🏆 *reviewed paper*

Automatisierte Generierung eines 3D-Baumkatasters am Beispiel des KIT Campus Nord

Jonas Hurst, Andreas Geiger

(Jonas Hurst, Institut für Automation und angewandte Informatik (IAI) Karlsruher Institut für Technologie (KIT), Herrmann-von-Helmholtz-Platz 1 76344 Eggenstein-Leopoldshafen, jonas.hurst9@kit.edu)
(Andreas Geiger, Institut für Automation und angewandte Informatik (IAI) Karlsruher Institut für Technologie (KIT), Herrmann-von-Helmholtz-Platz 1 76344 Eggenstein-Leopoldshafen, andreas.geiger@kit.edu)

1 ABSTRACT

Bei Baumkontrollen werden sogenannte Baumkataster angelegt, in denen Informationen über in einem bestimmten Bereich befindliche Bäume dokumentiert werden. In ihnen wird unter anderem die Gestalt der Bäume durch die Parametern Baumhöhe, Baumkronendurchmesser und Baumkronenumfang beschrieben. Im Rahmen dieses Beitrages wird ein in Python entwickeltes Analyse- und Transformationswerkzeug vorgestellt, welches erlaubt, Baumdaten in unterschiedlicher, strukturierter Form (CSV, XML) einzulesen, tabellarisch darzustellen und zu analysieren. Weiterhin ermöglicht es, aus den Gestalt beschreibenden Daten ein 3D-Modell abzuleiten und dieses im standardisierten Datenformat CityGML unter Verwendung des Vegetation-Moduls als geometrisches 3D Modell mit den vorhandenen Sachdaten zu exportieren.

Keywords: Digitalisierung, CityGML, 3D, Baumkataster, Python

2 EINLEITUNG

Im Rahmen der Verkehrssicherungspflicht ist diejenige Person, die Verfügungsgewalt über ein Grundstück ausübt, dazu verpflichtet sicherzustellen, dass von den auf dem Grundstück befindlichen Bäumen keine Gefahren ausgehen. Daraus resultiert, dass Bäume im öffentlichen Raum in regelmäßigen Kontrollen auf Standsicherheit und Beschädigungen überprüft werden müssen.

Dabei werden sogenannte Baumkataster angelegt, um der Dokumentationspflicht nachzukommen, in denen neben einer Bewertung jedes Baumes auf einem Grundstück dessen genaue Lage sowie Metadaten erfasst werden. Einige dieser Metadaten beschreiben den Baum in seiner Gestalt, z.B. die Baumhöhe, der Stammdurchmesser oder der Baumkronendurchmesser.

Die Daten eines Baumkatasters lassen sich abgesehen von der Dokumentation aber auch noch für andere Einsatzgebiete nutzen. Aus den erfassten Parametern lässt sich die Gestalt der Bäume in parametrisierter Form beschreiben und damit für die Visualisierung oder für Simulationsberechnungen einsetzen. In einer Abschattungsanalyse kann beispielsweiße der Schatten berechnet werden, der von einem Baum auf ein Gebäude geworfen wird. Diese Informationen können für eine Solarpotentialanalyse oder für thermische Gebäudesimulationen genutzt werden. Um solche Berechnungen durchführen zu können ist ein digitales dreidimensionales Modell der Bäume notwendig. Viele Softwarelösungen zum Verwalten solcher Baumkataster unterstützen einen Export dahingehend jedoch nicht. Das am Campus Nord des KIT eingesetzte Programmsystem Arbokat® unterstützt beispielsweiße lediglich einen Export in die Formate CSV, PDF, KML, Shapefile und INGRADAweb, sodass die Entwicklung eines Werkzeuges, welches aus den Baumkatasterdaten dreidimensionale Geometrien ableitet und diese im standardisierten CityGML-Format exportiert, notwendig ist.

3 GRUNDLAGEN

3.1 Baumkataster

Ein Baumkataster ist ein raumbezogenes Verzeichnis, in dem Informationen zu Bäumen innerhalb eines bestimmten Gebietes erfasst sind (Stadt Stuttgart, 2020). Ein solches Gebiet kann zum Beispiel ein einzelnes Grundstück sein, aber auch ein gesamtes Stadtgebiet umfassen. Sie werden zu Dokumentationszwecken regelmäßig stattfindender Baumkontrollen angelegt. Solche Kontrollen müssen regelmäßig von sowohl öffentlichen als auch privaten Grundbesitzern stattfinden, die Verkehr eröffnen oder auf einem Grundstück zulassen. Der Grund hierfür ist, dass ein Baumbesitzer der Verkehrssicherungspflicht nachzukommen muss, die er gegenüber Dritten hat (Wessolly und Erb, 2014). Bei diesen Baumkontrollen wird der Zustand des Baumes erfasst und bewertet, um beispielsweise Sach- und Personenschäden durch baumkrankheitsbedingte Sturmschäden zu vermeiden. Im Jahr 2004 hat die Forschungsgesellschaft Landschaftsentwicklung Landschaftsbau e.V. (FLL) erstmals die Anforderungen an Baumkontrollen in den Baumkontrollrichtlinien

ISBN 978-3-9504173-8-8 (CD), 978-3-9504173-9-5 (print)
Editors: Manfred SCHRENK, Vasily V. POPOVICH, Peter ZEILE, Pietro ELISEI, Clemens BEYER, Judith RYSER, Christa REICHER, Canan ÇELIK

normiert. Diese Baumkontrollrichtlinien der FLL wurden über gerichtliche Urteile dem aktuellen Stand der Technik der Baumkontrolle zugeordnet. Mittlerweile wurden die Baumkontrollrichtlinien überarbeitet, sodass diese seit 2010 in einer neuen, aktualisierten Fassung zur Verfügung stehen (Forschungsgesellschaft Landschaftsentwicklung Landschaftsbau e.V., 2010). Während dieser Baumkontrollen werden Daten über die untersuchten Bäume erfasst, wie z.B. der Gesundheitszustand, der Standort, die Höhe, der Stammdurchmesser oder der Baumkronendurchmesser.

Alle während der Baumkontrolle erfassten Daten werden danach zu Dokumentationszwecken im Baumkataster festgehalten. In ihnen werden also neben Daten zum Gesundheitszustand des Baumes und Standortinformationen und auch Daten über die Gestalt des Baumes festgehalten.

Es existieren einige Software-Programme, um die Erstellung und Verwaltung eines digitalen Baumkatasters zu erleichtern. Die am Campus Nord des KIT verwendete Software zur Erstellung und Verwaltung des digitalen Baumkatasters heißt Arbokat® und wird von der Firma iNovaGIS und dem Sachverständigenbüro Peter Klug konzipiert und entwickelt. Dabei werden alle in den Baumkontrollrichtlinien der FLL festgelegten Kriterien berücksichtigt (iNovaGIS, 2016).

