

Hochschule für Technik und Wirtschaft Dresden
Fakultät Geoinformation
Masterstudiengang Geoinformatik / Management

Masterarbeit

Untersuchungen zur automatisierten Ableitung von Grünflächen im urbanen Raum basierend auf offenen Daten

Eingereicht von
Theodor Rieche
Seminargruppe: 18/067/71
Matrikelnummer: 46576

1. Gutachter: Prof. Dr.-Ing. Martina Müller
2. Gutachter: Dr.-Ing. Robert Hecht

Eingereicht am: 15.09.2021

Inhaltsverzeichnis

| | |
|--|-----------|
| Inhaltsverzeichnis..... | II |
| Abkürzungsverzeichnis..... | IV |
| Vorwort..... | V |
| 1 Einführung..... | 1 |
| 1.1 Bedeutung von Stadtgrün | 1 |
| 1.2 Projekt meinGrün | 2 |
| 1.3 Generierung der Polygonbasis..... | 3 |
| 1.3.1 Aktueller Stand | 3 |
| 1.3.2 Anforderungen | 4 |
| 1.4 Stand der Wissenschaft | 5 |
| 1.5 Forschungsfragen..... | 6 |
| 2 Untersuchungsgebiet und verfügbare Datenquellen | 7 |
| 2.1 Untersuchungsgebiet | 7 |
| 2.2 OpenStreetMap | 8 |
| 2.3 ALKIS Bestandsdaten ohne Eigentümerdaten | 10 |
| 2.4 Stadt Dresden – kommunale Flächen | 11 |
| 2.5 Weitere Datenquellen | 11 |
| 3 Grundlagen zur automatisierten Ableitung von Grünflächen..... | 13 |
| 3.1 Barrieren im Kontext von Grünflächen | 13 |
| 3.2 Arbeiten mit PostGIS | 16 |
| 3.3 Maschinelles Lernen | 20 |
| 4 Methodik..... | 22 |
| 4.1 Analyse der Datenquellen | 22 |
| 4.2 Ableitungen von Barrieren aus OpenStreetMap Daten..... | 22 |
| 4.3 Ableitung eines OpenStreetMap Landnutzungsdatensatzes | 23 |
| 4.4 Ground-Truth | 24 |
| 4.5 Ermittlung der Barrierenwahrscheinlichkeit | 26 |
| 4.6 Prozess der Vermaschung..... | 26 |
| 4.7 Modelle für Zugänglichkeit und Grünheit..... | 27 |
| 5 Umsetzung für das Projekt meinGrün | 29 |
| 5.1 Technologie / Arbeitsumgebung..... | 29 |
| 5.2 Datenhaltung und Datenimport | 31 |
| 5.3 Ableitungen von Barrieren aus OpenStreetMap Daten..... | 33 |

| | | |
|--|---|--------------|
| 5.4 | Ableitung eines OpenStreetMap Landnutzungsdatensatzes | 36 |
| 5.5 | Durchführung eines Ground-Truth | 44 |
| 5.6 | Ermittlung der Barrierenwahrscheinlichkeit | 49 |
| 5.7 | Prozess der Vermaschung..... | 50 |
| 5.8 | Modelle für Zugänglichkeit und Grünheit..... | 53 |
| 6 | Evaluation der Ergebnisse | 55 |
| 6.1 | Explorative Datenanalyse OpenStreetMap | 55 |
| 6.2 | Landnutzungsdatensatz auf OpenStreetMap Basis..... | 57 |
| 6.3 | Ground-Truth | 64 |
| 6.4 | Barrieren-Wahrscheinlichkeit | 67 |
| 6.5 | Prozess der Vermaschung..... | 70 |
| 6.6 | Modelle für Zugänglichkeit und Grünheit..... | 75 |
| 6.7 | Laufzeit-Analyse | 77 |
| 6.8 | Übertragbarkeit..... | 78 |
| 7 | Zusammenfassung und Ausblick | 79 |
| Glossar | | VI |
| Quellenverzeichnis | | VII |
| Abbildungsverzeichnis..... | | X |
| Quelltextverzeichnis | | XI |
| Formelverzeichnis | | XI |
| Tabellenverzeichnis..... | | XI |
| Anlagenverzeichnis | | XIII |
| Erklärung über die eigenständige Erstellung der Arbeit..... | | XXXVI |

Abkürzungsverzeichnis

| | |
|----------|---|
| ALKIS | Amtliches Liegenschaftskatasterinformationssystem |
| AOI | area of interest, Untersuchungsgebiet |
| CPU | central processing unit |
| CRS | coordinate reference system |
| DB | Datenbank |
| EPSG | European Petroleum Survey Group Geodesy |
| ESRI | Environmental Systems Research Institute |
| FME | Feature Manipulation Engine, © 2021 Safe Software Inc |
| GeoJSON | Geographic JavaScript Object Notation, Dateiformat |
| GIS | Geoinformationssystem |
| HTW | Hochschule für Technik und Wirtschaft Dresden |
| IÖR | Leibniz-Institut für ökologische Raumentwicklung Dresden e.V. |
| ISO | International Organization for Standardization |
| LC | land cover, Landbedeckung |
| LU | land use, Landnutzung |
| LTS | long-term support |
| OGC | Open Geospatial Consortium |
| ÖSD | Ökosystemdienstleistung |
| OSM | OpenStreetMap |
| PL/pgSQL | Procedural Language/PostgreSQL Structured Query Language |
| POI | Point of Interest |
| QGIS | Quantum GIS |
| RAM | Random-Access Memory, Arbeitsspeicher |
| SQL | Structured Query Language |
| UUID | universally unique identifier |
| VGI | volunteered geographic information |
| WFS | Web Feature Service |
| WGS84 | World Geodetic System 1984 |

Vorwort

Aufgrund meiner Tätigkeit als wissenschaftliche Hilfskraft am Leibniz-Institut für ökologische Raumentwicklung e.V. in Dresden kam ich verstärkt mit Fragestellungen zu Ökologie und Stadtgrün und deren Untersuchung mit Methoden der Geoinformatik in Berührung. Durch meine Mitarbeit im Projekt meinGrün konnte ich vielfältige Erfahrungen sammeln. Ein besonderes Interesse gilt dabei den Möglichkeiten der Analyse- und Prozessierung von Geodaten, die Möglichkeiten der Nutzung von OpenStreetMap als Datenquelle sowie Themen mit Bezug zu Data Science und maschinellem Lernen.

In einen größeren Zusammenhang gesetzt tangiert die vorgelegte Masterarbeit wichtige Fragen unserer heutigen Zeit. Wie kann auch in Zukunft noch ein hohes Maß an Lebensqualität für die Stadtbevölkerung erreicht werden? Welche Grünflächen stehen in Städten zur Verfügung? In diesem Spannungsfeld ist die Thematik dieser Arbeit einzuordnen.

Die vorgelegte Masterarbeit entstand in Zusammenarbeit zwischen der Hochschule für Technik und Wirtschaft Dresden und dem Leibniz-Institut für ökologische Raumentwicklung Dresden e.V. Ein besonderer Dank geht an Prof. Dr.-Ing. Martina Müller sowie Dr.-Ing. Robert Hecht. Beide haben mich in den letzten Jahren unterstützt und auf meinem Weg begleitet. Weiterhin danken möchte ich meiner Familie und meinen Freunden für die Unterstützung und den Rückhalt zur Erstellung dieser Arbeit.

Theodor Rieche

Dresden, den 14. September 2021

1 Einführung

1.1 Bedeutung von Stadtgrün

Die positive Wirkung von Natur und Grünflächen auf den Menschen ist bekannt und wird durch den Ansatz der Ökosystemdienstleistungen (ÖSD) untersucht und nachgewiesen. Herausgearbeitet werden unter anderem ÖSD wie saubere Luft, lokale Klimaregulation, Ruhe und Erholung, welche förderlich für eine gute Lebensqualität der Bevölkerung sind (Bolund/Hunhammar 1999; Perrings et al. 2010; Krellenberg et al. 2021). Im urbanen Raum ist diese positive Wirkung von besonderer Bedeutung, um einen oft erhöhten Geräuschpegel, eine schlechtere Luftqualität sowie einen hohen Grad an versiegelten Flächen kompensieren zu können. Gleichzeitig sind Städte im Vergleich zu ländlichen Räumen an Grünflächen pro Einwohner eher unversorgt (TC 2020; BMUB 2017, S. 7). Menschen ziehen bewusst in die Städte, ein gesellschaftlicher Prozess der Urbanisierung hält weiterhin an (siehe Abbildung 1). Zu den Ursachen dieser Entwicklung gehören vielfältigere Kultur- und Freizeitangebote wie auch Arbeitsplätze und Karrierechancen in urbanen Räumen.

