel selbst in Punkte umgewandelt und die Höhenwerte von Fuß in Meter umgerechnet.

Die Lage und Höhe der Punkte (rot) der Kachelgrenze, die exakt in der Mitte der „inneren“ und der „umliegenden“ Punkte liegen, werden aus dem Mittelwert der Nachbarpunkte bestimmt. Diese markieren genau den Rand der Kachel (schwarzes Rechteck) und sind für benachbarte Kacheln identisch. Deswegen werden sie als feste Punkt, die nicht verändert werden dürfen, in die TIN-Berechnung eingebracht. Somit ist sichergestellt, dass es an den Grenzen benachbarter Kacheln keinen Höhenversatz gibt.

Das TIN wird aus den Punkten mit der Delaunay Triangulation berechnet. In FME gibt es die Option einer Punktfilterung bei der TIN-Erzeugung. Diese bewirkt eine Überprüfung der Punkte in Bezug auf das bestehende DGM und eliminiert diese, falls sie keine neuen Informationen für das bestehende DGM enthalten. Dies ist von Vorteil bei großen bis sehr großen Punktmengen, da so überflüssige Punkte bei einer TIN-Berechnung ausgelassen werden und die Berechnung optimiert abläuft. Eine Filterung hat allerdings zur Folge, dass die Struktur des DGMs unregelmäßig wird, wodurch in diesem Fall die Grenzpunkte benachbarter Kacheln nicht identisch wären (vgl. Abbildung 5.5 (b)). Unter solchen Bedingungen ist es unmöglich, einen homogenen Übergang der verschiedenen Kacheln zu generieren. Für die Berechnung wird demzufolge die Punktfilterung ausgestellt. Somit bleiben die Punkte gleichmäßig erhalten (vgl. Abbildung 5.5 (a)) und ein homogener Übergang an



(a) regelmäßiges Gitter (b) unregelmäßiges Gitter

Abbildung 5.5: Darstellung der triangulierten Punkte an der Kachelgrenze zwischen lila und blauen Bereich; (a) alle Punkte gehen in die TIN-Generierung ein; identische Punkte an dem Kachelübergang; (b) gefilterte Punkte gehen in die TIN-Generierung ein; Punkte der Kachelgrenzen stimmen nicht überein.

den Kachelgrenzen ist gewährleistet.[4]

Schritt 4: Speicherung im CityGML-Format

Zum Schluss wird das TIN in CityGML geschrieben. Dazu werden zusätzlich zur Klasse *TINRelief* die Klassen *ReliefFeature* und *CityModel* geschrieben, um ein vollständiges CityGML-Dokument zu erhalten, welches lokal gespeichert wird. Am Ende der Berechnung existiert für jede DGM-Kachel ein eigenes TIN.

Um aus den 35 153 Geländemodellen ein Geländemodell zu erhalten und um mit diesem weiterzuarbeiten, werden die einzelnen Dokumente in die Oracle-Datenbank importiert. Dazu wird das Importer/Exporter Tool verwendet. Dieses liest die Dokumente ein und speichert die Informationen automatisch in die richtigen Tabellen des CityGML-Schemas ab. Für weitere Berechnungen können kleine Gebiete des Geländemodells aus der Datenbank gelesen werden und die Höheninformationen verwendet werden.

Das Ziel, ein DGM für das gesamte Stadtgebiet von New York im CityGML-Format zu generieren, wird umgesetzt. Dafür werden 35 153 Kacheln mit jeweils ca. 150 000 Dreiecken berechnet. Insgesamt werden für das gesamte DGM 5 413 934 270 Dreiecke erzeugt. Abgespeichert in GML benötigen die Dokumente 1,45 TB. Die lange Berechnungszeit von nahezu 10 Tagen könnte verkürzt werden, indem Kacheln, die lediglich Nullwerte enthalten, bei der TIN-Generierung unberücksichtigt bleiben. Ca. die Hälfte (vg.l Kapitel 4.3.1) der Fläche des Rasterbildes enthält vermutlich keine Höheninformation. Es ist ebenfalls in Abbildung 5.2 (a) erkennbar, dass die schwarzen Bereiche in der oberen linken Hälfte und in der unteren rechten Ecke vermutlich *no data value* enthalten, d. h. keine realen Höhenwerte, wodurch diese unberücksichtigt bleiben könnten. Da allerdings Gewässer ebenfalls solche Werte aufweisen können, werden solche DGM-Kacheln trotzdem für diese Arbeit generiert.

[4] http://docs.safe.com/ (30.05.2015)

6 Gebäudemodell inklusive Adressen und Flurstücke

6.1 Definition in CityGML

Im Folgenden wird auf das CityGML-Modul *Building* eingegangen. Die Modellierung der Gebäude, die Anreicherung dieser mit zusätzlichen Attributen und die Übertragung der Höhe aus dem DGM wird erläutert. Des Weiteren werden die Flurstücke in diesem Kapitel behandelt.

6.1.1 Gebäude und Adressen im Modul Building

Gebäude

Das *Building* Modul (siehe Abbildung 6.1) ist von allen thematischen Modulen in CityGML am komplexesten. Es ermöglicht die Darstellung von Gebäuden und Gebäudeteilen in den LoD 0 bis LoD 4 [Gröger et al., 2012].

Die zentrale Klasse des Moduls ist *_AbstractBuilding*. Sie ist eine Unterklasse der abstrakten Klassen *_Site* und *_CityObject*. Wie schon in Kapitel 2.2 auf Seite 15 beschrieben, bekommt sie die Attribute und Relationen ihrer Oberklassen vererbt und gibt diese auch an ihre Unterklassen *Building* und *BuildingPart* weiter. Über die Aggregations-Hierarchie kann ein *_AbstractBuilding* aus beliebig vielen *BuildingParts* bestehen. Zusammengehörige Gebäudekomplexe wie Fabrikgebäude, die aus mehreren einzelnen Gebäuden bestehen, werden jedoch über *CityObjectGroups* zusammengefasst [Gröger et al., 2012], [SIG3D, 2014].

Mit dem LoD steigt der Detaillierungsgrad der Gebäude. Bei dem LoD 0 des *Building* Moduls wird der Gebäudeumriss in 3D als *FootPrint* und die Dachaußenkante als *RoofEdge* mit absoluter Höhe als horizontale Polygone in der Geometrie *MultiSurfaceType* dargestellt. Die äußere Form eines Gebäudes wird in dem LoD 1 dargestellt. Hierbei handelt es sich um einen Extrusionskörper mit horizontalen Boden- und Dachflächen und vertikalen Wandflächen. Es besteht die Möglichkeit, Gebäude in den Geometrien *Solid* und/oder *MultiSurface* zu speichern. Für die LoDs 1 bis 4 besteht die Option, Gebäude-Gelände-Schnittlinien zu dem jeweiligen Gebäude unter dem Attribut *lodXterrainIntersection* als *MultiCurve* zu speichern. Diese können aus verschiedenen nicht zusammenhängenden Liniensegmenten bestehen und müssen nicht geschlossen sein. Hierbei werden außenliegende Gebäudeinstallationen (*BuildingInstallation*) wie Balkone oder Außentreppen, berücksichtigt. Die *TerrainIntersectionCurve* hat keinen Einfluss auf das DGM in Auflösung und Genauigkeit, viel mehr ist es das Ziel, dass das DGM direkt an die *TerrainIntersectionCurve* anschließt.

Ab dem LoD 2 werden architektonische Details dargestellt. Dies können zum Beispiel prototypische Dachformen und Gebäudeinstallationen sein. Laut [SIG3D, 2014] wird empfohlen, Gebäude vorzugsweise in der Geometrie *Solid*, und nicht als *MultiSurface*, abzuspeichern. Die *BoundarySurface*

mit der Geometrie *MultiSurface* kann über *XLinks* mit dem Volumenkörper referenziert werden.

Für die Begrenzungsfläche gibt es die folgenden unterschiedenen semantischen Objekte:

- RoofSurface
- WallSurface
- GroundSurface
- ClosureSurface
- CeilingSurface
- InteriorWallSurface
- FloorSurface
- OuterCeilingSurface
- OuterFloorSurface

Als Gebäudeinstallationen (*BuildingInstallation*) werden Gebäudeteile der Außenhülle, dazu zählen u. a. Balkone, Schornsteine und Außentreppen, repräsentiert. Sie können eine beliebige Geometrie vom Typ *Geometry* haben. Das LoD 3 gibt die höchste Detaillierungsstufe der Außenhülle eines Gebäudes wieder. Es werden zusätzlich reale Dachformen, Türen und Fenster dargestellt. Letztere werden als Unterklassen, *Window* und *Door*, der Klasse *_Opening* gespeichert. Innenräume von Gebäuden werden im Detaillierungsgrad LoD 4 realisiert. Dazu werden die zusätzlichen Klassen *Room*, *IntBuildingInstallation* und *BuildingFurniture* verwendet [Gröger et al., 2012], [SIG3D, 2014].

Zusätzlich benötigte Attribute werden wie in allen thematischen Modulen als generische Attribute oder als ADEs gespeichert [Gröger et al., 2012].

Adressen
Gebäuden (Klasse *Building*), Gebäudeteilen (Klasse *BuildingPart*) und Türen (Klasse *Door* erst ab LoD 3 verfügbar), können nach dem CityGML-Standard beliebig viele Adressen zugewiesen werden (vgl. Abbildung 6.1). Die *Address*-Objekte können optional mit einem Punkt verortet werden. Dieser kann sowohl in 2D als auch 3D in der Geometrie *multiPoint* vorliegen und dient dazu, die Position des Eingangs zu lokalisieren. Das generische Schema wird von dem OASIS Consortium bereitgestellt. Die Adressen können direkt im Gebäude-Objekt oder in einem eigenständigen Dokument gespeichert werden. In letzterem Fall wird von den Gebäude- bzw. Tür- Objekten per *XLink* auf die Adresse referenziert [Gröger et al., 2012], [SIG3D, 2014], [xAL Standard, 2002].

6.1.2 Flurstücke im Modul LandUse

Die thematische Modellierung, die für die Repräsentation der Flurstücke verwendet wird, ist das Modul *LandUse* [Gröger et al., 2012]. Für eine detaillierte Beschreibung des Moduls sei auf Kapitel 9.1 verwiesen.

6.2 Verwendete Daten

6.2.1 Gebäude

Der Datensatz *Building*[1] ist im WGS84 Koordinatensystem gegeben (siehe Kapitel 4.2). Er wird vom *Department of Information Technology and Telecommunications* (DoITT) zur Verfügung gestellt. Die enthaltenen Gebäude liegen als Grundrisspolygone aus dem Jahr 2014 vor. Insgesamt sind 1 082 005 Gebäudepolygone in diesem Datensatz enthalten. In der Regel werden nur Gebäude erfasst, deren Grundfläche über 500 Quadratfuß und deren Höhe zusätzlich über 12 Fuß beträgt

[1] https://data.cityofnewyork.us/Housing-Development/Building-Footprints/tb92-6tj8(14.10.2014)

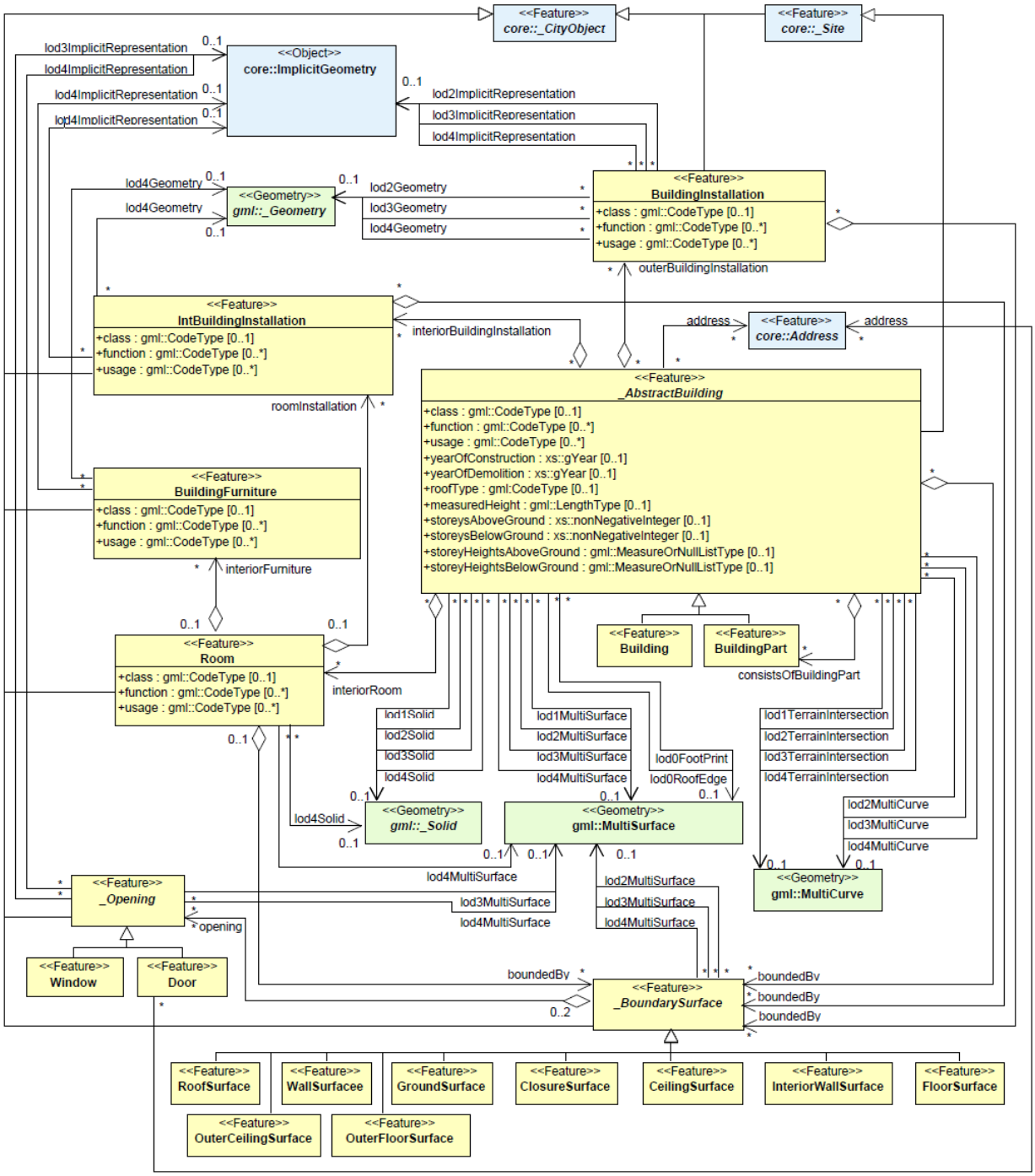


Abbildung 6.1: CityGML-UML-Diagramm des Moduls *Building* [Gröger et al., 2012]

oder die eine *Building Identification Number* (BIN) haben. Der Grundriss gibt die äußere Gebäudekante ohne Überhänge an. Allerdings werden Carports, die an die Hauptgebäude anschließen, mit in den Grundriss aufgenommen. Garagen, die als solche erkannt werden, sind mit im Datensatz enthalten, auch wenn sie keine BIN und nicht die entsprechende Größe haben. In diesem Fall wird eine einheitliche Garagen-BIN, je nach Stadtteil, vergeben. 95% der erhobenen Gebäude liegen mit einer Lagegenauigkeit von 2 Fuß vor .
In Tabelle 6.1 werden die Attribute des Datensatzes der Gebäude dargestellt, die in das New Yorker Stadtmodell übernommen werden sollen. Im Anhang B.1 ist ein Überblick der Originalattribute und deren Bedeutung gegeben.

6.2.2 Adressen

Der Datensatz *AddressPoints*[2] wird vom *Department of Information Technology and Telecommunications* (DoITT) zur Verfügung gestellt. Die Lage der Punkte der Adressen ist so gewählt, dass sie ca. 5 Fuß innerhalb der Gebäude in Richtung der zugehörigen Straßenfront liegen, somit aber nicht den Eingang bzw. die Tür symbolisieren. Die Adressen aus dem Datensatz *AddressPoints* werden über die jeweilige BIN mit den Daten der zugehörigen *Buildings* verknüpft. Dabei ist es möglich, dass ein Gebäude mehrere Adressen hat. Allerdings gibt es in den Datensätzen 1 082 005 Gebäude und lediglich 941 526 Adressen. Von den 1 082 005 Gebäuden sind insgesammt 215 080 dem Gebäudetyp Garage zugeordnet, welche keine eigene Adresse haben. Somit erhalten mindestens 119 399 Häuser keine Adresse aus dem Datensatz. Der Datensatz der Adressen beinhaltet Straßennamen, Hausnummer und die Postleitzahl als Attribute.

6.2.3 Flurstücke

Die Flurstücke[3] aus der sogenannten *Pluto-Datenbank* werden durch die *Department of Finance's Digital Tax Map* (DFDTM) bereitgestellt. Der Datensatz gliedert sich in zwei Bereiche. Zum Einen in die Datei *MapPLUTO*, welche die Flurstücke (*tax lots*) mit zusätzlichen weitreichenden Informationen enthält. Zum Anderen die Datei *MappingLot*, wobei es sich um Grundstücke, die nicht in der Steuerkarte enthalten sind, beispielsweise Grünanlagen handelt. Diese dienen lediglich einer vollständigen kartographischen Darstellung aller Flurstücke. Sie enthalten nur Informationen über Borough, Block und Lot Nummer (BBL). Beide Bereiche liegen als zweidimensionale Polygone in dem NAD83 Koordinatenreferenzsystem vor. Die Bedeutung der einzelnen Attribute sowie deren Codierung kann im *PLUTO DATA DICTIONARY* [4] nachgelesen werden. Zum Beispiel sind mit den Attributen *owner_name* und *owner_type* der Name des Eigentümers (hierbei handelt es sich meist um juristische Personen) und die Art des Eigentums, wie städtisches, angegeben. Auch ist die Anzahl der Grunddienstbarkeiten eines Grundstücks oder eine mögliche Zusammenlegung, Teilung oder Umformung eines Grundstücks und deren Zeitpunkt, in den Attributen aufgelistet. Die verwendeten Attribute sind in Tabelle 6.4 aufgelistet.

[2] https://data.cityofnewyork.us/City-Government/NYC-Address-Points/4iq4-tuhq (13.10.2014)
[3] http://www.nyc.gov/html/dcp/html/bytes/dwn_pluto_mappluto.shtml (13.10.2015)
[4] http://www.nyc.gov/html/dcp/pdf/bytes/pluto_datadictionary.pdf?r=1 (06.07.2015)

6.3 Umsetzung

6.3.1 Gebäude und Adressen

Zuerst werden die Gebäude in das benötigte Koordinatensystem NAD83 für die Lage und NAVD88 für die Höhe transformiert, wobei hier zunächst eine Höhe von 0 Metern angenommen wird (siehe Kapitel 4.2). Bei Gebäuden, die keine Gebäudehöhenwerte im Attribut *citygml_measured_height* enthalten, wird eine Standardhöhe von einem Fuß eingetragen. Dies betrifft insgesamt 5 653 Gebäude, die hauptsächlich weder den Gebäuden noch den Garagen zugeordnet werden können. Die Gebäudehöhen und die Grundhöhen werden von Fuß in Meter umgerechnet. Zusätzlich wird für die Gebäude jeweils die Grundfläche (*building_base_area*), sowie das Volumen (*building_volume*) berechnet, um weitere Nutzungen zu ermöglichen, z. B. für Energiebedarfsrechnungen und Heizbedarfsermittlungen. Die vorhandenen Attribute werden teilweise in die in CityGML vordefinierten Attribute umgeschrieben, sowie als eigene generische Attribute gespeichert. Zusätzlich enthalten die Objekte auch eine *externalReference*, die auf den originalen Datensatz sowie das originale Objekt verweist. Eine Übersicht der Attribute sowie ihrer Codierung findet sich im Anhang B.1. Für Gebäude werden somit die in Tabelle 6.1 und Tabelle 6.2 aufgelisteten Attribute im Modell gespeichert, sofern sie für ein Gebäude vorhanden sind. Dabei sind in Tabelle 6.1 die Attribte aus dem Originaldatensatz übernommen und in Tabelle 6.2 neue Attribute auf den Originaldaten erzeugt, beispielsweise wurde die Grundfläche des Gebäudes berechnet und im Attribute *building_base_area* hinterlegt. Die violett geschriebenen Attribute sind in CityGML vordefiniert, die restlichen wurden generisch erzeugt.

Tabelle 6.1: Original-Attribute der Gebäude [Notation: CityGML]

- building_identification_number
- building_volume
- borough_block_lot_number
- citygml_measured_height
- citygml_measured_height_units
- citygml_usage
- citygml_year_of_construction
- doitt_id
- geom_source
- ground_elevation
- last_modify_date
- last_status_type
- last_status_1
- source_id
- status

Tabelle 6.2: Neu erzeugte Attribute der Gebäude [Notation: CityGML]

- building_base_area
- building_volume_units
- externalReference.externalObject.name
- externalReference.informationSystem
- gml_id
- gml_name
- gml_parent_id
- ground_elevation_units

Es besteht die Möglichkeit, Gebäuden in CityGML Adressen zuzuweisen. Diese Verknüpfung erfolgt über das Attribut *building_identification_number* (BIN). Es ist in den Datensätzen der Gebäude sowie der Adressen gegeben und fungiert für die Gebäude (nicht für Garagen) als eindeutige Identifikation. Da jede Adresse nur eine BIN enthält, kann jede Adresse auch maximal einem Gebäude zugeordnet werden. Ein Gebäude hingegen kann mehrere Adressen enthalten. Die Adressen werden

intern im CityGML-Dokument der Gebäude in den entsprechenden Objektdefinitionen gespeichert. Dazu werden die Adressinformationen im *xAL*-Format in das CityGML-Attribut *citygml_address* geschrieben. Die Adressen werden zusätzlich als Punktobjekte im Stadtmodell dargestellt.
Von den insgesamt 941 526 Adressen können lediglich 938 834 Adresspunkte einem Gebäude zugeordnet werden. Die 2 692 verbleibenden Adresspunkte enthalten BINs, die zu keinem Gebäude passen. 201 713 Gebäude, also über ein Fünftel, erhält keine Adresse.

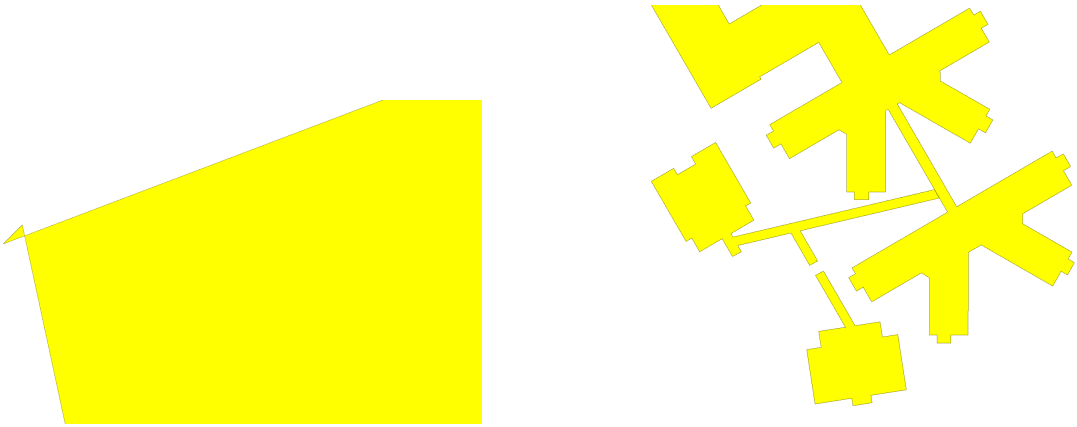
Steht nur ein Gebäude auf einem Flurstück, werden dem Gebäude noch ausgewählte Attribute aus dem Datensatz der Flurstücke[3] hinzugefügt. Die Überprüfung erfolgt unter Filterung des Attributes *number_of_buildings*, das die Anzahl der Gebäude auf dem Flurstück angibt. Es werden nur Attribute aus dem Pluto-Datensatz ausgewählt, die einen direkten Bezug zu den Gebäuden haben. Beispielsweise beschreibt das Attribut *PLUTO_condominium_number* die Anzahl der Wohnungen in einem Gebäude. Das Attribut *PLUTO_office_area* gibt die als Büro genutzte Fläche eines Gebäudes an. Da nicht alle Gebäude eine Adresse aus dem Datensatz *AddressPoints* erhalten, werden die Adressen des Flurstücks als Attribut *PLUTO_lot_address* den Gebäuden übergeben. Insgesamt erhalten so 10 700 weitere Gebäude eine Adresse, die keine Adresse von den Adresspunkten bekommen haben. In Tabelle 6.3 sind die verwendeten Attribute der Flurstücke aufgelistet. Die Bedeutung der einzelnen Attribute, sowie deren Codierung, kann im *PLUTO DATA DICTIONARY* [5] nachgelesen werden.