3.2 CityGML

Das Datenformat City Geography Markup Language (CityGML) ist das Datenmodell, in das das Baumkataster überführt werden soll. Es ist ein semantisches Datenmodell zum Speichern und Austauschen digitaler dreidimensionaler (3D) Stadtmodelle (Gröger et al. 2012).

Das CityGML-Datenmodell wird seit 2002 von der Special Interest Group 3D (SIG 3D) entwickelt. Die Arbeitsgemeinschaft besteht aus mehr als 70 Firmen, Gemeinden und Forschungseinrichtungen. Ziel ist, ein standardisiertes Datenformat zum Speichern von 3D-Stadtmodellen zu schaffen, um den Austausch von Stadtmodellen zu erleichtern (Kolbe, 2009). Im Jahr 2008 wurde das CityGML-Datenformat (damals in der Version 1.0.0) auch vom Open Geospatial Consirtium als Standard (OGC) anerkannt. Aktuell liegt CityGML in der Version 2.0.0 vor und wird von vielen Städten weltweit (z.B. Berlin, München, Paris, Monaco, Istanbul, Doha) dazu verwendet, 3D-Stadtmodelle bereitzustellen (Gröger et al. 2012).

CityGML verbindet dabei eine geometrische Modellierung von Städten mit einer semantischen Modellierung. Das bedeutet, dass ein Stadtmodell im CityGML-Format sich nicht nur zur Visualisierung einer Stadt eignet, da nicht nur eine geometrische Repräsentation von einzelnen Objekten einer Stadt vorliegt, sondern dabei die einzelnen Objekte auch klassifiziert und aggregiert werden können, sowie mit verschiedenen, das Objekt näher beschreibenden, Attributen versehen werden können. Eine solche semantische Modellierung ermöglicht in Kombination mit der geometrischen Modellierung weitreichende Analysen (Gröger et al. 2012).

Das CityGML-Format ist modular aufgebaut. Es besteht aus dem zwingend erforderlichen Core-Modul sowie aus thematisch gegliederten Erweiterungs-Modulen. Das Core-Modul umfasst alle dem Datenmodell zugrundeliegenden Konzepte und Komponenten, die von jedem der Erweiterungsmodule genutzt werden. Mit den thematisch voneinander abgegrenzten Erweiterungsmodulen findet dann die eigentliche Modellierung der Objekte einer Stadt statt. Das Stadtmodell besteht dann aus der Gesamtheit der einzeln modellierten Objekte. In CityGML 2.0 existieren Erweiterungsmodule für z.B. Gebäude, Brücken und Stadtmöbelierung. Es existiert außerdem ein Modul für Vegetation, welches im Folgenden genauer vorgestellt wird (Gröger et al. 2012).

Unterschieden werden im Vegetation-Modul zwei verschiedene Klassen von Vegetation. Während die Klasse PlantCover dazu verwendet wird, eine größere Ansammlung von Vegetation zu modellieren wie z.B. eine Wiese oder einen Wald, können über die Klasse SolitaryVegetationObject einzelne Vegetations-Objekte beschrieben werden wie z.B. einzelne Bäume, Sträucher oder Büsche. Beide Klassen erben von der abstrakten Klasse _VegetationObject, die wiederum von der abstrakten Klasse _CityObject des Core-Moduls erbt. Abbildung 1 zeigt das UML-Klassendiagramm des CityGML-Vegetation-Moduls.

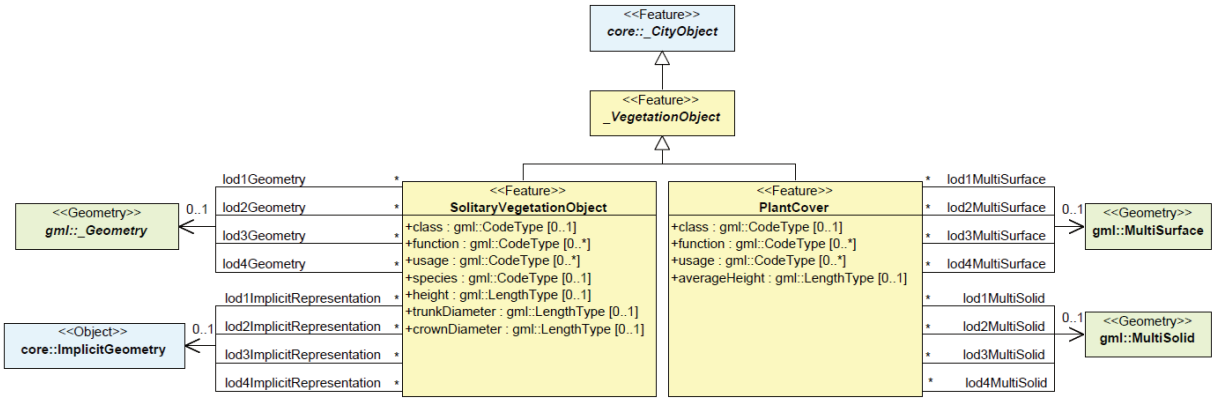


Abbildung 1: UML-Klassendiagramm des CityGML Vegetation-Moduls (Gröger et al. 2012)

Ein Objekt der Klasse PlantCover besitzt die Attribute class, welches die Vegetationsgruppe beschreibt, der das PlantCover-Objekt angehört. Die Attribute function und usage beschreiben den angedachten und tatsächlichen Zweck eines Objektes. Zudem besitzt die Klasse PlantCover das Attribut averageHeight, in dem die durchschnittliche Höhe der Vegetationsgruppe gespeichert wird (Gröger et al. 2012).

Auch ein Objekt der Klasse SolitaryVegetationObject hat die Attribute function und usage, die hier die gleiche Verwendung haben. Das Attribut class wird bei SolitaryVegetationObject jedoch dazu verwendet, um das Vegetations-Object genauer zu klassifizieren (z.B. Baum, Busch, Gras). Ein Objekt der Klasse SolitaryVegetationObject besitzt außerdem das Attribut species, in welchem der botanische Fachname der Spezies (z.B. Abies alba für Weißtanne) als CodeType gespeichert ist. Das bedeutet, dass jeder Baumspezies über eine sogenannte Codelist ein Zahlencode zugeordnet ist, und nur dieser Zahlencode im CityGML-Stadtmodell gespeichert wird (Gröger et al. 2012). Diese Codelist ist in der CityGML-Spezifikation definiert, sodass die verwendeten Codes programmunabhängig interpretiert werden können.