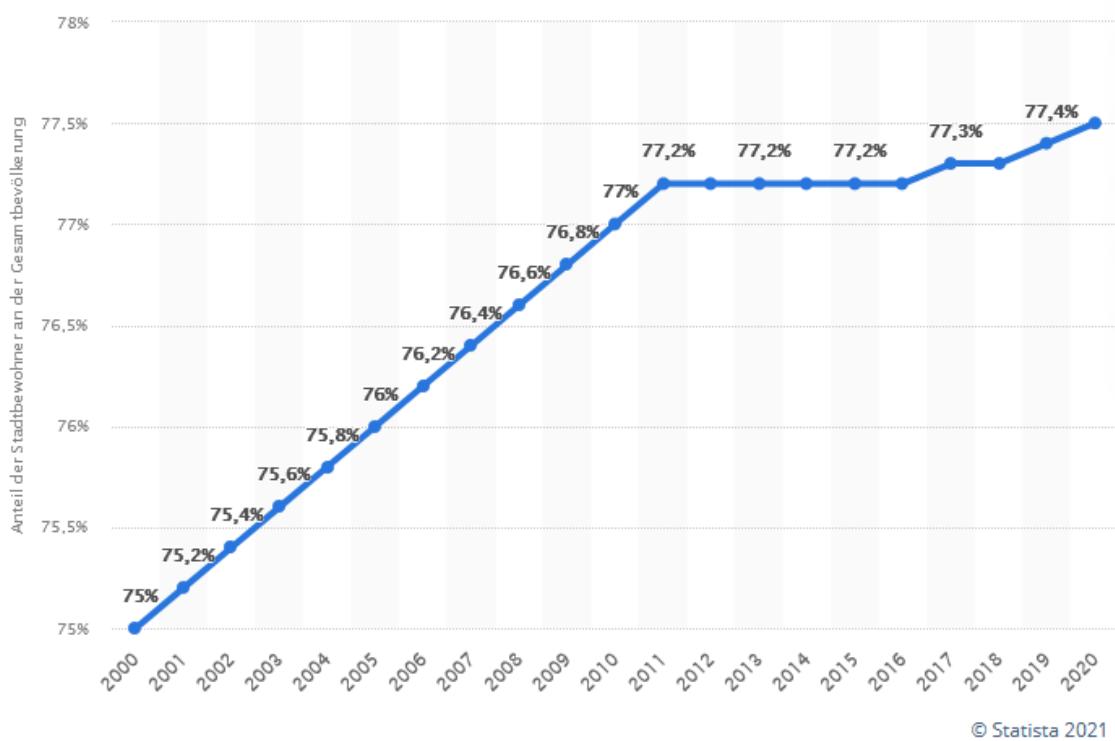


Abbildung 1 Urbanisierungsgrad: Anteil der Stadtbewohner an der Gesamtbevölkerung in Deutschland in den Jahren von 2000 bis 2020 (statista 2021)

Die COVID-19 Pandemie hat die hohe Bedeutung von öffentlich zugänglichen Grünflächen in urbanen Räumen ebenfalls aufgezeigt.

1.2 Projekt meinGrün

Im Kontext von städtischen Grünflächen und deren Nutzung durch die Bevölkerung bewegt sich das Forschungsprojekt mit dem Titel meinGrün. Im Zeitraum von 2018 bis 2021 wurden in einem deutschlandweiten Projektverbund – federführend koordiniert durch das IÖR Dresden – eine Infrastruktur von Diensten sowie eine WebApp entwickelt und erprobt (mg 2021). Die WebApp soll den Nutzerinnen und Nutzern die Möglichkeit bieten, nach geeigneten Grünflächen in der Umgebung zu suchen und eine möglichst grüne oder leise Route vom aktuellen Standort beginnend vorzuschlagen. Zur Recherche geeigneter Grünflächen stehen vielfältige Kriterien bereit. Begonnen werden kann über die Einstiegspunkte „Schnellsuche“, „Aktivitätensuche“ oder „Kriteriensuche“ (siehe Abbildung 2). Damit soll die urbane Bevölkerung die öffentlichen Grünflächen und deren Nutzungsmöglichkeiten bewusster erleben können. Die Funktionen basieren auf offenen amtlichen Daten, frei verfügbaren Fernerkundungsdaten und auf Daten aus den sozialen Netzwerken¹. Über aggregiertes und anonymisiertes Feedback können die kommunalen Aufgabenträgern bei Planung und Verwaltung unterstützt werden. (mg 2021; TC 2018)

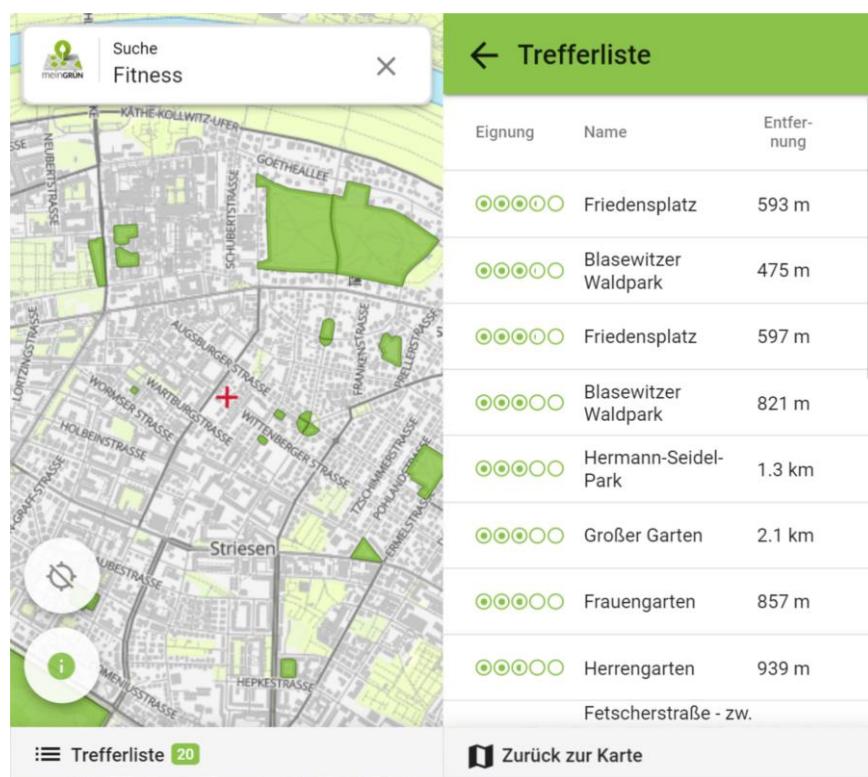


Abbildung 2 Bildschirmabzug der meinGrün-App nach Aktivitätensuche (meinGrün 2021)

Das Projekt stellt die Nutzerperspektive beziehungsweise die Bedürfnisse der Nutzerinnen und Nutzer in den Vordergrund. Mögliche Leitfragen könnten wie folgt formuliert werden:

¹ engl. „social media“

Welche Flächen sind zum Freunde treffen oder zum Fußball spielen geeignet? Gibt es dort Sitzbänke oder öffentliche Toiletten in der Nähe? Wie gelange ich am besten mit dem Fahrrad oder zu Fuß dorthin? Gibt es eine Route, welche besonders viel Grün entlang der Strecke bietet? Zum Einsatz kommen neuartige Bewertungsansätze für die Bildung von Grünflächen, die Beschreibung mithilfe Indikatoren sowie für die Routenberechnung. Durch die wissenschaftliche Untersuchung soll das bekannte Wissen zu Stadtgrün erweitert werden. Die Erfahrungen werden in einem Handbuch publiziert. Als Pilotstädte für die Entwicklung und Erprobung wurden Heidelberg und Dresden ausgewählt. In diesen Untersuchungsgebieten soll die WebApp ihre Funktionalität anbieten können. Das Projekt wird mit der Forschungsinitiative mFUND (Modernitätsfonds) vom Bundesministerium für Verkehr und digitale Infrastruktur (BMVI) gefördert.

1.3 Generierung der Polygonbasis

Zur Realisierung der Anforderungen an die WebApp wird für das Untersuchungsgebiet eine Polygonbasis benötigt, welche Polygone für öffentlich zugänglich Grünflächen bereitstellt. Sie werden durch Barrieren abgegrenzt (siehe 3.1). Diese Geometrien fungieren als Träger von Merkmalen für darauf aufbauende Bewertungsalgorithmen und sollen den Qualitätskriterien Lagegenauigkeit, Grünheit (greeness) und öffentlich Zugänglichkeit (publicness) genügen. Die Einschätzung der Ausstattung dieser Grünfläche bezüglich ausgewählter Kriterien (mit Wegen, Bänken, Grillplätzen etc.) wie auch die Eignung zum Ausüben von Aktivitäten (Lesen, Freunde treffen, Fussball spielen etc.) soll mit diesen Geometrien verknüpft werden. Sie bilden eine Basis für ein Großteil der Funktionalität der WebApp (siehe auch Grünflächen in Abbildung 2).

1.3.1 Aktueller Stand

Der aktuelle Stand der Polygonbasis im Projekt baut auf einer Realisierung mit der proprietären Software FME² auf. Der Prozessablauf findet sich dort graphisch visualisiert in mehreren sogenannten Workbenches gespeichert. Der aktuelle Ansatz nutzt als Datenquellen sowohl amtliche Daten wie auch OSM-Daten. Er beruht nicht vollständig auf offenen Daten. Die Nutzung und Extraktion entsprechender key-value aus OpenStreetMap Daten bringt Potenzial für eine noch exaktere Ableitung und höhere Detailschärfe mit sich. So lässt sich unter anderem die Nutzung von Landnutzungspolygonen und Verwaltungseinheiten optimieren. Letztere sind nicht zwingend zur Bildung der Polygonbasis erforderlich. Erprobt können weiterhin Ansätze zur Vorhersage bezüglich der Grünheit und der (öffentlichen) Zugänglichkeit. Abbildung 3 zeigt die aktuelle Polygonbasis für das Untersuchungsgebiet Dresden, visualisiert nach Grünflächentypen.