Tabelle 6.3: Gebäudeattribute aus dem PLUTO-Datensatz [Notation: CityGML]

- PLUTO_lot_address
- PLUTO_building_class
- PLUTO_land_use_category
- PLUTO_building_area
- PLUTO_building_area_units
- PLUTO_commercial_area
- PLUTO_commercial_area_units
- PLUTO_residential_area
- PLUTO_residential_area_units
- PLUTO_office_area
- PLUTO_office_area_units
- PLUTO_retail_area
- PLUTO_retail_area_units
- PLUTO_garage_area
- PLUTO_garage_area_units
- PLUTO_storage_area
- PLUTO_storage_area_units
- PLUTO_factory_area
- PLUTO_factory_area_units
- PLUTO_other_area
- PLUTO_other_area_units
- PLUTO_area_source
- PLUTO_number_of_floors
- PLUTO_proximity_code
- PLUTO_building_frontage
- PLUTO_building_frontage_units
- PLUTO_building_depth
- PLUTO_building_depth_units
- PLUTO_year_built
- PLUTO_year_built_code
- PLUTO_year_altered_1
- PLUTO_year_altered_2
- PLUTO_built_floor_area_ratio
- PLUTO_condominium_number
- PLUTO_units_residential

Die Gebäude liegen als Polygone vor, sollen aber als „Klötzchen“ bzw. *Solids* dargestellt werden. Dazu werden die Polygone um den Werte des Attributes *citygml_measured_height*, der Gebäudehöhe, extrudiert. Somit werden die Gebäude als *Solids* gespeichert.
Neun der Gebäudepolygone liegen als Aggregate vor. In Abbildung 6.2 sind zwei verschiedene Arten von Aggregaten dargestellt. In (a) ist zu erkennen, dass sich das Polygon selbst schneidet. In

[5] http://www.nyc.gov/html/dcp/pdf/bytes/pluto_datadictionary.pdf?r=1 (06.07.2015)



(a) (b)

Abbildung 6.2: Darstellung problematischer Polygone im Kontext der Geometrischen Primitive, hierbei handelt es sich um: (a) fehlerhaftes Polygon, schneidet sich selbst; (b) Aggregat, besteht aus zwei unzusammenhängenden Polygonen. Beide können nicht als Solids dargestellt werden

(b) ist ein Gebäude dargestellt, das aus mehreren nicht zusammenhängenden Polygonen besteht. *Solids* sind geometrische Primitive und bestehen aus *CompositSurfaces*, die sich weder selbst noch gegenseitig schneiden bzw. überlappen dürfen. Composite-Objekte stellen eine spezielle Form der komplexen Geometrie dar, da sie sich nur aus gleichartigen Geometrien zusammensetzen dürfen und geometrisch so miteinander verbunden sein müssen. Ein Composite-Objekt verhält sich wie ein Primitiv. Diese Eigenschaften würden für die extrudierten Polygone solcher Aggregat-Situationen nicht zutreffen, weshalb die Aggregate aufgelöst werden.

Im Folgenden wird erläutert, wie die Gebäude auf die Geländehöhe des DGMs gelegt werden. Ein Überblick des Ablaufs ist in Abbildung 6.3 in drei Schritten gegeben. Die Schritte 1 und 2 sind für Objekte, die lediglich aus einer Fläche (Straßen, Flurstücke, Parks, Gewässer und hydrologischen Infrastrukturen) bestehen, identisch[6]. In Schritt 3 wird explizit auf die Gebäude eingegangen. Dieser Schritt kann analog für die Bäume umgesetzt werden.

Schritt 1: Filterung der DGM-Kacheln

Da nur ein Teil der 35 153 Kacheln im Bereich von Objekten wie Straßen, Gebäuden und Gewässern liegt, werden ausschließlich diese Kacheln des DGMs aus der DB gelesen. Dazu wird in dem Workspace *DTM_tiles_for_Objects_on_DTM* eine räumliche Abfrage zwischen den Boundingboxen der DGM-Kacheln und der Objekte gemacht. Die gefilterten DGM-Kacheln werden einzeln an den folgenden Workspace übergeben.

Schritt 2: Höhenübertragung auf flächige Objekte

Im Workspace *Objects_on_DTM* werden die DGM-Kacheln mit den Objekten überlagert und diesen die Höheninformation übergeben. Dazu werden zunächst die flächenhaften Geometrien, die innerhalb einer DGM-Kachel liegen, aus der DB-Tabelle *SURFACE_GEOMETRY* gelesen. Für die

[6] Aus diesem Grund ist der eben erwähnte FME-Workspace in Zusammenarbeit mit B.Burger entstanden, auf die sich die ergänzende Arbeit bezieht.

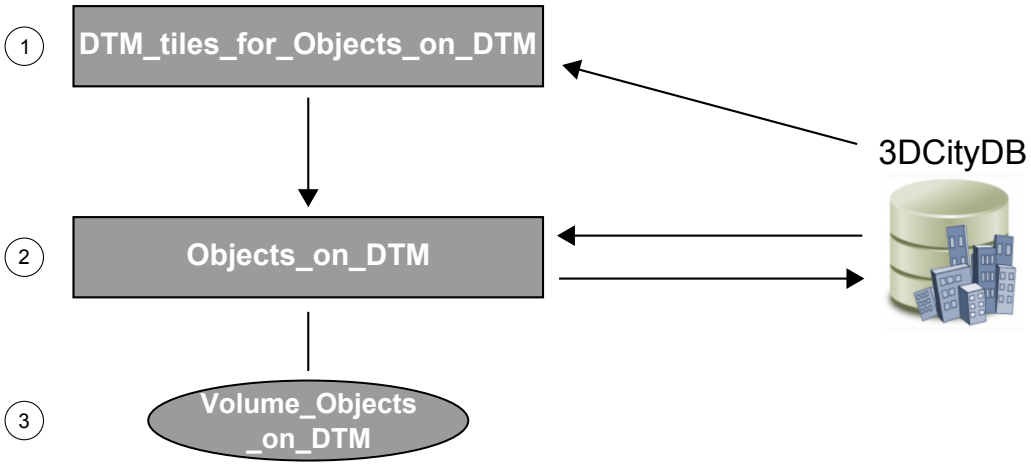
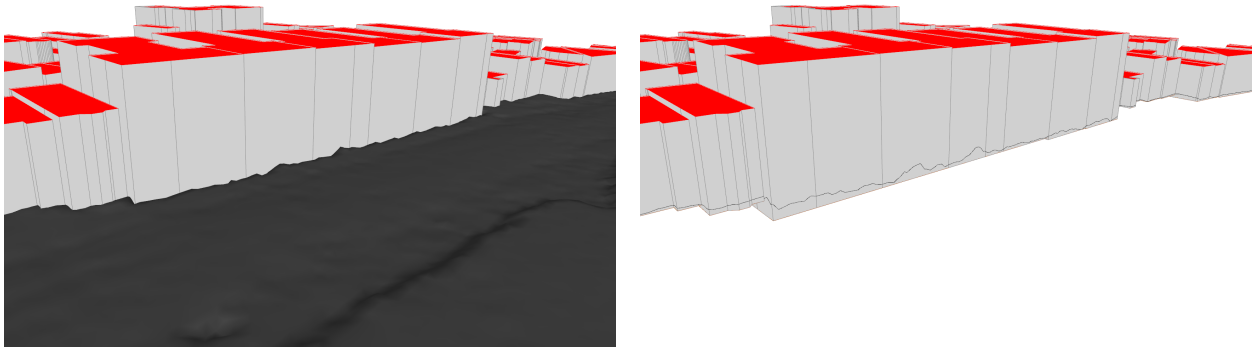


Abbildung 6.3: Ablaufdiagramm zur Übertragung der Höheninformation auf die Gebäude. Die Rechtecke stellen die einzelnen FME-Workspaces und die Ellipse einen Custom Transformer dar.

weitere Berechnung werden nur diejenigen Objekte zugelassen, deren Höhenwert 0 ist. Somit gehen bereits bearbeitete Objekte nicht wiederholt in die Berechnung ein. Auch werden nur Objekte berücksichtigt, die zumindest teilweise innerhalb der Kachel liegen. Dies ist beispielsweise bei längeren Straßen oder großen Flurstücken der Fall, die über den Bereich des DGMs hinausragen.
Da bisher ausschließlich die Boundingbox der DGM-Kachel verwendet wird, wird an dieser Stelle die Boundingbox der zugelassenen Objekte bestimmt und für diesen Bereich die einzelnen Dreiecksflächen des DGMs aus der DB gelesen. Dies ist nötig, damit die Objekte an den DGM-Kachelgrenzen nicht abgeschnitten werden oder z. T. keine Höhen übertragen bekommen, sondern alle Objekte zugelassen werden, die ganz oder teilweise in der DGM-Kachel liegen.
Anschließend werden den Objekte, die lediglich aus einer Fläche bestehen (Straßenflächen, Flurstücke, Parks, Gewässer und Gewässerstrukturen) wie auch die Grundflächen der Volumenobjekte (Gebäude und Bäume) ohne weitere Bearbeitung die Höhen des Geländemodells übertragen. Somit ist der Höhenverlauf dem Gelände angepasst und unregelmäßigen.

Schritt 3: Höhenübertragung auf Volumenobjekte

Volumenobjekte bestehen aus einer Grundfläche sowie weiteren Seiten- und Deckflächen. Die Grundflächen haben zunächst eine einheitliche Höhe von Null. Die weiteren Flächen eines Volumenkörpers variieren in ihrem Höhenwert. Auf dieser Grundlage werden in Schritt 2 (siehe oben!) ausschließlich den Grundflächen Höhen aus dem Geländemodell übertragen. Die zugehörigen Flächen werden aus der DB ausgelesen. Bei Gebäuden entsprechen diese den Wand- und Dachflächen, bei Bäumen den Baumstamm- und Baumkronenflächen. Die Grundflächen, die bereits auf das DGM gelegt wurden, erhalten anschließend ihre jeweilige minimale z-Koordinate als Höhenwert. So ist sichergestellt, dass die Objekte nicht über der Oberfläche schweben oder unter ihnen „durchgeschaut“ werden kann. Die z-Koordinaten der übrigen Flächen werden daraufhin an diese neue Höhe der Grundfläche angepasst, damit die Einzelflächen wieder ein in sich schließendes Volumenobjekt repräsentieren.
Abschließend werden die neu berechneten Geometrien der Objekte in der ursprünglichen Tabelle *SURFACE_GEOMETRY* geupdatet. Zusätzlich wird die Höhenkomponente der Boundingboxen der Objekte in der *CITYOBJECT* Tabelle aktualisiert.



(a) (b)

Abbildung 6.4: (a) Darstellung der Gebäude mit dem Gelände; (b) Darstellung der Gebäude mit der Schnittlinie (schwarze Linie) zwischen Gebäude und Gelände

Den Gebäuden wird zusätzlich die *TerrainIntersectionCurve* als Attribut übergeben. Dabei handelt es sich um die Schnittlinie der Gebäude mit dem DGM, welche berechnet und in der Tabelle *BUILDING* in der Spalte *LOD_1_TERRAIN_INTERSECTION* gespeichert wird. In Abbildung 6.4 (a) sind die Gebäude mit dem Gelände dargestellt. In Abbildung 6.4 (b) sind die Gebäude mit der Geländeschnittlinie abgebildet. Es ist erkennbar, dass die Gebäude teilweise unterhalb der Geländeoberflache liegen.

Ein stichprobenweiser Vergleich der Gebäudegrundhöhen, aus den Attributwerten und aus dem DGM, zeigt, dass sie bis auf ±2 Meter zusammenpassen. Beide Höhenangaben stammen aus der LiDAR Befliegung 2010. Es ist nicht genauer bekannt, wie die Grundhöhen der Gebäude für das Attribute ground_elevation bestimmt wurden. Bei den Grundhöhen aus dem DGM handelt es sich um den minimalen Höhenwert des DGM unterhalb des Gebäudes.

6.3.2 Flurstücke

Zunächst werden die Flurstücke in das benötigte Koordinatenreferenzsystem transformiert. Der überwiegende Teil der gegebenen Attribute der Flurstücke wird in CityGML übernommen. Dazu werden sie nach den entsprechenden *Pluto Data Dictionary* Informationen umbenannt. Die Bedeutung der einezelnen Attribute kann in dem *Pluto Data Dictionary*[3] eingesehen werden. Zusätzlich werden die Attributwerte von Fuß in Meter umgerechnet und *externalReferences* auf die Originaldaten angelegt. Da einige Flurstücke sehr große Flächen abdecken, werden Flurstücke über 40 000 m^2 geteilt. Diese sind mit dem Attribut *tiled_lot* gekennzeichnet. Jeder Teil eines großen Flurstücks erhält alle Attribute und hat somit identische Informationen wie die anderen Teile des originalen Flurstücks. Nach der Kachelung der Flurstücke wird eine zufällige *gml_id* angelegt, wodurch diese für jedes Flurstück bzw. jeden Teil eines Flurstücks unterschiedlich ist. Abschließend werden den Flurstücke noch die Höhen aus dem DGM übertragen. Dies erfolgt wie in Abschnitt 6.3.1 erläutert, in den Schritten 1 und 2. Die Unterscheidung zwischen *tax lot* und *mapping lot* erfolgt mit Hilfe des neu erzeugten Attributes *lot_type*.
In Tabelle 6.4 sind die verwendeten Attribute aufgelistet. Die violett geschriebenen Attribute sind in CityGML vordefiniert, die restlichen wurden generisch erzeugt.

Tabelle 6.4: Attribute der Flurstücke [Notation: CityGML]

- all_zoning_1
- all_zoning_2
- apportionment_bbl
- apportionment_date
- area_source
- assessed_value_land
- assessed_value_total
- basement_code
- block_number
- borough_block_lot_number
- borough_name
- building_area
- building_area_units
- building_class
- building_depth
- building_depth_units
- building_frontage
- building_frontage_units
- built_floor_area_ratio
- census_block
- census_tract
- city_concil_district
- citygml_class
- commercial_area
- commercial_area_units
- commercial_overlay1
- commercial_overlay2
- community_district
- condominium_number
- created_by
- created_date
- e_designation_number
- easements_number
- exempt_value_land
- exempt_value_total
- extension_code
- externalReference. informationSystem
- externalReference.name
- factory_area
- factory_area_units
- fire_company
- garage_area
- garage_area_units
- gml_id
- gml_name
- gml_parent_id
- historic_district_name
- irregular_lot_code
- land_use_category
- landmark_name
- last_modification_date
- last_modified_by
- last_modified_date
- limited_height_district
- lot_address
- lot_area
- lot_area_units
- lot_depth
- lot_depth_units
- lot_frontage
- lot_frontage_units
- lot_number
- lot_type
- number_of_buildings
- number_of_floors
- office_area
- office_area_units
- other_area
- other_area_units
- owner_name
- owner_type
- pluto_base_map_indicator
- police_precinct
- proximity_code
- residential_area
- residential_area_units
- retail_area
- retail_area_units
- sanborn_map
- school_district
- sperical_purpose_district_1
- sperical_purpose_district_2
- split_boundary_indicator
- storage_area
- storage_area_units
- tax_map
- tiled_lot
- units_residential
- units_total
- version_number
- year_altered_1
- year_altered_2
- year_built
- year_built_code
- zipcode
- zoning_district_1
- zoning_district_2
- zoning_district_3
- zoning_district_4
- zoning_map
- zoning_map_code

7 Bäume

Bäume tragen wesentlich zum Stadtbild bei und sollten auch in dem Stadtmodell von New York City nicht fehlen. Darüber hinaus kann die Erfassung des Baumbestandes einer Stadt die Grundlage für weiterführende Planungsaufgaben bieten. So können z. B. Bäume als Hindernisse bei der Routenwahl von Schwertransporten berücksichtigt oder auch die Baumpflege sinnvoll organisiert werden. Im folgenden Kapitel werden die verwendeten Daten sowie die Umsetzung in das CityGML-Format beschrieben.

7.1 Definition in CityGML

In CityGML werden Pflanzen im Vegetation-Modul (Abbildung 7.1) beschrieben. Das Vegetation-Modul besteht aus der abstrakten Klasse *_VegetationObject*, die sich in die Klassen *SolitaryVegetationObject* und *PlantCover* spezialisiert. Diese erben alle Attribute der übergeordneten Klassen *_VegetationObject* und der Wurzelklasse *_CityObject*. Dazu zählen u. a. *name* und *ExternalReferences*. Die Klassen *SolitaryVegetationObject* und *PlantCover* unterscheiden sich nach der Art des Bewuchses [Gröger et al., 2012].

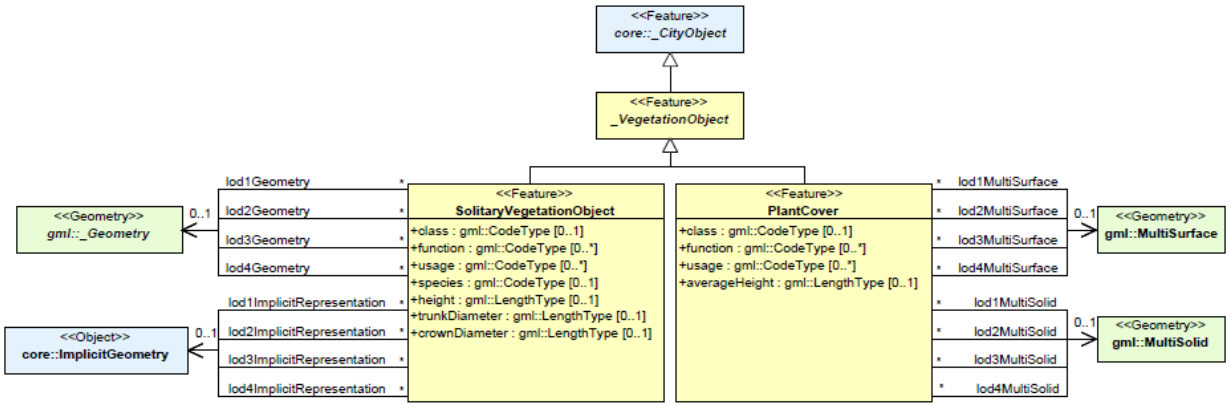


Abbildung 7.1: CityGML-UML-Diagramm des Moduls *Vegetation* [Gröger et al., 2012]

Bei einem *SolitaryVegetationObject* handelt es sich um eine einzelne Pflanze, zum Beispiel einen freistehenden Baum. Für diese Klasse sind die optionalen Attribute *class*, *function*, *usage*, *species*, *height*, *trunkDiameter* und *crownDiameter* vorgesehen. Das *SolitaryVegetationObject* kann in den LoD 1 bis LoD 4 mit einer expliziten Geometrie, in dem Typ *Geometry*, oder als implizite Geometrie, *ImplicitGeometry*, vorkommen. Die implizite Geometrie eignet sich für die Darstellung häufig vorkommender Objekte mit identischen Formen oder Geometrien. Dabei wird ein Prototyp definiert, der für die Darstellung mit dem Objekt verknüpft wird.

Die Klasse *PlantCover* stellt flächenhafte Pflanzengruppen, wie Wälder oder Wiesen, dar. Hierbei kommt es nicht auf die einzelne Pflanze an, sondern auf die Darstellung der Vielfalt der Pflanzengattungen der einzelnen Pflanzengruppe. Die Geometrie des *PlantCovers* Objektes im LoD 1 bis LoD 4 ist abhängig von der Höhe der Pflanze. Niedrige werden durch die Geometrie *MulitSurface* dargestellt, höher wachsende durch *MultiSolid*. Optional bestehende Attribute sind *class*, *function*, *usage* und *averageHeight*.
Zusätzliche Attribute können als generische Attribute angelegt werden. Die Werte der einzelnen Attribute sind in vorgegebenen Codelisten genauer definiert. Diese können verwendet oder neu erstellt werden [Gröger et al., 2012].

7.2 Verwendete Daten

Für die Standorte der Bäume liegen zwei verschiedene Datensätze vor, die beide vom New Yorker *Department of Parks and Recreation* zur Verfügung gestellt werden. Dabei handelt es sich um die Datensätze *Street Census Tree*[1] und *milliontreesNYC*[2] , wobei der erste Datensatz 30 000 Baumobjekte weniger enthält, jedoch über eine größere Anzahl an verarbeitbaren Baumattributen verfügt. Eine Kombination der beiden Datensätze ist nicht möglich, da sich sowohl die Lage der Bäume als auch ihre IDs unterscheiden. Deshalb wird nur der Datensatz *Street Census Tree* für die Modellierung der Bäume verwendet. Der Datensatz *Street Census Tree* stammt von einer Baumzählung von Freiwilligen aus dem Zeitraum von 2005 bis 2006. Die 620 000 Bäume sind nach den Boroughs geordnet in Datensätze aufgeteilt.

Die Bäume liegen als zweidimensionalen Punkte im Nordamerikanischen NAD83 Koordinatenreferenzsystem in Fuß vor. Zusätzlich zu den Koordinaten des Baumes enthält das Attribut *tree_in_relation_to_building* die Position des Baumes in Bezug auf das nächststehende Gebäude, inklusive dessen Adresse in Form der Attribute *next_building_street* und *next_building_number*. Die Straße, in der der Baum steht, ist in dem Attribut *street_of_tree_location* hinterlegt, sowie die Postleitzahl und der Stadtteil in den Attributen *zipcode* und *borough_name*. Die Attribute *cross_street_low* und *cross_street_high* geben die kreuzenden Straßen an der niedrigeren bzw. der höheren Adresse der Straße an.
Zusätzlich enthält der Datensatz baumspezifische Attribute wie Stammdurchmesser (*citygml_trunk_diameter*), Spezies (*citygml_species*) und Gesundheitszustand (*tree_condition*), der einzelnen Bäume. Der Stammdurchmesser wird in Brusthöhe, 4,5 Fuß oberhalb des Bodens, in Fuß gemessen. Teilt sich ein Stamm unterhalb 3 Fuß, werden die einzelnen Stämme getrennt gemessen und zu einem Stammdurchmesser addiert. Der Gesundheitszustand der Bäume unterteilt sich in 7 Klassen, die den Status von „Gesund und in besten Zustand“ über „teilweise abgestorbene Bäume“ bis „leere Plätze für Bäume“ repräsentieren. Die genaue Codierung ist im Anhang B.2 angegeben.

[1] https://data.cityofnewyork.us/Environment/Street-Tree-Census-Staten-Island-/pvq6-utmq (15.10.2014), https://data.cityofnewyork.us/Environment/Street-Tree-Census-Brooklyn-/ztcw-bzc8 (15.10.2014), https://data.cityofnewyork.us/Environment/Street-Tree-Census-Bronx-/bkyy-g74a (15.10.2014), https://data.cityofnewyork.us/Environment/Street-Tree-Census-Queens-/4wcq-bg6s (15.10.2014), https://data.cityofnewyork.us/Environment/Street-Tree-Census-Manhattan-/e6n3-m3vc (15.10.2014)
[2] http://www.milliontreesnyc.org/html/home/home.shtml (15.10.2014)

7.3 Umsetzung

Für die Darstellung der Bäume in CityGML werden die expliziten sowie die impliziten Geometrien umgesetzt. Die Überführung der Daten in die Semantik von CityGML ist für beide Geometrien identisch und wird im Folgenden beschrieben. Danach folgen Erläuterungen zur expliziten und impliziten Geometrie. Zum Schluss wird ein Vergleich anhand von Testdatensätzen vorgenommen.

