

3D-Punktwolken in PostgreSQL/PostGIS für Anwendungen in Geoinformationssystemen

THERESA MEYER¹ & ANSGAR BRUNN¹

Zusammenfassung: Neben dem dateibasierten Management von 3D-Punktwolken besteht seit einigen Jahren auch die Möglichkeit, diese gemeinsam mit anderen Geodaten in räumlichen Datenbanksystemen vorzuhalten und dabei von den Vorteilen der kombinierten Analyse von georeferenzierten Punktwolken-, Raster- und Vektordaten zu profitieren. In diesem Beitrag wird vorgestellt, wie die Punktwolkenerweiterung „Pointcloud“ für die PostgreSQL/PostGIS Datenbank mit ihren speziellen Methoden und der Integration von PostGIS-Funktionen genutzt werden kann, um auch räumliche Abfragen in 3D zu realisieren. Im Besonderen werden erweiterte Möglichkeiten zur Punktwolkenskachelung durch Kombinationen von Funktionen der Point Data Abstraction Library (PDAL) aufgezeigt und dahingehend untersucht, in wie weit sich unterschiedliche Kachelmethoden und -konfigurationen auf Speicherplatzbedarf, Abfrageperformance und Interaktionen zwischen Punktwolken und anderen 2D und 3D Geoobjekten in der Datenbank auswirken. Die Ergebnisse zeigen, dass eine individuell gewählte Speicherstruktur für eine Punktwolke unter Berücksichtigung ihrer dreidimensionalen räumlichen Ausdehnung und Punktdichte ausschlaggebend für einen geringen Speicherverbrauch und performante Abfragen in PostGIS-Anwendungen ist.

1 Einleitung

Als Quelle für räumliche Informationen gewinnen 3D-Punktwolken in Anwendungsfeldern der Geodäsie, Geoinformatik, Architektur und Bauarchäologie zunehmend Bedeutung. Obwohl moderne Messinstrumente wie Laserscanner, Unmanned-Aerial-Systems (UAS) und Mobile Mapping Systeme die Erzeugung von Punktwolken immer mehr vereinfachen und eine große Auswahl von anwendungsspezifischer Auswertesoftware zur Verfügung steht, stellt die effiziente Verwaltung von charakteristisch großen Punktwolkendaten im Alltagsbetrieb von Vermessungs- und Ingenieurbüros eine Herausforderung dar. Punktwolken dienen als Grundlage zur Lösung verschiedenster ingenieurtechnischer Aufgaben und haben schon lange Einzug in die Projekte auch von kleinen und mittleren Unternehmen gehalten. Sie werden z. B. zu Zwecken der BIM-konformen As-Built-Dokumentation von Gebäuden und Anlagen erzeugt (TANG et al. 2010), für Veränderungsdetektionen und Deformationsanalysen von Ingenieurbauwerken (MUKUPA et al. 2016) oder sie dienen zur hochauflösenden, dreidimensionalen Dokumentation von Urgelände (BURGER et al. 2015), weshalb sie mittlerweile auch zu einer unverzichtbaren Planungsgrundlage für Bauprojekte aller Art geworden sind.

Für die Verarbeitung, die Speicherung und den Austausch zwischen Anwendungsprogrammen werden für Punktwolken üblicherweise eigens entwickelte Dateiformate wie LAS (ASPRS 2013) PCD (RUSU & COUSINS 2011) oder E57 (HUBER 2011) genutzt, die es ermöglichen neben X-, Y- und Z-Koordinaten auch zusätzliche Punktattribute wie *Intensity*, *Return Number*, *Red*, *Green*,

¹ Hochschule für angewandte Wissenschaften Würzburg - Schweinfurt, Fakultät Kunststofftechnik und Vermessung, Röntgenring 8, D-97070 Würzburg, E-Mail: [Theresa.Meyer, Ansgar.Brunn]@fhws.de

Blue, NIR, Classification, Point Source ID verlustfrei komprimiert zu transportieren. Alternativ zur dateibasierten Verwaltung gibt es seit einigen Jahren jedoch auch die Möglichkeit 3D-Punktwolken gemeinsam mit anderen Geodaten in räumlichen Datenbanksystemen („Geodatenbanken“) vorzuhalten und dabei von der kombinierten Analyse von georeferenzierten Punktwolken-, Raster- und (2D- und 3D-) Vektordaten zu profitieren. Vorteil einer (Geo-) Datenbank-Lösung ist neben der zentralen Datenhaltung die Tatsache, dass räumliche und nicht räumliche Daten aller Art, wie Punktwolken, Rasterdaten, Vektordaten (z. B. Katasterdaten, Bestands- und Planungsdaten in Form von CAD- und BIM-Modellen), Fachdaten und administrative Daten jederzeit miteinander in Beziehung gesetzt werden können.

Dieser Beitrag handelt von 3D-Punktwolken in ihrer Rolle als Objekte in räumlichen Datenbanksystemen und speziell der Punktwolken-Erweiterung „Pointcloud“, die seit dem Jahr 2013 für das freie und quelloffene Geodatenbanksystem *PostgreSQL/PostGIS* entwickelt wird. *Pointcloud* erweitert PostGIS um neue Datentypen und eine Reihe eigener Methoden, die es ermöglichen 3D-Punktwolken in Datenbanktabellen zu speichern und mittels GIS-Tools zu analysieren. Im Besonderen geht es um die Möglichkeiten der zwei- und dreidimensionalen Kachelung von 3D-Punktwolken mittels Funktionen der *Point Data Abstraction Library (PDAL)*. Anhand eines beispielhaften PostGIS/Pointcloud-Workflows für Punktwolken-Abfragen im dreidimensionalen Raum, werden die Ergebnisse von Untersuchungen der Auswirkungen unterschiedlicher Kachel-Methoden und –Konfigurationen auf Speicherplatzbedarf, Abfrageperformance und Interaktionen zwischen Punktwolken und Vektordaten präsentiert. Es wird außerdem eine Auswahl von open-source Tools und Programmcode-Bibliotheken für die Bearbeitung und Analyse von 3D-Punktwolken vorgestellt und skizziert wie sich diese in datenbankbasierte Arbeitsabläufe integrieren lassen. Der Artikel schließt mit einer Einschätzung, in wie weit sich das Geoobjekt „3D-Punktwolke“ in die Strukturen der PostGIS-Datenbank integrieren lässt und einem Blick auf alternative Lösungen auch im Bereich kommerzieller Software.

2 3D-Punktwolken in GIS

Punktwolken werden z. B. für die Generierung von 3D-Gebäudemodellen zum Aufbau von 3D-Stadtmodellen (HAALA & BRENNER 1997), zur Vegetationsdetektion in urbanen Gebieten (HÖFLE & HOLLAS 2010), als Grundlage für Gelände- und Objekt-Modellierungen bei Hochwassersimulationen (MERWADE et al. 2008) oder für die Extraktion von Dach- und Fassadenflächen für Solarpotentialanalysen (VÖGTLE et al. 2005; JOCHEM et al. 2011) benötigt.