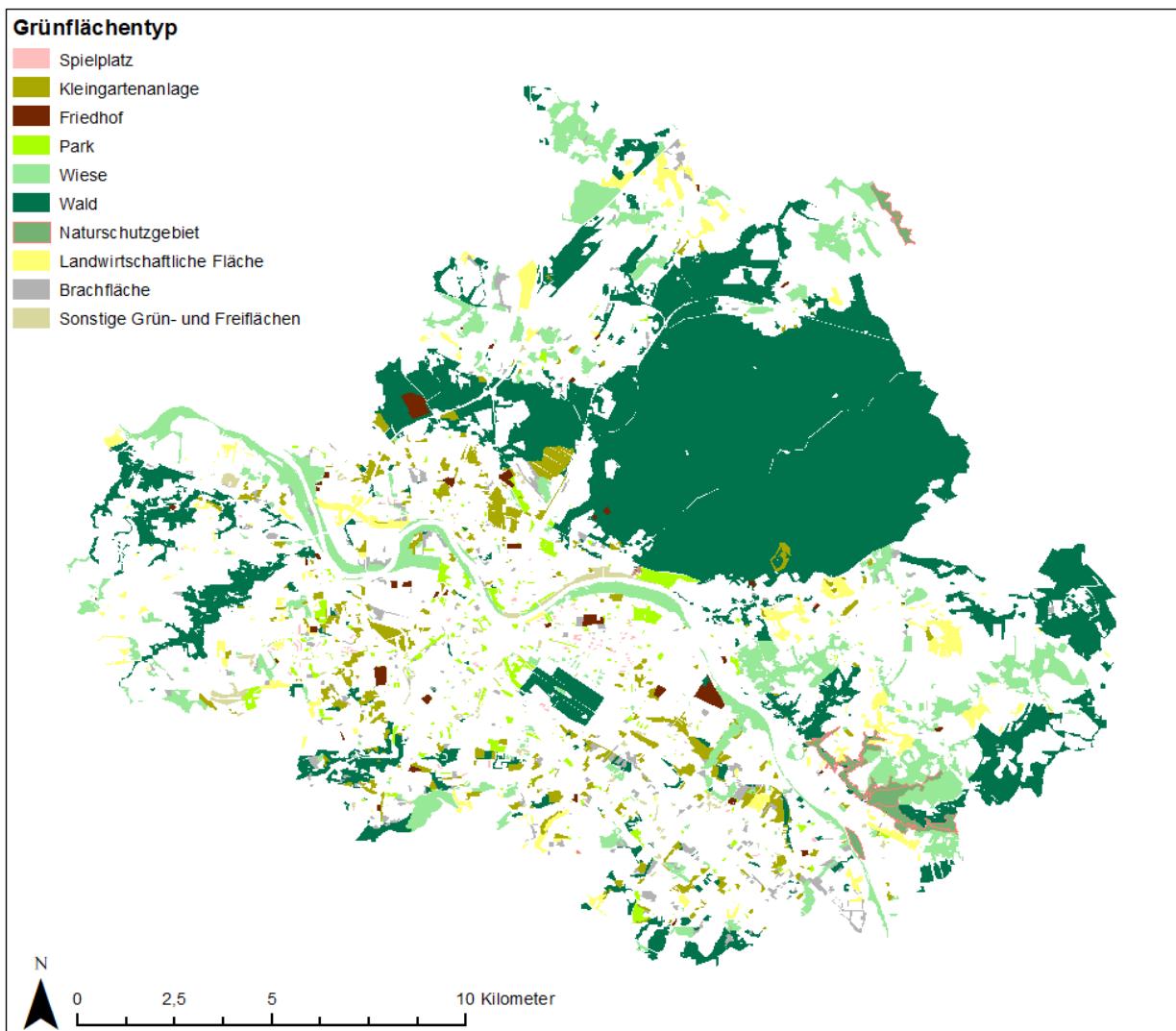


Abbildung 3 Grünflächentypen in Dresden (Handbuch 2021)

1.3.2 Anforderungen

Es sollen Anforderungen an einen zu generierenden Polygondatensatz definiert werden. Beginnend bei der Semantik sollen auch lizenzirechtliche Fragen, Geometrie und Topologie spezifiziert werden. Die Polygonbasis soll nur aus als öffentlich zugängliche Grünflächen erkannten Flächen bestehen³. Alle Polygone sollen über einen universellen und eindeutigen Identifikator verfügen (UUID). Der Datensatz soll aus offenen Datenquellen abgeleitet wer-

² Feature Manipulation Engine, © 2021 Safe Software Inc

³ Sogenannte „green targets“

den, um lizenzrechtliche Aspekte in der Nutzung zu vereinfachen. Die Verarbeitung ist mit Open Source Produkten zu realisieren. Die Polygone sollen in OCG-konformen Geometrien gemäß ISO 19125 / OGC Simple Features Standard⁴ vorliegen. Ein Geometriertyp in der zweiten Dimension (Polygon, Multipolygon) gewährleistet die Repräsentation der flächenhaften Ausprägung der Grünflächen. Der Datensatz soll für das Untersuchungsgebiet Dresden generiert werden und im Koordinatenreferenzsystem WGS84 (EPSG-Code 4326) definiert sein. Er muss überlappungsfrei sein, darf bezogen auf das Untersuchungsgebiet aber Löcher aufweisen.

1.4 Stand der Wissenschaft

Inzwischen existieren viele wissenschaftliche Studien, welche sich mit dem Projekt OpenStreetMap (OSM) auseinandersetzen. Nach (Arsanjani et al. 2015, S. Foreword) können die Studien grob in folgende Rubriken gegliedert werden: erstens in Studien, welche sich mit der Datenextraktion auseinandersetzen beziehungsweise OSM als freie Datenquelle nutzen. Zweitens in Studien zur Geschichte, zur Datenqualität oder zur Entwicklung des OSM-Projektes. Drittens zur Untersuchung von Mustern im OSM-Datenbestand. Viertens zu räumlichen und gesellschaftlichen Aspekten, welche sich in der Art der Kartierung ausdrücken können. Wie fünftens zur sozialen Interaktion innerhalb der OSM-Gemeinschaft.

Basierend auf OSM-Daten wurden Ansätze von DataScience und maschinellem Lernen gezeigt (Janakiev 2018; Thomas 2020; Nguyen 2021). Ludwig et al. 2021 thematisieren die Ableitung von Grünflächen hinsichtlich Grünheit und öffentliche Zugänglichkeit. Die Synthese von OpenStreetMap und Fernerkundungsdaten wird zum Beispiel in Vargas-Munoz et al. 2021 aufgezeigt. Ebenfalls können komplexe Vergleichsindikatoren auf Basis von OpenStreetMap berechnet werden ((Kirschner et al. 2020)). Eine Ableitung eines Landbedeckungsdatensatzes auf Basis von OpenStreetMap und Fernerkundungsdaten konnte in Schultz et al. 2017 vollzogen werden. Mögliche Ansätze zur Ableitung von Straßenblöcken sind in Sk53-osm 2015; lucasmation 2016 aufgezeigt.

OpenStreetMap ist in der Wissenschaft angekommen. Dies drückt sich in der steigenden Anzahl an Studien und durch die Verknüpfung mit anderen Datenquellen aus.

⁴ Siehe <https://www.ogc.org/standards/sfa>

1.5 Forschungsfragen

Folgende Forschungsfragen sollen in dieser Arbeit untersucht werden.

- Wie gut eignen sich OpenStreetMap Daten zur Ableitung von öffentlich zugänglichen Grünflächen im urbanen Raum?
- Welche in OpenStreetMap kartierten Landnutzungsübergänge bzw. key-value weisen wie wahrscheinlich auf eine Barriere hin?
- Wie lässt sich die Qualität der generierten Polygone messen?

Der Verfasser stellt folgende Hypothesen auf:

- OpenStreetMap-Daten sind weltweit verfügbar und bilden damit eine gute Grundlage für einen allgemeinen Algorithmus (Übertragbarkeit).
- OpenStreetMap-Daten sind aufgrund der freiwilligen und Community-getragenen Kartierung heterogen in Bezug auf Qualitätskriterien wie Vollständigkeit, Lagegenauigkeit, Aktualität.
- Trotz dieser Unschärfen im Datensatz lassen sich Erkenntnisse generieren.
- OpenStreetMap Daten können intrinsisch bezüglich innerer Muster in den Daten (Nachbarschaften, Topologie, ...) analysiert werden.
- Für die Validierung müssen weitere Referenzdaten (amtliche, Ground-Truth, Fernerkundungsdaten, ...) hinzugezogen werden.

2 Untersuchungsgebiet und verfügbare Datenquellen

Zu Beginn soll das Untersuchungsgebiet eingeführt werden. Es folgen Beschreibungen der in dieser Arbeit genutzten Datenquellen.

2.1 Untersuchungsgebiet

Das Untersuchungsgebiet ist projektintern definiert als Fläche der jeweiligen Stadt zuzüglich eines gepufferten Umlandsaumes⁵ mit einer Breite von fünf Kilometern. Dieser soll für die am Stadtrand wohnende Bevölkerung eine sinnvolle räumliche Funktionalität der App gewährleisten und verschiebt das Ende der Verfügbarkeit weiter ins Umland. Diese Arbeit konzentriert sich auf das Untersuchungsgebiet der meinGrün-Pilotstadt Dresden.

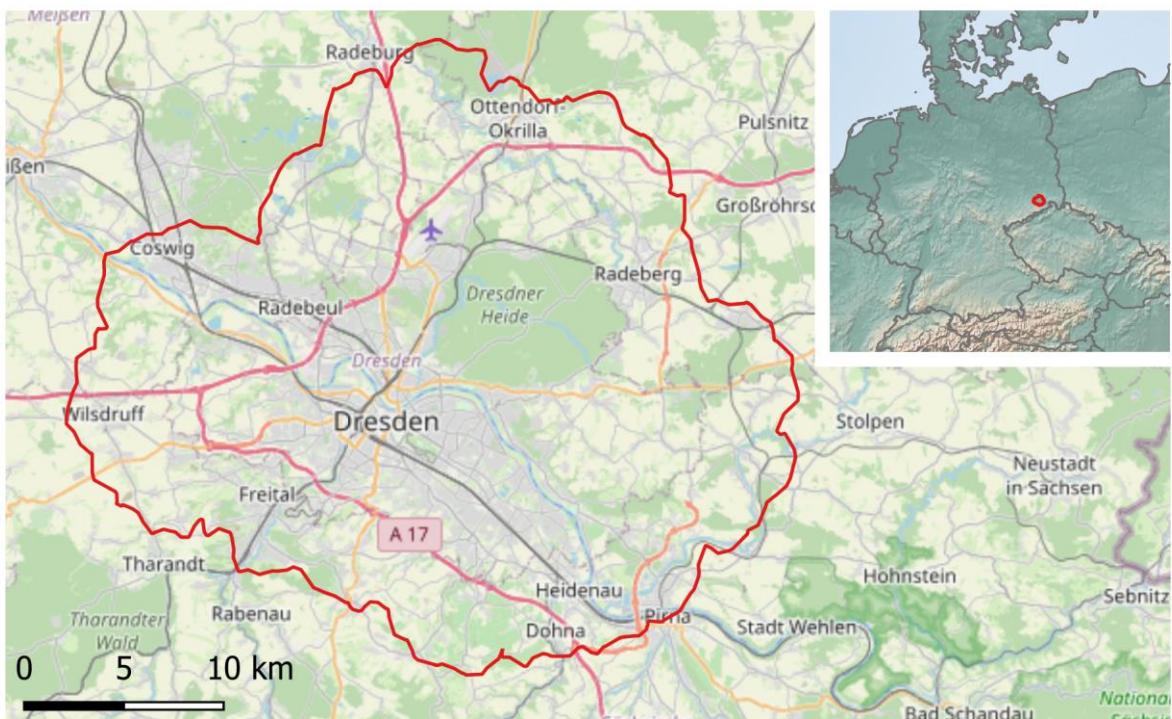


Abbildung 4 Untersuchungsgebiet Dresden mit Umlandsaum (natural earth 2021; mg 2021)