Konvertierung der Daten nach CityGML

Zunächst werden die Positionen in das Lagekoordinatensystem NAD83 und die Höhen in das NAVD88-Höhensystem transformiert (vgl. Kapitel 4.2). Aus dem Datensatz der Bäume werden die sinnvollen und informativen Attribute ausgewählt und diese teilweise vordefinierten CityGML-Attributen zugeordnet, oder neue Attribute als generische Attribute erzeugt. Im Anhang B.2 sind die ausgewählten Attribute mit Erläuterungen angegeben. Die Species der Bäume wurde mit Hilfe der *Tree Species List* des *Street Tree Census* in die vollständigen Lateinischen Namen umbenannt (siehe Anhang B.2). Für die Codierung der Attributwerte werden eigene Codelisten angefertigt und Codelisten aus dem original Datensatz verwendet. Diese befinden sich im Anhang B.2. Das Attribut *citygml_trunk_diameter* wird von Fuß in Meter umgerechnet. Damit sind die Längenangaben einheitlich in Meter angegeben, da auch das Koordinatensystem von Fuß in Meter transformiert wird. Zusätzlich erhalten die Objekte jeweils die zwingend nötige *gml_id* und *gml_parent_id*, sowie externe Referenzen, die auf die URL des Datensatzes und die *tree_id* innerhalb des Datensatzes verweisen. So kann für jedes Objekt nachverfolgt werden, aus welchem Datensatz es stammt und um welches Objekt es sich genau handelt.

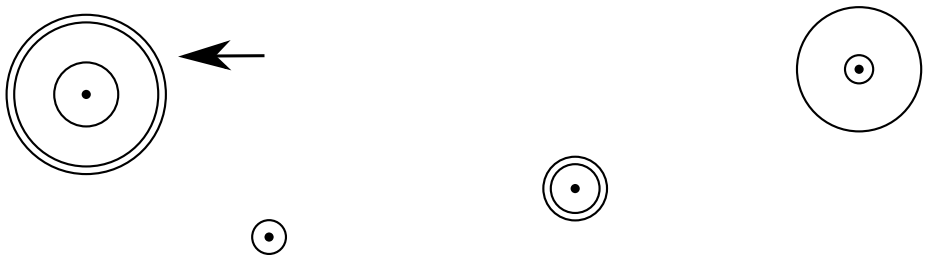


Abbildung 7.2: Problem: Standorte der Bäume liegen zum Teil übereinander

In Abbildung 7.2 geben die Punkte Standorte von Bäumen an, die schwarzen Kreise mit dem Punkt als Zentrum stellen die Durchmesser der einzelnen Bäume dar. Die linke Darstellung mit Pfeil markiert den Standpunkt von 3 Bäumen, daneben den eines Baums, die beiden rechten entsprechen jeweils dem Standpunkt zweier Bäume. Hierbei liegen angeblich alle drei Bäume auf dem selben Ort, haben aber verschiedene Durchmesser. Ein möglicher Grund hierfür könnte sein, dass die Daten von Freiwilligen gesammelt werden und die Position der Bäume in Bezug zum nächststehenden Gebäude angegeben wird, nicht aber Koordinaten für den realen Standort bestimmt werden. Dies wird deutlich, wenn die Standpunkte der Bäume visualisiert werden. In Abbildung 7.3 ist ein Screenshot aus GoogleEarth mit Baumobjekten (hellgrüne Lollipops) dargestellt. Die Koordinaten der Positionen der Bäume liegen auf der Straße. Ein Fehler bei der Transformation der Baumobjekte in das gewählte Koordinatensystem kann ausgeschlossen werden, da die Koordinaten lediglich von Fuß in Meter umgerechnet werden (vgl. Kapitel 4.2). Eine Information zu dem Abstand zwischen

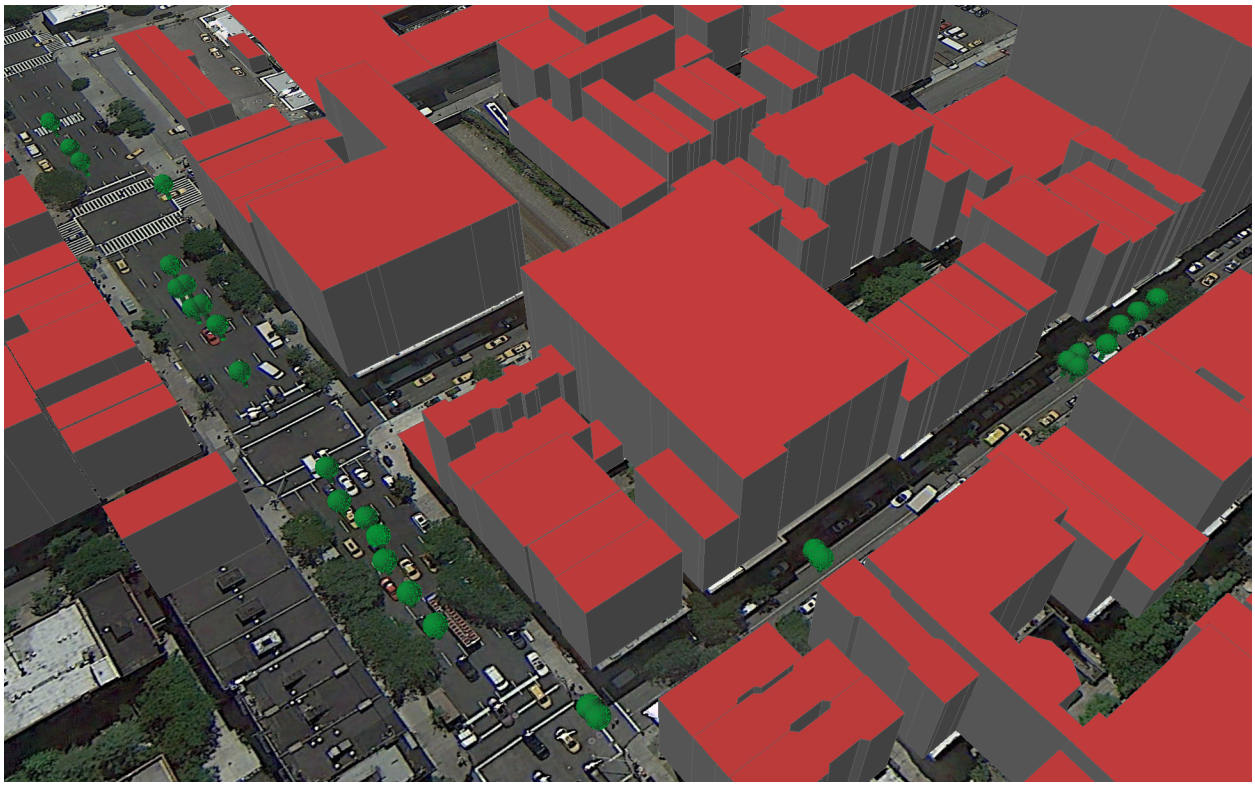


Abbildung 7.3: Problem: Standorte der Bäume liegen mitten auf der Straße

Baum und Straßenmittelachse in der Realität ist nicht gegeben. Zusätzlich kommt es häufiger vor, dass Baumobjekte auf dem gleichen Standort mit identischen Attributwerten vorkommen und sich nur in der *tree_id* unterscheiden.

Da es keine Angaben über die Höhe der Bäume gibt und diese auch nicht mit vertretbarem Aufwand herleitbar ist, wird eine einheitliche Baum-Geometrie gewählt, welche die Baum-Objekte im Stadtmodell darstellt. Dabei werden alle Bäume eines Standpunktes mit Hilfe eines einzelnen Baumobjektes visualisiert. Falls sich mehrere Bäume auf einem Standort befinden werden, von den 2 bis N Bäumen jeweils nur das Attribute *tree_id* in der *externalReference{}.name* hinterlegt. Für einen Baum der „Gruppe“ werden alle verfügbaren Attribute gespeichert, sowie eine *externalReference* als Verweis auf das dargestellte Objekt angelegt. Analog wird für den Fall, dass nur ein Baum auf einem Standort steht, vorgegangen. Somit ist sichergestellt, dass alle Bäume eines Standortes identifiziert und zurückverfolgt werden können. Der Baum aus der „Gruppe“ mit dem größten Stammdurchmesser (*citygml_trunk_diameter*) bzw. in zweiter Instanz mit der höchsten *tree_id* ist der Repräsentant und liefert alle seine verfügbaren Attribute an das Geometrieobjekt. Das erste Kriterium, den Baum mit dem größten Stammdurchmesser darzustellen, wurde gewählt, weil der Baum mit der Zeit im Umfang und der Höhe wächst, und da es möglicherweise die aktuellste Bestimmung des Baumes ist. Natürlich könnte mittlerweile der Baum entfernt worden sein, ein neuer Baum gepflanzt sein und somit ein veralteter Baum dargestellt sein. Falls das erste Kriterium versagt, greift das zweite, welches die höchste *tree_id* verwendet. In diesem Fall wird davon ausgegangen, dass die *tree_id* mit neueren Baumbestimmungen steigt. Diese Kriterien dienen lediglich dazu, dass ein Baum für die Repräsentation mehrerer Bäume ausgewählt wird.

Für die Geometrie der Bäume werden zwei verschiedene Methoden verwendet und im Folgenden

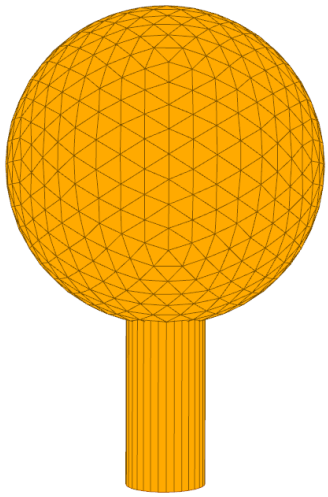


Abbildung 7.4: Lollipop Geometrie der Bäume

erläutert. Dabei ist die Darstellung der Baumobjekte identisch, der Stammdurchmesser der Bäume geht nicht in die Darstellung mit ein. Sie werden aus einem Zylinder als Baumstamm und einer Kugel als Krone in Form eines „Lollipops“ zusammengestellt (siehe Abbildung 7.4). Eine Baumgeometrie besteht aus 1 314 einzelne Polygonen. Diese werden zunächst mit FME in einer *CompositeSurface* zu einer Geometrie zusammengefasst und mit Hilfe verschiedener Transformer in die GML unterstützte Geometrie (*MultiSurface*) gespeichert.
Bei der Darstellung bleibt die Spezies der Bäume wie auch die Art (Laub- oder Nadelbaum) unberücksichtigt. Insgesamt werden 277 108 Baumobjekte in CityGML geschrieben. Die Höhenübertragung von dem DGM auf die Baumobjekte erfolgt analog zu dem in Kapitel 6.3.1 beschriebenen Ablauf der Schritte 1 bis 3. Hierauf wird an dieser Stelle nicht weiter eingegangen.

Implizite Geometrie

Weil ein identisches Geometrie-Objekt für alle Baumobjekte verwendet wird, besteht die Möglichkeit eine implizite Geometrie (*lod1ImplicitRepresentation*) zu wählen. Auf diese Weise kann eine Mehrfachspeicherung der identischen Geometrie umgangen und so Speicherplatz gespart werden. Der sogenannte Prototyp wird in einem lokalen Koordinatensystem erzeugt. Es wird eine direkte Bereitstellung angewendet. In dem Fall erfolgt die Speicherung des Prototyps intern in dem CityGML-Dokument als Geometrie des ersten Baumobjektes. Die weiteren Baumobjekte verweisen lediglich über einen *XLink* auf die Geometrie bzw. die *gml_id* des Prototypen. Zu jedem Baum-Standort wird ein Einfügepunkt für die prototypische Geometrie (=Referenzpunkt) im globalen Referenzsystem gespeichert. Über eine Transformationsmatrix wäre es möglich die Skalierung und Drehung jedes einzelnen Baumes einzustellen. Dies wird in dieser Arbeit nicht verwirklicht, weshalb alle Bäume gleich groß sind [Wanasky, 2011], [Gröger et al., 2012].
Die Umsetzung der impliziten Geometrie mit FME wird in zwei Schritten realisiert, da es noch keine vordefinierte Funktion für das Generieren impliziter Geometrien gibt. Zunächst werden die Bäume wie oben beschrieben in CityGML geschrieben und als Punkte gespeichert. Die Erzeugung

der Geometrie erfolgt wie im obigen Abschnitt *Konvertierung der Daten nach CityGML* erläutert. Der Ursprung des Geometrie-Objektes liegt in der Mitte des Stammes auf der Höhe Null. In einem zweiten Schritt wird die prototypische Geometrie und die Verlinkung nach dem Prinzip "suchen-und-ersetzen" eingefügt. Dazu wird die *lod1Geometry* in eine *lod1ImplicitRepresentation* mit einer *ImplicitGeometry* und *referencePoint* umgeschrieben. Die Geometrie des Prototypen wird in dem ersten Objekt des GML-Dokumentes statt einem *XLink* eingefügt.
Für die weitere Verarbeitung der Daten müssen diese in eine Datenbank mit dem CityGML-Schema importiert werden. Dazu brauchen die Objekte eine *envelope Bounding Box*. Diese ist in dieser generierten impliziten Darstellung nicht enthalten. Eine Möglichkeit für einen erfolgreichen Import wäre die Gruppierung der Bäume in einer *City Object Group*, da die Objekte in diesem Fall eine Bounding Box erhalten. Dieser Ansatz wurde nicht weiter verfolgt, sondern die Darstellung der Bäume in einer expliziten Geometrie für die gesamte Stadt New York.

Explizite Geometrie

Die Grundlagen der Generierung des CityGML Datensatzes erfolgt analog der impliziten Geometrie und ist in Abschnitt *Konvertierung der Daten nach CityGML* erläutert. Bei der zweiten Methode werden die Geometrien der Bäume als explizite Geometrien (*lod1Geometry*) gespeichert. Das bedeutet, dass jeder Standort von einem oder mehreren Bäumen ein Geometrie-Objekt erhält und dieses jeweils erzeugt und gespeichert werden muss. Dies ist sehr rechen- und speicherintensiv. Aus den ursprünglich 206 MB SHAPE-Dateien werden insgesamt 113 GB große CityGML-Dateien. Die Erzeugung der Geometrie erfolgt wie in Abschnitt *Konvertierung der Daten nach CityGML* erläutert. Um die GML-Dokumente generieren zu können, müssen die einzelnen Stadtteile gekachelt und diese Kacheln einzeln berechnet werden. Die Bäume von New York werden auf 168 Kacheln aufgeteilt und jede Kachel einzeln berechnet. Insgesamt werden 277 108 Geometrieobjekte erzeugt, um die Bäume zu repräsentieren.

Geometrievergleich

Für einen Ausschnitt mit 16 Bäumen sind sowohl explizite als auch implizite Geometrien in CityGML generiert worden. Die Dateigröße für die explizite Darstellung sind 6 887 KB, die implizite hingegen benötigt nur 383 KB. Ein Vorteil der impliziten Geometrie gegenüber der expliziten ist die Einsparung von Speicherplatz, weswegen die Geometrie nur einmal gespeichert werden muss und redundante Speicherung wegfällt. Die Geometrie der Bäume besteht aus 1 314 Polygonen. Für jedes Polygon werden in einer GML-Datei 10 Zeilen gespeichert und für jedes Objekt mit expliziter Geometrie identisch wiederholt. Je mehr Objekte eine identische Geometrie erhalten, desto mehr lohnt sich eine implizite Darstellung. Des Weiteren spielt auch die Anzahl der implizit modellierbaren Objekte, sowie die benötigte Anzahl verschiedener Prototypen eine Rolle. Ein Nachteil ist, dass die Darstellung des Prototypen bis auf die Größe und eine Drehung gleich aussehen bzw. keine unterschiedliche Textur angebracht werden kann.

8 Gewässer und hydrologische Infrastruktur

Im Folgenden Kapitel wird auf die Modellierung der Gewässer in CityGML und eine mögliche Veränderung dieses Moduls sowie auf den Datensatz der verwendeten Gewässer eingegangen. Des Weiteren wird die Umsetzung der Integration von Gewässern und dem DGM in das Stadtmodell erläutert und ein Ausblick auf mögliche Verbesserungen gegeben. In dem zweiten Teil des Kapitels wird das Modul *CityFurniture* und dessen Modellierung mit den Daten der hydrologischen Infrastruktur wie Stege oder Anlegestellen, dargelegt.

8.1 Gewässer

8.1.1 Gewässer im Modul WaterBody

In dem thematischen Modul *WaterBody* werden Gewässer modelliert. Gewässer sind dynamisch und können sich durch Gezeiten verändern. Zusätzlich sind sie unter anderem von Naturereignissen wie Hochwassern abhängig. Dynamische Ereignisse werden in dem CityGML-Modell nicht berücksichtigt. Somit handelt es sich hierbei nicht um ein vollständiges Modell für Gewässer, dennoch erfüllt es die Anforderungen für ein 3D-Stadtmodell. Ab dem LoD 2 besteht die Möglichkeit Wasserstände stufenweise durch die Klasse *WaterSurface* darzustellen und mit Hilfe des Attributes *WaterLevel* genauer zu beschreiben [Gröger et al., 2012].

Das thematische Modul *WaterBody* ist in Abbildung 8.1 dargestellt. Die Wurzelklasse ist *_CityObject*, sie vererbt ihre Attribute und Relationen auf die abstakte Klasse *_WaterObject* und diese wiederum auf ihre Unterklassen *WaterBody* und *_WaterBoundarySurface* (siehe hierzu Kapitel 2.2) [Gröger et al., 2012].

Die geometrische Darstellung der Gewässer variert je nach LoD. Im LoD 0 und LoD 1 werden diese stark generalisiert dargestellt, wobei kleinere Flüsse, Bäche und Seen ganz wegfallen. Gestreckte Gewässer, wie Flüsse, werden aus einem *Network* aus Liniensegmenten als *MulitCurve* dargestellt. Die Fließrichtung wird mit der Orientierung der Liniensegmenten von dem ersten zum letzten Punkt der Linien dargestellt. Bei flächigen Gewässern, wie Ozeane, Meere und größere Seen, werden die Wasseroberflächen in 3D mit der Geometrie *MulitSurface* repräsentiert. Ab dem LoD 1 ist es zudem möglich Gewässer als Volumen mit Hilfe der Geometrie *Solid* zu modellieren. Die in dieser Arbeit modellierten Gewässer werden einheitlich als *MultiSurface* im LoD 1 gespeichert.
In den LoD 2, LoD 3 und LoD 4 fällt die Generalisierung weg und es ist zudem möglich *_WaterBoundarySurfaces*, beziehungsweise die Unterklassen *WaterGroundSurface*, *WaterSurface* und *WaterClosureSurface*, einzufühen (siehe Abbildung 8.2). Die *WaterGroundSurface* beschreibt die untere Grenze des Gewässers, die mit der Geländeoberfläche unterhalb des Gewässers identisch ist. Die Klasse *WaterSurface* repräsentiert die Oberfläche des Gewässers. Diese kann dazu verwendet werden um verschiedene Wasserstände darzustellen wie Flut, Ebbe, Wasserstände von Hochwasser und Schiffbarkeit, die in dem Attribut *waterLevel* codiert sind. Hierfür existiert eine vorgefertigte Codeliste, die

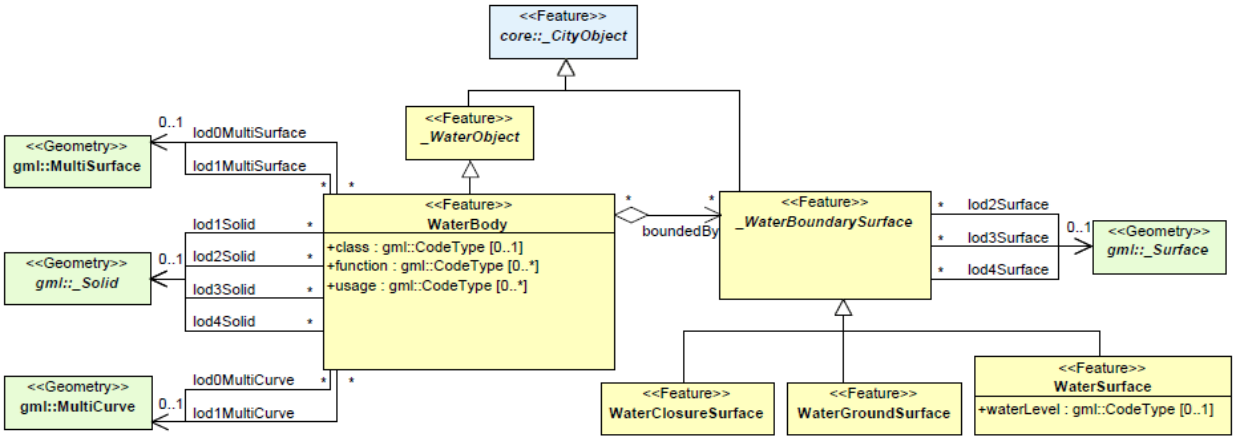


Abbildung 8.1: CityGML-UML-Diagramm des Moduls *WaterBody* [Gröger et al., 2012]

erweitert oder neu definiert werden kann. Gewässer, die nicht durch die Flächen von *WaterSurface* und *WaterGrundSurface* geschlossen sind, werden mit Objekten der Klasse *WaterClosureSurface* zu geschlossenen Volumenkörpern. Zusätzlich besteht die Möglichkeit unterschiedliche Gewässer voneinander zu trennen. Die Verknüpfung der Wasserflächen mit dem zugehörigen Volumen erfolgt über *XLinks* [Gröger et al., 2012].

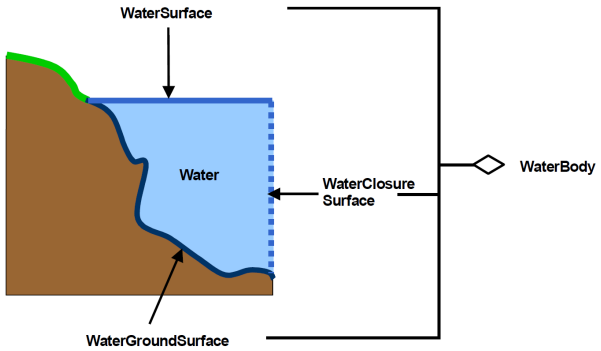


Abbildung 8.2: Darstellung des Wasserkörpers in CityGML [Gröger et al., 2012]

8.1.2 Datensatz der Gewässer

Die Gewässer in und um New York City sind in dem Datensatz *Hydro*[1] als zweidimensionale Polygone enthalten. Sie liegen im NAD83-Koordinatenreferenzsystem vor. Die Daten werden vom *Department of Information Technology and Telecommunications* (DoITT) aus photogrammetrischen Befliegungen bereitgestellt. Die angestrebte Auflösung der Bilder liegt bei 0,5 Fuß. In dem Datensatz sind zum Teil die Namen der einzelnen Gewässer sowie die Gewässerarten gegeben. In Anhang B.3 ist eine Übersicht über die Attribute der Gewässer gegeben.