Des Weiteren besitzt die Klasse SolitaryVegetationObject einige Attribute zur parametrisierten Beschreibung der Geometrie. So kann im Attribut height die Höhe des Vegetations-Objektes gespeichert werden, im Attribut trunkDiameter der Durchmesser des Baumstammes (falls das modellierte Objekt ein Baum ist) und im Attribut crownDiameter der Durchmesser der Baumkrone. Alle Attribute der Klassen PlantCover und SolitaryVegetationObject sind jedoch optional und nicht zwingend erforderlich, was auch die Modellierung sehr flexibel macht. So können z.B. mit der Klasse SolitaryVegetationObject Bäume als auch Sträucher und Büsche modellieren, obwohl z.B. das Attribut trunkDiameter bei Büschen nicht sinnvoll einzusetzen ist. Es macht aber auch dann eine thematische Modellierung möglich, wenn nicht alle Attribute eines Vegetation-Objektes erfasst sind oder erfasst werden können. Weiterhin ist es möglich, jedem Objekt der Klasse _CityObject eine beliebige Anzahl frei definierbarer Attribute hinzuzufügen, sogenannte Generic Attributes, um eine thematische Modellierung über Attribute zu ermöglichen die nicht vom Datenmodell vorgesehen sind (Gröger et al. 2012).

Zur Visualisierung der Vegetation sowie zur Durchführung geometrischer Analysen ist es außerdem möglich, jedem Objekt der Klasse PlantCover und SolitaryVegetationObject Geometrien im Level-Of-Detail (LOD) 1 bis LOD4 zuzuweisen (Gröger et al. 2012).

4 ANALYSE

In diesem Kapitel werden zunächst die Ausgangsdaten, die im CSV-Format vorliegen, analysiert. Anschließend wird untersucht, welche Anforderungen an das zu erstellende Softwarewerkzeug gestellt werden

4.1 Baumdaten im CSV-Format

Das am KIT Campus Nord zur Verwaltung eines digitalen Baumkatasters verwendete Programmsystem Arbokat® unterstützt einen Export der Baumdaten in das Datenformat Comma Seperated Values (CSV) (iNovaGIS, o.D.). Dabei handelt es sich um eine Textdatei zur Speicherung strukturierter Daten, die sich aufgrund der Einfachheit des Datenformats leicht automatisiert verarbeiten lässt.

Die von Arbokat® generierte CSV-Datei enthält alle im Baumkataster gespeicherten Bäume. Jede Zeile der Datei enthält dabei die Informationen für je einen Baum. Innerhalb einer Zeile stehen, durch Semikolons

ISBN 978-3-9504173-8-8 (CD), 978-3-9504173-9-5 (print)
Editors: Manfred SCHRENK, Vasily V. POPOVICH, Peter ZEILE, Pietro ELISEI, Clemens BEYER, Judith RYSER, Christa REICHER, Canan ÇELIK

voneinander getrennt, verschiedene Attribute, die den Baum näher in seinem Standort, seiner Gestalt und seinem Gesundheitszustand beschreiben. Um die einzelnen Baumattribute interpretieren zu können stehen in der ersten Zeile der CSV-Datei die Namen der Attribute. Für die Erstellung eines dreidimensionalen Baumkatasters sowohl als geometrisches als auch als parametrisiertes Modell im CityGML-Format sind dabei die folgenden Informationen wichtig, die in der CSV-Datei gespeichert sind:

- Genaue Position des Baumes: Rechts- und Hochwert im Referenzsystem Gauß-Krüger-Zone-3
- Lateinischer Name der Baumart
- Gestalt des Baumes: Baumhöhe, Baumkronendurchmesser, Stammumfang

Auffällig ist, dass den Bäumen in der CSV-Datei keine ID zugewiesen ist, über die ein Baum identifiziert werden könnte. Es existiert lediglich eine Baumnummer, die jedoch mehrfach vergeben wurde. Allerdings ist jedem Baum ein Bereich zugewiesen ist, in dem er steht und bei näherer Betrachtung fällt auf, dass die Baumnummer innerhalb eines Bereiches eindeutig ist. Somit ist es möglich, einen Baum anhand einer Kombination aus Bereichsnummer und Baumnummer eindeutig im gesamten Datensatz zu identifizieren.

4.2 Anforderungen an das Programm

Im Allgemeinen soll das in Python geschriebene Programm Baumdaten verschiedener Datenformate einlesen und verarbeiten können. Die Daten sollen analysiert, validiert und aus ihnen ein parametrisches als auch geometrisches Baummodell generiert werden können. Dieses soll im CityGML-Format exportiert werden können. Diese allgemeinen Anforderungen, die im Folgenden konkretisiert werden, legen die Rahmenbedingungen für die Entwicklung des Programms fest.

- Einlesen der Baumdaten: Das Programm soll Baumdaten in zweierlei Datenformaten einlesen und verarbeiten können. Es soll Baumdaten im CSV- als auch im XML-Format einlesen, verarbeiten und interpretieren können. Dabei soll eine gewisse Flexibilität beim Einlesen der Daten hinsichtlich ihrer Modellierung im jeweiligen Datenformat möglich sein.
- Graphische Benutzeroberfläche: Das Programm soll zur vereinfachten Bedienbarkeit über eine graphische Benutzeroberfläche verfügen. In ihr sollen die eingelesenen Baumdaten tabellarisch angezeigt werden, damit der Benutzer sich einen Überblick über den Datensatz verschaffen kann. Über die Benutzeroberfläche sollen alle Funktionen des Programms aufgerufen werden können.
- Analyse der Daten: Die eingelesenen Daten sollen analysiert werden können, damit der Benutzer sich einen Überblick über die Qualität der Daten schaffen kann. Die Daten sollen sowohl nach ihrer ID als auch nach ihrer Position auf Duplikate überprüft werden können. Weiterhin soll eine Validierung der Gestalt beschreibenden Parameter der Bäume stattfinden.
- Ableitung einer Referenzhöhe: Da in Baumkatastern häufig keine Referenzhöhen zu den Bäumen gespeichert sind, diese aber für weitere Analysen wichtig ist, soll es eine Möglichkeit geben, für jeden Baum im Datensatz eine Referenzhöhe aus einem DGM abzuleiten. Diese soll in der Datenstruktur gespeichert und bei der Weiterverarbeitung der Daten zu berücksichtigt werden.
- Export in das CityGML-Datenformat: Das Programm soll die Möglichkeit bieten, das Baumkataster im standardisierten, semantischen Datenformat CityGML unter Verwendung des Vegetation-Moduls zu exportieren. Dabei soll ein parametrisiertes als auch ein geometrisches Modell des Baumes erstellt und exportiert werden.

5 KONZEPT UND IMPLEMENTIERUNG

In diesem Kapitel soll zunächst das theoritische Konzept, auf welches das Softwarewerkzeug aufbaut, vorgestellt werden. Im Anschluss wird kurz auf die technische Umsetzung dieses Konzeptes eingegangen.