Das Gebiet ist strukturiert durch verdichtete urbane Bebauung und den Flusslauf der Elbe. Im Norden befindet sich ein großes zusammenhängendes Waldgebiet der Dresdner Heide. Das Relief ist in Ausrichtung zum Flusslauf abschüssig. Eingeschlossen sind neben dem Gebiet der Stadt Dresden auch Flächen angrenzender Landkreise. Dieses Untersuchungsgebiet (AOI Dresden⁶) ist die Grundlage für Analysen und Fragestellungen dieser Arbeit.

⁵ engl. „outskirt“

⁶ engl. „area of interest“

2.2 OpenStreetMap

OpenStreetMap ist ein auf freiwillige Partizipation ausgerichtetes Community-Projekt⁷ und hat das Ziel einen globalen Geodatensatz zu allen Objekten der Erdoberfläche und einigen weiteren Themen⁸ zusammenzutragen. Im OSM-Projekt generierte Geodaten lassen sich als VGI bezeichnen. Stehen zur Nutzung neben verschiedenen gerasterten Basiskarten unzählige thematische Karten auf OpenStreetMap-Basis bereit, so ist es eine zentrale Datenbank für die Datenhaltung im Hintergrund, welches den Kern des Projektes ausmacht. Inzwischen existiert eine Vielzahl an Editoren und Tools, mit denen neue Daten importiert oder bereits bestehende Daten analysiert werden können. Wie so oft folgen neue Projekte innerhalb des „OpenStreetMap-Ökosystems“ dem Grundsatz der Open Source Bewegung und stehen für die Gemeinschaft und die Öffentlichkeit zur freien Nutzung bereit. Im Jahr 2004 entstand durch Steve Coast die Initiative, eine eigene freie Datenquelle für Geodaten zu beginnen. Hintergrund waren erschwerte lizenzirechtliche Zugänge zu vorhandenen Daten des britischen Ordnance Survey. Mithilfe von GPS-fähigen Navigationsgeräten einer Vielzahl an Mitwirkenden sollte und wurde über die Zeit ein Datensatz an Geodaten zusammengetragen, der als eigene Datenquelle und Nachweis für die eigenständige Erfassung fungiert. Zentral ist dabei die sogenannte „on-the-ground“-Regel zur Erfassung in-situ. Das Projekt wuchs stetig an (siehe Abbildung 5). (OSM history 2021; Arsanjani et al. 2015, S. Foreword)

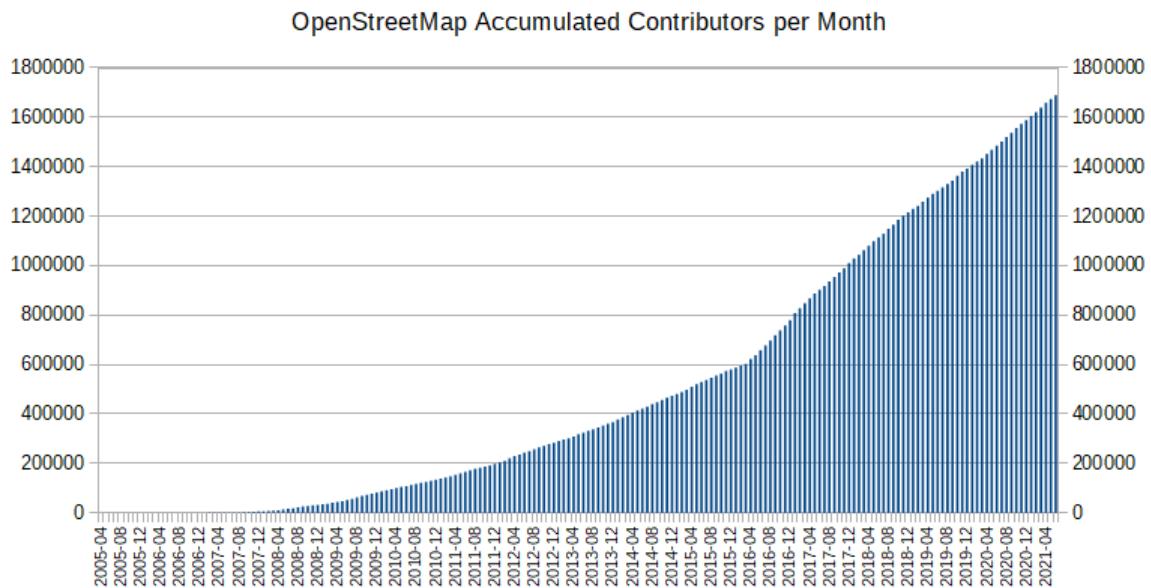


Abbildung 5 Akkumulierte OSM-Nutzerzahlen nach Monat (OSM stats 2021)

⁷ ähnlich der Wikipedia

⁸ Genannt seien exemplarisch Grenzverläufe von administrativen Einheiten oder Schutzgebieten, wie auch unterirdisch verlaufende Kanäle etc.

Inzwischen zählen auch das Interpretieren und Vektorisieren von freigegebenen Fernerkundungsdaten, wie auch die Zusammenarbeit mit amtlichen datenhaltenden Stellen zu den zulässigen Datenquellen. Die OSM-Datenbank steht unter der selbst für diesen Zweck entwickelten Lizenz Open Data Commons Open Database License (ODbL) v1.0⁹.

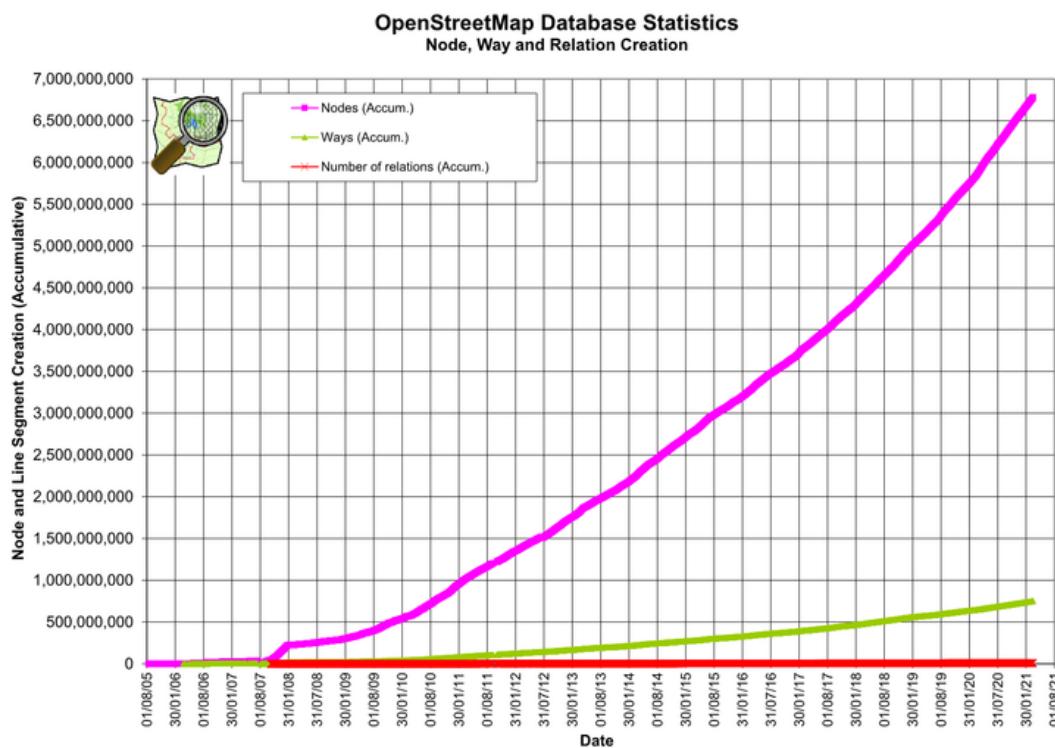


Abbildung 6 Generierung von OSM Geometrien nach Zeit (OSM stats 2021)

Die Geometrie ist durch ein topologisch-strukturiertes Datenbankmodell definiert (siehe Ausschnitt in Anlage 1). Sie wird in die Typen „Nodes“, „Ways“ und „Relations“ gegliedert (siehe Abbildung 6) und ist nicht GIS-konform im Sinne der ISO 19125 / OGC Simple Features Standard (Andrae/Zipf 2013, S. 17). Einzig die „Nodes“ sind mit ihren Attributen „latitude“ und „longitude“ Träger von Koordinaten. Diese liegen im CRS WGS84 (EPSG-Code 4326) vor. „Ways“ sind Linienzüge bestehend aus einer geordneten Folge an „Nodes“ und können geschlossen oder offen sein. Sie referenzieren dabei explizit auf die ID des jeweiligen „node“ und können in einem GIS Linien oder Polygone repräsentieren. „Relations“ stellen einen umspannenden Metatyp zur Gruppierung von beliebig vielen „ways“ und „nodes“ dar, beispielsweise zur Erfassung von Linien des öffentlichen Nahverkehrs oder europäischen Fernwanderwegen. Semantisch werden Objektarten dynamisch und offen über sogenannte „key-value“ ermöglicht. Dabei können beliebig viele „key-value“ simultan gesetzt werden.