[1] https://data.cityofnewyork.us/Environment/Hydrography/drh3-e2fd (14.10.2015)

Diese werden in die folgenden Arten unterteilt:

- Lake/ Reservoir
- Pond
- River
- Stream wider than 8 feet
- Wetland
- Beach/ Shoreline
- Bay/ Ocean

8.1.3 Höhenübertragung auf Gewässer

Die Gewässerpolygone enthalten ursprünglich keine Höheninformation. Über Erfahrungswerte – wie die Wasseroberfläche eines Sees liegt unterhalb der Geländekante, da er sonst abfließt – können implizite Höheninformationen abgeleitet werden. Die Implizite Höheninformation drückt die Relation zu anderen Objekte aus, in diesem Fall zwischen Wasseroberfläche und Gelände. Sie ist nicht für alle Objekte vorhanden (und sind abhängig von der betrachteten Gegend). So können, im Gegensatz zu Straßen, über Wiesen keine Informationen zur die Neigung oder Krümmung abgeleitet werden [Koch, 2006].
Werden die Geometrien der zweidimensionalen Objekte in das DGM eingegliedert und diesen eine Höhe zugewiesen, wird es *Integration von Objekten und DGM* genannt. Grob kann die Integration in zwei Arten aufgeteilt werden. Zum Einen in die geometrische, wobei die Objekte in das DGM integriert werden. Zum Andern in die semantische Integration. Wobei sich hierbei die Lage der Objekte verändern kann. Dies kann beispielsweise der Fall sein, wenn die Semantik regelt, dass eine Straße nicht in einem Gewässer enden darf.

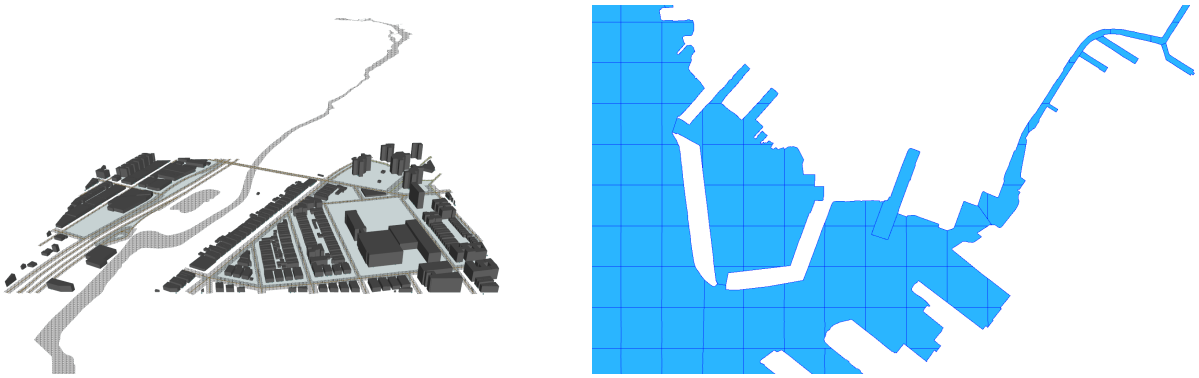
[Schilling et al., 2007] zeigen eine effiziente Lösung der Integration von zweidimensionalen Geodaten und DGM. Da es sich bei dem DGM um eine durchgehende Oberfläche mit lediglich eingefügten Flächen der 2D-Objekte handelt, tritt kein Problem des Höhenversatzes auf. Die Integration lässt sich grob in drei Schritt einteilen. Zunächst werden die Eckpunkte der Objekt-Polygone in das TIN in der Lage eingefügt und bekommen eine Höhe zugewiesen. Somit gehören die Eckpunkte (*Constrained Nodes*) nun zum TIN. In dem nächsten Schritt werden die *Constrained Nodes* zu Dreiecken verbunden und überflüssige Punkte innerhalb eines Objektes aus der Vermaschung entfernt. Zusätzlich wird die korrekte Höhe der Punkte bestimmt. Abschließend werden die Dreiecke, die Objekte enthalten, den Layern der Objekte zugewiesen. Somit ist die Integration der Objekte und des DGMs erledigt. Über eine optionale Rechnung können die Flächen geglättet werden.

Koch (2006) hingegen befasst sich mit der semantischen Integration zweidimensionaler Objekte und Digitaler Geländemodelle, welche auf der Methode einer Triangulation basiert. Semantisch korrekt bedeutet in diesem Fall, dass je nach Objekt bestimmte geometrische Bedingungen eingehalten werden. Ziel ist es eine semantisch korrekte Repräsentation topographischer Objekte, wie u. a. Flüsse und Seen, zu generieren. Dazu werden die verwendeten Daten zunächst mathematisch optimiert. Zusätzlich zu der geometrischen Integration der Objekte wird auch die Semantik der Objekte berücksichtigt. So werden beispielsweise Seen als horizontale Flächen generiert. Flüsse hingegen werden mit einem abnehmenden Höhenverlauf in Fließrichtung modelliert. Im Gegensatz zu einem See werden Flüsse mit Hilfe individuell geneigter Ebenen modelliert. Auch werden Nachbarschaftsbeziehungen berücksichtigt, beispielsweise zwischen Gewässerflächen und angrenzendem Gelände, da das Gelände von der Höhe her über der Wasserfläche liegt. U. a. werden mit Hilfe der Nachbarschaftsbeziehungen

abrupte Veränderungen im Geländemodell vorgebeugt. Die semantischen Informationen werden als Regeln formuliert und gehen als solche in die Berechnung mit ein.
Es werden zwei Varianten durchgeführt. Die erste verändert lediglich den Höhenwert um ein semantisch korrektes Ergebnis zu erreichen, da die Lagekoordinaten als fehlerfrei angenommen werden. In der zweiten Variante gehen die Genauigkeiten der Lagekoordinaten in die Berechnung mit ein. Aus diesem Grund kann es zu einer Veränderung der ursprünglichen Lage dieser kommen. Für die Umsetzung der semantischen Regeln werden Bedingungs- und Beobachtungsgleichungen einer Ausgleichung formuliert. [Koch, 2006].

Die Anpassung der Gewässer in dieser Arbeit erfolgt stark vereinfacht im Vergleich zu den oben genannten Beispielen. Zunächst werden alle Gewässer, ohne Unterscheidung ihrer Art, in CityGML als *MultiSurface* abgespeichert.
Wenn die Gewässer im CityGML-Format in die *3DCityDB* importiert werden, dann werden die Polygone im Ganzen wieder aus der DB exportiert. In Abbildung 8.4 (a) ist dieses dargestellt. Es ist erkennbar, dass die übrigen Objekte bei dem Export auf den Ausschnitt angepasst sind (siehe Straßen). Der Fluss wird allerdings in seiner gesamten Länge exportiert. Die Höhenübertragung wird für alle Objekte, die ganz oder teilweise in einer DGM-Kachel (250×250 m) enthalten sind, durchgeführt (vgl. Kapitel 6.3.1). Da viele dieser Objekte über den Rand der Kachel herausragen, wird letztendlich der Ausschnitt des DGMs an die Boundingbox der Objekte angepasst. In einem solchen Fall würde ein sehr großer DGM-Ausschnitt aus der DB exportiert, der u. U. bei einer Berechnung an die Grenzen des verfügbaren Arbeitsspeichers stoßen würde. Nicht zuletzt, da es sich bei dem Export des DGMs um die Boundingbox der Objekte handelt und nicht um den Umriss der Objekte. Dies ist in Abbildung 8.3 dargestellt. Das rote Rechteck stellt die benötigte Größe des DGM Ausschnitts dar. Im Vergleich dazu bildet das grüne Rechteck den benötigten Ausschnitt einer DGM-Kachel ab. Dieses wird damit umgangen, dass die Gewässer ab einer Größe von 40 000 m^2 in Kacheln von ca. 200×200 m aufgeteilt werden. Gewässer, die durch die Kachelung, kleiner als 2 000 m^2 sind, werden einem anderen Stück des Gewässers zugeordnet. So wird eine unnötige kleinteilige Gewässerstruktur umgangen.

Abbildung 8.3: rot: Boundingbox um ein gesamtes Gewässerpolygon; grün: Boundingbox um den benötigten Ausschnitt



(a) (b)

Abbildung 8.4: (a) Gewässerpolygone werden aus der DB im Ganzen exportiert, restlichen Objekte werden an Ausschnitt angepasst, erkennbar an Straßenpolygonen; (b) Gewässer werden für die weitere Berechnung gekachelt

Bisher werden in dem OGC-Standard von CityGML lediglich Gewässer im Gesamten modelliert. Da Gewässer – v. a. Meer oder Flüsse – eine sehr große Ausdehnung aufweisen, sollte überlegt werden, ob die Version CityGML 3.0 nicht derart angepasst wird, dass Gewässer ebenfalls durch mehrere kleinere Gebiete repräsentiert werden können. Dafür könnte beispielsweise eine zusätzliche Klasse *WaterBodyPart* als Kindelement der Klasse *WaterBody* eingeführt werden. Durch die Aggregations-Hierachie könnte ein Objekt der Klasse *WaterBody* aus beliebig vielen Objekten der neuen Klasse *WaterBodyPart* bestehen. Die Attribute wie auch die Relationen würden von der Klasse *WaterBody* an die neue Klasse *WaterBodyPart* weitergegeben werden.

a) flächenhafte Gewässer

See/Teich

Die implizite Höheninformation eines Sees oder Teiches sagt aus, dass es sich hierbei um eine annähernd horizontale Fläche handelt. Für die Objekte der Gewässer wird dies in der vorliegenden Arbeit vereinfacht angenommen und modelliert. Zunächst werden die Höhenwerte des Rands[2] der Gewässer mit Hilfe des DGMs ermittelt. Dies ist im wesentlichen in Kapitel 6.3.1 in den Schritten 1 und 2 erläutert. In Abbildung 8.5 ist die Wasseroberfläche eines Sees, der auf das Geländemodell gelegt ist, stark überhöht dargestellt. An dem Rand erkennt man besonders gut, wie unregelmäßig dieser in seinem Höhenverlauf ist. Den Rand inklusive der gesamten innen liegenden Fläche gilt es zu glätten bzw. eben zu gestalten. Das Loch in der Seeoberfläche ist durch eine Insel innerhalb des Sees verursacht. Für die Modellierung wird die Wasseroberfläche dieser Objekte knapp unter (3cm) den minimalen Höhenwert seiner Randpunkte gesetzt. Dies ist nötig, damit die Höhe der Wasserfläche unterhalb der Geländefläche liegt. Sonst würde das Gewässer abfließen, bzw. das Gelände würde noch dem Gewässer zugeordnet werden [Koch, 2006].

Zusätzlich können Gewässerflächen und DGM integriert werden. Dazu wird der Rand des Gewässers (inklusive neuer Höhenwerte) bei einer erneuten TIN-Generierung als *Breaklines* eingeführt, um auf diese Weise das DGM an Stellen von Wasserflächen zu glätten bzw. Gewässer und DGM

[2] hier handelt es sich um die gekachelten Gewässer, d. h. die Ränder des Polygons müssen nicht zwangsläufig die Ränder des realen Gewässers sein

zu integrieren. Dazu werden auch alle TIN Punkte innerhalb des gekachelten Gewässerpolygons[2] entfernt. In Abbildung 8.6 (a) sind die Punkte des TINs zusehen, wobei Punkte des DGMs, deren Lagekoordinaten innerhalb des See-Polygons liegen, entfernt sind. Abbildung 8.6 (b) ist der Übergang von Gelände und Wasseroberfläche nach der Integration überhöht dargestellt.

Abbildung 8.5: Darstellung der Wasseroberfläche mit Geländehöhen (10-fach überhöht

Meer
Das DEM[3] gibt die Höhenwerte in Fuß über dem Meeresspiegel an. Referenz für das vertikale Datum ist das NAVD88 (vgl. Kapitel 4.2). Das DEM wurde von dem *Department of Environmental Protection* bearbeitet, so dass Gewässer bis zu 2 000 Fuß um die fünf Stadtteile enthalten sind. Für das 3D-Stadtmodell von New York werden die Flächen des Meeres stark vereinfacht und lediglich auf eine Höhe von Null Metern gelegt.

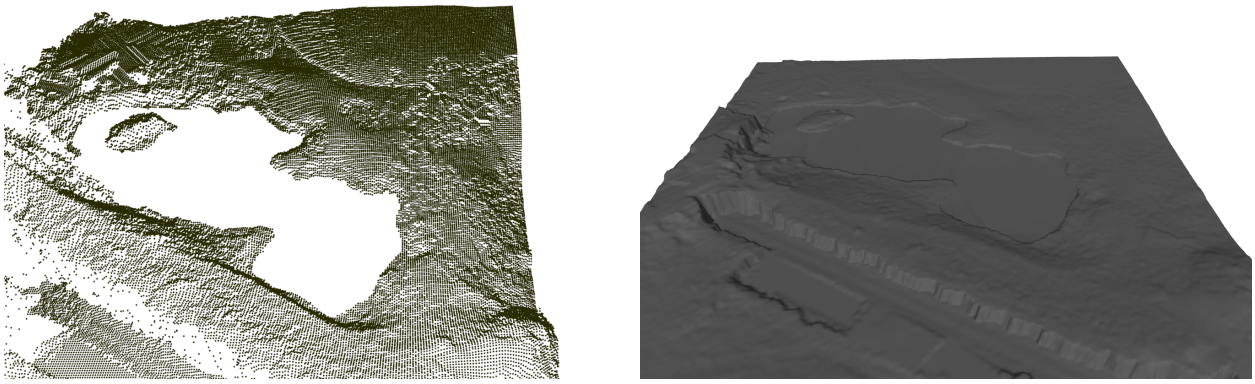
b) gestreckte Gewässer

In dieser Arbeit werden langgezogene Gewässer, beispielsweise Flüsse, auf die selbe Art modelliert wie die flächenhaften Gewässer (=Seen und Teiche) (siehe Abschnitt 8.1.3).

Generell gilt für Fließgewässer, dass das Niveau in Fließrichtung des Flusses abnimmt. Dies modelliert [Koch, 2006] mit Hilfe von geneigten Teilebenen, wobei die Fließrichtung des Gewässers bekannt ist. Bei den in dieser Arbeit verwendeten Daten handelt es sich hingegen ausschließlich um Polygone ohne Angabe zur Fließrichtung. Da die Gewässer ohne Rücksicht auf Verlauf und Seitenverhältnisse in Segmente aufgeteilt werden (siehe Abbildung 8.4 (b)), müsste die Fließrichtung zunächst bestimmt werden.
Um diese zu bestimmen, werden Höhenangaben zu den Gewässern benötigt. Die meisten Flüsse des Datensatzes erhalten bei einer Überlagerung mit dem DGM keinen erkennbaren Höhenverlauf. Dies ist zum Einen dadurch bedingt, dass das DEM (aus welchem das DGM abgeleitet wird, siehe Kapitel 5) vor dem Veröffentlichen u. a. im Bereich der Gewässer bearbeitet wurde, weil es beim Laserscanning über Gewässern aufgrund von Totalreflexionen zu Datenlücken kommen kann. Zum Anderen liegt der Fokus des DEM nicht auf einer realitätsnahen Repräsentation der Gewässer, sondern des Festlandes.
Neben der Tatsache, dass Fließgewässer keine größeren Höhenunterschiede aufweisen, mündet die Mehrheit dieser entgegen physikalischer Gesetze scheinbar mit ansteigenden Höhen in das Meer. Der *Richmond Creek* beispielsweise stellt dabei eine Ausnahme dar. Dieser fließt von ca. 30 Meter Höhe auf Meeresniveau.

Im Folgenden wird eine Möglichkeit erläutert, wie die Fließrichtung der vorliegenden Gewässer, die einen Höhenverlauf wie der *Richmond Creek* aufweisen, bestimmt werden kann. Da die Gewässer

[3] https://data.cityofnewyork.us/City-Government/1-foot-Digital-Elevation-Model-DEM-/dpc8-z3jc (15.10.2014)



(a) (b)

Abbildung 8.6: Integration Gewässer und Gelände: (a) Darstellung der Geländepunkte des TINs, in der Wasserfläche liegende Punkte sind gelöscht; (b) Darstellung der Wasseroberfläche enthält Geländehöhen aus DGM

ohne Rücksicht auf Verlauf und Seitenverhältnisse in Segmente aufgeteilt wurden (siehe Abbildung 8.4 (b)), würde man zunächst die minimale Höhe für die jeweiligen Gewässersegmente bestimmt (vgl. Kapitel 8.1.3). In einem weiteren Schritt würden die Gewässersegmente wieder zu einem Fluss vereint werden, wodurch der Fluss im Gesamten, aber mit Höhensprüngen zwischen den einzelnen Segmenten vorliegen würde. Die grobe Fließrichtung könnte somit aus den Maximal- und Minimalwerten bestimmt werden. Dabei entspricht der Maximalwert der Quelle und der Minimalwert der Einmündung in das Meer. Um schließlich die Höhensprünge zu glätten, könnte eine Ausgleichung nach Koch durchgeführt werden [Koch, 2006].
Die Umsetzung der eben erwähnten Ideen zur Repräsentation würden den Rahmen der vorliegenden Arbeit sprengen. Diese bieten allerdings eine gute Grundlage, um darauf aufzubauen und die Einebnung der Gewässer des New Yorker Stadtmodells durch zuführen.

8.2 Hydrologische Infrastruktur

Mit Hydrologischer Infrastruktur sind Bauten an Gewässern gemeint. Hierzu zählen u. a. Ufermauern und Anlegestellen.

8.2.1 Hydrologische Infrastruktur im Modul CityFurniture

Unbewegliche Objekte eines Stadtmodells, wie Straßenlaternen, Litfaßsäulen und Straßenschilder werden mit dem Modul *CityFurniture* (siehe Abbildung 8.7) modelliert. Sie dienen v. a. der Analyse, Planung und Navigation von Verkehr [Gröger et al., 2012].
Die Klasse *CityFurniture* ist eine Unterklasse von der Klasse *_CityObject* und erbt somit deren Attribute, sowie die Möglichkeiten *ExternalReferences* oder zusätzliche generische Attribute anzulegen. Vordefinierte Attribute sind class, function und usage. *class* klassifiziert Objekte wie Straßenlaterne, Verkehrsschilder oder Mülleimer. *function* teilt die Objekte ein und zu welchem thematischen Gebiet sie gehören, wie beispielsweise Architektur oder Verkehr. *usage* beschreibt die tatsächliche Nutzung der Objekte.

Die räumliche Darstellung der Objekte der Klasse *CityFurniture* kann als explizite Geometrie erfolgen oder als implizite Geometrie. Implizite Geometrien eignen sich besonders für die Darstellung identischer Geometrien, wie Straßenlaternen. In dem vorliegenden Fall, der Darstellung hydrologischer Infrastrukturen wie Pier oder Anlegesteg, wird eine explizite Geometrie gewählt, da die Objekte sich voneinander unterscheiden. Zusätzlich besteht die Möglichkeit für jedes LoD, eine *TerrainIntersectionCurve* einzufügen, um die 3D Objekte in die Geländeform einzupassen [Gröger et al., 2012].

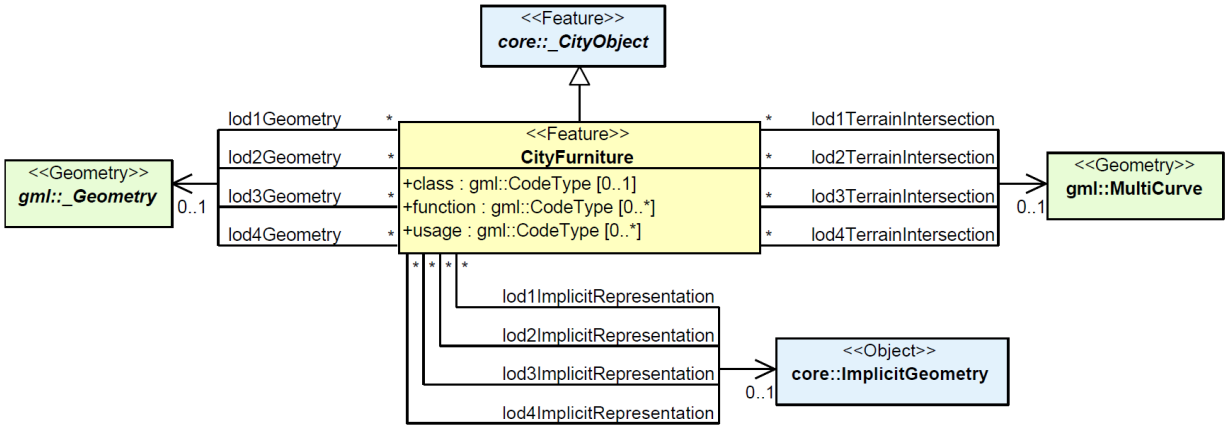


Abbildung 8.7: CityGML-UML-Diagramm des Moduls *CityFurniture* [Gröger et al., 2012]

8.2.2 Datensatz der hydrologische Infrastruktur

Die Daten über die hydrologische Infrastruktur[4] werden vom *Department of Information Technology and Telecommunications* bereit gestellt. Der Datensatz liegt im WGS84 Koordinatensystem als zweidimensionale Polygone vor. Insgesamt gibt es 2 464 hydrologische Infrastruktur Objekte. Es wird zwischen *pier*, *jetty* und *seawall* unterschieden. Weiterhin ist noch ein Attribut *Elevation* gegeben. Die Attributwerte erstrecken sich von -2,8 bis 251,9. Da weder eine Einheit (vermutlich Fuß) angegeben ist noch um welche Höhe es sich handelt, wird diese Attribut im weiteren Verlauf nicht weiter beachtet.

8.2.3 Umsetzung und Höhenübertragung der hydrologischen Infrastruktur

Die Objekte der hydrologischen Infrastruktur werden zunächst in das Modell-Koordinatenreferenzsystem transformiert. Die zwingend nötigen Attribute wie *gml_id* und *gml_parent_id* werden angelegt und die Objekte in CityGML in der *LoD1Geometry* als Polygone gespeichert. Die Höhenübertragung erfolgt wie in Kapitel 6.3.1 beschrieben in den Schritten 1 und 2. Für die hydrologische Struktur gibt es drei verschiedene Arten (*pier*, *jetty* und *seawall*). In Anhang B.4 ist ein Überblick über die verwendeten Attribute gegeben.
Um ebenfalls das Modul *CityFurniture* in das Modell zu integrieren, wird der Datensatz der hydrologischen Infrastruktur ausgewählt.

[4] https://data.cityofnewyork.us/Environment/Hydrography/drh3-e2fd (14.10.2014)

9 Parks

In diesem Kapitel wird das CityGML-Modul *LandUse* erläutert. Dieses wird für die Modellierung der Parkanlagen wie auch für die Flurstücke (Kapitel 6) verwendet. Im Anschluss werden die beiden voneinander unabhängigen Datensätze über Parkanlagen erläutert und über eine einheitliche ID kombiniert. Hierbei wird besonderer Wert auf nicht-redundante Datenspeicherung gelegt.

9.1 Definition in CityGML

Das thematische Modul *LandUse* in CityGML (siehe Abbildung 9.1), beschreibt die Oberfläche der Erde. Dabei wird zwischen LandUse, die durch Menschen geschaffene Erdoberfläche, wie Siedlungsgebiet, Flurstücke und räumliche Planungsobjekte, und der natürlichen Bodenbedeckung *LandCover*, wie Sand, Wald und Grasland unterschieden [Gröger et al., 2012].

Die Klasse *LandUse* stammt von der Wurzelklasse *_CityObject* ab und erbt u. a. die Attribute *name* und *gml_id*. Das *LandUse*-Objekt muss eine 3D-Geometrie besitzen und wird im LoD 0 bis 4 als *MultiSurface* modelliert.
Für die optionalen Attribute *class*, *function* und *usage* können die vorgegebenen Codelisten von CityGML verwendet oder neue definiert werden [Gröger et al., 2012]. Mit Hilfe dieses Moduls werden die Flurstücke (siehe Kapitel 6.3.2) und die Parkanlagen umgesetzt.