Im Umgang mit Punktwolken sind zwei grundsätzliche Fälle zu unterscheiden. Lassen sich die Punkthöhen (Z-Koordinaten) als Funktion der Lage modellieren besitzt, eine Punktwolke die charakteristische Eigenschaft von 2,5D Objekten, bei denen für jede Kombination von X und Y nur ein einziger Z-Wert existiert. Solche Punktwolken resultieren i. d. R. aus luftgestützten Erfassungsmethoden und können auch im Rasterformat gespeichert sowie mit herkömmlichen Raster-Tools in einem GIS verarbeitet werden. Handelt es sich dagegen um unstrukturierte Punktwolken mit ausgeprägtem 3D-Charakter, wie sie z. B. aus terrestrischem oder mobilem Laserscanning hervorgehen, entsteht der Bedarf nach „richtigen“ 3D-Speicherstrukturen und –

Operationen, da eine Rasterkonvertierung einen enormen und irreversiblen Informationsverlust zur Folge hätte.

Allgemeiner Standard für GIS-Anwendungen in 3D ist das *Feature-Geometry-Modell*, das dreidimensionale Koordinaten erlaubt und eine Reihe von Klassen für 3D-Oberflächen, Dreiecksvermaschungen (TINs), 3D-Körper und 3D-Topologie bereitstellt. Auch das erweiterte Datenbankschema *SQL/MM Spatial* unterstützt seit der dritten Version dreidimensionale Koordinaten und umfasst, im Gegensatz zum *Simple-Feature-Modell*, entsprechende 3D-Methoden. Während es bewährte und standardisierte Modelle und Strukturen für Raster- und vor allem Vektordaten in Geodatenbanksystemen gibt, ist die Art und Weise wie die Integration von originalen 3D-Punktwolken zu handhaben ist jedoch noch nicht spezifiziert oder standardisiert. Derzeit widmet sich die OGC-Arbeitsgruppe „Point Cloud“ der Definition von Anforderungen an Interoperabilität und der Implementierung von Standards im Anwendungsfeld von 3D-Punktwolken (OGC 2018).

3 Pointcloud-Erweiterung für PostgreSQL/PostGIS

Obwohl PostGIS mit POINT Z einen Geometrietyp für Punkte mit dreidimensionalen Koordinaten bereitstellt, ist dieser Datentyp für die Speicherung von 3D-Punktwolken ungeeignet. Grund hierfür ist schlicht die Tatsache, dass die Anzahl von Punkten in einer Punktwolke aus z. B. terrestrischem Laserscanning mit bis zu mehreren Milliarden Punkten viel zu groß ist, als dass jeder Punkt in einer einzelnen Tabellenzeile gespeichert werden könnte. Für die Speicherung von 3D-Punktwolken bei gleichzeitig effizientem Zugriff auf alle Punktattribute in einer PostgreSQL-/PostGIS Datenbank wurde deshalb die Erweiterung *Pointcloud* entwickelt. Sie erweitert PostGIS um die zwei neuen Objekttypen *PcPoint* und *PcPatch*. Räumlich benachbarte Punkte werden zu mehreren hundert bis tausenden als sog. *Patches* (Kacheln/ Blöcke) in Form von Byte-Arrays organisiert. Jedes Punktwolken-Patch entspricht einer Instanz der Klasse *PcPatch* und wird in einer eigenen Tabellenzeile gespeichert. Die inhaltliche Struktur von Patches wird durch ein XML-Schemadokument beschrieben. Es enthält genaue Informationen darüber, welche Punkt-Attribute an welcher Stelle und mit welchem Datentyp vorliegen. Beispielsweise könnte ein Intensitätswert je nach Auflösung als 4-Byte Integer oder in einem einzelnen Byte gespeichert und X-, Y- und Z-Koordinaten durch Double-Werte oder als 4-Byte Integer mit Skalierungsfaktor repräsentiert werden.

3.1 Erweiterte Punktwolkenkachelung mit PDAL-Funktionen

Punktwolken werden beim Schreiben in die Postgres-Datenbank praktischerweise in Patches gekachelt, damit sie in Tabellen komprimiert gespeichert und gleichzeitig effizient abgefragt werden können. Die Punktwolkenkachelung wird standardmäßig während des Imports im Zuge der Ausführung von Kommandozeilen-Anweisungen und Funktionen der *Point Data Abstraction Library* (PDAL) realisiert. PDAL stellt zwei grundsätzliche Funktionen zur Punktwolkenkachelung bereit:

- Ungleichmäßige Kachelung mit *filters.chipper()*:
Diese Funktion bündelt benachbarte Punkte in einzelne nicht überlappende Patches. Die

maximale Anzahl von Punkten in einem Patch wird durch einen Vorgabewert für den Funktions-Parameter *capacity* festgelegt. Als Standardwert sind 5000 Punkte pro Kachel vorgesehen (PDAL, 2018). Die Punkte werden so sortiert, dass eine lineare Ordnung im zweidimensionalen Raum entsteht. Die Z-Koordinaten bleiben gänzlich unberücksichtigt. Die sortierte Punktliste wird dann in einzelne Patch-Arrays gemäß der vorgegebenen Kapazität unterteilt. Der Algorithmus prüft für jedes Array dessen Ausdehnung in X- und Y-Richtung und unterteilt neu, falls diese zu unausgewogen ist.

- Gleichmäßige Gitter-Kachelung mit *filters.splitter()*:
Bei Anwendung dieser Funktion werden Punktwolkenpunkte in quadratische Patches, mit einer vom Anwender vorgegebenen Seitenlänge (in Meter) gebündelt. Der Kachelursprung wird entweder zufällig vom Algorithmus gewählt oder kann in Form einer X- und Y-Koordinate festgelegt werden. Auf diese Weise entstehen Patches mit derselben Ausdehnung in der X/Y-Ebene, die sich jedoch in der Anzahl ihrer Punkte unterscheiden.

Unabhängig von der gewählten Kachel-Methode und Patch-Größe muss sich der Anwender der Tatsache bewusst sein, dass die Punktwolken-Kachelung mit *filters.chipper()* und *filters.splitter()* in beiden Fällen keine 3D- sondern vielmehr eine 2D-Kachelung ist. Die individuelle Ausdehnung einer Punktwolke entlang der Z-Achse wird nicht berücksichtigt. Das hat zur Folge, dass auch Punkte, die aufgrund unterschiedlicher Z-Werte im 3D-Raum unter Umständen sehr weiter auseinanderliegen dennoch demselben Patch-Array zugeordnet werden. Besonders für Punktwolken mit ausgeprägtem 3D-Charakter können auf diese Weise sehr unregelmäßige und unausgewogene Patches entstehen (Abb. 1). Um Punktwolkenpatches auch in ihrer Ausdehnung entlang der Z-Achse zu limitieren, bietet sich ein neuartiger Kachel-Prozess unter Hinzunahme der PDAL-Funktion *filters.range()* an. Die Funktion ist ein Bereichsfilter, der in Anwendung für die Dimension ‚Z‘ eine Punktwolke in beliebig viele Höhenschichten (z. B. 1 m oder 5 m) unterteilt. In Kombination mit *filters.chipper()* und *filters.splitter()* kann *filters.range()* dazu verwendet werden eine um ‚Z‘ erweiterte dreidimensionale Punktwolkenkachelung zu erzeugen (Abb. 2).