⁹ Siehe <https://www.openstreetmap.org/copyright>

2.3 ALKIS Bestandsdaten ohne Eigentümerdaten

Die exakte Bezeichnung des Datensatzes lautet „WFS SN ALKIS Vereinfacht“. Er wird vom sächsischen GeoSN bereitgestellt und lässt sich als Web Feature Service optimal in QGIS analysieren¹⁰. Unterstützt werden die OGC Versionen WFS 1.1.0 und WFS 2.0.0 (GeoSN 2021). Die Geodaten des Amtlichen Liegenschaftskatasterinformationssystem (ALKIS) sind Bestandteil des AAA-Modells der AdV¹¹. Die Geometrien werden im CRS ETRS89/ UTM Zone 33N (EPSG-Code 25833) angeboten. Die Nutzung des Dienstes ist kostenfrei. Voraussetzung ist die Namensnennung mit folgendem Wortlaut: „Staatsbetrieb Geobasisinformation und Vermessung Sachsen (GeoSN)“ oder kurz „GeoSN“ (<https://geoportal.sachsen.de/cps/geosn.html>) (GeoSN 2021). Der Datensatz enthält rund 2,8 Millionen Flurstücksgeometrien von Sachsen. Angereichert sind diese durch verschiedene Attribute (siehe Abbildung 7). 135558

| Objekt | Wert |
|--------------|---|
| Flurstueck | |
| idflurst | DESNALK0q8002KLt |
| (abgeleitet) | |
| (Aktionen) | |
| idflurst | DESNALK0q8002KLt |
| flstkennz | 140209_00259000200 |
| land | Freistaat Sachsen |
| landschl | 14 |
| gemarkung | Altstadt II |
| gemaschl | 140209 |
| flur | NULL |
| flurschl | NULL |
| flstnrzae | 259 |
| flstnrnen | 2 |
| regbezirk | NUTS 2-Region Dresden |
| regbezschl | 146 |
| kreis | Kreisfreie Stadt Dresden |
| kreisschl | 14612 |
| gemeinde | Stadt Dresden |
| gmdschl | 14612000 |
| oid | DESNALK0q8002KLtFL |
| aktualit | 2015-10-22Z |
| flaeche | 11002 |
| abwrecht | Kein abweichender Rechtszustand |
| lagebeetxt | Parkstraße |
| tntxt | Sport-, Freizeit- und Erholungsfläche;11002 |

Abbildung 7 Bildschirmabzug ALKIS WFS – Datenschema / Attribute

Von Relevanz für diese Arbeit ist das Attribut „tntxt“. Es gibt Auskunft über Nutzung und Nutzungsschlüssel. Die Aktualität ist ebenfalls Teil des Datenschemas.

¹⁰ Die deutlich aufwendigere Nutzung des NAS XML-Formates für ALKIS-Daten entfällt dadurch.

¹¹ Arbeitsgemeinschaft der Vermessungsverwaltungen der Länder der Bundesrepublik Deutschland

2.4 Stadt Dresden – kommunale Flächen

Dieser Datensatz ist das Resultat einer Anfrage der Fraktion der Piraten im Stadtrat Dresden. Das Wissen um Flächen in kommunaler Hand wurde als Voraussetzung angesehen, um Nutzungspläne im Stadtrat für diese Flächen einbringen zu können. Dem Antrag wurde stattgegeben. Die Antwort stellt eine Auflistung der Flurstücke in kommunaler Hand dar. Diese wurden anschließend mit den Flurstücksgrenzen verschneitten um einen Geodatensatz zu generieren (siehe Abbildung 8). Der Datensatz enthält einige kleinere Lücken. Insgesamt ist die Vollständigkeit dennoch sehr gut. Das Attribut „eigentum_status“ kann genutzt werden. Die Daten sind von 2020. (Piraten-Dresden 2021) Grün wurden hier kommunale Flächen visualisiert (hellrot sind nicht kommunale Flächen, siehe Abbildung 8).

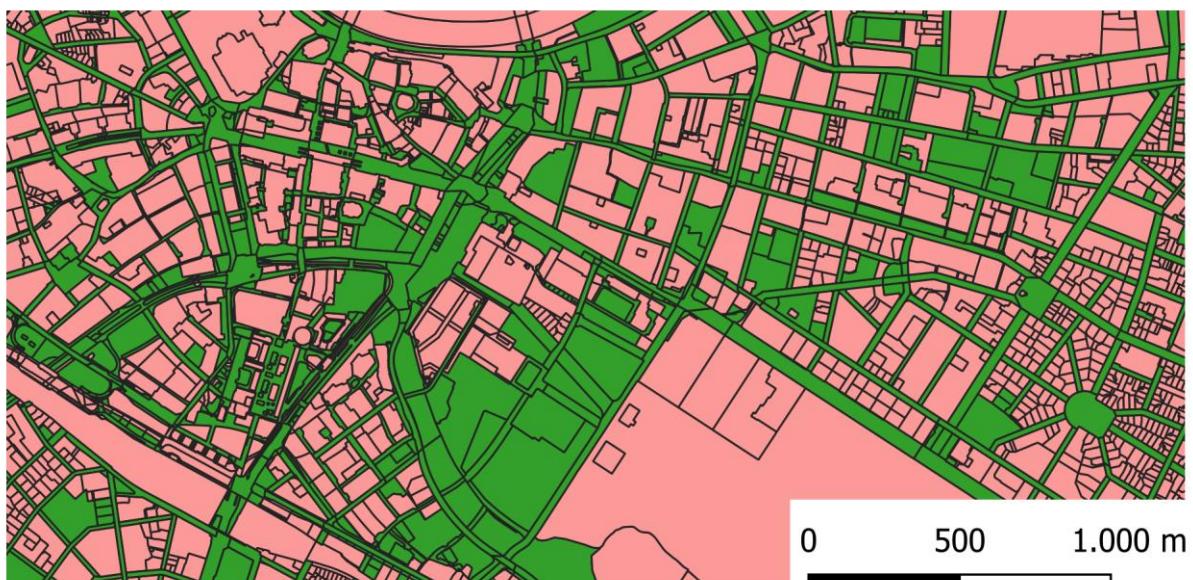


Abbildung 8 Datensatz kommunale Flächen Dresden

2.5 Weitere Datenquellen

Für diese Arbeiten wurden weitere potenzielle Datenquellen analysiert und geprüft. Im Rahmen des Projektes „Essbarer Stadtteil (Dresden-) Plauen (ESP)¹²“ wurde im Frühjahr 2020 für den Stadtteil Dresden-Plauen eine fachkundige Kartierung essbarer Gehölze durchgeführt. Ebenfalls wurden private Flächen erfasst, sowie Potenzialflächen für das Bepflanzen mit essbaren Gehölzen in der Zukunft. Aufgrund von einer nicht gegebenen Vollständigkeit und Qualität bezüglich der privaten Flächen entfiel dieser Datensatz als Referenz für den Aspekt Zugänglichkeit. Die „Staatliche Schlösser, Burgen und Gärten Sachsen gemeinnützige GmbH“ (SBG) verwaltet unter anderem Park- und Gartenanlagen, wie den Großen Gar-

¹² Siehe <https://essbarer-stadtteil.de/>

ten in Dresden. Eine Anfrage bezüglich Geodaten der verwalteten Flächen wurde gestellt. Eine Herausgabe der Nummern betroffener Flurstücke erschien möglich, unterliegt aber noch einer Prüfung hinsichtlich der Rechte der tatsächlichen Eigentümer der Flächen. Eigene Geometrien werden laut Auskunft aktuell nicht geführt.

3 Grundlagen zur automatisierten Ableitung von Grünflächen

Einige für diese Arbeit notwendigen Grundlagen sollen in diesem Kapitel gelegt werden.

3.1 Barrieren im Kontext von Grünflächen

Für den Kontext dieser Arbeit stellt die Auseinandersetzung mit Barrieren in Bezug zu Grünflächen eine herausragende Bedeutung dar. Zunächst soll die Verwendung des Begriffes „Barriere“ in unterschiedlichen wissenschaftlichen Disziplinen vorgestellt werden. Es folgen Konzepte aus der Nutzerperspektive zur Nutzung von Grünflächen.

Ein Blick in verschiedene Wissenschaftsgebiete führt zunächst zur Architektur und Landschaftsarchitektur. Barrieren gilt es im Sinne einer angestrebten „Barrierefreiheit“ planerisch zu vermeiden. So soll möglichst allen Menschen ein gleicher Zugang gegeben werden können. Neben baulichen Aspekten wird dieser Begriff analog beim Zugang zu öffentlichen Dokumenten oder Online-Auftritten auch auf andere Aspekte der Sinneswahrnehmung (wie die Wahl verschiedener Schriftgrößen oder das Existieren einer Audio-Version) übernommen. Im übertragenen Sinne wird von „Sprachbarrieren“ gesprochen. In der Ökologie führt der Begriff „Barriere“ hin zur geographischen und zur ökologischen Isolation. Während bei geographischer Isolation Barrieren in der Realwelt unter anderem durch Ozeane oder Gebirge auftreten können (durch einen Ozean voneinander isolierte Inseln), ist bei ökologischer Isolation die Koexistenz verschiedener Spezies in ihren jeweils besetzten ökologischen Nischen innerhalb eines Ökosystems beschrieben. Während Art A beispielsweise Schichten von Baumwurzeln bewohnt, hat sich eine Art B die Baumkronen als ökologische Nische gesucht. Barrieren stellen in diesem Zusammenhang unterschiedlich ausgeprägte biotische und abiotische¹³ Umweltfaktoren dar (Patten/Auble 1981). Der Begriff einer Barriere ist aus geographischer oder stadtplanerischer Sicht in der Definition eines Straßenblocks¹⁴ oder Häuserblocks inkludiert. Die umliegenden Straßenzüge stellen die Abgrenzung beziehungsweise eine Barriere für einen Straßenblock dar. Prinzipiell erfordert jedes Polygon das Vorhandensein eines äußeren Ringes¹⁵ als Begrenzung. Im administrativen Sinne können Barrieren auch als (Verwaltungs-) Grenzen interpretiert werden.