5.1 Konzept

Den Ablauf des entwickelten Programmes zeigt in schematischer Darstellung der Programmablaufplan in Abbildung 2. Es ist möglich, sowohl CSV- als auch XML-Dateien zu interpretieren und in eine räumliche Datenbank, die als interne Datenstruktur fungiert, zu überführen. Von dort aus werden die Daten in der GUI angezeigt, können analysiert, mit Höheninformationen angereichert und ins CityGML-Format exportiert werden.

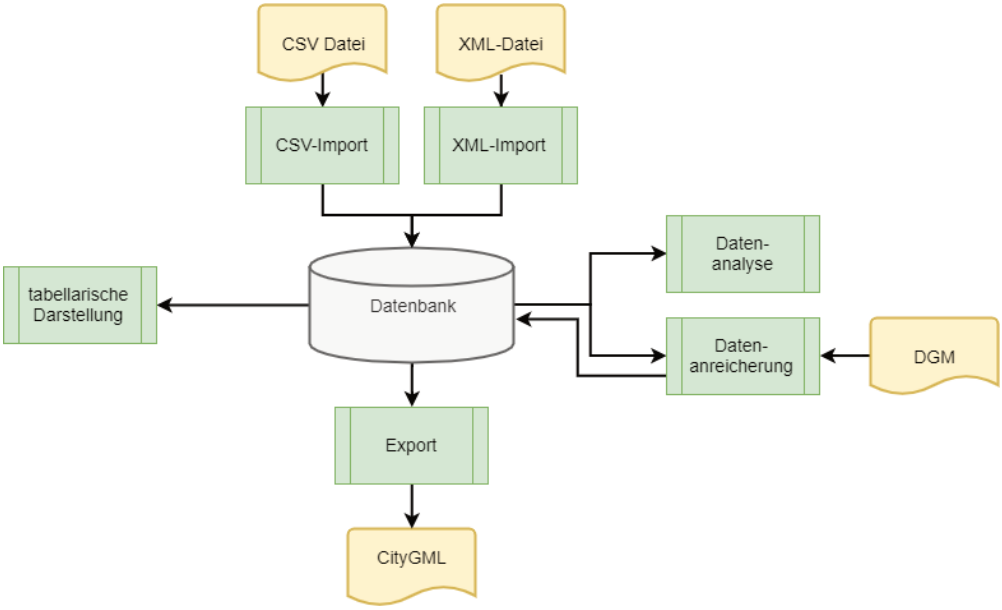


Abbildung 2: Programmablaufplan (eigene Darstellung)

Während der tabellarischen Anzeige der Daten hat der Benutzer die Möglichkeit, die Reihenfolge der Tabellenspalten anzupassen sowie einzelne Tabellenspalten ein- und auszublenden. Weiterhin hat er die Möglichkeit, die Spaltenwerte auf- und absteigend zu sortieren, um so einen besseren Überblick über die Daten erhalten zu können.

Während der Datenanalyse ist es möglich, die Daten auf zweierlei Arten auf Duplikate zu untersuchen. Einerseits wird dazu die ID des Datensatzes untersucht, wobei ein Duplikat potentiell dann vorliegt, wenn eine ID mehrmals im Datensatz vergeben wurde. Andererseits ist es möglich, den Datensatz auf Duplikate zu untersuchen, indem die Position eines Baumes betrachtet wird. Ist der Abstand zweier Bäume des Datensatzes geringer als ein zuvor vom Benutzer eingestellter Schwellwert, handelt es sich potentiell um ein Duplikat. Werden Duplikate festgestellt hat der Benutzer die Möglichkeit, die Duplikate von einem späteren Export ins CityGML-Datenformat auszuschließen.

Eine weitere Analysemöglichkeit der Daten ist die Validierung der Gestalt beschreibenden Baumparameter (Baumhöhe, Stammdurchmesser, Baumkronendurchmesser). Dafür sind verschiedene Kriterien implementiert, anhand der die Validität der Parameter geprüft werden kann. Bäume werden als ungültig eingestuft, wenn folgende Kriterien verletzt werden:

- Es existiert ein Wert für Baumhöhe, Stammdurchmesser und Baumkronendurchmesser
- Die Werte für Baumhöhe, Stammdurchmesser und Baumkronendurchmesser sind größer als null
- Der Stammdurchmesser ist größer als der Baumkronendurchmesser
- Der Baumkronendurchmesser ist größer als die Baumhöhe
- Der Stammdurchmesser ist größer als die Baumhöhe

Für jeden als invalide eingestuften Baum wird dem Nutzer anschließend in einer kurzen Meldung auf der Benutzeroberfläche die ID des ungültigen Baumes ausgegeben, sowie eine kurze Erklärung, warum diese invalide ist. Beim späteren Export der Daten ins CityGML-Datenformat werden die als invalide eingestuften Bäume zwar als parametrisiertes Modell exportiert, jedoch nicht als geometrisches, da für sie kein geometrisches Modell erzeugt werden kann.

Bei der Datenanreicherung ist es möglich, die vorhandenen Baumdaten mit weiteren Informationen anzureichern, um die vorhandenen Daten entweder um weitere Informationen zu erweitern oder um vorhandene Informationen zu verbessern. Es ist dabei möglich, eine Referenzhöhe für jeden Baum aus einem digitalen Geländemodell (DGM) abzuleiten oder eine einheitliche Referenzhöhe für jeden Baum festzulegen, falls dem Nutzer kein DGM vorliegt (was aber nur in flachen Gebieten sinnvoll sein kann).

Zur Ableitung der Referenzhöhe wird in einem ersten Schritt das digitale Geländemodell in die räumliche Datenbank importiert. Jeder Höhenreferenzpunkt des DGMs entspricht dabei einer Zeile einer neu

ISBN 978-3-9504173-8-8 (CD), 978-3-9504173-9-5 (print)
Editors: Manfred SCHRENK, Vasily V. POPOVICH, Peter ZEILE, Pietro ELISEI, Clemens BEYER, Judith RYSER, Christa REICHER, Canan ÇELIK

angelegten Tabelle. Diese Tabelle hat die Lage des Höhenreferenzpunktes als 2D-Geometrie gespeichert, die Höhe als numerisches Attribut. In einem zweiten Schritt wird zunächst für jeden Baum untersucht, ob er innerhalb der konvexen Hülle aller Referenzpunkte liegt. Ist das nicht der Fall, kann zu diesem Baum keine Referenzhöhe aus diesem DGM abgeleitet werden, da zur Position des Baumes keine Höheninformationen im DGM vorliegen und keine Extrapolation der Höhe erfolgen soll. Liegt der Baum aber innerhalb der Konvexen Hülle, werden die vier Höhenreferenzpunkte, die dem Baum am nächsten liegen, extrahiert. Zwischen ihnen wird anschließend mit der Inverse-Distance-Weighting-Methode interpoliert, um die Referenzhöhe des Baumes zu bestimmen. Dabei handelt es sich um eine gewichtete Mittelwertbildung, wobei räumlich näher liegende Datenwerte stärker gewichtet werden als weiter entfernte Datenwerte (Bill, 2016)