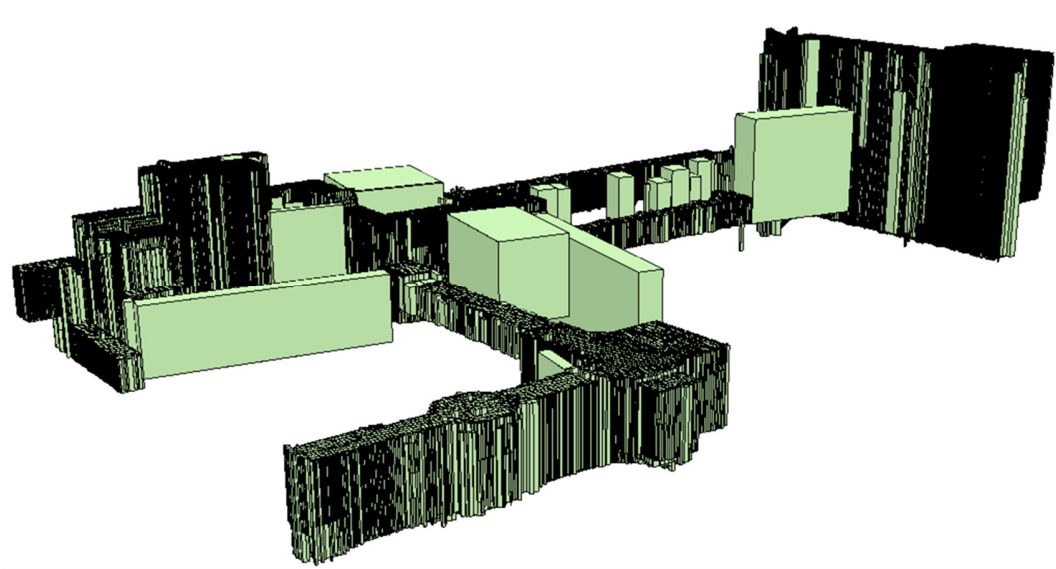


Abb. 1: Kachelung einer Punktwolke mit 3 Millionen Punkten mit `filters.chipper()` und Kapazität 100

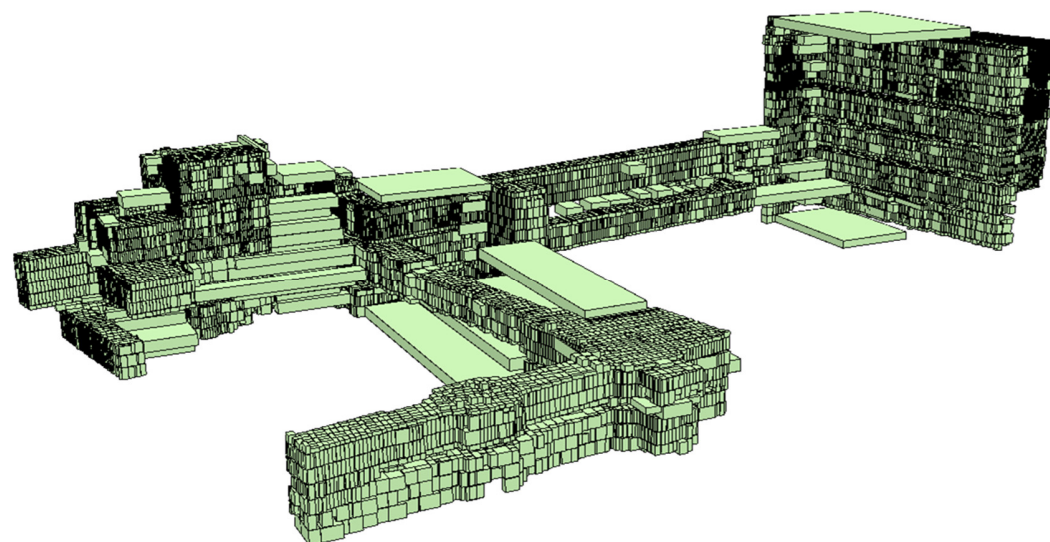


Abb. 2: Kachelung einer Punktwolke mit 3 Millionen Punkten mit `filters.chipper()`, Kapazität 100 und Beschränkung auf 1m entlang der Z-Achse

3.2 Pointcloud-Funktionen und Integration von PostGIS

Sämtliche Funktionen der Pointcloud-Erweiterung beziehen sich auf die beiden zentralen Pointcloud-Objekte `PcPoint` und `PcPatch`. `PcPoints` spielen insgesamt eine eher untergeordnete Rolle, denn es ist zwar möglich `PcPoint`-Tabellen nach gültigem Schema anzulegen und mit Punkten zu füllen, allerdings werden diese Einzelpunkte in erster Linie nur für spezielle Abfragen, z. B. als Zwischenschritt zur Konvertierung einer kleinen Ergebnismenge von Punktwolken-Punkten in PostGIS-Punkte vom Typ `POINT Z` benötigt. Ganze Punktwolken werden in `PcPatch`-Tabellen verwaltet und zu diesem Zweck stellt Pointcloud eine Reihe spezieller Funktionen bereit. Dazu gehören u. a. solche zum Erzeugen und Auflösen, Umsortieren und Komprimieren/

Dekomprimieren von Patches (z. B. *PC_MakePatch()*, *PC_Union()*, *PC_Sort()* und *PC_Compress()*) oder Funktionen zum Abruf von genauen Informationen über deren Inhalt (z. B. *PC_Summary()*, *PC_NumPoints()* oder *PC_PatchMax()* und *PC_PatchMin()*). Darüber hinaus gibt es die Pointcloud-Funktionen *PC_FilterGreaterThan()*, *PC_FilterLessThan()*, *PC_FilterBetween()* und *PC_FilterEquals()*, zum Filtern von Punkten auf Basis von Attributwerten wie Intensität, Klassifizierungscode, Z-Wert oder Zeitstempel.