Zur Abgrenzung geeigneter Flächen wurde im Projekt „meinGrün“ nun das Konzept einer Barriere entwickelt. Intension war es, die geometrische Abgrenzung der Zielgeometrien hinsichtlich für die Nutzerinnen und Nutzer der App wahrnehmbarer Barrieren zu repräsentie-

¹³ Belebte und unbelebte

¹⁴ engl. „city block“

¹⁵ In PostGIS ableitbar mit ST_ExteriorRing()

ren. Diese befinden sich physisch auf der Erdoberfläche und sollen konzeptionell das Ausüben einer Aktivität einschränken beziehungsweise begrenzen. So würden die Aktivitäten „Fußball spielen“ oder „spazieren gehen“ über eine Mauer oder ein Zaun hinweg nicht möglich sein. Bei einem geöffneten Tor könnten „Joggen“ oder „Rad fahren“ aber fortgesetzt werden. Barrieren lassen sich also aus Nutzersicht basierend auf dem Ausüben von Aktivitäten definieren. Die folgende Tabelle 1 listet die im Projekt enthaltenen Aktivitäten und ihre Kategorie in der meinGrün-App auf. Ergänzt wurde eine eigene Bewertung bezüglich des erwartbaren Aktionsraumes beim Ausüben der Aktivität.

| Aktivität | Kategorie in meinGrün App | Aktionsraum (eigene grobe Einteilung) |
|--------------------------|---------------------------|---------------------------------------|
| Basketball spielen | Aktiv im Grünen | Eher kleinräumiger |
| Federball spielen | | Eher kleinräumiger |
| Fitness | | Eher kleinräumiger |
| Frisbee spielen | | Eher kleinräumiger |
| Fußball spielen | | Eher kleinräumiger |
| Joggen | | Eher weitläufiger |
| Rad fahren | | Eher weitläufiger |
| Skaten | | Spezialfall |
| Tischtennis | | Eher kleinräumiger |
| Volleyball spielen | | Eher kleinräumiger |
| Entspannen | Entspannt im Grünen | Eher kleinräumiger |
| Essen und Trinken | | Eher kleinräumiger |
| Freunde treffen | | Eher kleinräumiger |
| Gassi gehen | | Eher weitläufiger |
| Grillen | | Eher kleinräumiger |
| Lesen | | Eher kleinräumiger |
| Natur beobachten | | Eher kleinräumiger |
| Sonnenbaden | | Eher kleinräumiger |
| Spazieren gehen | | Eher weitläufiger |
| Auf den Spielplatz gehen | | Eher kleinräumiger |

Tabelle 1 Definierte Aktivitäten im Projekt meinGrün (mit Aktionsraum)

Ein Aktivitäten-spezifischer Aktionsraum (als räumliche Ausdehnung während des Ausübens) kann definiert werden und verdeutlicht die Komplexität der Realwelt. Menschen orientieren sich an ihren jeweiligen auf Erfahrung basierenden mentalen Karten und denken dabei in Räumen von unterschiedlicher Dimension. Der Ansatz „Placial versus Spatial“ untersucht konzeptionell jene Verschiebung von klassischen raumbezogenen und auf starren Abgrenzungen basierenden Analyseeinheiten (Landkreis, Stadtteil, Straßenblock, etc.) hin zu einem dynamischen individuellen Beschreiben jener Orte der Menschen, mit denen sie sich identifizieren. So entsteht ein vielschichtiges Modell, welches den Versuch unternimmt, die Menschen in ihrer Perspektive, also in ihrer mentalen Karte, abzuholen und ihre Bedürfnisse untersuchen zu können. (Young 2014) Orte oder Zielangaben können konkrete Point of Interests wie ein bestimmtes Restaurant oder ein markanter Baum sein, welche sich in der Geoinformatik vielleicht als Punktkoordinate ausdrücken lassen. Möglich wären andererseits auch Ortsbeschreibungen welche sich in größeren Räumen vollziehen. Beispiele können das Nennen eines Parks oder Waldgebietes sein („Ich gehe in den Großen Garten“, „Lass uns in der Dresdner Heide wandern gehen“) oder zum Beispiel eines Stadtteils („Ich komme aus der Dresdner Neustadt“). Denkbar wären auch die Identifikation und das Denken in Gebieten links- oder rechtsseitig eines großen Flusses. Oder die Unterteilung in „obere“ und „untere“ Stadt etc. Jeder Mensch ist individuell, auch bezogen auf die eigene Mobilität und Vorlieben beim Erleben von Grünflächen. Die Betrachtung der Aktivitäten aus dem Projekt meinGrün kann in diesen Kontext eingeordnet werden. (Jones et al. 2020; Hardy 2000)

In Tabelle 1 wurde eine eigene einfache Typisierung hinsichtlich des Aktionsraumes durchgeführt. Skaten kann dabei in einem Skatepark eher kleinräumig stattfinden, auf öffentlichen Wegen und Straßen aber auch weite Distanzen zurücklegen. Im Projekt meinGrün wurde zu Beginn ein Multi-Level-Ansatz zur Modellierung von zwei hierarchischen Ebenen favorisiert. Durch sogenannte Zielgeometrien und Sub- /Teilgeometrien sollte die Möglichkeit entstehen, verschiedene Indikatoren zur Ausstattung der Grünflächen wie auch Ausübung von Aktivitäten in der jeweils geeigneten Dimension zu berechnen. Würde für die Aktivitäten „Fahrrad fahren“ oder „Joggen“ eventuell ein vollständiger Park als Ziel zurückgegeben, so würden für „Fußball spielen“ oder „Lesen“ Teilflächen als geeignete Ziele generiert werden. Subgeometrien hätten als Barriere Wege und Landnutzungswechsel nutzen können, während Zielgeometrien vorzugsweise durch Straßen, Bahnstrecken und Flüsse definiert worden wären. Konzeptionell ein Versuch, die komplexe Wirklichkeit herunterzubrechen, erwies sich dieser Ansatz für die technische Implementierung und das methodische Vorgehen im Projekt nicht als durchführbar. Etabliert wurde demzufolge ein einfacher Polygon-Datensatz zur Modellierung.

3.2 Arbeiten mit PostGIS

Da viele Prozessschritte mit den SQL-basierten Funktionen von PostGIS durchgeführt wurden, soll dem Arbeiten mit PostGIS ein eigener Abschnitt gewidmet sein. Neben Möglichkeiten und Vorteilen sollen einige allgemeine Hinweise¹⁶ gegeben werden. Dabei sollen auf räumliche Operationen, UUID, FUNCTION, auf das Arbeiten mit QGIS und mit OSM-Daten sowie auf Laufzeit und Performance und Docker eingegangen werden. Eine Schreibweise in Großbuchstaben weist auf die Bedeutung als PostgreSQL oder PostGIS Befehlswort hin.

PostGIS erweitert PostgreSQL um Funktionalitäten für das Arbeiten mit Daten mit Raumbezug. Neben räumlichen Indizes (SPATIAL INDEX) seien die leistungsfähigen ST-Funktionen erwähnt¹⁷, welche teilweise auf dem Paket GEOS¹⁸ aufbauen. PostGIS setzt die Simple Features Access Spezifikation der OGC um. Es steht unter der freien GNU General Public License Lizenz. Generell stehen auch alle Möglichkeiten von PostgreSQL zur Verfügung. Einfache Analysen lassen sich oft schnell durchführen¹⁹. Große Datenmengen können gut abgefragt und verknüpft werden. Datenmodelle im objektrelationalen DB-Schema sind für ihre jeweiligen Anwendungsfälle adaptierbar, es lassen sich unkompliziert neue Tabellen oder Attribute beziehungsweise Spalten anlegen. PostgreSQL und PostGIS sind Open Source Projekte und finden global eine große Verbreitung. Getragen und weiterentwickelt werden sie durch eine Community. (PostGIS 2021; FOSSGIS 2021)

Im Vergleich zur FME liegt der Fokus in der Verarbeitung weniger auf dem „Datenfluss“ eines einzelnen Features bzw. Objekt, sondern vorwiegend auf dem Arbeiten mit Relationen, also Tabellen. Dennoch kann durch adäquate ORDER BY Sortierung und dank Schleifen oder einem sogenannten CURSOR auf die Reihenfolge der Features während einer Verarbeitung Einfluss genommen werden. Abfragen in PostGIS folgen einem deklarativen beschreibenden Programmierparadigma, doch kann durchaus mit der Art und Weise der Formulierung einer Abfrage auf die im Hintergrund ablaufende Verarbeitung und die Performance eingewirkt werden. (Obe/Hsu 2011, S. 241) Wichtig ist, nur wirklich benötigte Daten auszuwählen und die Menge an zu durchsuchenden Features sinnvoll einzuschränken. Werden Geometrien mit ST_Intersection() verschnitten, könnte eine Prüfung mit ST_Intersects() in der WHERE-Klausel für eine kürzere Laufzeit sorgen. Räumliche Indizes können geometrische beziehungsweise topologische Operationen beschleunigen. Anstelle einer Verwendung von zeilenweise arbeitenden CURSOR oder LOOP (in einer Funktion oder Prozedur) kann geprüft werden, ob analog eine SELECT, INSERT INTO, UPDATE SET oder DELETE Anweisung das gleiche Resultat liefert. Nach Durchführen von ST_Intersection() oder