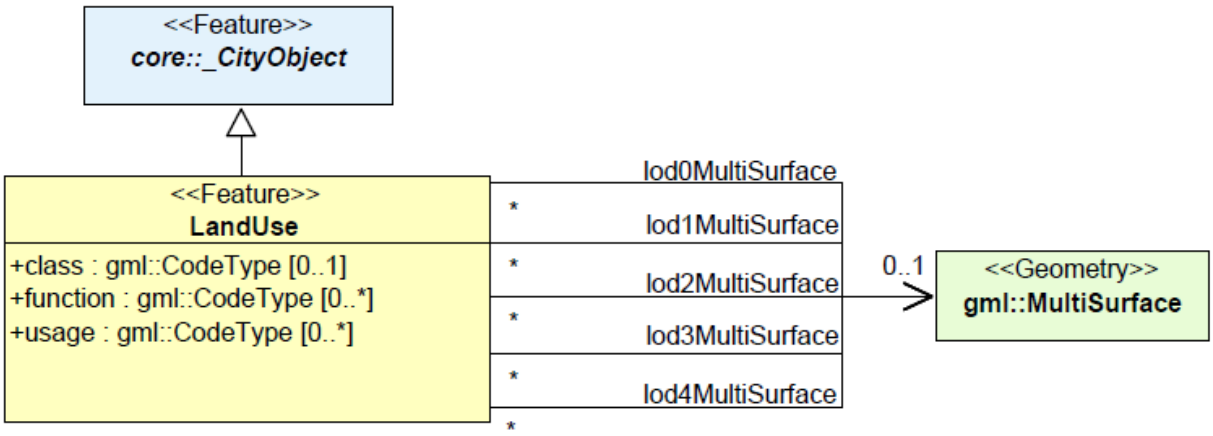


Abbildung 9.1: CityGML-UML-Diagramm des Moduls *LandUse* [Gröger et al., 2012]

9.2 Verwendete Daten

Für die Repräsentation der Parks stehen zwei verschiedene Datensätze zur Verfügung. Sie werden vom *Department of Information Technology and Telecommunications* (DoITT) sowie dem *Depart-*

ment of Parks and Recreation (DPR) bereitgestellt.
Die Geometrien der Daten sind in beiden Datensätzen als zweidimensionale Polygone gegeben. Attribute wie Name des Parks und Parknutzung sind in beiden Datensätzen gegeben. Zusätzlich liegt eine einheitliche *park_number* für jeden Park in beiden Datensätzen vor. Der Datensatz *PARK*[1] ist von seinen Umrissen der Parkanlagen detaillierter dargestellt als der Datensatz *DPR_ParksProperties*. Zusätzlich liegen die Parkgrenzen sowie innerhalb liegende Elemente wie Spielplätze oder Sportanlagen getrennt vor. Unter anderem beinhaltet er auch Grünstreifen an Straßenflächen. Der Datensatz *DPR_ParksProperties*[2] enthält zusätzlich Informationen zur Adresse inklusive Stadtteil und ob die Parkanlagen an ein Gewässer angrenzen.
In der folgenden Tabelle sind in den ersten beiden Spalten die verwendeten Attribute der jeweiligen Datensätze aufgelistet. Die Codelisten der Parks sind in Anhang B.5 mit Bedeutung aufgelistet.

Park [DoITT]	**DPR_ParksProperties** [DPR]	**CityGML Name**
PARK_NAME	SIGNNAME	gml_name
FEATURE_CODE		citygml_function
LANDUSE (lückenhaft)	TYPECATEGORY	citygml_usage/citygml_class
PARKNUMBER	GISPROPNUMBER	park_number
	WATERFRONT	waterfront
	BOROUGH	borough_name
	LOCATION	location
		park_area
		park_area_units
		part_of_park_id
detaillierter aufgelöste Geometrie	*grober aufgelöste Geometrie*	

Tabelle 9.1: Attribute der Datensätze der Parkanlagen [CityGML Notation] [1,2]

9.3 Umsetzung

Von den Datensätzen werden zunächst jeweils interessante Attribute ausgewählt und diese in vordefinierte CityGML-Attribute umbenannt oder als generische Attribute neu erzeugt. Da in beiden Datensätzen größtenteils die selben Parkanlagen enthalten sind, aber unterschiedliche Attribute dazu vorliegen, sollen die Daten kombiniert werden. Ziel ist es, dabei alle doppelt vorkommenden Parkgebiete zusammenzufassen und redundante Sachdaten und Geometrien zu vermeiden.

Die Kombination der beiden Datensätze ist über eine einheitliche *park_number* möglich. Dazu werden die Außengrenzen der Parks aus dem Datensatz *PARK* mit den Polygonen des Datensatzes *DPR_ParksProperies_001* zusammengeführt, sodass die Geometrie aus dem Datensatz *PARK*

[1] https://data.cityofnewyork.us/Recreation/M-m-kas-Meadow/8c29-4nxt (16.10.2015)
[2] https://data.cityofnewyork.us/City-Government/Parks-Properties/rjaj-zgq7 (15.10.2014)

zusätzlich zu den eigenen die Attribute aus dem Datensatz *DPR_ParkProperties_001* erhält. Vorrangig wird die Geometrie aus dem Datensatz *PARK* gewählt, da diese detaillierter ist als die Geometrie des anderen Datensatzes.
In Tabelle 9.1 sind die verwendeten Attribute in den ersten beiden Spalten angegeben. Die blau geschrieben Attribute werden im Fall einer Kombination vorrangig verwendet, da diese vollständiger sind, im Vergleich der beiden Datensätze. In der dritten Spalte sind die endgültigen Attributnamen angegeben. In dem Anhang B.5 sind die Codierungen für die Attribute *citygml_class*, *citygml_function* und *citygml_usage* angegeben.

In einem weiteren Schritt werden die Flächen innerhalb eines Parkes, wie Fußballfeld oder Golfplatz, mit den Parkgrenzen verschnitten und so die Zusammengehörigkeit der Parks ermittelt. Das Attribut *part_of_park_id* zeigt auf die *gml_id* der zugehörigen Parkgrenze eines Objektes. Bei den *externalReferences* wird besonders darauf geachtet, dass nur auf Datensätze und Objekte verwiesen wird, die letztendlich auch in dem End-Objekt enthalten sind.

Insgesamt liegen im Orginal 14 674 Parkobjekte in beiden Datensätzen vor. Durch die Kombination der Daten bleiben 10 562 Objekte übrig. Um den Parkanlagen von dem DGM die Höhen zu übertragen, werden diese, wie auch die Gewässer und Flurstücke zuvor, ab einer Größe von 40 000m^2 in Segmente aufgeteilt. Das Attribut *tiled_park* vermerkt, ob ein Park-Objekt geteilt wurde oder nicht. Die Höhenübertragung von dem DGM auf die Parkflächen erfolgt wie in Kapitel 6.3.1 in den Schritten 1 und 2 beschreiben. Eine weitere Bearbeitung der Höhenwerte ist nicht nötig.

10 Visualisierung der Ergebnisse

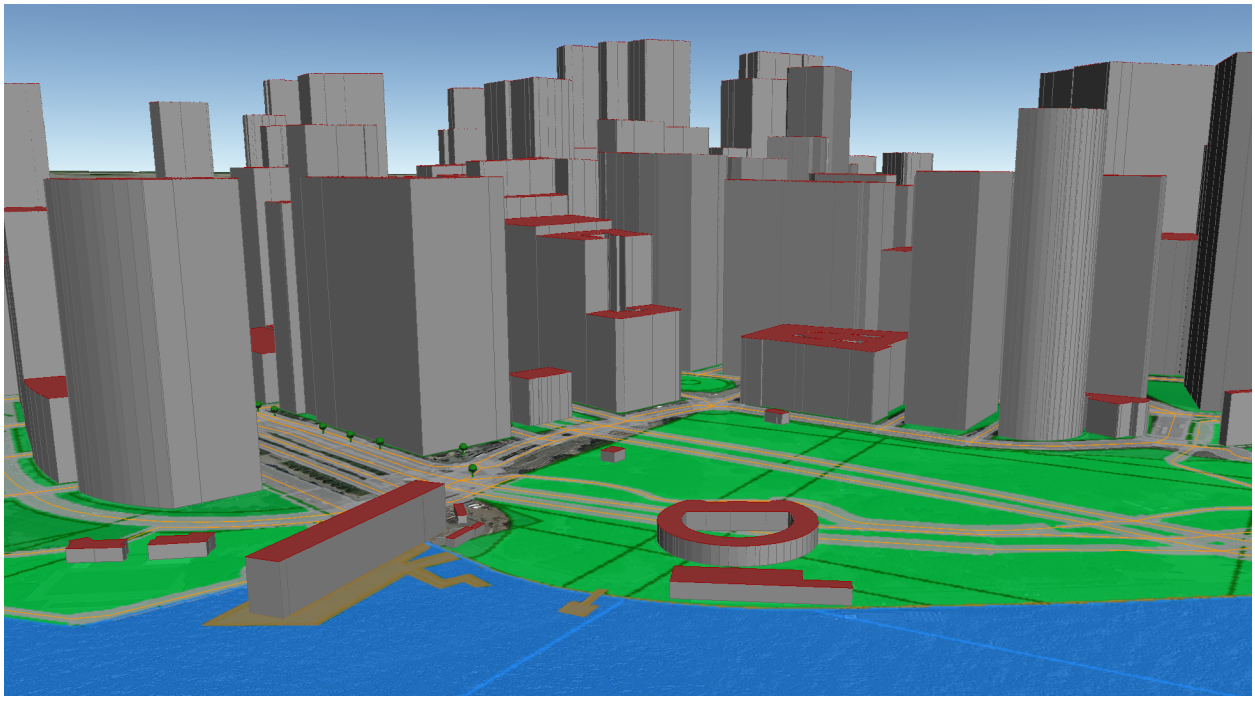


Abbildung 10.1: Ausschnitt des Stadtmodells enthält die generierten Objekte: Gebäude, Flurstücke, Gewässer, Straßen, hydrologische Infrastruktur und Bäume

In Abbildung 10.1 ist ein Ausschnitt des generierten Stadtmodells mit seinen Objekten zu sehen. Die Objekte können als KML-Dateien beispielsweise im *3DCityDB Webclient* oder auch in Google Earth visualisiert werden. Im Folgenden wird hierauf eingegangen und verschiedene Ausschnitte von Objekten dargestellt.

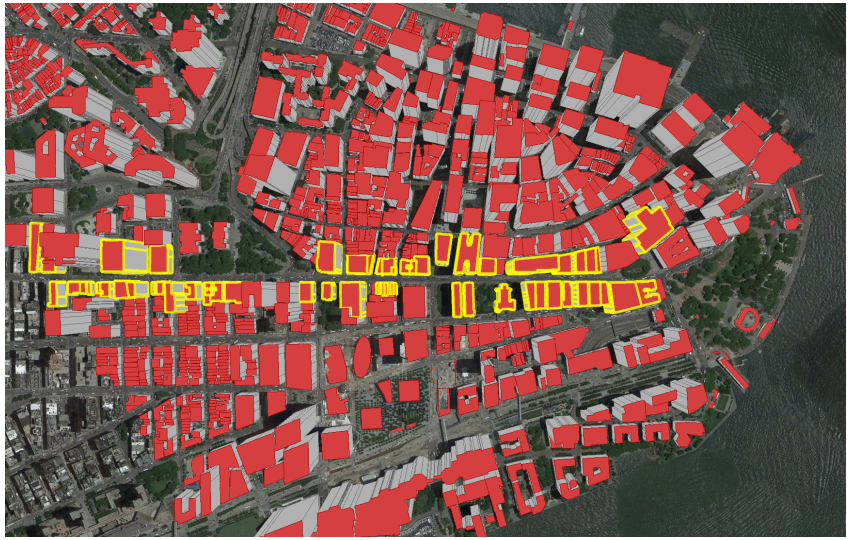
Für die Visualisierung des Stadtmodells im *3DCityDB Webclient* mit dem Google Earth-Plugin wird ein Ausschnitt generiert. Bei den Daten handelt es sich um KML-Dateien, welche die Geometrie der einzelnen Objekte repräsentieren. Um zusätzliche Abfragen auf Grundlage der Attribute der Objekte zu gestalten, werden diese in Spreadsheets hinterlegt. Im Folgenden werden Beispiele hierfür gezeigt.
In Abbildung 10.2 (a) werden alle Häuser selektiert, die laut ihrer Adresse am Broadway liegen. Dabei ist zu erkennen, dass längst nicht alle Gebäude, die am Broadway liegen, markiert sind. Dies ist möglich da ein Gebäude mehrere Adressen haben kann, die in verschiedenen Straßen liegen, beispielsweise bei einem Eckhaus. In den Spreadsheets wird nur die erste gespeicherte Addresse hinterlegt. Außerdem besteht auch die Möglichkeit, dass für ein Gebäude keine Adresse vorhanden ist. In der Abbildung 10.2 (b) sind die Gebäude markiert, die höher als 200 Meter sind. Die Gebäudehöhen können sowohl für Notfallszenarien (wie werden höheren Stockwerke

erreicht?) als auch für Abschattungsanalysen für GNSS-Positionierung und Verschattungsanalysen für Solarpotentiale interessant sein.
In Abbildung 10.2 (c) werden die Bäume, deren Position auf dem Gehweg liegt, gelb hervorgehoben. Selektiert wird hierbei nach dem Attribut *pit*. In diesem Ausschnitt sieht man, dass die Position der Bäume mitten auf der Verkehrsfläche der Straße angegeben ist, auch wenn die eigentliche Position laut dem Attribut *pit* auf dem Gehweg sein sollte. In Abbildung 10.2 (d) sind sämtliche Bäume markiert, deren Gesundheitszustand hervorragend ist. Diese Abfragen können für die Planung der Pflege der Stadtbäume genutzt werden. Hilfreiche Abfragen wären, ob beispielsweise abgestorbene Äste oder ganz abgestorbene Bäume entfernt werden müssen.
Für die Gebäude werden Flurstückinformationen hinterlegt, wenn nur ein Gebäude auf einem Flurstück steht. In Abbildung 10.3 sind diese Flurstücke markiert. Es ist zu sehen, dass dies für einen Großteil der Flurstücke zutrifft und die Gebäude um viele interessante Attribute erweitert werden können.

Zusätzlich ist es möglich, die KML-Dateien auch in Google Earth darzustellen, wobei die zugehörigen Attribute der Objekte in sogenannten „Ballons“ aufgelistet werden. Die Werte der Attribute stehen entweder offen da, wie die GML_ID, oder können durch einen Mouse-Over-Effekt eingeblendet werden. Eine Abfrage in Google Earth ist im Gegensatz zur Visualisierung im *3DCityDB Webclient* nicht möglich.
In Abbildung 10.4 (a) sind die Gebäude als extrudierte Körper dargestellt. Dabei werden die Wandflächen grau und die Dachflächen rot dargestellt. Die hellgrauen senkrechten Linien unterteilen die Wandflächen in Segmente. Diese werden durch die Punkte, die den Gebäudegrundriss definieren, erzeugt. Im Ballon sind die Attribute zu dem Gebäude-Objekt aufgelistet.
In Abbildung 10.4 (b) sind die Bäume dargestellt. Diese werden als sogenannte Lollipops modelliert. Die einzelnen Flächen der Kugel und des Zylinders sind erkennbar. Die dunkle Färbung der unteren Hälfte der Kugel ist vom Blickwinkel auf das Objekt abhängig und ist nicht immer sichtbar. Die Position der Bäume liegt auf der Straßenfläche. Im Satellitenbild ist erkennbar, dass rechts und links von der Straße noch mehr Bäume stehen, die nicht modelliert sind. Der Grund dafür ist, dass im Datensatz lediglich Bäume vorhanden sind, die auf öffentlichen Straßenflächen stehen, jedoch keine, die in Parkanlagen oder auf privaten Grundstücken wachsen.

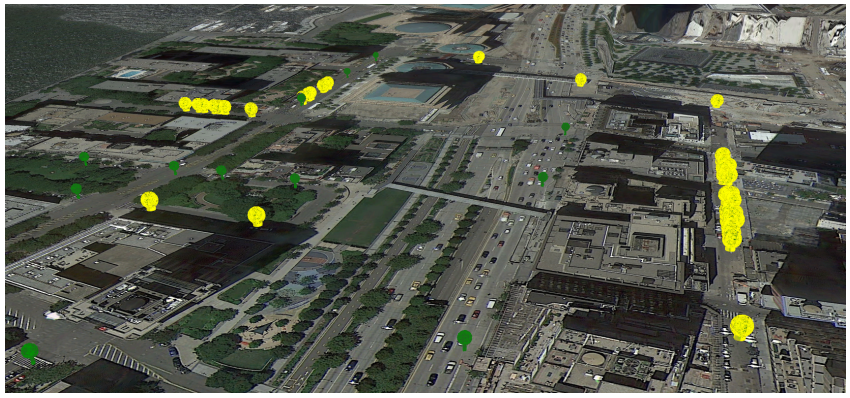
In Abbildung 10.5 (a) sind die Parkanlagen und in (b) die Flurstücke dargestellt. Beide werden in CityGML dem Modul *LandUse* zugeordnet und können somit mit dem Importer/Exporter Tool nicht getrennt aus der DB exportiert werden. In Google Earth werden beide Datensätze in einem Layer angezeigt. Dies ist in Abbildung 10.5(a) erkennbar, da die Transparenz bei einer Überlagerung der Flächen abnimmt. Das rührt daher, dass für Parkanlagen sowohl geometrische Informationen aus den Datensätzen der Parks wie auch der Flurstücke enthalten sind. Des Weiteren sind die Kachelgrenzen für zu große Flächen beider Datenquellen sichtbar, welche für die Generierung erforderlich waren. Über eine Filterung des Attributes *citygml_class* können Parkflächen von den Grundstücken selektiert werden. Wie in Abbildung 10.5 (a) abgebildet, werden Grünstreifen den Parkflächen zugeordnet. In Abbildung 10.5 (b) ist die Vielfalt der vorhanden Attribute für die Flurstücke erkennbar.

Die Gewässer werden in Abbildung 10.6 (a) dargestellt. Wie schon bei den Flurstücken und Parkflächen erwähnt, werden auch Gewässer zu großen Ausmaßes gekachelt. Dies ist bei der Wasserfläche

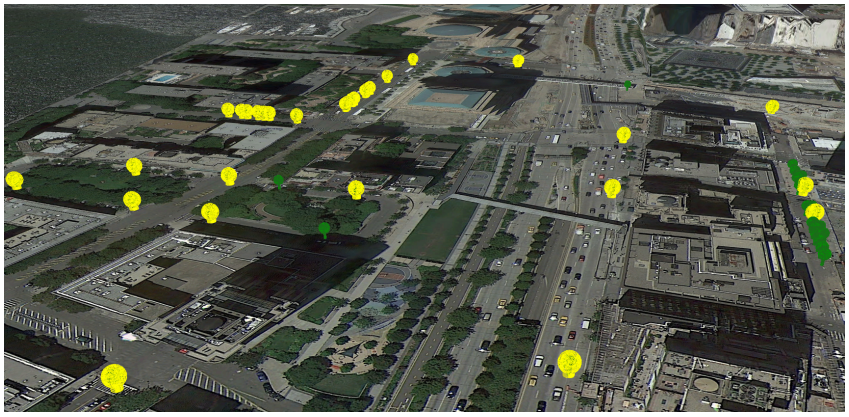


(a) Häuser am Broadway

(b) Gebäude höher als 200 Meter



(c) Bäume stehen auf Gehweg



(d) Gesunder Baum

Abbildung 10.2: Die „gehighlighteten “ Objekte stellen das jeweilige Ergebnis der folgenden Abfragen dar:
(a) Selektiere alle Gebäude mit $ADDRESS_STREET = BROADWAY$
(b) Selektiere alle Gebäude mit $citygml_measured_height > 200$
(c) Selektiere alle Bäume mit $pit = 1$
(d) Selektiere alle Bäume mit $tree_condition = 1$

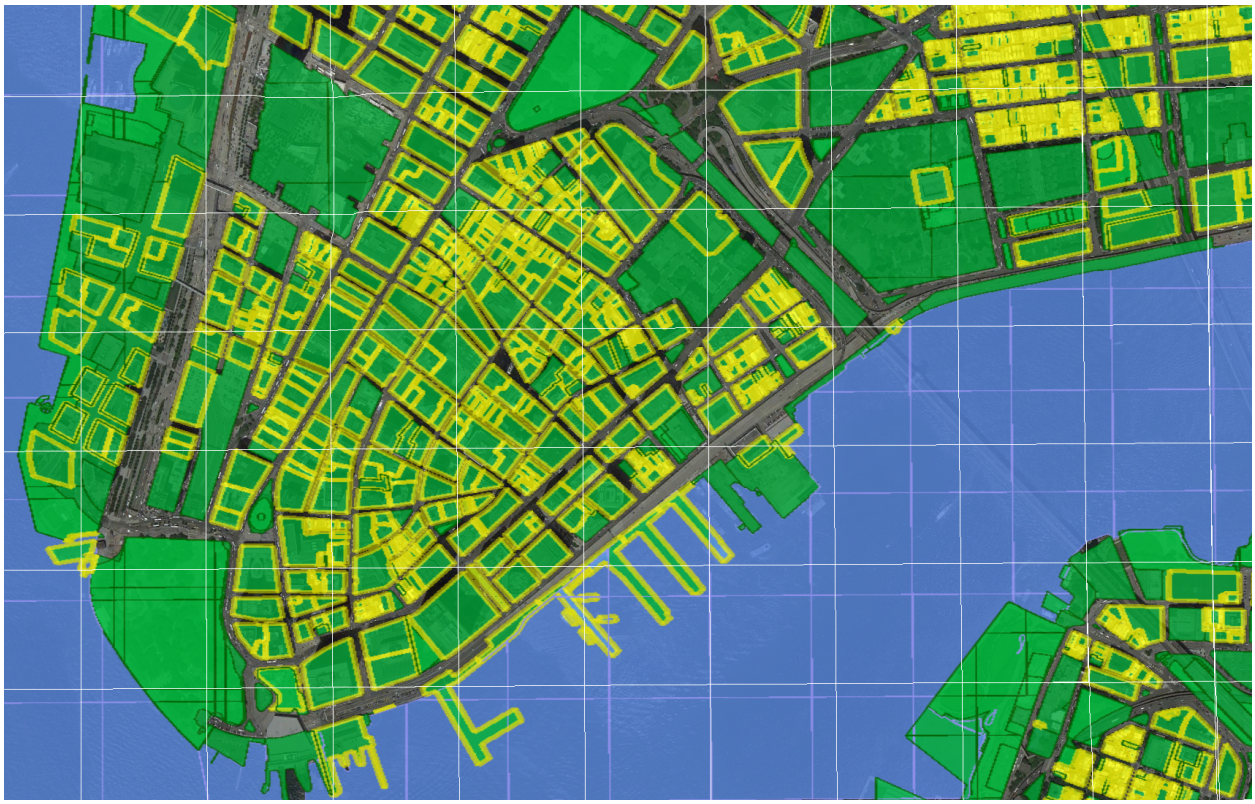
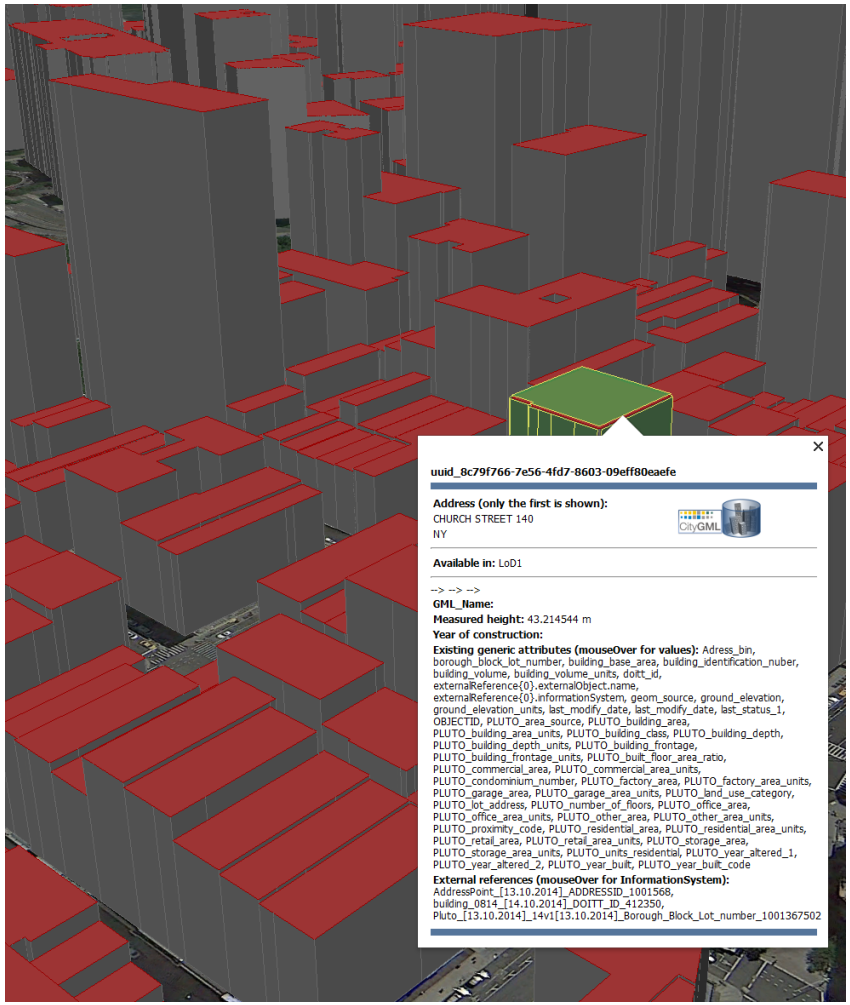