Erst mit der zusätzlichen Erweiterung *Pointcloud_PostGIS* werden Interaktionen zwischen Pointcloud- und PostGIS-Objekten möglich. Unter Verwendung der Funktion *PC_Intersects()* können Patches dahingehend überprüft werden, ob sie die Geometrie eines PostGIS-Objekts überschneiden. Mit *PC_Intersection()* werden neue Patches erzeugt, die die Ergebnismenge einer Überschneidung zwischen einem PcPatch und z. B. einem Polygon darstellen. Sowohl *PC_Intersects()* als auch *PC_Intersection()* akzeptieren ausschließlich 2D-Objekte und ermöglichen deshalb keine räumlichen Abfragen in 3D. Die Funktion *PC_EnvelopeGeometry()* gibt die zweidimensionale Begrenzung eines Patches als PostGIS-Polygon zurück und *Pc_BoundingDiagonalGeometry()* ermöglicht durch die Rückgabe der 3D-Boundingbox-Diagonalen eines Patches als LINESTRING Z den Aufbau eines 3D-Indexes.

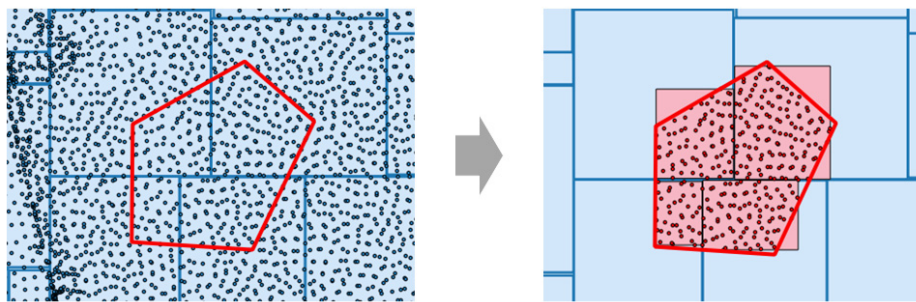


Abb. 3: Anwendung der Pointcloud_PostGIS-Funktion *PC_Intersection()*

3.3 Exemplarischer Workflow einer mehrstufigen Abfrage in 3D

Für Intersection-Abfragen mit PcPatch-Objekten bietet Pointcloud in Kombination mit der Pointcloud_Postgis-Erweiterung nur die Methoden *PC_Intersects()* und *PC_Intersection()*. Beide Funktionen akzeptieren aber nur PcPatch- und zweidimensionale PostGIS-Objekte und keine 3D-Körper vom Typ POLYHEDRALSURFACE Z. Somit können Pointcloud-Punkte zunächst nur projiziert auf die Grundrissebene abgefragt werden. Für die Selektion einer Punktwolken-Teilmenge, die sich innerhalb eines 2,5D-Körpers im 3D-Raum befindet, kann die Beschränkung auf den zweidimensionalen Raum durch eine mehrstufige Abfrageprozedur umgangen werden:

1. Mit der Pointcloud-Funktion *PC_BoundingDiagonalGeometry()* werden die Diagonalen der 3D-Boundingboxen aller Patches als PostGIS 3D-Linien vom Typ LINE Z erzeugt.
2. Um performante Abfragen in 3D zu ermöglichen wird für die Diagonalen bzw. deren 3D-Boundingboxen ein 3D-Index aufgebaut.

3. Durch Anwendung des Index-Operators &&& werden die 3D-Boundingboxen aller Patch-Diagonalen selektiert, welche die 3D-Boundingbox eines beliebigen 3D-Objekts vom Typ POLYHEDRALSURFACE Z überschneiden.
4. Die 3D-Boundingboxen der Patch-Diagonalen werden mit der PostGIS-Funktion *Box3D()* in Volumenkörper konvertiert, sodass der exakte Test auf Überschneidung unter Verwendung der Operation *ST_3DIntersects()* ausgeführt werden kann (nur mit SFCGAL Backend).

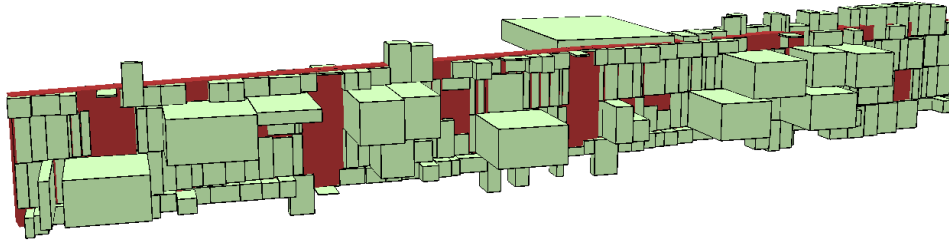


Abb. 4: Ergebnis von *ST_3DIntersects()*

5. Mit einem Inner Join über ein mitgeführtes Schlüsselattribut (ID) werden die selektierten Volumenkörper auf die ursprünglichen PcPatch-Objekte zurückgeführt.
6. Die vorselektierten Punktwolken-Patches werden erst jetzt mit Hilfe der speziellen Pointcloud-Funktion *PC_Intersection()* auf die Begrenzungen des Abfrage-Objekts in X und Y „zugeschnitten“.

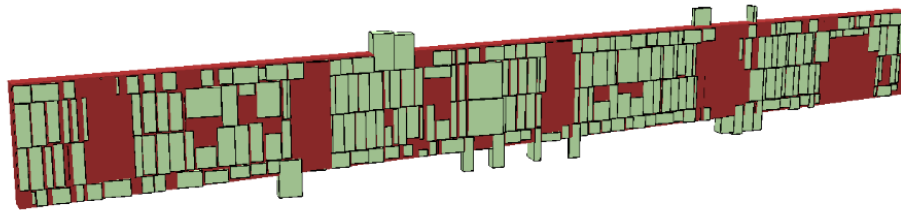


Abb. 5: Ergebnis von *PC_Intersection()*

7. Je nach Kachel-Methode verbleiben in den Patches immer noch 3D-Punkte, die ober- bzw. unterhalb des Abfrage-Objekts liegen. Für Abfragen mit 2,5D-Objekten bietet sich deshalb die Pointcloud-Funktion *PC_FilterBetween()* an. Mit ihr lässt sich die minimale und maximale Z-Ausdehnung der finalen Ergebnis-Patches exakt beschränken.