Beim Export des Baumkatasters in das CityGML-Format wird jeder Baum, der im Baumkataster vorhanden ist, als SolitaryVegetationObject ins Stadtmodell exportiert. Dabei wird einerseits ein parametrisiertes Baummodell erstellt und exportiert. Der Nutzer trifft dazu vor dem Export eine Zuordnung über die GUI, in welcher er die Baumattribute des programminternen Datenmodells in der Datenbank den Attributen der CityGML-Klasse SolitaryVegetationObject zuordnet. Andererseits wird auch ein geometrisches Baummodell erstellt und exportiert, das aus den Gestalt beschreibenden Baumparametern errechnet wird. Der Nutzer hat dabei die Möglichkeit, den verschiedenen LODs unterschiedliche geometrische Baummodelle zuzuweisen. Diese geometrischen Baummodelle unterscheiden sich erheblich in ihrer Komplexität. Je nach Komplexität der Geometrie wird dabei nach Laub- oder Nadelbaum differenziert und unterschiedliche Geometrien generiert. Alle möglichen Arten der Baumgeometriemodelle, die der Nutzer den LODs zuweisen kann, sind im Folgenden nach aufsteigender Geometriekomplexität geordnet aufgelistet:

- Linie (Abbildung 3a): Es wird an der Position jedes Baumes eine senkrecht im Raum stehende Linie generiert. Der Anfangspunkt der Linie liegt dabei auf Höhe der Erdoberfläche, der Endpunkt auf Höhe der Baumhöhe über der Erdoberfläche.
- Zylinder (Abbildung 3b): Es wird an der Position jedes Baumes ein senkrecht auf Höhe der Erdoberfläche stehender Zylinder generiert. Die Höhe des Zylinders entspricht dabei der Höhe des Baumes, der Durchmesser des Zylinders entspricht dem Durchmesser der Baumkrone.
- (texturiertes) Rechteck (Abbildung 3c): Ein (texturiertes) Rechteck im Raum wird auf Höhe der Erdoberfläche an der Position jedes Baumes generiert. Die Höhe des Rechtecks entspricht der Höhe des Baumes, die Breite entspricht dem Baumkronendurchmesser. Auf Wunsch des Nutzers können pro Baum mehrere, sich in der Mitte überkreuzende Rechtecke generiert werden, um die Sichtbarkeit auf das Baummodell aus verschiedenen Blickwinkeln zu ermöglichen.
- Kontur beschreibende Polygone (Abbildung 3d): Es werden Polygone generiert, die die Kontur des Baumes beschreiben. Diese Polygone überkreuzen sich in der Mitte, um die Sichtbarkeit auf das Baummodell aus verschiedenen Blickwinkeln zu ermöglichen.
- Stark vereinfachte Baumgeometrie (Abbildung 3e): Zur Repräsentation des Baumstammes wird ein Quader generiert. Länge und Breite dieses Quaders entsprechen dem Stammdurchmesser. Für die Baumkrone eines Laubbaumes wird ein Würfel oberhalb des Quaders generiert, für Nadelbäume eine vierseitige Pyramide. Sowohl bei Quader als auch bei Pyramide entsprechen deren Längen und Breiten dem Baumkronendurchmesser. Die Summe der Höhen von Quader und Würfel bzw. Quader und Pyramide entspricht der Höhe des Baumes.
- Vereinfachte Baumgeometrie (Abbildung 3f): Zur Repräsentation des Baumstammes wird ein Zylinder auf Höhe der Erdoberfläche generiert. Dessen Durchmesser entspricht dem Stammdurchmesser. Für die Baumkrone eines Laubbaumes wird eine Kugel, für Nadelbäume ein Kegel oberhalb des Zylinders generiert. Kugeldurchmesser bzw. Kegelgrundflächendurchmesser entsprechen dem Baumkronenradius. Die Summe von Zylinderhöhe und Kugeldurchmesser bzw. Kegelhöhe entspricht der Baumhöhe.