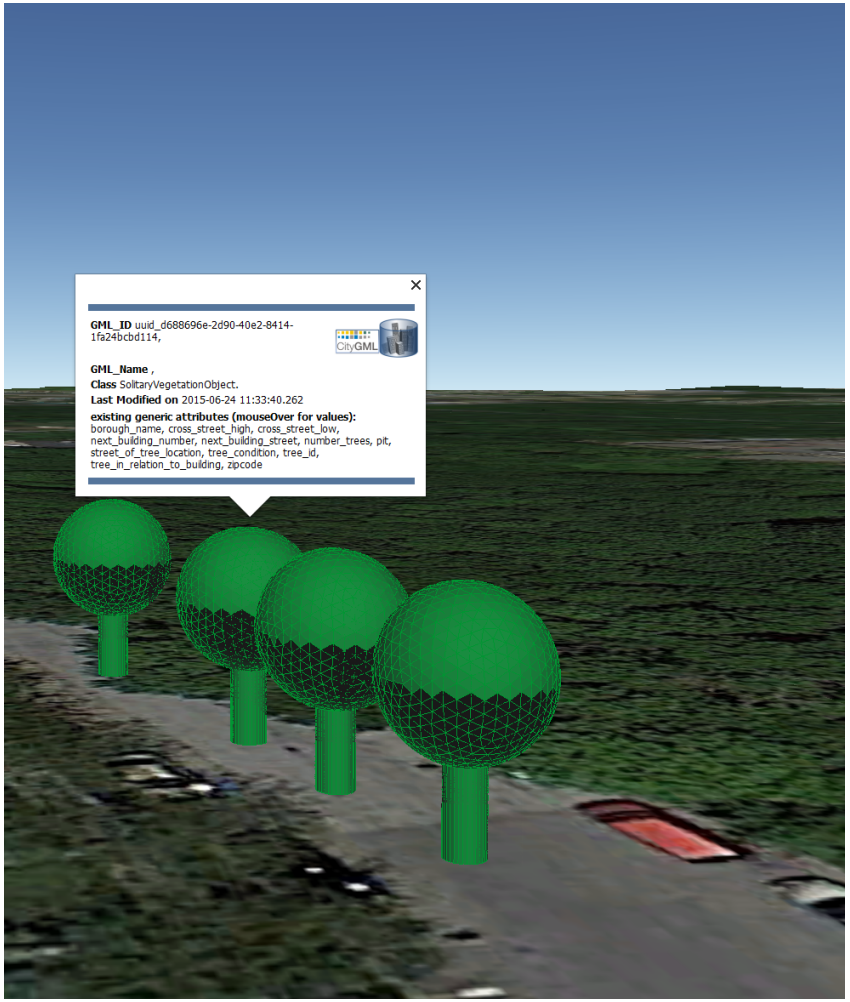
Abbildung 10.3: Selektiere alle Flurstücke mit $number_of_buildings = 1$

auf der rechten Seite der Fall. Der See auf der linken Seite überschreitet den Grenzwert nicht und wird somit im Gesamten modelliert. In Abbildung 10.6 (b) sind die hydrologischen Infrastrukturen dargestellt. In diesem Fall handelt es sich um eine Anlegestelle, an der im Satellitenbild Boote liegen.

Die Abfragen und Bilder zeigen wie umfangreich der Informationsgehalt dieses Stadtmodelles ist. Jedoch ist in dem *3DCityDB Wbclient* keine Abfrage in Kombination verschiedener Layer möglich. Die möglichen Abfragen erstrecken sich bisher nur auf die Abfrage und Kombination der Attribute. Für ein Stadtmodell gilt: es ist nur so gut wie seine Daten sind. Je mehr Objekte mit möglichst vielen Attributen verfügbar sind um so reichhaltiger und aussagekräftiger werden die abgeleiteten Informationen. Das generierte Stadtmodell dient als Grundlage für weitere Anwendungen und Analysen.

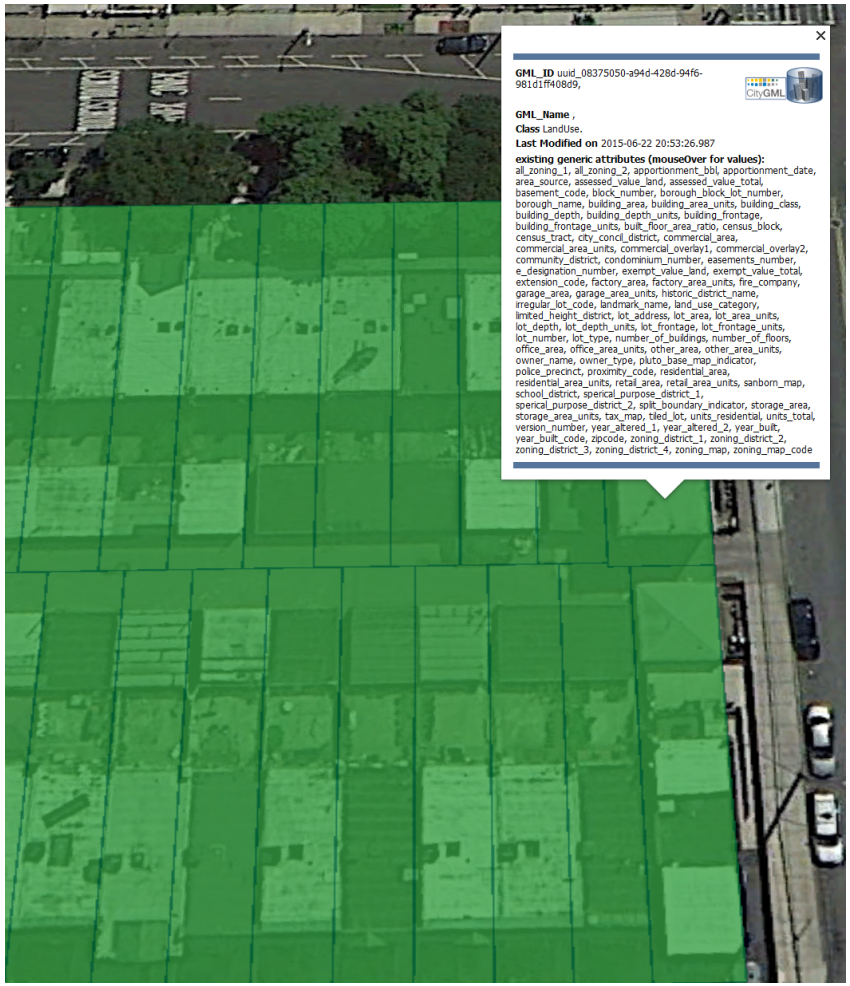


(a) Gebäude



(b) Bäume

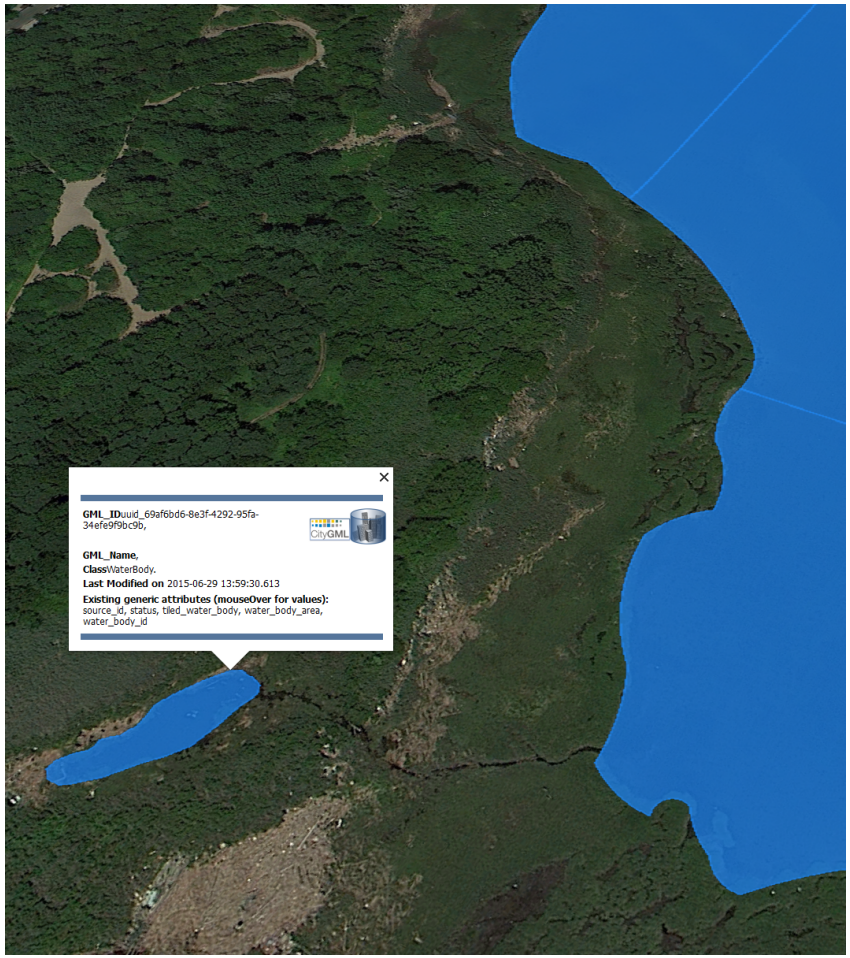
Abbildung 10.4: Geometrie und Attribute der Gebäude und Bäume in Google Earth visualisiert



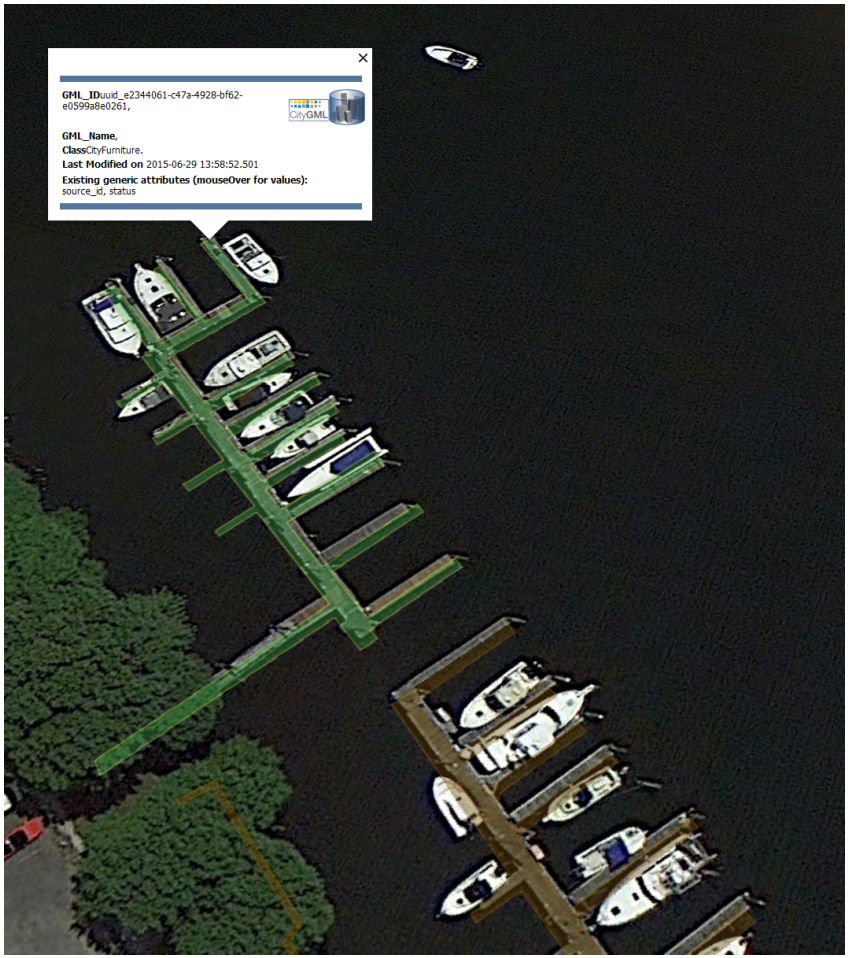
(a) Flurstücke

(b) Park

Abbildung 10.5: Geometrie und Attribute der Flurstücke und Parks in Google Earth visualisiert



(a) Gewässer



(b) Gewässerstrukturen

Abbildung 10.6: Geometrie und Attribute der Gewässer(-strukturen) in Google Earth visualisiert

11 Fazit und Ausblick

Verfasser: B. Burger und B. Cantzler

Das Ziel der zwei Masterarbeiten, ein semantisches 3D-Stadtmodell für die gesamte Stadt New York im CityGML-Format im LoD 1 zu erstellen, wurde erfolgreich umgesetzt. Dieses beinhaltet die folgenden Objektarten, die in der 3DCityDB vorliegen: Bäume, Flurstücke, Gebäude inklusive Adressen, Gewässer(-strukturen), Straßen, Parks, Zonierung. Dabei befasste sich die vorliegende Arbeit mit der der Generierung des DGMs, der Bäume, der Flurstüke, den Gebäuden inklusive der Adressen, den Parks sowie der Gewässer und der hydrologischen Infrastruktur. Die ergänzende Arbeit von B. Burger beschäftigte sich ebenfalls mit der Erzeugung des DGMs und weiterhin mit den Straßen und der Zonierung.

In dieser Arbeit wurde zunächst ein Überblick über den Begriff *Open Data* gegeben sowie ein Vergleich der Handhabung dieser in Hamburg und New York gemacht. Des Weiteren wurde die Erzeugung des DGMs aus einem DEM erläutert. Die Herausforderung bestand darin, dass das TIN nicht im Ganzen erzeugt wurde, sondern mit Hilfe von 35 153 einzelnen Kacheln mit einer Größe von ca. 250 x 250 m. Diese müssen exakt aneinander anschließen und dürfen keinen Sprung in der Höhe aufweisen. Die Gebäude wurden mit zusätzlichen Attributen aus dem Datensatz der Adressen sowie ausgewählter Attribute der Flurstücke angereichert. Zusätzlich wurde am Beispiel der Gebäude die Höhenübertragung aus dem DGM auf die einzelnen Objekte gezeigt. Die Modellierung der Bäume erfolgte als implizite und explizite Geometrien in Form von Lollipops. Anschließend erfolgte die Integration des DGMs und der Gewässer. Die Generierung der Grünanlagen erfolgte durch die Kombination zweier voneinander unabhängiger Datensätze mit verschiedenen Attributen und Objekten. Schließlich wurden die Ergebnisse in *Google Earth* und im *cloud-basierten 3D-Webclient* visualisiert.

Nicht alle Berechnungen konnten bis zum Abgabedatum der Arbeiten abgeschlossen werden. Das größte Problem bereitete dabei das digitale Geländemodell. Wie in Kapitel 4.3.1 bereits erwähnt, dauerte die Berechnung des gekachelten TIN und dessen Speicherung im CityGML-Format nahezu 10 Tage. Der Import dieser 35.153 Kacheln in die DB benötigte weitere 20 Tage – unter Zuhilfenahme der Import/Export-Software. Dieser Zeitraum beinhaltet jedoch nicht die Aktivierung der räumlichen und normalen Indizes in der DB. Letztere wurden innerhalb von 22 Stunden, die räumlichen werden hingegen – zum Zeitpunkt der Abgabe – bereits seit einem Monat aktiviert. Dabei muss allerdings erwähnt werden, dass es sich bei dem Computer, der die Oracle-Datenbank beinhaltet, nicht um einen Hochleistungsserver handelt (vgl. Kapitel 2.3).

Um in Zukunft solch eine Problematik – die Erzeugung der räumlichen Indizes für rund 5 Mrd. Dreiecksflächen – zu vermeiden, könnten die Daten in der DB partitioniert werden. Demzufolge würden Tabelleinhalte auf mehrere Tablespaces verteilt werden, wodurch Indizes nicht mehr für eine einzige

große Tabelle, sondern jeweils für die Partitionen angelegt werden können. Dies verbessert die Handhabbarkeit großer Daten und hat keinen Einfluss auf Abfragen. Speziell für die Aufteilung räumlicher Indizes gibt es in Oracle Spatial das *range partitioning* [Oracle, 2010].

Des Weiteren wäre zu überlegen, inwieweit es sinnvoll ist, ein DGM mit seinen einzelnen Dreiecksflächen und nicht als ein gesamtes TIN in der 3DCityDB abzuspeichern. Da Oracle sogar einen eigenen Datentyp für TINs anbietet (SDO_TIN), sollte davon ausgegangen werden können, dass dieser ein TIN effizienter verwaltet als eine Tabelle mit einer Unmenge an einzelnen Dreiecksflächen. Demzufolge sollte überlegt werden, ob CityGML bzw. die 3DCityDB in Zukunft diesen Datentyp unterstützt. Dabei sollte jedoch kein einzelnes großes TIN für z.B. die gesamte Stadt New York erzeugt werden, sondern vorzugsweise mehrere kleinere TINs, wobei darauf zu achten ist, dass nebeneinander liegende Kachelgrenzen exakt aneinander anschließen.

Nachdem zum Zeitpunkt der Abgabe dieser Arbeit die Indizes für das DGM noch nicht vollständig aktiviert sind, konnten die Objekte des Stadtmodells lediglich testweise auf das DGM gelegt werden, wodurch sie ihre Höheninformation erhalten. Sobald die Aktivierung der Indizes erfolgreich durchgeführt sein wird, kann dieser Prozess gestartet werden, der einzeln für jede Kachel und die darin liegenden Objekte angestoßen wird (vgl. Kapitel 6.3.1). Diese Berechnung könnte allerdings ungefähr zwei Monate in Anspruch nehmen, da bereits zehn Kacheln eine Stunde benötigen.

In dieser Arbeit wurden alle Datensätze semantisch und geometrisch für das Stadtmodell in CityGML generiert und in eine Datenbank importiert. Aus dieser können Ausschnitte mit den gewünschten Objekten gelesen und in Google Earth oder dem *3D-Webclient* visualisiert oder auf andere Weise weiterverwendet werden. Der Fokus lag auf der Erstellung eines gesamten 3D-Stadtmodells mit einer Vielzahl an Objektarten und nicht auf den geometrischen Feinheiten. Demzufolge stellt es eine gute Grundlage dar auf der weiterführende Arbeiten aufbauen können.
für weiterführende Arbeiten dar, auf der aufgebaut werden kann.

Die Darstellung der Bäume könnte vereinfacht werden, da bei der verwendeten Repräsentation sehr viele Flächen benötigt wurden. Denkbar wäre z. B. die Krone der Bäume nicht durch eine angenäherte Kugel – bestehend aus 1 314 Flächen (inklusive der Flächen des Zylinders) – sondern mit Hilfe eines Dodekaeders darzustellen, der lediglich 12 Flächen benötigt. Zusätzlich könnte die Geometrie des Stammes durch z. B. einen Quader generalisiert werden. Durch die Vereinfachung können die Rechenzeiten optimiert wie auch die Ladezeiten bei der Visualisierung und die Zeit der Generierung reduziert werden.

Da die Gewässer überwiegend als sehr große Flächen vorliegen, werden sie in Kacheln unterteilt, um sie verarbeiten zu können. Des Weiteren werden diese auf das DGM gelegt, um ihnen die entsprechenden Höheninformation zu übertragen. Dabei erhalten flächenhafte Gewässer ein einheitliches Niveau der Wasseroberfläche. Für Fließgewässer könnte in Zukunft eine Integration der Gewässer und des DGM vorgenommen werden. Dadurch könnte die geometrische Repräsentation semantisch korrekt dargestellt werden.

Die Gebäude enthalten im Datensatz schon reichhaltige Informationen, diese werden mit den Attributen der Flurstücke erweitert. Allerdings liegen diese noch nicht für alle Gebäude vor. Zusätzlich lassen sich aus den gegebenen Informationen weitere ableiten, beispielsweise die Volumenbestimmung eines Gebäudes. Das Modell der Gebäude könnte noch ausgebaut werden, indem weitere Datensätze

mit diesem kombiniert werden, so könnten weitreichende Analysen und Abfragen ermöglicht werden. Beispielsweise könnte über die Kombination der Einwohnerzahl, die Anzahl der Wohnungen und den Flächen über Wohnflächen ein durchschnittlicher Wert der Wohnfläche pro Kopf bestimmt werden.

Die Visualisierung der Daten ist in Google Earth flächendeckend für NYC möglich. Im 3D-Webclient wird hingegen lediglich ein Ausschnitt zur Demonstration wiedergegeben, da die Spreadsheets nicht für größere Datenmengen (> 2 Mio. Zellen) ausgelegt sind. Der virtuelle Globus basiert auf einem Google Earth Plugin, da mit Hilfe dessen mehrere Layer, d.h. Datensätze, gleichzeitig dargestellt werden können. Jedoch soll das Modell in naher Zukunft ebenfalls mit Hilfe des virtuellen Globus Cesium (WebGL-basiert) repräsentierbar sein. Dieser bietet u.a. die Vorteile, dass es sich um ein *Open Source*–Produkt handelt und ein eigenes Geländemodell eingebunden werden kann, wodurch Objekte eines 3D-Stadtmodells mit ihren absoluten Höhen dargestellt werden können.

Die erwähnten Rechenzeiten sind zwar immens, jedoch muss berücksichtigt werden, dass sowohl die Daten als auch die dementsprechenden Rechenzeiten sich nicht auf einen kleinen Ausschnitt, sondern auf die gesamte Stadt New York mit einer Fläche von ca. 1.000 km^2 beziehen. Für dieses Ausmaß wurden flächendeckend für die oben genannten Objektarten sowohl CityGML- als auch KML-Dateien erzeugt, welche ca. 3,5 Mio. Objekte, knapp 200 Attribute und ca. 1,6 TB (CityGML – wovon das DGM allein 1,45 TB umfasst) bzw. 19 GB (KML – exklusive DGM) Speicher aufweisen.