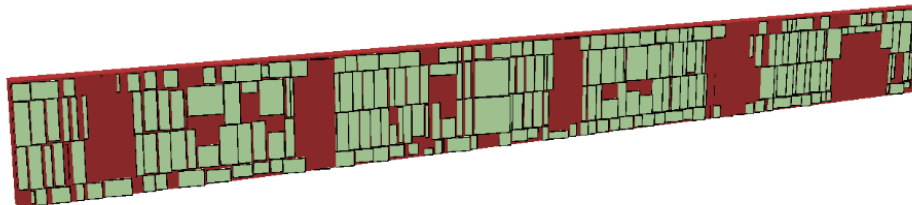


Abb. 6: Ergebnis von *PC_FilterBetween()*

Als Ergebnis dieser Abfragen liegt eine Auswahl von Punktwolken-Patches vor, welche ausschließlich 3D-Punkte beinhalten, die im Inneren oder auf der Außenseite eines Abfragekörpers

im 3D-Raum liegen. Es könnte sich dabei z. B. um ein Szenario handeln, bei dem eine Punktwolke und ein 3D-Gebäudemodell interagieren. Das Abfrageobjekt könnte einen Wandkörper im Inneren eines Gebäudes darstellen und es sollen alle 3D-Punkte aus einem terrestrischen Laserscan selektiert werden, die sich auf diesem Wandkörper befinden, damit sie getrennt vom Rest der Punktwolke weiterverarbeitet und analysiert werden (z. B. um Ausrichtung und Neigung as-built zu ermitteln und den Planungsdaten gegenüberzustellen). Auswahlpunkte können mit PDAL aus der Datenbank in ein Punktwolkenformat exportiert, mit externen Funktionen und direkter Datenbankanbindung weiterverarbeitet oder als PostGIS-Punkte vom Typ POINT Z konvertiert werden. Letzteres ermöglicht auch punktgenaue Abfragen mit beliebigen 3D-Objekten in PostGIS, funktioniert praktisch aber nur für unverhältnismäßig kleine Selektionsmengen von 3D-Punkten, die kaum noch dem Wesen einer 3D-Punktwolke entsprechen.

3.4 Vor- und Nachteile unterschiedlicher Kachel-Methoden

Die Entscheidung für eine Bestimmte Punktwolken-Patch-Größe und Kachel-Methode wird dem Nutzer überlassen. In der PDAL-Dokumentation gibt es nur die Empfehlung, dass der Speicherplatzbedarf eines einzelnen Patches die standardmäßige Seitengröße von 8 KB einer PostgreSQL-Datenbank nicht überschreiten sollte und dass dies für die meisten LiDAR-Punktwolken mit Patchgrößen von 400 bis 600 Punkten noch erreicht wird (PDAL 2018). Der Standardwert bei Anwendung von *filters.splitter()* entspricht jedoch mit 5000 Punkten etwa dem 10-fachen dieser Empfehlung.

Für den Anwender stellt sich unweigerlich die Frage welche Kachel-Methode (regelmäßig oder unregelmäßig? 2D oder 3D?) am geeignetsten für welche Art von Punktwolke, Projekt-/Untersuchungsgebiet und die konkrete Aufgabenstellung ist. Untersuchungen des Speicherplatzbedarfs bei Anwendung von *filters.chipper()* und *filters.splitter()* in verschiedenen Konfigurationen, haben gezeigt, dass die Punkt-Kapazität bei Anwendung von *filters.chipper()*, bzw. die Gitterweite bei *filters.splitter()* möglichst groß sein sollte, wenn es unabhängig von weiteren Interaktionen und Operationen in der Geodatenbank nur um die verlustfreie Minimierung von Punktwolken-Daten geht. Je mehr 3D-Punkte in einem Patch gebündelt werden, desto weniger Patches entstehen insgesamt und desto effektiver wirkt die standardmäßige Kompressionsmethode „dimensional“ innerhalb jedes einzelnen Patches. Bei dieser Kompressionsmethode werden die Punkt-Arrays nicht punkt- sondern dimensionsweise sortiert, sodass für jede Punktdimension/Punktattribut die geeignetste Kompressionstechnik (z. B. Lauflängencodierung) angewandt werden kann (BOUNDLESS 2015).

Auch eine 3D-Kachelung wirkt sich deutlich positiv auf die Größe der Punktwolken-Tabellen in der Datenbank aus. Es ist anzunehmen, dass dies ebenfalls in der Kompression der Punkte innerhalb der Patches begründet liegt. Je ähnlicher die Ausprägungen von Punktmerkmalen wie Koordinaten, RGB- oder Intensitätswerte usw. sind, desto stärker lassen sich diese komprimieren. Räumlich eng benachbarte Punkte haben ähnliche Eigenschaften und Punkte, die innerhalb von „echten“ 3D-Patches gebündelt werden liegen näher beieinander als 3D-Punkte, die gemeinsam in 2D-Patches organisiert werden. Vergleiche von 2D- und 3D-Kachelungen, mit und ohne Kompression haben das bestätigt.

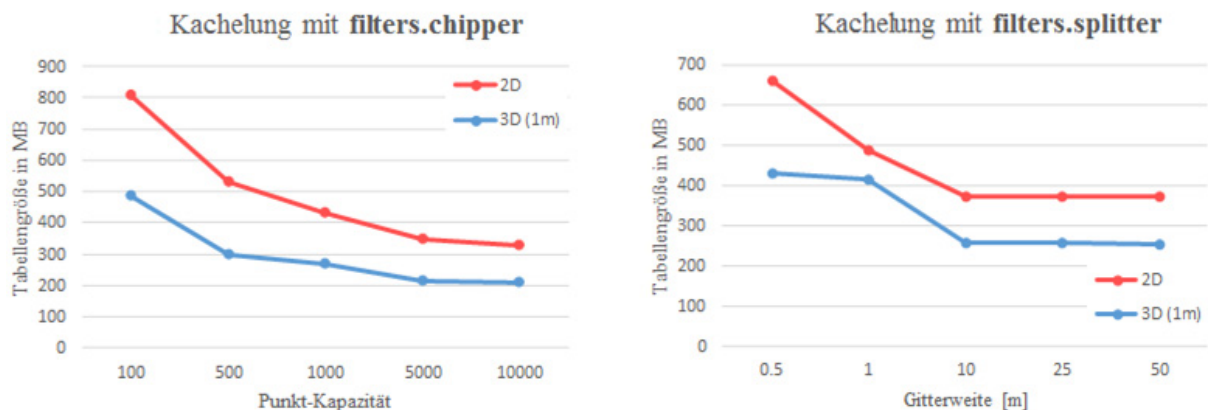


Abb. 7: Auswirkung unterschiedlicher Kachelmethoden in 2D und 3D auf den Speicherplatzbedarf von Pointcloud-Tabellen für eine Beispielpunktwolke (ca. 33 Millionen 3D-Punkte mit RGB-Werten).