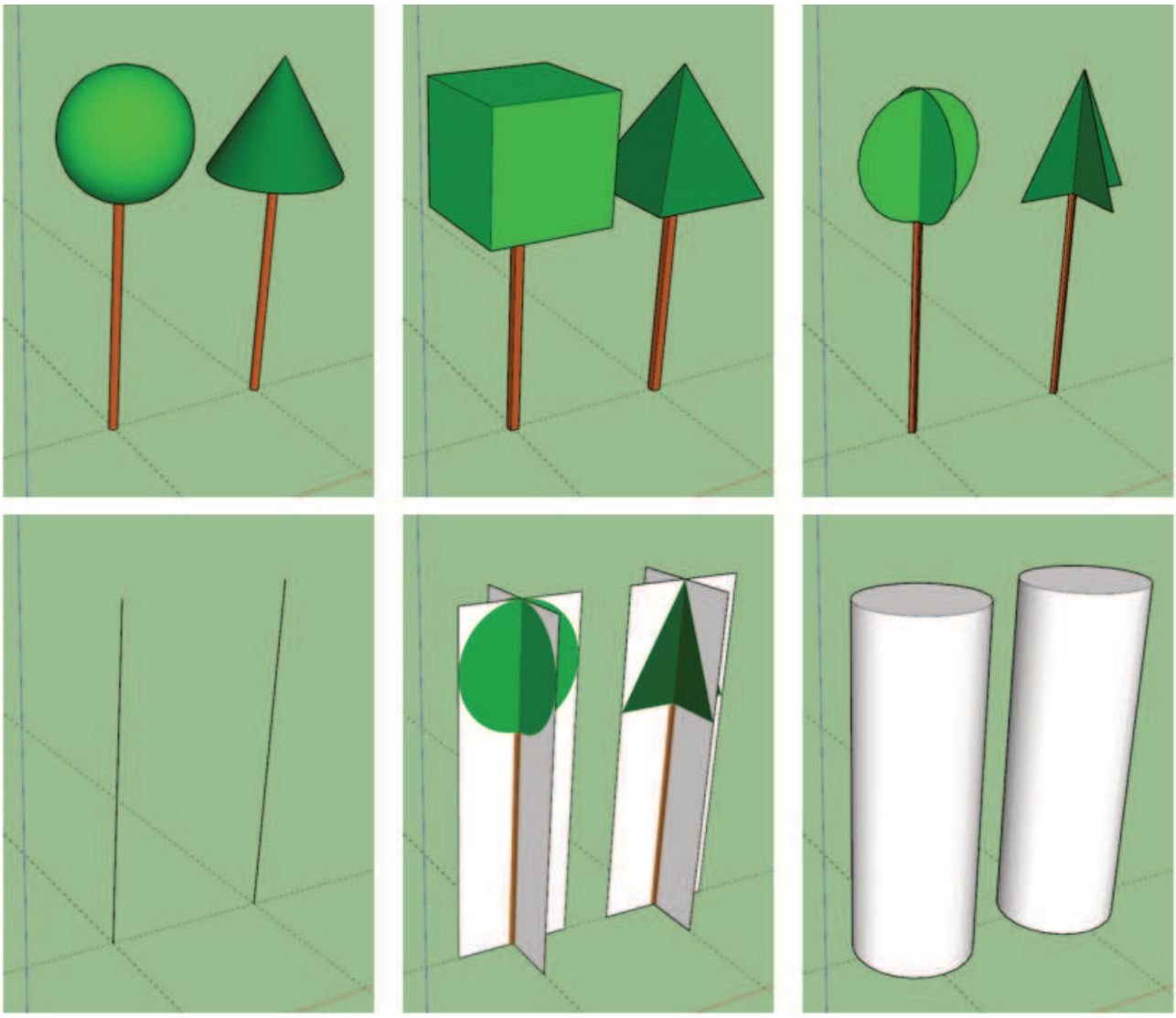


Abbildung 3a-f: Die zu erzeugenden Geometrien in den verschiedenen Detaillierungsgraden für Laub- und Nadelbäume (eigene Darstellungen)

Da das von CityGML verwendete Geometriemodell nur ebene und keine gekrümmten Flächen zulässt (Gröger et al. 2012), müssen Kugel-, Zylinder- und Kegeloberflächen durch Kombination von Polygonen angenähert werden. Der Benutzer kann dabei die Genauigkeit dieser Annäherung bestimmen, wobei die Komplexität der Geometrien mit steigender Annäherungsgenauigkeit ebenfalls zunimmt, da zu einer Steigerung der Genauigkeit die Anzahl der Flächen zunimmt.

5.2 Implementierung

In diesem Abschnitt wird kurz dargelegt, wie das zuvor vorgestellte Konzept technisch umgesetzt wird. Zur Implementierung des hier vorgestellten Programmes wird die Programmiersprache Python der Version 3 verwendet.

Als Datenbank, die zur Verwaltung des internen Datenmodells genutzt wird, kommt SQLite zum Einsatz. Bei SQLite handelt es sich um ein frei verfügbares SQL-Datenbanksystem, und hat den Vorteil, dass keine Serverarchitektur benötigt wird sondern die komplette Datenbank als seine Datei auf der Festplatte gespeichert ist (SQLite, 2020a). Unterstützung für SQLite-Datenbanken ist direkt in die Python-Standardbibliothek implementiert (Python Software Foundation, 2020), somit sind keine zusätzlichen Softwarebibliotheken nötig, um SQLite-Funktionalität in einem Python-Programm nutzen zu können. In dieser Anwendung verwendet SQLite zusätzlich die Erweiterung SpatiaLite, eine quelloffene Programmbibliothek, die SQLite um Funktionalitäten einer räumlichen Datenbank erweitert (Furieri, 2020).

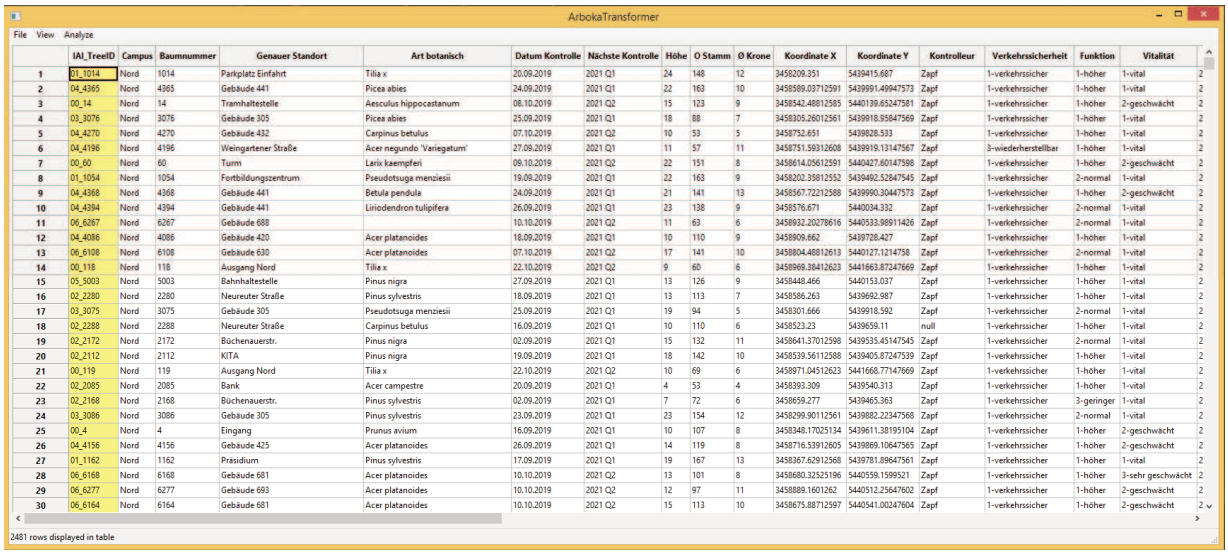
ISBN 978-3-9504173-8-8 (CD), 978-3-9504173-9-5 (print)
Editors: Manfred SCHRENK, Vasily V. POPOVICH, Peter ZEILE, Pietro ELISEI, Clemens BEYER, Judith RYSER, Christa REICHER, Canan ÇELIK

Die graphische Benutzeroberfläche wird mit der Bibliothek wxPython realisiert. Der Programmcode für eine Standardbenutzeroberfläche wird dazu mit dem Programm wxFormBuilder automatisiert generiert. Diese Standardbenutzeroberfläche wird dann durch eigenen Code ergänzt und angepasst.

Da die im Baumkataster vergebenen Baumnummern lediglich innerhalb eines Bereiches eindeutig sind, ist es möglich, eine ID aus einer Kombination der Bereichsnummer und der Baumnummer zu generieren. Dem Nutzer wird also beim Import der Daten die Möglichkeit gegeben, zwei Spalten auszuwählen, die dann miteinander durch einen Unterstrich verbunden werden. Hat ein Baum in Bereich 04 beispielsweiße die Baumnummer 4394, wird daraus die ID 04_4394 generiert. Diese ID wird in einer neuen Spalte IAI_TreeID in der Datenbank gespeichert. Da bei der weiteren Benutzung des Programmes häufig auf einzelne Bäume in der Datenbank zugegriffen werden muss, wird aus Gründen der Effizienz ein Index für die Spalte IAI_TreeID generiert.