Das erzeugte 3D-Stadtmodell von NYC stellt eine gute Grundlage für weitere Arbeiten dar. Da in dieser Arbeit die Generierung der Datensätze für die gesamte Stadt NY maßgeblicher war als z.B. eine detaillierte Repräsentation der Geometrie, könnte letztere beispielsweise in Bezug auf die Straßen oder Gewässer perfektioniert werden. Darüber hinaus könnte das Modell mit weiteren geometrischen- oder Sachinformationen – aus dem NYC Open Data Portal – angereichert werden. Dies wäre sowohl für LoD 1 als auch für höhere Detaillierungsstufen in Betracht zu ziehen. In den Kapiteln „Visualisierung der Ergebnisse“ beider Masterarbeiten wurden bereits einige Beispiele aufgezeigt, für welche Anwendungen das Stadtmodell z.B. mit Hilfe des 3D-Webclient genutzt werden könnte. Überdies könnten weitere Analysen, z.B. Sonneneinstrahlungsanalysen, die unabhängig von den Attributen einer Stadt, allerdings abhängig vom CityGML-Format sind, durchgeführt werden. Da das erzeugte *3D-Stadtmodell der Stadt New York auf* Open Data basiert, soll es in Zukunft ebenfalls der Öffentlichkeit frei zur Verfügung stehen.

Anhang

A Herkunft der verwendeten Datensätze im Überblick

B Codelisten/ Informationen Attribute

- Gebäude
- Bäume
- Spezies Liste Bäume
- Hydrologische Infrastruktur
- Gewässer
- Parks

C FME-Workspaces zum Digitalen Geländemodell

D Geometrie und Attribute der einzelnen Objektarten

A Herkunft der verwendeten Datensätze im Überblick

Tabelle A.1: Überblick über die Herkunft der Datensätze.

Datensatz	**Name des Eingangsdatensatzes**	**Behörde**
Adressen	NYC Address Points	DoITT
Bäume	Street Tree Census (Staten Island) Street Tree Census (Brooklyn) Street Tree Census (Bronx) Street Tree Census (Queens) Street Tree Census (Manhattan)	DPR
DGM	1 foot Digital Elevation Model (DEM)	DEP & DoITT
Flurstücke	MapPLUTO – Bronx MapPLUTO - Brooklyn MapPLUTO – Manhattan MapPLUTO - Queens MapPLUTO - Staten Island	DFDTM
Gebäude	Building Footprints	DoITT
Gewässer	Hydrography	DoITT
Gewässerstrukturen	Hydrography-structures	DoITT
Parks	Mèmèkas Meadow Parks Properties	DoITT DPR
Straßen	LION File-Geodatabase	DCP
Zonierung	Borough Boundaries Census Tracts 2010 Zip Code Boundaries	DCP DCP DoITT

DCP Department of City Planning

DEP Department of Environmental Protection

DoITT Department of Information Technology & Telecommunications

DPR Department of Parks and Recreation

DFDTM Department of Finance's Digital Tax Map

B Codelisten

B.1 Übersicht über originalen Attribute der Gebäude [CityGML Notation]

CityGML Attribute	Beschreibung
building_identification_nuber	Building Identification Number, eindeutig zuordenbare Identifikationsnummer der Gebäude; Ausnahme bei Garagen, hier Nummern 1.000.000, 2.000.000, 3.000.000, 4.000.000, 5.000.000 je nach Stadtteil
borough_block_lot_number	Borough (1. Ziffer), Block (2.-6.Ziffer) und Lot (7.-10. Ziffer) Nummer
citygml_function	Unterscheidung zwischen Gebäude und Garage (5110)
citygml_measured_height	Höhe des Daches [Fuß]; berechnet aus Differenz Gebäudehöhe und Höhe aus interpolierten TIN; Höhe über Boden, nicht über dem Meeresspiegel
citygml_usage	Nutzung des Gebäudes, Unterscheidung zwischen Polizei (0), Feuerwehr (1), Schule (2), Krankenhaus (3) und Flughafen (13)
citygml_year_of_construction	Jahr der Erbauung des Gebäudes
doitt_id	ID von Department gegeben
geom_source	Gibt die Quelle der Daten an, photogrammetrisch oder manuell
ground_elevation	Bodenhöhe, ermittelt aus LiDAR-Daten und über photogrammetrische Methoden; Vergleich der Werte des Attributes *GROUND_ELEVATION* und den Werten aus dem interpolierten LiDAR-Daten erfolgt erst mit einheitlichem Koordinatensystem

B.2 Attribute und Codelisten der Bäume [CityGML Notation]

CityGML Attribute	Beschreibung
borough_name	Stadtbezirk
citygml_trunk_diameter	Durchmesser auf Brusthöhe (4,5 Fuß) [m]
citygml_species	Baumspezies
cross_street_high	Kreuzende Straße am Ende der höheren Adresse
cross_street_low	Kreuzende Straße am Ende der niedrigeren Adresse
externalReference{}. informationSystem	Verweist auf Quelle des Datensatzes
externalReference{}.name	Verweist auf Objekt aus Datensatz
gml_id	
gml_parent_id	
next_building_number	Hausnummer des nächst stehenden Gebäudes zum Baum
next_building_street	Straße des nächst stehenden Gebäudes zum Baum
number_trees	Anzahl der Bäume einer Adresse
pit	1 sidewalk 2 continuous 3 lawn
street_of_tree_location	Straße, in der Baum steht
tree_condition	Gesundheitszustand (siehe Tabelle tree_condition)
tree_id	Original ID
tree_in_relation_to_building	1 Front 2 Side 3 Rear 4 Across 5 Adjacent 6 Median 7 Side/Across 8 Side/Median 9 Assigned
zipcode	Postleitzahl

tree_condition

Code	Bedeutung
1	Excellent. Full, well balanced crown and limb structure; leaves normal size color; no dead or broken branches; trunk solid, bark intact.
2	Good. Crown uneven or misshapen; some mechanical damage to bark or trunk; some signs of insects or disease; leaves somewhat below normal size and quantity; some dead or broken branches (less than half of the tree).
3	Poor: large dead limbs with over one-half of the tree already dead or removed; large cavities; drastic deformities; leaves significantly below normal size and quantity; severe insect or disease damage.
4	Dead. Dead tree; leaves absent; twigs brittle.
5	Shaft. All branches removed; trunk left standing; sprouts may or may not be evident.
6	Stump. Stump shorter than breast height; leaves entirely absent or present only on stump sprouts.
7	Empty Tree Pit. Nothing there at all.

Tree Species List

Trees Count! New York City 2005 Street Tree Census

Species Code	Common Name	Scientific Name	
UNK	UNKOWN		
AC	MAPLE, OTHER	ACER	SPECIES
ACCA	MAPLE, HEDGE	ACER	CAMPESTRE
ACGI	MAPLE, AMUR	ACER	GINNALA
ACNE	BOXELDER	ACER	NEGUNDO
ACPA	MAPLE, JAPANESE	ACER	PALMATUM
ACPL	MAPLE, NORWAY	ACER	PLATANOIDES
ACPLCR	MAPLE, NORWAY-CR KNG	ACER	PLATANOIDES CRIMSON KING
ACPS	MAPLE, SYCAMORE	ACER	PSEUDOPLATANUS
ACRU	MAPLE, RED	ACER	RUBRUM
ACSA1	MAPLE, SILVER	ACER	SACCHARINUM
ACSA2	MAPLE, SUGAR	ACER	SACCHARUM
AECA	HORSECHESTNUT, RED	AESCULUS	X CARNEA
AEHI	HORSECHESTNUT	AESCULUS	HIPPOCASTANUM
AIAL	TREE OF HEAVEN	AILANTHUS	ALTISSIMA
ALGL	ALDER, EUROPEAN	ALNUS	GLUTINOSA
AM	SERVICEBERRY, OTHER	AMELANCHIER	SPECIES
BE	BIRCH, OTHER	BETULA	SPECIES
CABE	HORNBEAM, EUROPEAN	CARPINUS	BETULUS
CA	CATALPA	CATALPA	
CACA	HORNBEAM, AMERICAN	CARPINUS	CAROLINIANA
CA	HICKORY	CARYA	
CEAT	CEDAR, ATLAS	CEDRUS	ATLANTICA
CECA	REDBUD, EASTERN	CERCIS	CANADENSIS
CEJA	KATSURA TREE	CERCIDIPHYLLUM	JAPONICUM
CEOC	HACKBERRY	CELTIS	OCCIDENTALIS
CHVI	FRINGE TREE	CHIONANTHUS	VIRGINICUS
CLLU	YELLOWOOD	CLADRASTIS	LUTEA
COCO2	TURKISH HAZELNUT	CORYLUS	COLURNA
COFL	DOGWOOD, FLOWERING	CORNUS	FLORIDA
COKO	DOGWOOD, KOUSA	CORNUS	KOUSA
COMA	CHERRY, CORNELIAN	CORNUS	MAS
CR	HAWTHORN, OTHER	CRATAEGUS	SPECIES
DIVI	PERSIMMON, COMMON	DIOSPYROS	VIRGINIANA
EUUL	HARDY RUBBER TREE	EUCOMMIA	ULMOIDES
FAGR	BEECH, AMERICAN	FAGUS	GRANDIFOLIA
FASY	BEECH, EUROPEAN	FAGUS	SYLVATICA

FR	ASH, OTHER	FRAXINUS	SPECIES
FRAM	ASH, WHITE	FRAXINUS	AMERICANA
FRPE	ASH, GREEN	FRAXINUS	PENNSYLVANICA
GIBI	GINKGO	GINKGO	BILOBA
GLTR	HONEYLOCUST	GLEDITSIA	TRIACANTHOS
GYDI	KENTUCKY COFFEETREE	GYMNOCLADUS	DIOICUS
HADI	SILVERBELL	HALESIA	DIPTERA
HAVI	WITCH HAZEL	HAMAMELIS	VIRGINIANA
ILSP	HOLLY SPECIES	ILEX	SPECIES
JUNI	WALNUT, BLACK	JUGLANS	NIGRA
JUVI	EASTERN REDCEDAR	JUNIPERUS	VIRGINIANA
KOPA	GOLDENRAIN TREE	KOELREUTERIA	PANICULATA
LA	LARCH	LARIX	
LAWA	GOLDEN-CHAIN TREE	LABURNUM	WATERERI
LIST	SWEETGUM	LIQUIDAMBAR	STYRACIFLUA
LITU	TULIP TREE	LIRIODENDRON	TULIPIFERA
MA1	MAGNOLIA, OTHER	MAGNOLIA	SPECIES
MA2	CRABAPPLE	MALUS	SPECIES
MAAC	CUCUMBER TREE	MAGNOLIA	ACUMINATA
MAAM	MAACKIA,AMUR	MAACKIA	AMURENSIS
MAGR	MAGNOLIA, SOUTHERN	MAGNOLIA	GRANDIFLORA
MAPO	OSAGE-ORANGE	MACLURA	POMIFERA
MAPU	APPLE	MALUS	PUMILA
MEGL	DAWN REDWOOD	METASEQUOIA	GLYPTOSTROBOIDES
MO	MULBERRY	MORUS	
NYSY	BLACKGUM	NYSSA	SYLVATICA
OSVI	EASTERN HOP HORNBEAM	OSTRYA	VIRGINIANA
OXAR	SOURWOOD	OXYDENDRUM	ARBOREUM
PATO	ROYAL PAULOWNIA	PAULOWNIA	TOMENTOSA
PHAM	AMUR CORKTREE	PHELLODENDRON	AMURENSE
PI1	SPRUCE, OTHER	PICEA	SPECIES
PI2	PINE, OTHER	PINUS	SPECIES
PIAB	SPRUCE, NORWAY	PICEA	ABIES
PINI	PINE, AUSTRIAN	PINUS	NIGRA
PIPU	SPRUCE, COLORADO	PICEA	PUNGENS
PIRE	PINE, RED	PINUS	RESINOSA
PIRI	PINE, PITCH	PINUS	RIGIDA
PIST	PINE, EASTERN WHITE	PINUS	STROBUS
PISY	PINE, SCOTCH	PINUS	SYLVESTRIS
PITH	PINE, JAPANESE BLACK	PINUS	THUNBERGII
PLAC	LONDON PLANETREE	PLATANUS	ACERIFOLIA
PO	POPLAR, OTHER	POPLAR	SPECIES
PODE	COTTONWOOD, EASTERN	POPULUS	DELTOIDES
PONI	POPLAR, LOMBARDY	POPULUS	NIGRA
PR	CHERRY, OTHER	PRUNUS	SPECIES
PRCE	PLUM, PURPLELEAF	PRUNUS	CERASIFERA
PRSE1	CHERRY, BLACK	PRUNUS	SEROTINA
PRSE2	CHERRY, KWANZAN	PRUNUS	SERRULATA KWANZAN

PRVISH	CHOKECHERRY, SHUBERT	PRUNUS	VIRGINIANA SHUBERT
PYCA	PEAR, CALLERY	PYRUS	CALLERYANA
PZME	DOUGLAS FIR	PSEUDOTSUGA	MENZIESII
QU	OAK, OTHER	QUERCUS	SPECIES
QUAC	OAK, SAWTOOTH	QUERCUS	ACUTISSIMA
QUAL	OAK, WHITE	QUERCUS	ALBA
QUBI	OAK, SWAMP WHITE	QUERCUS	BICOLOR
QUCO	OAK, SCARLET	QUERCUS	COCCINEA
QUIM	OAK, SHINGLE	QUERCUS	IMBRICARIA
QULA1	OAK, TURKEY	QUERCUS	LAEVIS
QULA2	OAK, LAUREL	QUERCUS	LAURIFOLIA
QUMA1	OAK, BUR	QUERCUS	MACROCARPA
QUPA	OAK, PIN	QUERCUS	PALUSTRIS
QUPH	OAK, WILLOW	QUERCUS	PHELLOS
QURO	OAK, ENGLISH	QUERCUS	ROBUR
QURU	OAK, NORTHERN RED	QUERCUS	RUBRA
ROPS	BLACK LOCUST	ROBINIA	PSEUDOACACIA
SA	WILLOW SPECIES	SALIX	SPECIES
SAAL	SASSAFRAS	SASSAFRAS	ALBIDUM
SOAL	MOUNTAINASH, KOREAN	SORBUS	ALNIFOLIA
STJA1	SNOWBELL, JAPANESE	STYRAX	JAPONICA
STJA2	JAPANESE PAGODA TREE	STYPHNOLOBIUM	JAPONICUM
SYRE	JAPANESE TREE LILAC	SYRINGA	RETICULATA
TADI	BALDCYPRESS	TAXODIUM	DISTICHUM
TI	LINDEN, OTHER	TILIA	SPECIES
TIAM	LINDEN, AMERICAN	TILIA	SPECIES
TICO	LINDEN, LITTLE LEAF	TILIA	CORDATA
TITO	LINDEN, SILVER	TILIA	TOMENTOSA
TSCA	HEMLOCK, EASTERN	TSUGA	CANADENSIS
UL	ELM, OTHER	ULMUS	SPECIES
ULAM	ELM, AMERICAN	ULMUS	AMERICANA
ULPA	ELM, CHINESE	ULMUS	PARVIFOLIA
ULPR	ELM, ENGLISH	ULMUS	PROCERA
ULPU	ELM, SIBERIAN	ULMUS	PUMILA
ZESE	ZELKOVA, JAPANESE	ZELKOVA	SERRATA

B.3 Übersicht über Attribute der Gewässer [CityGML Notation]

CityGML Attribute	**Beschreibung**
citygml_class	2600 Lake/Reservoir 2610 Pond 2620 River 2630 Stream wider than 8 feet 2640 Wetland 2650 Beach/Shoreline 2660 Bay/Ocean
gml_name	
source_id	
status	
tiled_water_body	Gibt an ob ein Gewässer in Kacheln aufgeteilt wurde
water_body_area	Gibt die gesamte Fläche des (ungeteilten) Gewässers an
water_body_id	Identisch für Kacheln eines aufgeteilten Gewässers

B.4 Übersicht über Attribute der hydrologischen Infrastruktur [CityGML Notation]

CityGML Attribut	Bedeutung
citygml_class	1 pier 2 jetty 3 seawall
gml_id	
gml_parent_id	
status	
source_id	

B.5 Codeliste der Parks [CityGML Notation]

citygml_class

Code	Bedeutung
1010	Greenstreets
1020	Mall
1030	Park
1040	Triangle/Plaza

citygml_function

Code	Bedeutung
4900	Base Ball
4910	Court
4920	Football Field
4930	Soccer Field
4940	Golf Course
4950	Pool
4960	Running Track
4970	Skating Rinks
4980	Park Boundary

citygml_usage

Code	Bedeutung
2010	Event Area
2020	Lage Park Area
2030	Pending Acquisitions privately owned public space
2040	Retired N/A
2050	Triangle/Plaza
2060	Tracking Only
2070	Strip
2080	School Yard to Playground
2090	Buildings/Institutions
2100	Community Park
2110	Flagship Park
2120	Garden
2130	Historic House Park
2140	Jointly Operated Playground
2150	Lot
2160	Nature Area
2170	Neighborhood Park
2180	Parkway
2190	Playground
2200	Recreation Field/ Courts
2210	Undeveloped
2220	Waterfront Facility

C FME-Workspaces zum Digitalen Geländemodell

Name des Workspace	Kurzbeschreibung
DTM_in_OracleDB	erstellt DB-Tabellen für das Rasterbild und speichert dieses als gesamtes und gekachelt ab
DTM	Ausgangs-Workspace zur DGM-Generierung
DTM_TIN	berechnet das TIN für jeweils eine Kachel und speichert dieses im CityGML-Format ab
DTM_Buffer_from_OracleDB	liest aus der DB einen Ausschnitt des Rasterbilds aus, der für die TIN-Berechnung nötig ist

Name des Custom Transformer	Kurzbeschreibung
Tiling	bildet die Schleife, um das DGM kachelweise zu generieren

D Geometrie und Attribute der einzelnen Objektarten

In der ergänzenden Arbeit wurden die Straßen sowie die Zonierung des 3D-Stadtmodells generiert. Die Geometrien der Straßen werden in AbbildungD.1 in GoogleEarth visualisiert. In sogenannten Ballons werden die hinterlegten Attribute und ihre Werte der jeweiligen Straße aufgelistet.

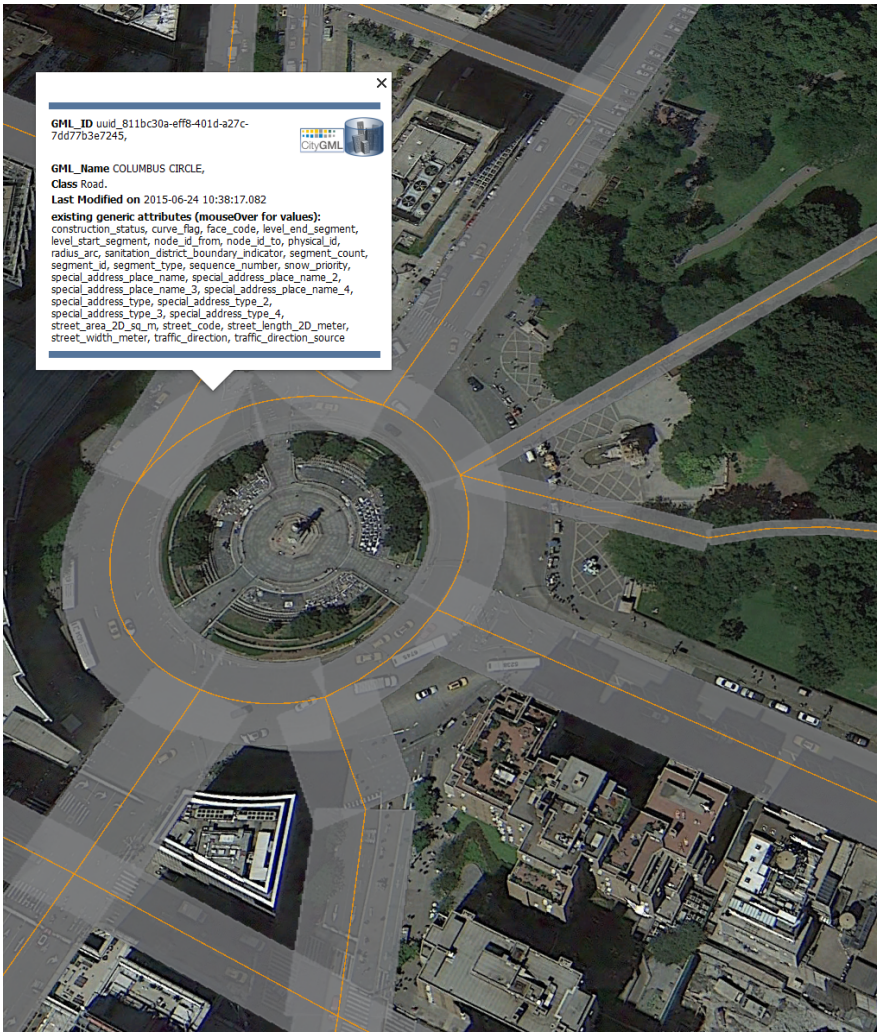


Abbildung D.1: Geometrie und Attribute der Straßen in GoogleEarth visualisiert

Abbildungsverzeichnis

2.1 Beispielhafte Skizzen der Zellzerlegung, CSG und B-Rep . . . 14
2.2 CityGML UML-Diagramm des *Core* Moduls . . . 16
2.3 CityGML-UML-Diagramm der Top Level Klassenhierarchie . . . 17
2.4 Aggregat, Complex und Composite am Beispiel Flächen . . . 18
2.5 Die von CityGML definierten Detaillierungsstufen (LoD0–LoD4) . . . 19

4.1 Flächenhafte Ausdehnung der Stadt New York. . . . 37
4.2 Allgemeines Ablaufdiagramm der Datenbearbeitung. . . . 40

5.1 CityGML-UML-Diagramm des Moduls *Relief* . . . 44
5.2 Digitales Geländemodell von New York . . . 45
5.3 Übersicht des Ablaufs zur Generierung des DGMs . . . 46
5.4 Schematische Darstellung der Kachelgrenze . . . 48
5.5 Darstellung des Gitters an der Kachelgrenze . . . 49

6.1 CityGML-UML-Diagramm des Moduls *building* . . . 53
6.2 Darstellung problematischer Polygone . . . 57
6.3 Ablaufdiagramm zur Übertragung der Höheninformation auf die Gebäude . . . 58
6.4 Schnittlinie Gelände – Gebäude . . . 59

7.1 CityGML-UML-Diagramm des Moduls *vegetation* . . . 61
7.2 Standorte der Bäume . . . 63
7.3 Standorte der Bäume . . . 64
7.4 Geometrie der Bäume . . . 65

8.1 CityGML-UML-Diagramm des Moduls *waterbody* . . . 68
8.2 Darstellung des Wasserkörpers in CityGML definiert . . . 68
8.3 Boundingbox um ein gesamtes Gewässerpolygon . . . 70
8.4 Kachelung der Gewässer . . . 71
8.5 Darstellung der Wasseroberfläche enthält Geländehöhen aus DGM . . . 72
8.6 Integration Gewässer und Gelände . . . 73
8.7 CityGML-UML-Diagramm des Moduls *CityFurniture* . . . 74

9.1 CityGML-UML-Diagramm des Moduls *landuse* . . . 75

10.1 Ausschnitt des generierten Stadmodells . . . 79
10.2 Visualisierung der Gebäude und Bäume . . . 81
10.3 Visualisierung der Flurstücke . . . 82
10.4 Visualisierung der Gebäude und Bäume . . . 83
10.5 Visualisierung der Flurstücke und Parks . . . 84
10.6 Visualisierung der Gewässer(-strukturen) . . . 85

D.1 Geometrie und Attribute der Straßen in GoogleEarth visualisiert . . . 109

Tabellenverzeichnis

4.1 Ursprüngliche Koordinatenreferenzsysteme 35
4.2 Koordinatenreferenzsystem 36
4.3 Überblick der Datensätze 38
4.4 Rechenzeiten der Datensätze 39

6.1 Übersicht über Gebäude Attribute (original) 55
6.2 Übersicht über Gebäude Attribute (neu erzeugt) 55
6.3 Übersicht über Gebäude Attribute (aus PLUTO-Datensatz) 56
6.4 Übersicht über verwendete Attribute der Flurstücke 60

9.1 Attribute der Datensätze der Parkanlagen 76

A.1 Überblick über die Herkunft der Datensätze. 93

Literaturverzeichnis

Literatur

[Abel et al., 2014]
Abel, J., Vries, E. de, Engelhardt, K., Geiger, C., Hansen, J., Heimstädt, T., Knopp, A., Lippa, B., Meineke, C. und Winter, J.-P.: *Auszug aus dem Hintergrundbericht: Smart Country - Digitale Strategien für Regionen.* Internet & Gesellschaft Co:llaboratory e.V. 2014.