Werden Abfrageprozeduren für dieselbe Punktwolke in verschiedenen Kachelungen durchlaufen lassen sich auch Unterschiede hinsichtlich Ausführungsdauer und Ergebnismenge erkennen. Dabei ist zwischen Abfragen auf Patch-Level und solchen auf Einzelpunkt-Level zu unterscheiden. Vorselektionen von ganzen Patches auf Basis von Index-Operatoren und approximierten Geometrien sind als erste Näherungen zu betrachten (Stufe 1). In einer nächsten Verfeinerungsstufe wird die Kandidatenmenge von Patches dann mit der originalen Geometrie eines Abfrageobjekts betrachtet (Stufe 2) und erst im letzten Schritt mit Pointcloud-Funktionen auf Punkt-Level-Genauigkeit ausgewählt (Stufe 3) (vgl. Workflow aus Kap. 4.3). Bei jeder Kachelmethode ist die Auswahlmenge nach Stufe 1 am größten und wird dann in Stufe 2 bis zur finalen Ergebnismenge nach dem Durchlauf von Stufe 3 weiter reduziert. Die gleiche Abfrageprozedur liefert für jede Kachelmethode letztendlich dieselbe Punktmenge, dennoch sind deutliche Unterschiede in den Zwischenergebnissen nach einzelnen Filterstufen erkennbar, die sich auf die Performance von Anwendungen in der Praxis auswirken können. Je kleiner die Anzahl von Punkten in einem Patch, desto feiner die Kachelung und desto präziser sind die Ergebnismengen aber desto größer ist die Gesamtanzahl von Patches und der Aufwand für einzelne Operationen in der Datenbank.

Die Größe von Unterschieden beim Vergleich von verschiedenen 2D- und 3D-Kachelmethoden ist abhängig von der individuellen Punktwolken-Geometrie (vor allem vom Verhältnis von Ausdehnung in der X/Y-Ebene und zur Ausdehnung entlang der Z-Achse) und von der, des Abfrage-Objekts. Die Diagramme in Abbildung 8 zeigen die deutlichen Unterschiede in den Ergebnismengen der Filterstufen 1 und 2 für eine Auswahl von 3D-Punkten innerhalb eines Quaders im 3D-Raum aus einer Punktwolke mit ca. 3 Millionen Punkten. Diese Beispiel-Punktwolke erstreckt sich über mehrere Stockwerke im Inneren eines Gebäudes (siehe auch Abb. 1 u. 2) und hat daher einen sehr ausgeprägten 3D-Charakter. Die Ergebnisse von 2D- und 3D-Kachelmethoden, jeweils mit unterschiedlich großen Patches zeigen, dass sich eine 3D-Kachelung für möglichst präzise Ergebnismengen von Anfragen auf Patch-Level grundsätzlich besser geeignet ist als die standardmäßige 2D-Variante.

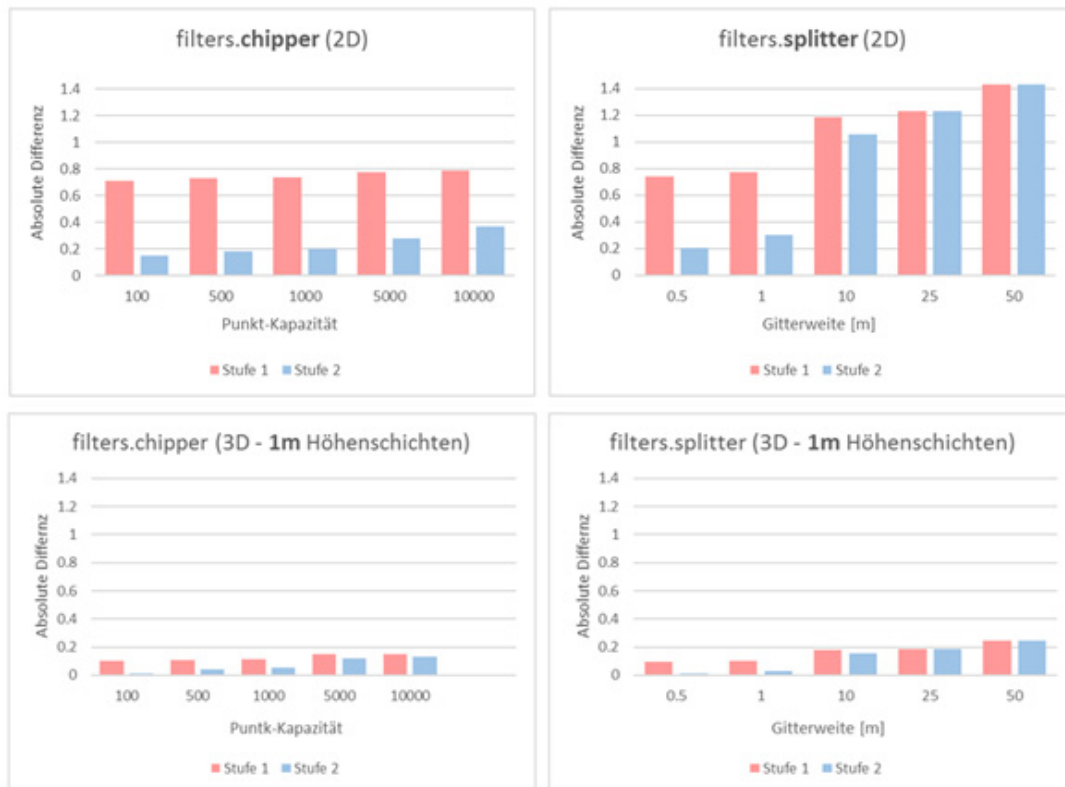


Abb. 8: Auswirkung unterschiedlicher Kachelmethoden in 2D und 3D auf die Ergebnismengen von Punktanfragen der Stufen 1 und 2 für eine Beispelpunktwolke mit ca. 3 Millionen Punkten.

4 Alternative Lösungen zur Integration von 3D-Punktwolken in GIS

Die Aufbereitung und Weiterverarbeitung von originalen 3D-Punktwolken ist keine klassische GIS-Aufgabe, sondern wird zu großen Teilen in spezieller Software für Punktwolken, Laserscanning, digitaler Photogrammetrie und/ oder 3D-Modellierung umgesetzt. *CloudCompare* (CLOUDCOMPARE, 2018) ist beispielsweise eine sehr vielseitige open-source Punktwolken-Software, die jedoch keine Programmier- oder Datenbankschnittstelle bereitstellt. Für eine hersteller- und programmunabhängige Aufbereitung, Verarbeitung und Analyse von Punktwolken existieren eine Reihe von freien Bibliotheken, die u. a. dafür genutzt werden können, den Funktionsumfang von offenen datenbankbasierten Punktwolken-Managementsystemen zu erweitern.