¹⁶ Im Sinne von „best practise“ Empfehlungen

¹⁷ Siehe https://postgis.net/docs/PostGIS_Special_Functions_Index.html

¹⁸ GEOS - Geometry Engine, Open Source, <https://trac.osgeo.org/geos>

¹⁹ Zum Beispiel unter Nutzung von Befehlen wie COUNT, SUM oder GROUP BY

`ST_Difference()` kann das Ergebnis in verschiedenen (Single- und/oder Multi-) Geometrietypen vorliegen. Dies muss beachtet werden. Gegebenenfalls die Ausgabe-Daten mit `ST_GeometryType()`, `ST_Dimension()` oder `ST_CollectionExtract()` entsprechend auf die gewünschten Geometrietypen reduzieren. Bei Untersuchungen von topologischen Beziehungen kann durch ein „größer als“ Vergleichsoperator bezogen auf existierende (Integer-) IDs das Resultat ohne Duplikate ausgegeben werden. Ansonsten entsteht jede resultierende Geometrie zweimal, für beide Richtungen ($A \rightarrow B$ und $B \rightarrow A$). Wenn möglich für alle Tabellen in der DB ein einheitliches CRS nutzen, gegebenenfalls transformieren. Dies ist relevant für Berechnungen von metrischen Längen oder Flächen. Prinzipiell ist es immer erstrebenswert jedem Feature eine UUID zuzuweisen. Dies ist Voraussetzung zur Visualisierung einer PostGIS-Tabelle in QGIS. Die Erweiterung „uuid-ossp“ hat sich bewährt. Wichtig ist eine neue UUID besonders für Verarbeitungsschritte, in denen Geometrien aufgeteilt oder vereinigt wurden. Nachteil dieser Art von UUID ist, dass keine einfachen Vergleiche basierend auf Ganzzahlen möglich sind. Außerdem erwartet auch die App QField eine Integer-ID, diese wird stattdessen automatisch generiert. Vorteil ist, dass nicht der schon generierte Datensatz permanent auf Eindeutigkeit einer neu gewählten ID abgeprüft werden muss, beispielsweise um die größte noch nicht vergebene Ganzzahl zu finden. Die UUID basiert unter anderem auf der Uhrzeit (siehe Quelltext 1) (uuid-ossp 2021; Obe/Hsu 2011, S. 146).

```
uuid_in(md5(random()::text || clock_timestamp()::text)::cstring) AS "id"
```

Quelltext 1 Generieren einer UUID

Beim Arbeiten mit einer FUNCTION in der Sprache PL/pgSQL stehen viele bekannte Strukturen und Befehle wie Schleifen und Entscheidungen zur Verfügung.

```
CREATE OR REPLACE FUNCTION functionname (
    parameter_list
)
RETURNS TABLE ( column_list )
LANGUAGE plpgsql
AS $$

DECLARE
    -- variable declaration
BEGIN
    -- body
END; $$
```

Quelltext 2 Struktur einer FUNCTION mit PL/pgSQL

Sie wird im gesamten als eine TRANSACTION auf die Datenbank angewandt. Ein COMMIT innerhalb einer FUNCTION ist nicht zulässig. Als „Workaround“ bei erforderlichem COMMIT für Zwischenergebnisse hat sich das Aufteilen in mehrere Funktionen als mögliche Lösung gezeigt. Diese können direkt im SQL zeilenweise ausgeführt werden. Ein COMMIT ist nun möglich (siehe Quelltext 3).

```
SELECT public.funktion1();
COMMIT;
SELECT public.funktion2();
COMMIT;
SELECT public.funktion3();
```

Quelltext 3 möglicher „Workaround“ mit mehreren Funktionen und COMMIT

Zum Beobachten der Variablen und zum Debuggen bei Fehlern hat sich unter anderem das Ausgeben von Inhalten per RAISE NOTICE Nachricht angeboten. Für das Zurückgeben von Zeitstempeln unbedingt TIMEOFDAY() anstelle now() verwenden. Ersteres verwendet auch innerhalb derselben Transaktion die korrekte Zeit, bei letzterem werden jeweils identische Zeiten egal in welcher Codezeile innerhalb einer Transaktion zurückgegeben. Das Arbeiten mit PostGIS und QGIS ist sehr simple und leistungsfähig. Über eine DB-Verbindung können sämtliche Tabellen mit Raumbezug schnell geladen und präzise visualisiert werden. Hier liegen die Stärken von QGIS, beispielsweise gegenüber dem Abfrage-Tool pgAdmin. Die Layer-Funktion „Objektanzahl anzeigen“ in QGIS nur sparsam verwenden beziehungsweise gleich wieder deaktivieren. Ansonsten entstehen beim Öffnen des QGIS-Projektes zu Beginn viele Datenbankzugriffe und eine deutlich verzögerte Ladezeit. Im Falle einer abgestürzten oder nicht erfolgreich terminierten Verarbeitung in PostGIS bietet sich das manuelle Entfernen aller Tabellen an, die unter Umständen unvollständig oder defekt sind. Dies ermöglicht QGIS ein korrektes Aufbauen der DB-Verbindung und ein erfolgreiches Laden aller Tabellen. Geometriespalten mit Geometrien unterschiedlicher Dimensionen (ST_GeometryCollection) werden in mehrere Layer für den Import nach QGIS aufgeteilt. Wurden in der DB das Datenschema einer bereits in QGIS befindlichen Tabelle verändert (durch neue oder geänderte Spalten), so ist sinnvoll, diese Tabelle völlig neu zu importieren. Ein Exportieren des Stils erleichtert das Laden der Symbolisierung um den vorherigen Stand wiederherzustellen. Ändern sich nur Datenzeilen, also Features, werden diese oft in Echtzeit automatisch neu generiert. Im anderen Fall kann in QGIS manuell aktualisiert werden. Der durch die Erweiterung „uuid-ossp“ generierte Datentyp „uuid“ kann von QGIS gelesen und sowohl mit in eine ESRI-Shapefile oder eine GeoJSON exportiert werden. Während auch Tabellen mit „hstore“ und „json“ Attributen in QGIS geladen und korrekt dargestellt werden, scheitert der Export mit einer Fehlermeldung und erzeugt leeren Dateien (siehe Tabelle 2). Helfen könnte ein Umwandeln in „text“ oder „varchar“ Datentypen.

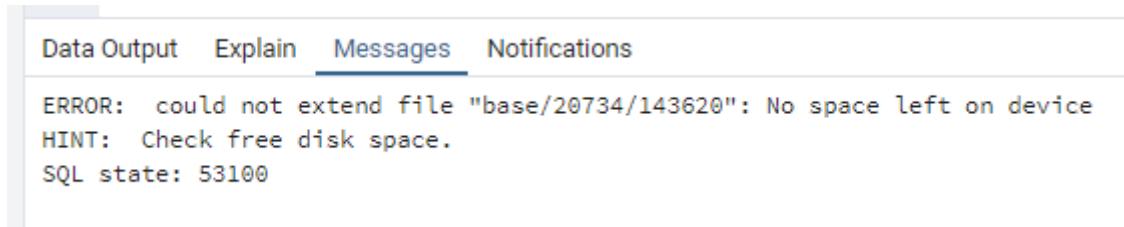
| Datentyp aus DB | Import nach QGIS | Export GeoJSON | Export Shapefile |
|-----------------|------------------|----------------|------------------|
| uuid | ✓ | ✓ | ✓ |
| json | ✓ | ✗ | ✗ |
| hstore | ✓ | ✗ | ✗ |

Tabelle 2 Verwendung spezieller Datentypen (Auswahl)

Zum Arbeiten mit OSM-Daten bringt das Tool osm2pgsql einige Erleichterungen mit. Bedeutend ist die Migration aus dem topologisch strukturierten OSM-Datenschema in ein GIS-fähiges Zielschema. So liegen in der DB nicht nur die originären Nodes, Ways und Relations vor, sondern auch Punkte, Linien und Polygone. Letztere sind für eine weitere Verwendung in der Geoinformatik besonders gut geeignet. Die Erweiterung hstore wird²⁰ erfordert und sollte installiert werden. Im OSM-Datenschema ist die „osm_id“ jeweils bezogen auf Nodes, Ways und Relations eindeutig. Die Migration hin zu Simple Feature Geometrien macht eine neu gezogene Aufteilung in Punkte, Linien und Polygone erforderlich. Dadurch ist die Eindeutigkeit der „osm_id“ nicht mehr gegeben. Aus diesem Grund führt osm2pgsql negative IDs ein, welche die Eindeutigkeit im Zielschema wiederherstellen (osm2pgsql-Manual 2021, S. 2). Bei einem Export von OSM-Daten in das Format ESRI-Shapefile ist die Limitierung der Attribute auf 255 zu berücksichtigen (ArcGIS 2021). Der Aspekt der Laufzeit und Ressourcennutzung sollte während einer (Geo-) Verarbeitung Priorität haben. PostgreSQL und PostGIS verfügen über eine Reihe von Möglichkeiten die erwartbare Laufzeit abschätzen und wenn möglich reduzieren zu können. Mit der Anweisung EXPLAIN SELECT kann eine Bewertung der geschätzten Laufzeit ausgegeben werden. Ebenfalls finden sich Werte für die belegte Speichermenge beziehungsweise Länge einer verarbeiteten Datenzeile. Es existiert im Hintergrund ein sogenannter „query planner“, der eine Abfrage in ihre Teile zerlegt und nach der besten Strategie zum Ausführen sucht (Obe/Hsu 2011, S. 242). Um eine DB-Tabelle zu sichten, kann auch die limitierte Rückgabe der ersten Zeilen sinnvoll sein (LIMIT 1 oder LIMIT 10), um Ressourcen des Computers zu schonen²¹. Bei der Nutzung einer PostGIS-Datenbank mit Docker kann auf verschiedene bereits existierende Docker Images zurückgegriffen. Diese sind vorkonfiguriert und können sofort in einem Docker Container gestartet und genutzt werden. Zum Aufbau einer Datenbankverbindung von pgAdmin oder QGIS aus ist zu beachten, dass diese auf dem Hostbetriebssystem laufen. Von dort sind die Zugangsdaten andere als zum Beispiel mit dem Kommandozeilen-Tool psql, welches „in“ dem laufenden Container im Terminal ausgeführt werden kann. Generell ist das Arbeiten mit Docker strukturiert und einfach. Für verschiedene Projekte können beliebig viele Datenbankcontainer parallel initialisiert werden, und jeweils die aktuell benötigten gestartet werden. Zu beachten ist die standardmäßige Limitierung des Speicherplatzes eines Containers auf 10 GB. Bei Bedarf kann dies über die daemon.json Konfigurationsdatei geändert werden (siehe Abbildung 9 und Abbildung 10).