[Almarabeh & AbuAli, 2010]
Almarabeh, T. und AbuAli, A.: "A General Framework for E-Government: Definition Maturity Challenges, Opportunities, and Success". *European Journal of Scientific Research* 39 (1), S. 29–42 (2010).

[Andrae, 2009]
Andrae, C.: *OpenGIS essentials – Die Geo-Standards von OGC und ISO im Überblick. Spatial Schema – ISO19107 und ISO19137 vorgestellt und erklärt.* Wichmann Verlag, 2009.

[Antonarakis et al., 2008]
Antonarakis, A., Richards, K. und Brasington, J.: "Object-based land cover classification using airborne LiDAR". *Remote Sensing of Environment* 112 (6), S. 2988–2998 (2008).

[Bloomberg und Merchant, 2012]
Bloomberg, M. R. und Merchant, R. N.: *Open Data Policy and Technical Standards Manual.* Version Final 1.2.1. NYC Department of Information Technology & Telecommunications, 2012.

[Bundesministerium des Innern, 2014]
Bundesministerium des Innern: *Digitale Verwaltung 2020 - Regierungsprogramm 18. Legislaturperiode.* Bundesministerium des Innern, 2014.

[Coors et al., 2013]
Coors, V., Holweg, D., Ekkehard, M. und Petzhold, B.: *3D-Stadtmodelle – Projekte im Raum.* InGeoForum und Kommission 3D-Stadtmodelle der Deutschen Gesellschaft für Kartographie (DGfK) und der Deutschen Gesellschaft für Photogrammetrie, Fernerkundung und Geoinformation (DGPF), 2013.

[de Lange, 2013]
de Lange, N.: *Geoinformatik in Theorie und Praxis.* 3. Auflage. Springer-Verlag Berlin Heidelberg, 2013.

[Dietrich, 2010]
DIETRICH, D.: "Offene Daten für Deutschland - Chancen und Herausforderungen für die Zukunft". In: *„Offene Staatskunst"Bessere Politik durch „Open Government"?* 1. Internet & Gesellschaft Co:llaboratory, Oktober 2010, S. 49–79.

[Dobusch et al., 2011]
DOBUSCH, L., FORSTERLEITNER, C. und HIESMAIR, M.: *Freiheit vor Ort - Handbuch kommunale Netzpolitik.* open source PRESS, 2011.

[Dobusch et al., 2012]
DOBUSCH, L., PAWEL, S., POMBERGER, G. und RIEDL, R.: "Open Government Data: eine Initiative der Open-Commons-Region Linz". *HMD Praxis der Wirtschaftsinformatik* 49 (1), S. 74–83 (2012).

[Döllner et al., 2006]
DÖLLNER, J., KOLBE, T. H., LIECKE, F., SGOUROS, T. und TEICHMANN, K.: "The Virtual 3D City Model of Berlin – Managing, Integrating and Communicating Complex Urban Information". *Proceedings of the 25th International Symposium on Urban Data Management UDMS 2006 in Aalborg, Denmark* (2006).

[van Essen, 2008]
ESSEN, R. van: "Maps Get Real: Digital Maps evolving from mathematical line graphs to virtual reality models". In: *Advances in 3D Geoinformation Systems.* Hrsg. von OOSTEROM, P. van, ZLATANOVA, S., PENNINGA, F. und FENDEL, E. Springer-Verlag, 2008. Kap. 1, S. 3–18.

[EU-Richtlinie, 2007]
EUROPÄISCHES PARLAMENT UND DER RAT DER EUROPÄISCHEN UNION: *Richtlinien – Richtlinie 2007/2/EG des europäischen Parlaments und des Rates vom 14. März 2007 zur Schaffung einer Geodateninfrastruktur in der Europäischen Gemeinschaft (INSPIRE)*, 2007.

[xAL Standard, 2002]
Extensible Address Language (xAL) Standard Description Document for W3C DTD/Schema. Version 2.0. summary of Changes: New release of xAL DTD/Schema that supports postal services elements, namespaces and new structures. Customer Information Quality Technical Committee [CIQ TC], http://www.oasis-open.org/committees/ciq 2002.

[Gröger et al., 2012]
GRÖGER, G., KOLBE, T. H., NAGEL, C. und HÄFELE, K.-H.: *OGC City Geography Markup Language (CityGML) Encoding Standard.* Version 2.0.0, Open Geospatial Consortium, 2012.

[Groß et al., 2014]
GROSS, M., HABBEL, F.-R. und SIEGFRIED, T.: *Petitionspapier-Open Data in Kommunen.* Deutscher Städte- und Gemeindebund, Kommunale Gemeinschaftsstelle für Verwaltungsmanagement, Bundes-Arbeitsgemeinschaft der Kommunalen IT-Dienstleister e.V. 2014.

[Herzberg, 2013]
HERZBERG, J.: "Open Government - Versuch einer Begriffsbestimmung". *eNewsletter Wegweiser Bürgergesellschaft* 06/2013 (2013).

[Jahnke, 2013]
Jahnke, M.: "Nicht-Photorealismus in der Stadtmodellvisualisierung für mobile Nutzungskontexte". Dissertation. Technische Universität München, 2013.

[Jahnke et al., 2011]
Jahnke, M., Berger, T., Donaubauer, A. und Krisp, J.: "Nicht fotorealistische Darstellung von 3D-Stadtmodellen". In: *HMD Praxis der Wirtschaftsinformatik*. Bd. 3. Gabler Verlag, 2011, S. 101–112.

[Kahmen, 2006]
Kahmen, H.: *Angewandte Geodäsie: Vermessungskunde*. 20. Auflage. de Gruyter, 2006.

[Klessmann et al., 2012]
Klessmann, J., Denker, P., Schieferdecker, I. und Schulz, S. E.: *Open Government Data Deutschland - Eine Studie zu Open Government in Deutschland im Auftrag des Bundesministerium des Innern*. Bundesministerium des Innern, 2012.

[Koch, 2006]
Koch, A.: "Semantische Integration von zweidimensionalen GIS-Daten und digitalen Geländemodellen". Dissertation. Universität Hannover, 2006.

[Kolbe, 2009]
Kolbe, T. H.: "Representing and Exchanging 3D City Models with CityGML". In: *Proceedings of the 3rd International Workshop on 3D Geoinformation, Seoul, Korea. Lecture Notes in Geoinformation & Cartography*. Hrsg. von Lee, J. und Zlatanova, S. Springer Verlag, 2009.

[Kolbe et al., 2015]
Kolbe, T. H., Yao, Z., Nagel, C., Kunde, F., Willkomm, P., Hudra, G., Müftüoglu, A. und Herreruela, J.: *3D City Database for CityGML – Documentation*. Version 3.0.0, 2015.

[Löwner et al., 2014]
Löwner, M. O., Benner, J. und Gröger, G.: "Aktuelle Trends in der Entwicklung von CityGML 3.0". In: *Deutsche Gesellschaft für Photogrammetrie, Fernerkundung und Geoinformation (DGPF) Tagungsband 23*. 2014.

[Löwner et al., 2013]
Löwner, M. O., Casper, E., Becker, T., Benner, J., Gröger, G., Gruber, U., Häfele, K.-H., Kaden, R. und Schlüter, S.: "CityGML 2.0 – ein internationaler Standard für 3D-Stadtmodelle, Teil 2: CityGML in der Praxis". *Zeitschrift für Geodäsie, Geoinformation und Landmanagement, 2/2013*, S. 131–143 (2013).

[von Lucke & Geiger, 2010]
Lucke, J. von und Geiger, C. P.: *Open Government Data - Frei verfügbare Daten des öffentlichen Sektors (Gutachten für die Deutsche Telekom AG zur T-City Friedrichshafen)*. Deutsche Telekom Institute for Connected Cities, Zeppelin University Friedrichshafen, 2010.

[Oracle, 2010]
Oracle Spatial Partitioning: Best Practices. An Oracle White Paper. Oracle, 2010.

[Pfeifer, 2003]
PFEIFER, N.: "Oberflächenmodelle aus Laserdaten". *VGI – Österreichische Zeitschrift für Vermessung und Geoinformation* 91 (4), S. 243–252 (2003).

[Schilling et al., 2007]
SCHILLING, A., BASANOW, J. und ZIPF, A.: "Vector based Mapping of Polygons on Irregular Terrain Meshes for Web 3D Map Services". *3rd International Conference on Web Information Systems and Technologies (WEBIST). Barcelona, Spain* (2007).

[Sester et al., 2007]
SESTER, M., KIELER, B. und GÖSSELN, G. von: "Semantische und Geometrische Integration von Geodaten". In: *Kartographie als Baustein moderner Kommunikation, Symposium 2007 in Königslutter am Elm, Kartographische Schriften.* Bd. 14, Kirschbaum Verlag Bonn, 2007, S. 53–58.

[SIG3D, 2014]
SIG3D: *Handbuch für die Modellierung von 3D Objekten – Teil 2: Modellierung Gebäude (LoD1, LoD2 und LoD3).* Version 2.0.1 DE. SIG3D AG Qualität, 2014.

[Smart et al., 2011]
SMART, P. D., QUINN, J. A. und JONES, C. B.: "City Model Enrichment". *ISPRS Journal of Photogrammetry and Remote Sensing* 66, S. 223–234 (2011).

[Stadler, 2007]
STADLER, A.: "Kohärenz von Geometrie und Semantik in der Modellierung von 3D Stadtmodellen". In: *Entwicklerforum Geoinformationstechnik 2007 - Junge Wissenschaftler forschen.* Shaker Verlag, Aachen 2007, S. 167–181.

[von Lucke, 2010]
VON LUCKE, J.: "Open Government - Öffnung von Staat und Verwaltung (Gutachten für die Deutsche Telekom AG zur T-City Friedrichshafen)". In: *Entdeckung, Erkundung und Entwicklung 2.0: Open Government, Open Government Data und Open Budget 2.0.* Deutsche Telekom Institute for Connected Cities, Zeppelin University Friedrichshafen, 2010, S. 53–104.

[Wanasky, 2011]
WANASKY, S.: "Prototypische Objekte bei 3D-Stadtmodellen am Beispiel des Standards CityGML". Dissertation. Technische Universität München & Eidgenössische Technische Hochschule Zürich, 2011.

[Yao und Kolbe, 2014]
YAO, Z. und KOLBE, T. H.: "Cloud-basierter Webclient für kollaboratives Arbeiten auf semantischen 3D-Stadtmodellen". In: *Deutsche Gesellschaft für Photogrammetrie, Fernerkundung und Geoinformation (DGPF) Tagungsband 23.* 2014.

Internetquellen

[1]
ADVISORY COUNCIL: *opendefinition.* URL: `http://opendefinition.org/od/`. Aufrufdatum: 02.12.2014.

[2]
APP:CONTEST 2014 Hamburg: *APP:CONTEST 2014 Hamburg*. URL: AdrianUlrich, HandelskammerHamburghttp://www.app-contest-hamburg.de/. Aufrufdatum: 28.05.2015.

[3]
Bayerisches Staatsministerium der Finanzen, für Landesentwicklung und Heimat: *Open Data Portal des Freistaats Bayern*. URL: https://opendata.bayern.de/;jsessionid=E3C4AC093A01D4FADB5AF513956B33CF?0. Aufrufdatum: 04.06.2015.

[4]
Bundesamt für Kartographie und Geodäsie: *Geoportal.de*. URL: http://www.geoportal.de/DE/Geoportal/geoportal.html?lang=de. Aufrufdatum: 06.06.2015.

[5]
Bundesamt für Kartographie und Geodäsie: *Geoportal.de*. URL: http://www.geoportal.de/DE/GDI-DE/INSPIRE/Direktive/Data-Specs/data-specs.html?lang=de. Aufrufdatum: 06.06.2015.

[6]
Bundesministerium des Innern: *GovData - Das Datenportal für Deutschland*. URL: https://www.govdata.de/. Aufrufdatum: 03.06.2015.

[7]
Bundesministerium des Innern: *GovData - Das Datenportal für Deutschland*. URL: https://www.govdata.de/dl-de/by-2-0. Aufrufdatum: 07.06.2015.

[8]
Dietrich, *Bundeszentrale für politische Bildung*. URL: http://www.bpb.de/gesellschaft/medien/opendata/64055/was-sind-offene-daten. Aufrufdatum: 03.12.2014.

[9]
Fraunhofer-Gesellschaft: *Fraunhofer IAIS*. URL: http://www.iais.fraunhofer.de/ogd4kommunen.html. Aufrufdatum: 03.12.2014.

[10]
Freie Hansestadt Bremen: *Transparenzportal Bremen*. URL: http://transparenz.bremen.de/sixcms/detail.php?gsid=bremen02.c.730.de. Aufrufdatum: 04.06.2015.

[11]
GeoRepository – Geomatic Solutions: *NAD83 / New York Long Island*. URL: http://georepository.com/crs_32118/NAD83-New-York-Long-Island.html. Aufrufdatum: 05.03.2015.

[12]
GeoRepository – Geomatic Solutions: *NAD83 / New York Long Island (ftUS)*. URL: http://georepository.com/crs_2263/NAD83-New-York-Long-Island-ftUS.html. Aufrufdatum: 05.03.2015.

[13]
GeoRepository – Geomatic Solutions: *NAVD88 height*. URL: http://georepository.com/crs_5703/NAVD88-height.html. Aufrufdatum: 05.03.2015.

[14]
GeoRepository – Geomatic Solutions: *WGS84/Pseudo-Mercator*. URL: http://georepository.com/crs_3857/WGS-84-Pseudo-Mercator.html. Aufrufdatum: 05.03.2015.

[15]
Innenministerium Baden-Württemberg: *Open Data Portal Baden-Württemberg*. URL: https://opendata.service-bw.de/web/guest/home. Aufrufdatum: 04.06.2015.

[16]
Keserü, J. und Heise, C.: *Sunlight Foundation*. URL: https://sunlightfoundation.com/blog/2015/06/08/open-data-without-germany-the-g7-summit-and-the-bumpy-road-to-transparency/. Aufrufdatum: 16.06.2015.

[17]
Landeshauptstadt München. Referat für Stadtplanung und Bauordnung: *Das Münchner Stadtmodell*. URL: http://www.muenchen.de/rathaus/Stadtverwaltung/Referat-fuer-Stadtplanung-und-Bauordnung/Stadt-und-Bebauungsplanung/Stadtmodell.html#/. Aufrufdatum: 05.04.2015.

[18]
Ministerium des Innern, für Sport und Infrastruktur des Landes Rheinland-Pfalz: *Open-Government-Data-Portal Rheinland-Pfalz*. URL: http://www.daten.rlp.de/. Aufrufdatum: 04.06.2015.

[19]
Ministerium für Inneres und Kommunales des Landes Nordrhein-Westfalen: *opennrw mitreden - mitmachen - mitgestalten*. URL: https://open.nrw/de/startseite. Aufrufdatum: 04.06.2015.

[20]
Niedersächsisches Ministerium für Umwelt, Energie und Klimaschutz: *Das Niedersächsische Umweltportal – Behördliche Umweltinformation auf einen Klick*. URL: http://numis.niedersachsen.de/. Aufrufdatum: 04.06.2015.

[21]
Obama, B.: *Transparency and Open Government - Memorandum for the Heads of Executive Departments and Agencies*. URL: https://www.whitehouse.gov/the_press_office/Transparency_and_Open_Government/. Aufrufdatum: 27.05.2015.

[22]
odnauthor: *8 Open Government Data Prinzipien Vol. 3 - „Zeitnah“*. URL: http://opendata-network.org/2010/03/8-open-government-data-prinzipien-zeitnah/. Aufrufdatum: 28.05.2015.

[23]
Open Knowledge Foundation Deutschland e.V.: *Offene Daten*. URL: https://offenedaten.de. Aufrufdatum: 03.12.2014.

[24]
OpenStreetMap Deutschland: *Was ist OpenStreetMap?* URL: http://www.openstreetmap.de/. Aufrufdatum: 06.04.2015.

[25]
Oracle Deutschland: *Oracle Spatial und Oracle Locator*. URL: http://www.oracle.com/de/products/database/options/spatial/index.html. Aufrufdatum: 13.04.2015.

[26]
Oracle Help Center. URL: http://docs.oracle.com/cd/B28359_01/appdev.111/b28398/geor_operations.htm#CIHCHDFF. Aufrufdatum: 23.06.2015.

[27]
Safe Software Inc.: *FME Documentation*. URL: http://docs.safe.com/. Aufrufdatum: 30.05.2015.

[28]
Safe Software Inc.: *Safe Software - FME*. URL: http://www.safe.com/. Aufrufdatum: 13.04.2015.

[29]
Senatsverwaltung für Wirtschaft, Technologie und Forschung: *Berlin Open Data*. URL: http://daten.berlin.de/. Aufrufdatum: 04.06.2015.

[30]
Statistische Bundesamt, Wiesbaden: *GENESIS-Online Datenbank*. URL: https://www-genesis.destatis.de/genesis/online. Aufrufdatum: 06.06.2015.

[31]
Sunlight Foundation: *Ten Principles for OpeOpen Up Government Information*. URL: http://sunlightfoundation.com/policy/documents/ten-open-data-principles/. Aufrufdatum: 26.05.2015.

[32]
The City of New York: *Datensatz AddressPoints*. URL: https://data.cityofnewyork.us/City-Government/NYC-Address-Points/4iq4-tuhq. Aufrufdatum: 13.10.2014.

[33]
The City of New York: *Datensatz Building*. URL: https://data.cityofnewyork.us/Housing-Development/Building-Footprints/tb92-6tj8. Aufrufdatum: 14.10.2014.

[34]
The City of New York: *Datensatz Hydrography*. URL: https://data.cityofnewyork.us/Environment/Hydrography/drh3-e2fd. Aufrufdatum: 14.10.2014.

[35]
The City of New York: *Datensatz Hydrography Structure*. URL: https://data.cityofnewyork.us/Environment/hydrography-structures/ab93-5cvg. Aufrufdatum: 14.10.2014.

[36]
The City of New York: *Datensatz Park*. URL: https://data.cityofnewyork.us/Recreation/M-m-kas-Meadow/8c29-4nxt. Aufrufdatum: 16.10.2014.

[37]
The City of New York: *Datensatz Park Properties*. URL: https://data.cityofnewyork.us/City-Government/Parks-Properties/rjaj-zgq7. Aufrufdatum: 15.10.2014.

[38]
THE CITY OF NEW YORK: *Datensatz Pluto*. URL: http://www.nyc.gov/html/dcp/html/bytes/dwn_pluto_mappluto.shtml. Aufrufdatum: 13.10.2014.

[39]
THE CITY OF NEW YORK: *Datensatz Tree Bronx*. URL: https://data.cityofnewyork.us/Environment/Street-Tree-Census-Bronx-/bkyy-g74a. Aufrufdatum: 15.10.2014.

[40]
THE CITY OF NEW YORK: *Datensatz Tree Brooklyn*. URL: https://data.cityofnewyork.us/Environment/Street-Tree-Census-Brooklyn-/ztcw-bzc8. Aufrufdatum: 15.10.2014.

[41]
THE CITY OF NEW YORK: *Datensatz Tree Manhattan*. URL: https://data.cityofnewyork.us/Environment/Street-Tree-Census-Manhattan-/e6n3-m3vc. Aufrufdatum: 15.10.2014.

[42]
THE CITY OF NEW YORK: *Datensatz Tree Queens*. URL: https://data.cityofnewyork.us/Environment/Street-Tree-Census-Queens-/4wcq-bg6s. Aufrufdatum: 15.10.2014.

[43]
THE CITY OF NEW YORK: *Datensatz Tree Staten Island*. URL: https://data.cityofnewyork.us/Environment/Street-Tree-Census-Staten-Island-/pvq6-utmq. Aufrufdatum: 15.10.2014.

[44]
THE CITY OF NEW YORK: *NYC 1 foot Digital Elevation Model*. URL: https://data.cityofnewyork.us/City-Government/1-foot-Digital-Elevation-Model-DEM-/dpc8-z3jc. Aufrufdatum: 15.10.2014.

[45]
THE CITY OF NEW YORK: *NYC OpenData*. URL: https://nycopendata.socrata.com/. Aufrufdatum: 20.10.2014.

[46]
THE CITY OF NEW YORK: *NYC.gov Terms of Use*. URL: http://www1.nyc.gov/home/terms-of-use.page. Aufrufdatum: 01.06.2015.

[47]
TRANSPARENZPORTAL HAMBURG: *Datensatz ALKIS*. URL: http://suche.transparenz.hamburg.de/dataset/alkis-liegenschaftskarte-ausgewahlte-daten-hamburg. Aufrufdatum: 07.06.2015.

[48]
TRANSPARENZPORTAL HAMBURG: *Datensatz Park*. URL: http://suche.transparenz.hamburg.de/dataset/parkanlagen. Aufrufdatum: 28.05.2015.

[49]
TRANSPARENZPORTAL HAMBURG: *Datensatz Verkehr*. URL: http://suche.transparenz.hamburg.de/dataset/strassen-hh-sib. Aufrufdatum: 06.06.2015.

[50]
TRANSPARENZPORTAL HAMBURG: *Datesatz Adressen*. URL: http://suche.transparenz.hamburg.de/dataset/alkis-adressen-hamburg. Aufrufdatum: 07.06.2015.

[51]
TRANSPARENZPORTAL HAMBURG: *Datesatz Badegewässer*. URL: http://suche.transparenz.hamburg.de/dataset/basis-gewassernetz-fur-hamburg. Aufrufdatum: 07.06.2015.

[52]
TRANSPARENZPORTAL HAMBURG: *Datesatz Bäume*. URL: http://suche.transparenz.hamburg.de/dataset/strassenbaumkataster-hamburg. Aufrufdatum: 07.06.2015.

[53]
TRANSPARENZPORTAL HAMBURG: *Datesatz Gewässernetz*. URL: http://suche.transparenz.hamburg.de/dataset/basis-gewassernetz-fur-hamburg. Aufrufdatum: 07.06.2015.

[54]
TRANSPARENZPORTAL HAMBURG: *Datesatz Verwaltungsgrenzen*. URL: http://suche.transparenz.hamburg.de/dataset/alkis-verwaltungsgrenzen-hamburg. Aufrufdatum: 07.06.2015.

[55]
TRANSPARENZPORTAL HAMBURG: *Digitales Geländemodell*. URL: http://suche.transparenz.hamburg.de/dataset/digitales-hohenmodell-hamburg-dgm-1. Aufrufdatum: 07.06.2015.

[56]
TRANSPARENZPORTAL HAMBURG: *Stadtmodell*. URL: http://suche.transparenz.hamburg.de/dataset/3d-stadtmodell-hamburg. Aufrufdatum: 07.06.2015.

[57]
TRANSPARENZPORTAL HAMBURG: *Transparenzportal Hamburg*. URL: https://transparenz.hamburg.de/. Aufrufdatum: 28.05.2015.

[58]
WEB3D CONSORTIUM: *What is X3D?* URL: http://www.web3d.org/x3d/what-x3d. Aufrufdatum: 06.04.2015.

[59]
WIRTSCHAFT UND TECHNOLOGIE GMBH, B. P. für: *Berlin 3D-Downloadportal*. URL: http://www.businesslocationcenter.de/berlin3d-downloadportal/documents/terms.de.html. Aufrufdatum: 04.07.2015.

Selbstständigkeitserklärung

Hiermit erkläre ich, die vorliegende Arbeit selbständig verfasst zu haben und keine anderen als die in der Arbeit angegebenen Quellen und Hilfsmittel benutzt zu haben.

....., den		.....
Ort	Datum	Unterschrift