Im Zusammenhang mit PostGIS/Pointcloud ist an vorderster Stelle die PDAL-Bibliothek zu nennen. Sie realisiert den praktischen Datenbankimport von Punktwolken durch Verarbeitungsanweisungen (sog. *PDAL-Pipelines*), die in JSON-Syntax geschrieben und über die Kommandozeile ausgeführt werden. PDAL ermöglicht es im Zuge einer Pipeline-Ausführung zusätzliche sog. *Filter*-Funktionen auf die Daten anzuwenden. Dazu gehören z. B. Berechnungen aus Punktnachbarschaften wie Normalenvektoren, Eigenwerte oder Voxel-Schwerpunkte. Andere Funktionen dienen dazu Punktwolken mit georeferenzierten Bildern zu colorieren, Dreiecksnetze

zu erstellen oder Punktwolken in andere Bezugssysteme zu transformieren. Es existiert auch eine Verbindung zwischen PDAL und der *Point Cloud Library* (PCL) (PCL, 2018), deren Fokus noch mehr auf der Entwicklung von neuen Algorithmen zur Verarbeitung von Punktwolken liegt. PDAL unterstützt außerdem die Python-Programmiersprache, denn eigene Python-Funktionen sind auf einfache Weise in PDAL-Pipelines einzubinden und mit der *PDAL*-Erweiterung für Python ist es wiederum auch möglich PDAL-Funktionen in Python-Anwendungen auszuführen. Eine weitere Programmbibliothek speziell für 3D-Daten und Punktwolken ist *Open3D* (ZHOU et al. 2018), die neben grundlegenden Verarbeitungsalgorithmen auch Möglichkeiten zur Visualisierung von Punktwolken inklusive Animationen umfasst. *Open3D*-Funktionen sind ebenfalls über eine Python-Erweiterung nutzbar und damit auch für die Verarbeitung von Punktwolken in der PostGIS-Datenbank. Die *LAStools* (RAPIDLASSO 2018) (nur teilweise frei) und *LibLAS*-Bibliothek (LIBLAS 2018) sind speziell für den Umgang mit LiDAR-Punktwolken im LAS-Format entwickelt. Programmbibliotheken können 3D-Punktwolken um zusätzliche Punkt-Attribute wie z. B. Klassifizierungscodes erweitern und damit den Informationsgehalt innerhalb einer Datenbank erhöhen.

Auch im Bereich proprietärer und kommerzieller Software gibt es Lösungen zur Integration von 3D-Punktwolken in Geoinformationssysteme und Geodatenbanken. In *ESRI's ArcGIS*-Produkten können Punktwolken im LAS-Format mit Hilfe von *LAS-Datasets* verwaltet, analysiert und angezeigt werden (ESRI 2016). In einem solchen Dataset sind Verweise auf mehrere, miteinander in Beziehung stehende LAS-Punktwolken in einem beliebigen Dateiverzeichnis gespeichert. Dadurch ist ein zentraler Zugriff auf Punktwolken, die verteilt in einzelnen Dateien vorliegen möglich. In Kombination mit Werkzeugen und Funktionspaletten zur Punktwolkengenerierung aus Luftbilddaten (den sog. *Ortho Mapping Tools*), deckt ArcGIS als ein einzelnes System bereits große Teile des Gesamtworkflows zur Erzeugung, Verarbeitung und GIS-basierten Analyse von z. B. UAV-Produkten ab. Die *LAStools*-Software-Suite von *Rapidlasso* umfasst eine große Auswahl von Punktwolken-Kommandozeilen-Tools, die auch als Plugin den Funktionsumfang von ArcGIS, *QGIS* und *Erdas Imagine* erweitern können und ebenfalls auf einzelne LAS-Dateien referenzieren (RAPIDLASSO 2018).

Ein ähnliches Produkt ist *OPALS* (Orientation and Processing of Airborne Laser Scanning data), entwickelt von der TU Wien (OPALS 2017a). *OPALS* unterstützt verschiedenste Raster- und Vektorformate und fokussiert ebenfalls LiDAR-Punktwolken, auch wenn es nicht auf die Verarbeitung von solchen beschränkt ist. Die Software deckt mit ihren verschiedenen Modulen und Funktionen u. a. zur Georeferenzierung, Qualitätskontrolle, Klassifikation, Filterung und Geländemodellierung weitgehend die gesamte Punktwolken-Verarbeitungskette, dank des integrierten Datenmanagers auch für sehr große Datenmengen, ab. Für die direkte Anbindung an GIS-Software ist *OPALS* über Python nutzbar und für *QGIS* existiert die spezielle Erweiterung *QPALS* (OPALS 2017b). Schließlich unterstützt auch die kommerzielle Geodatenbank *Oracle Spatial* mit dem Paket *SDO_PC_PKG* ebenfalls die Speicherung und Verarbeitung von n-dimensionalen Punktwolken. Wie bei Pointcloud für PostGIS werden Punktwolken in Blöcke benachbarter Punkte mit einer vorgegebenen Punkt-Kapazität gekachelt. Im Gegensatz zur PostGIS-Datenbank sind mit der Funktion *PC_Clip()* Selektionen von Teilmengen aus einer Punktwolke sowohl mit 2D- (Polygonen) als auch mit 3D-Geometrien (Quadern) möglich (ORACLE 2017).

5 Fazit & Ausblick

Punktwolken können als Objekte in Geodatenbanken integriert werden und es gibt viele Beispiele für den Nutzen von originalen 3D-Punktwolken in GIS. Auf Basis der räumlichen Nähe von Punktwolken-Punkten zu anderen Geoobjekten können durch räumliche Verbindung Informationen aus einer 3D-Punktwolke auf andere Geoobjekte übertragen werden und umgekehrt. Auch alltägliche Ingenieurprojekte, die nicht zwangsläufig klassischen GIS-Anwendungen zuzuordnen sind können durch das Management von Punktwolken in räumlichen Datenbanken optimiert werden. Projektablaufe profitieren von der kombinierten Speicherung und Analyse von großen 3D-Punktwolken mit anderen Projektdaten (ob administrativ oder fachlich mit und ohne Raumbezug) sowie den Vorzügen von zentraler Datenhaltung und Mehrbenutzerbetrieb.

Die PostGIS-Datenbank ermöglicht mit der Pointcloud-Erweiterung das Anlegen von Punktwolkentabellen. Bei der praktischen Arbeit im Hinblick auf 3D-Anwendungen hat PostGIS/Pointcloud jedoch die Einschränkung, dass Operationen mit Pointcloud-Objekten nur mit klassisch zweidimensionalen PostGIS-Features möglich sind. Außerdem ist für den Anwender die Frage nach einer geeigneten Punktwolken-Kachelung ein noch wenig beleuchtetes Thema. Es gibt gewisse Standardwerte die aber nicht zwangsläufig die beste Wahl für jede Art von Punktwolke sind. Die Punktwolken-Erweiterung Pointcloud wurde vor dem Hintergrund von LiDAR-Punktwolken aus Airborne Laserscanning für große Gebiete im kleinen Maßstab entwickelt und für solche Anwendungen sind grobe Kachelungen und Operationen in 2D ausreichend.