Für den Fall, dass in den eingelesenen Daten keine ID vorhanden ist, ist es außerdem möglich, die von SQLite automatisiert generierte ROWID als ID zu verwenden. Es handelt sich dabei um eine Integer-Zahl, die automatisiert von SQLite vergeben wird, über die jede Zeile in einer Datenbanktabelle angesprochen werden kann (SQLite, 2020b).

Im Datenansichtsfenster des Programms werden nach dem Einleseprozess die Daten aller Bäume tabellarisch angezeigt, wie Abbildung 4 zeigt. Vom Programm automatisiert generierte Spalten (wie z.B. die der ID) werden dabei mit gelber Farbe hinterlegt um dem Nutzer anzuzeigen, dass diese Spalte nicht dem Baumkataster entstammt.



	IAI_TreeID	Campus	Baumnummer	Genauer Standort	Art botanisch	Datum Kontrolle	Nächste Kontrolle	Höhe	Ø Stamm	Ø Krone	Koordinate X	Koordinate Y	Kontrolleur	Verkehrssicherheit	Funktion	Vitalität	
1	01_1014	Nord	1014	Parkplatz Einfahrt	Tilia x	20.09.2019	2021 Q1	24	148	12	3458209.351	5439415.687	Zapf	1-verkehrssicher	1-höher	1-vital	2
2	04_4365	Nord	4365	Gebäude 441	Picea abies	24.09.2019	2021 Q1	22	163	10	3458589.03712591	5439991.49947573	Zapf	1-verkehrssicher	1-höher	1-vital	2
3	00_14	Nord	14	Tramhaltestelle	Aesculus hippocastanum	08.10.2019	2021 Q2	15	123	9	3458542.48812585	5440139.65247581	Zapf	1-verkehrssicher	1-höher	2-geschwächt	2
4	03_3076	Nord	3076	Gebäude 305	Picea abies	25.09.2019	2021 Q1	18	88	7	3458305.26012561	5439918.95847569	Zapf	1-verkehrssicher	1-höher	1-vital	2
5	04_4270	Nord	4270	Gebäude 432	Carpinus betulus	07.10.2019	2021 Q2	10	53	5	3458752.651	5439828.533	Zapf	1-verkehrssicher	1-höher	1-vital	2
6	04_4196	Nord	4196	Weingartener Straße	Acer negundo 'Variegatum'	27.09.2019	2021 Q1	11	57	11	3458751.59312608	5439919.13147567	Zapf	3-wiederherstellbar	1-höher	1-vital	2
7	00_60	Nord	60	Turm	Larix kaempferi	09.10.2019	2021 Q2	22	151	8	3458614.05612591	5440427.60147598	Zapf	1-verkehrssicher	1-höher	2-geschwächt	2
8	01_1054	Nord	1054	Fortbildungszentrum	Pseudotsuga menziesii	19.09.2019	2021 Q1	22	163	9	3458202.35812552	5439492.52847545	Zapf	1-verkehrssicher	2-normal	1-vital	2
9	04_4368	Nord	4368	Gebäude 441	Betula pendula	24.09.2019	2021 Q1	21	141	13	3458567.72212588	5439990.30447573	Zapf	1-verkehrssicher	1-höher	2-geschwächt	2
10	04_4394	Nord	4394	Gebäude 441	Liriodendron tulipifera	26.09.2019	2021 Q1	23	138	9	3458576.671	5440034.332	Zapf	1-verkehrssicher	2-normal	1-vital	2
11	06_6267	Nord	6267	Gebäude 688		10.10.2019	2021 Q2	11	63	6	3458932.20278616	5440533.98911426	Zapf	1-verkehrssicher	2-normal	1-vital	2
12	04_4086	Nord	4086	Gebäude 420	Acer platanoides	18.09.2019	2021 Q1	10	110	9	3458909.662	5439728.427	Zapf	1-verkehrssicher	1-höher	1-vital	2
13	06_6108	Nord	6108	Gebäude 630	Acer platanoides	07.10.2019	2021 Q2	17	141	10	3458804.48812613	5440127.1214758	Zapf	1-verkehrssicher	2-normal	1-vital	2
14	00_118	Nord	118	Ausgang Nord	Tilia x	22.10.2019	2021 Q2	9	60	6	3458969.38412623	5441663.87247669	Zapf	1-verkehrssicher	1-höher	1-vital	2
15	05_5003	Nord	5003	Bahnhaltestelle	Pinus nigra	27.09.2019	2021 Q1	13	126	9	3458448.466	5440153.037	Zapf	1-verkehrssicher	1-höher	1-vital	2
16	02_2280	Nord	2280	Neureuter Straße	Pinus sylvestris	18.09.2019	2021 Q1	13	113	7	3458586.263	5439692.987	Zapf	1-verkehrssicher	1-höher	1-vital	2
17	03_3075	Nord	3075	Gebäude 305	Pseudotsuga menziesii	25.09.2019	2021 Q1	19	94	5	3458301.666	5439918.592	Zapf	1-verkehrssicher	2-normal	1-vital	2
18	02_2288	Nord	2288	Neureuter Straße	Carpinus betulus	16.09.2019	2021 Q1	10	110	6	3458523.23	5439659.11	null	1-verkehrssicher	1-höher	1-vital	2
19	02_2172	Nord	2172	Büchenauerstr.	Pinus nigra	02.09.2019	2021 Q1	15	132	11	3458641.37012598	5439535.45147545	Zapf	1-verkehrssicher	2-normal	1-vital	2
20	02_2112	Nord	2112	KITA	Pinus nigra	19.09.2019	2021 Q1	18	142	10	3458539.56112588	5439405.87247539	Zapf	1-verkehrssicher	1-höher	1-vital	2
21	00_119	Nord	119	Ausgang Nord	Tilia x	22.10.2019	2021 Q2	10	69	6	3458971.04512623	5441668.77147669	Zapf	1-verkehrssicher	1-höher	1-vital	2
22	02_2085	Nord	2085	Bank	Acer campestre	20.09.2019	2021 Q1	4	53	4	3458393.309	5439540.313	Zapf	1-verkehrssicher	1-höher	1-vital	2
23	02_2168	Nord	2168	Büchenauerstr.	Pinus sylvestris	02.09.2019	2021 Q1	7	72	6	3458659.277	5439465.363	Zapf	1-verkehrssicher	3-geringer	1-vital	2
24	03_3086	Nord	3086	Gebäude 305	Pinus sylvestris	23.09.2019	2021 Q1	23	154	12	3458299.90112561	5439882.22347568	Zapf	1-verkehrssicher	2-normal	1-vital	2
25	00_4	Nord	4	Eingang	Prunus avium	16.09.2019	2021 Q1	10	107	8	3458348.17025134	5439611.38195104	Zapf	1-verkehrssicher	1-höher	2-geschwächt	2
26	04_4156	Nord	4156	Gebäude 425	Acer platanoides	26.09.2019	2021 Q1	14	119	8	3458716.53912605	5439869.10647565	Zapf	1-verkehrssicher	1-höher	2-geschwächt	2
27	01_1162	Nord	1162	Präsidium	Pinus sylvestris	17.09.2019	2021 Q1	19	167	13	3458367.62912568	5439781.89647561	Zapf	1-verkehrssicher	1-höher	1-vital	2
28	06_6168	Nord	6168	Gebäude 681	Acer platanoides	10.10.2019	2021 Q2	13	101	8	3458680.32525196	5440559.1599521	Zapf	1-verkehrssicher	1-höher	3-sehr geschwächt	2
29	06_6277	Nord	6277	Gebäude 693	Acer platanoides	10.10.2019	2021 Q2	12	97	11	3458889.1601262	5440512.25647602	Zapf	1-verkehrssicher	1-höher	2-geschwächt	2
30	06_6164	Nord	6164	Gebäude 681	Acer platanoides	10.10.2019	2021 Q2	15	113	10	3458675.88712597	5440541.00247604	Zapf	1-verkehrssicher	1-höher	2-geschwächt	2

Abbildung 4: Datenansichtsfenster des Programmes unter Windows 8.1 mit tabellarischer Anzeige der Baumdaten des KIT Campus Nord (eigene Darstellung)

Über einen Rechtsklick auf eine Spaltenüberschrift hat der Benutzer die Möglichkeit, die Tabelle nach dieser Spalte auf- oder absteigend zu sortieren oder diese Spalte auszublenden. Außerdem kann er per Drag-and-Drop die Reihenfolge der Spalten verändern. Über das Hauptmenü können alle wichtigen Funktionen des Programmes aufgerufen werden, wie z.B. das Einlesen einer Datei mit Baumdaten, den Export eines Datensatzes als CityGML, die Analyse der Daten und die Erweiterung der Daten. Zum jetzigen Zeitpunkt ist lediglich ein CityGML-Export des parametrisierten Baummodells möglich, Generierung und Export eines geometrischen Baummodells ist noch nicht möglich.

6 FAZIT UND AUSBLICK

Wie in diesem Beitrag gezeigt wurde, wird mit diesem Programm ein Werkzeug geschaffen, das es ermöglicht, strukturierte, nicht-standardisierte Baumdaten in das standardisierte, semantische Datenmodell CityGML zu überführen. Da das Datenformat der OGC-Standard für 3D-Stadtmodelle ist, kann es von vielen Programmen eingelesen und weiterverarbeitet werden. Es stellt also das Bindeglied zwischen Programmen zur Baumkatasterverwaltung und Programmen der Visualisierung und Analyse von Stadtmodellen dar.

Im weiteren Verlauf der Entwicklung dieses Programmes soll zunächst der Export der 3D-Baumgeometrien in den verschiedenen LODs realisiert werden. Der Nutzer soll außerdem auswählen können, ob weitere

Attribute, die zwar im Baumkataster erfasst aber nicht von der CityGML-Klasse SolitaryVegetationObject abgebildet werden können, als generische Attribute ins Datenmodell eingefügt werden sollen. Es sollen außerdem weitere Funktionen zur Anreicherung der Baumdaten implementiert werden. So soll es möglich sein, die im Baumkataster gespeicherte Baumhöhe mithilfe eines Laserscanning-Datensatzes aus einer Befliegung des Geländes zu verifizieren. Um realitätsgetreuere 3D-Modelle der Bäume zu erstellen, ist es außerdem notwendig die Höhe des Laubansatzes des Baumes zu kennen. Diese wird allerdings bei den Baumkontrollen häufig nicht erfasst und ist daher nicht im Baumkataster gespeichert. Daher soll untersucht werden, ob es möglich ist, diese ebenfalls aus den Laserscanning-Daten der Befliegung abzuleiten.

7 LITERATURVERZEICHNIS

BILL; RALF: Grundlagen der Geoinformationssysteme, 6., völlig neu bearbeitete und erweiterte Auflage, Berlin: Wichmann, 2016

FORSCHUNGSGESELLSCHAFT LANDSCHAFTSENTWICKLUNG LANDSCHAFTSBAU E.V.: Baumkontrollrichtlinien. Richtlinien für die Regelkontrollen zur Überprüfung der Verkehrssicherheitvon Bäumen, 2010

GRÖGER, G., KOLBE, T.H., NAGEL, C., HÄFELE, K.-H.: OGC City Geography Markup Language (CityGML) Encoding Standard, 2012

INOVAGIS: Arbokat® Baumkataster, URL: http://inovagis.com/wp-content/uploads/2016/11/Arbokat_Infoflyer.pdf, letzter Zugriff: 08.01.2020, 2016

INOVAGIS: Benutzerhandbuch Arbokat® Baumkataster, o.D.

KOLBE, T.H.: Representing and Exchanging 3D City Models with CityGML. In: 3D geo-information sciences, pp. 15-31, Berlin: Springer, 2009

FURIERI, A.: SpatiaLite, URL: https://www.gaia-gis.it/fossil/libspatialite/home, letzter Zugriff: 22.01.2020

PYTHON SOFTWARE FOUNDATION: sqlite3 - DB-API 2.0 interface for SQLite databases, URL https://docs.python.org/3.6/library/sqlite3.html, letzter Zugriff: 22.01.2020.

SQLITE: About SQLite, URL: https://www.sqlite.org/about.html, letzter Zugriff: 22.01.2020a

SQLITE: Rowid Tables, URL: https://www.sqlite.org/rowidtable.html, letzter Zugriff: 16.01.2020b

STADT STUTTGART (2020). Baumkataster - Grünflächenmanagement. URL: https://www.stuttgart.de/item/show/13557/1, letzter Zugriff: 21. 01. 2020

WESSOLY, L., ERB, M.: Handbuch der Baumstatik und Baumkontrolle [Neuausg.], Berlin: Patzer, 2014

ISBN 978-3-9504173-8-8 (CD), 978-3-9504173-9-5 (print)
Editors: Manfred SCHRENK, Vasily V. POPOVICH, Peter ZEILE, Pietro ELISEI, Clemens BEYER, Judith RYSER, Christa REICHER, Canan ÇELIK