In diesem Beitrag wurde gezeigt, wie eine erweiterte Punktwolkenskachelung unter Berücksichtigung der Punktwolkenausdehnung entlang der Z-Achse realisiert werden kann, indem Funktionen der PDAL-Bibliothek kombiniert zum Einsatz kommen. Es lässt sich feststellen, dass eine solche 3D-Kachelung tendenziell positive Auswirkungen auf den Speicherplatzbedarf von Punktwolkentabellen in PostGIS hat und dass diese Art der Punktwolkenorganisation wesentlich präzisere Punktmengen in frühen Filterschritten bei mehrstufigen Abfrageprozeduren für 3D-Anwendungen liefert.

Im Kontext von Themen wie „Indoor-GIS“ oder die Integration von *Building Information Modeling* (BIM) und GIS, kommt dem Thema „3D“ in GIS eine ganz neue Bedeutung und auch Relevanz zu. Punktwolken haben individuelle Eigenschaften und werden für unterschiedliche Zwecke erfasst. Diese Umstände müssen bei der Wahl einer geeigneten Punktwolken-Kachelung und den in einem System zur Verfügung stehenden Operationen berücksichtigt werden.

Es gibt bereits umfangreiche Vergleiche von Datenbank-Lösungen, bei denen die Verwaltung von unterschiedlich großen LiDAR-Punktwolken in verschiedenen Systemen (PostGIS, Oracle Spatial u. MonetDB) getestet und einander gegenübergestellt wurden (VAN OOSTEROM et al. 2015). Noch eingehendere Untersuchungen der Auswirkungen von verschiedenen Punktwolken-Arten in verschiedenen Größen mit unterschiedlichen Kachelungen vor allem für „3D-relevante“ Projektszenen im großen Maßstab (z. B. 3D-Punktwolken in einem „Campus-GIS“) wären aufschlussreich.

6 Literaturverzeichnis

- ANDRAE, C., 2009: OpenGIS essentials: Die Geo-Standards von OGC und ISO im Überblick: Spatial Schema. Wichmann.
- ASPRS, 2013: LAS Specification Version 1.4 – R13: The American Society for Photogrammetry & Remote Sensing.
- BRINKMANN, T., 2013: Geodatenbanksysteme in Theorie und Praxis. Wichmann.
- BURGER, S., ELFLEIN, A. & VÖLTER, U., 2015: Photo-based-scanning als Erweiterung des ingenieurgeodätischen Leistungsspektrums. ZfV - Zeitschrift für Geodäsie, Geoinformation und Landmanagement, 140(4), 197-205.
- BOUNDLESS, 2015: Homepage Open Geo Suite 4.6:
<https://connect.boundlessgeo.com/docs/suite/4.6/index.html>
- CLOUDCOMPARE, 2018: Projekthomepage: <https://www.danielgm.net/cc/>
- ESRI, 2016: Dokumentation ArcGIS Desktop: <http://desktop.arcgis.com/de/arcmap/10.3/manage-data/las-dataset/what-is-a-las-dataset.htm>
- HAALA, N. & BRENNER, C., 1998: Interpretation of Urban Surface Models Using 2D Building Information. Computer Vision and Image Understanding, 72, 204-214.
- HÖFLE, B., RUTZINGER, M., GEIST, T. & STÖTTER, J., 2006: Using airborne laser scanning data in urban data management - set up of a flexible information system with open source components. Proceedings UDMS 2006: Urban Data Management Symposium. Aalborg, Denmark, 7.11-7.23 (on CD-ROM).
- HÖFLE, B. & HOLLAUS, M., 2010: Urban Vegetation Detection using high Density Full-Waveform Airborne LIDAR Data – Combination of Object-based Image and Point Cloud Analysis. ISPRS TC VII Symposium, 2010 .
- HÖFLE, B. & JOCHEM, A., 2012: 3D-Laserscanning-Punktwolken und GIS – Aktuelle Entwicklungen. gis.SCIENCE, 2, 91-100.
- HUBER, D., 2011: The ASTM E57 File Format for 3D Imaging Data Exchange: Proceedings of the SPIE Vol. 7864A, Electronics Imaging Science and Technology Conference (IS&T), 3D Imaging Metrology, Vol. 7864A.
- JOCHEM, A., HÖFLE, B. & RUTZINGER, M., 2011: Extraction of Vertical Walls from Mobile Laser Scanning Data for Solar Potential Assessment. Remote Sensing, 3, 650-667.
- LIBLAS, 2018: Projekthomepage: <https://liblas.org/>
- MUKUPA, W., ROBERST, G. W., HANCOCK, C. M., AL-MANASIR, K., 2016: A Review of the Use of Terrestrial Laser Scanning Application for Change Detection and Deformation Monitoring of Structures. Survey Review.
- MERWADE, V., COOK, A. & COONROD, J., 2008: GIS techniques for creating river terrain models for hydrodynamic modeling and flood inundation mapping. Environmental Modelling & Software, 23, 1300-1311.
- OGC, 2018: Homepage der OGC Arbeitsgruppe "Point Cloud":
<http://www.opengeospatial.org/projects/groups/pointcloudwg>
- OPALS, 2017a: Homepage von OPALS: <https://geo.tuwien.ac.at/opals/html/index.html>
- OPALS, 2017b: Homepage von OPALS und QPALS:
https://geo.tuwien.ac.at/opals/html/usr_qpals.html

- ORACLE, 2017: Homepage von Oracle:
https://docs.oracle.com/database/121/SPATL/sdo_pc_pkg-clip_pc.htm#SPATL1190
- OTEPKA, J., GHUFFAR, S., WALDHAUSER, C., HOCHREITER, R. & PFEIFER, N., 2013: Georeferenced Point Clouds: A Survey of Features and Point Cloud Management. *ISPRS International Journal of Geo-Information*, **2**, 1038-1065.
- PCL, 2018: Projekthomepage: <http://pointclouds.org/>
- PDAL, 2018: Projekthomepage: <https://pdal.io/index.html>
- RAPIDLASSO, 2018: Homepage von Rapidlasso GmbH: <https://rapidlasso.com/lastools/>
- RUSU, R. & COUSINS, S., 2011: 3D is here: Point Cloud Library (PCL). *IEEE International Conference on Robotics and Automation*, 1-4.
- TANG, P., HUBER, D., AKINCI, B., LIPMAN, R. & LYTLE, A., 2010: Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques: *Automation in Construction*, **19**, 829-843.
- VAN OOSTEROM, P., MARTINEZ-RUBI, O., IVANOVA, M., HORHAMMER, M., GERINGER, D., RAVADA, S., TIJSSEN, T., KODDE, M., GONCALVES, R., 2015. Massive point cloud data management: Design, implementation and execution of a point cloud benchmark: *Computers & Graphics*, **49**, 92-125.
- VOEGTLE, T., STEINLE, E. & TOVARI, D., 2005: Airborne Laserscanning Data for Determination of suitable areas for Photovoltaics. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, **36**(3/W19), 215-220.
- ZHOU, Q., PARK, J. & KOLTUN, V., 2018: Open3D: A Modern Library for 3D Data Processing. *arXiv: 1801.09847*.