²⁰ zusätzlich zur Erweiterung postgis

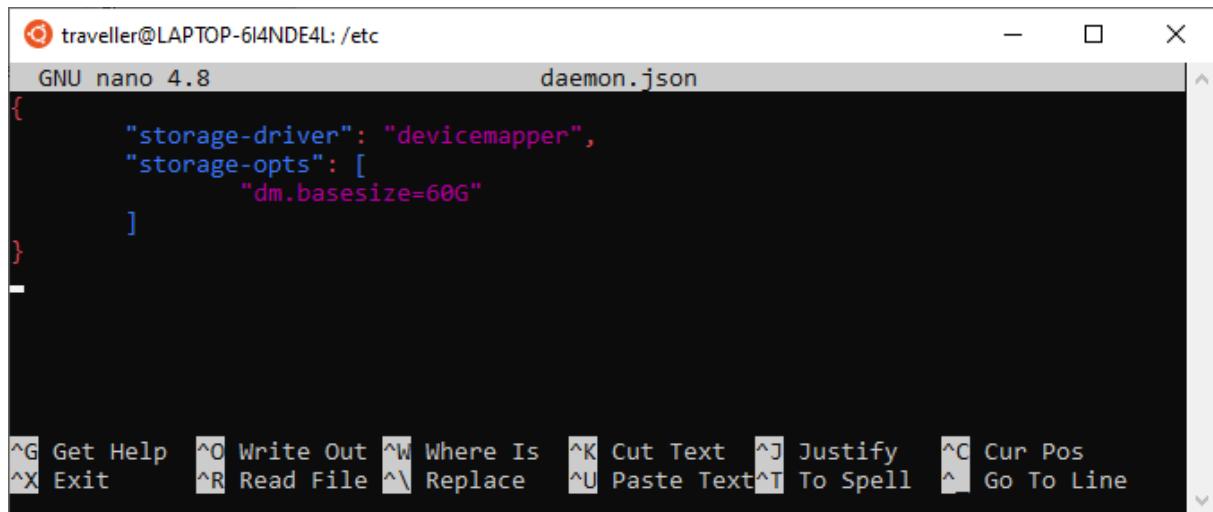
²¹ Vergleiche Python-Bibliothek Pandas den Befehl dataframe.head()



```
Data Output Explain Messages Notifications

ERROR: could not extend file "base/20734/143620": No space left on device
HINT: Check free disk space.
SQL state: 53100
```

Abbildung 9 Fehlermeldung in PostgreSQL zu fehlendem Speicherplatz



```
traveller@LAPTOP-6I4NDE4L:/etc
GNU nano 4.8          daemon.json

{
    "storage-driver": "devicemapper",
    "storage-opts": [
        "dm.basesize=60G"
    ]
}

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos
^X Exit      ^R Read File  ^\ Replace   ^U Paste Text ^T To Spell  ^| Go To Line
```

Abbildung 10 Wahl der Limitierung des Speicherplatzes pro Container

3.3 Maschinelles Lernen

Maschinelles Lernen ist die Disziplin zum Erkennen von Mustern in Daten, um darauf aufbauend Vorhersagen und Zuordnungen treffen zu können. Intrinsische Muster in einem mehrdimensionalen Datensatz sind für Menschen oft nicht erkennbar. Unterschiedliche mathematische Algorithmen können sich einer solchen Aufgabe annähern und stehen für diese Aufgaben bereit. Es handelt sich um eine „Datenwissenschaft“ und lässt sich grundlegend in unüberwachtes und überwachtes Lernen gliedern. Bei letzterem werden bereits existierende Ausgabewerte als Referenz übergeben und können für Training und Testung eines Algorithmus genutzt werden. Es muss also bereits Wissen über ein bestimmtes Gebiet oder Teil des Datensatzes vorhanden sein. In Abhängigkeit der zu erreichenden Qualität ist anschließend die Vorhersage für andere, noch nicht genutzte Datensätze möglich. Sinnvoll kann auch eine Aufteilung in einen Trainingsdatensatz und einen Testdatensatz sein, um innerhalb des Datensatzes eine Validierung der erreichbaren Qualität ermöglichen zu können. Die Form des überwachten Lernens für eine Vorhersage stellt eine von insgesamt drei grundlegenden Herangehensweisen dar. Datensätze müssen aufbereitet und in eine geeignete Metrik übersetzt werden. In einem Modell wird zum Beispiel eine binär codierte Ausgabe zur Vorhersage

von zwei Zuständen („grün“ – 1, „nicht grün“ – 0) genutzt. Eingabeveriablen aus vorhandenen Daten müssen ausgewählt und ebenfalls in sinnvolle Wertebereiche überführt werden. Sie dienen als n-dimensionaler Merkmalsraum als mathematische Grundlage für die Algorithmen. (Ng/Soo 2018; Raschka/Mirjalili 2021; Nguyen/Zeigermann 2021)

Mit der freien Python-Bibliothek „scikit-learn“ steht ein wirkmächtiges Werkzeug für viele Fälle des maschinellen Lernens bereit und lässt sich unter anderem in einem „Jupyter Notebook“ einfach anwenden.

4 Methodik

Das methodische Vorgehen soll im folgenden Kapitel dargelegt werden.

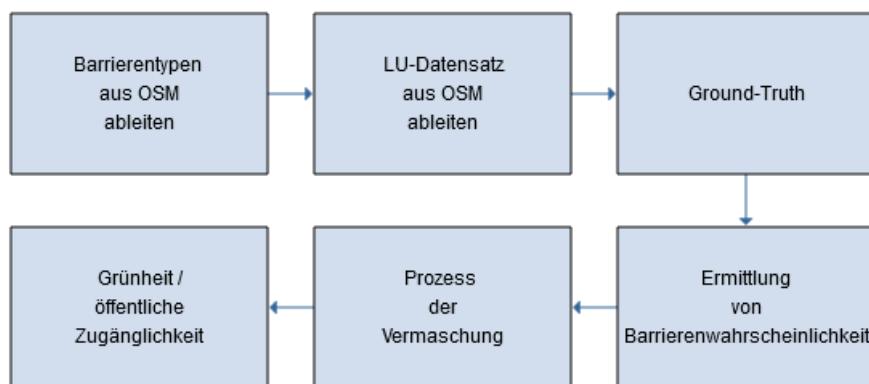


Abbildung 11 Methodisches Vorgehen

4.1 Analyse der Datenquellen

Die verfügbaren Datenquellen sind hinsichtlich ihrer Eignung zu analysieren. Relevant sind Kriterien wie Vollständigkeit, Granularität des Datenmodells zu Eigentumsdaten beziehungsweise Nutzungstypen, sowie die Aktualität. Bei einer nicht ausreichenden Datenlage kann einer späterer Ground-Truth zur Erfassung vor Ort notwendig werden. OSM-Daten werden aufgrund der Fragestellungen bezüglich offener Daten zum Einsatz kommen.

4.2 Ableitungen von Barrieren aus OpenStreetMap Daten

Geeignete Barrieren sind aus OSM zu identifizieren und als einzelne Datensätze abzuleiten. Dabei erfolgt eine Zuordnung von key-value zu den Barrierentypen Bahnverkehr, Straßen, Wege, Wasserwege und Barrieren. Der Typ „Barriere“ orientiert sich am OSM key „barrier“ und repräsentiert bereits in OSM explizit erfasste Barrieren wie Mauern, Zäune oder Hecken. In einem späteren Arbeitsschritt werden Wechsel der Landnutzung beziehungsweise Landbedeckung abgeleitet (siehe 4.3). Es werden folgende Annahmen als sicheres Wissen getätigt:

- Straßen sind immer Barrieren (Barrieren-Wahrscheinlichkeit p=1)
- Schienengebundener Verkehr (Bahnverkehr) ist immer eine Barriere (p=1)
- In OSM explizit mit dem key „barrier“ versehene Linien (wie Zäune, Mauern, Hecken, ...) sind immer Barrieren (p=1)
- Wasserwege / Fließgewässer jeglicher Breite sind immer Barrieren (p=1)

- Tunnel und Brücken sind immer keine Barrieren ($p=0$)

Zu unsicherem Wissen wird dagegen die Barrierenwahrscheinlichkeit von Wegen und von einem Wechsel der Landnutzungs-/ Landbedeckungsklasse gezählt. Die abgeleiteten Barrirendatensätze enthalten ausschließlich Liniengeometrien.

4.3 Ableitung eines OpenStreetMap Landnutzungsdatensatzes

Zur Untersuchung der Wechsel von Landnutzungsklassen wird ein überlappungs- und lochfreier Datensatz auf Basis von OSM benötigt. Diese topologischen Anforderungen erfüllt das OSM Datenmodell für Flächen nicht. Für eine einfachere Vergleichbarkeit und Übertragbarkeit sollen die einzelnen OSM key-value in schon existierende Schemata zur Einteilung von Landnutzung/ Landbedeckung der Erdoberfläche eingeteilt werden. Gleichzeitig erleichtert diese Aggregation die späteren Untersuchungen, es wird Komplexität reduziert. Zu Beginn soll das Flächenschema von „OSM Landuse/ Landcover“ von GIScience²² angewandt werden. Die Zuordnung der OSM key-value in das Zielschema ist dabei vorgegeben. Anschließend wird das Flächenschema des IÖR-Monitors (siehe Abbildung 12) für eine eigene Klassentransformation genutzt werden. Der Schwerpunkt soll auf relevanten Klassen bezüglich Vegetation und Grünflächen liegen, für die auch das IÖR-Monitor Flächenschema angepasst werden kann. Eine Restklasse wird eingeführt, um alle nicht von OSM abgedeckten Flächen aufzunehmen.

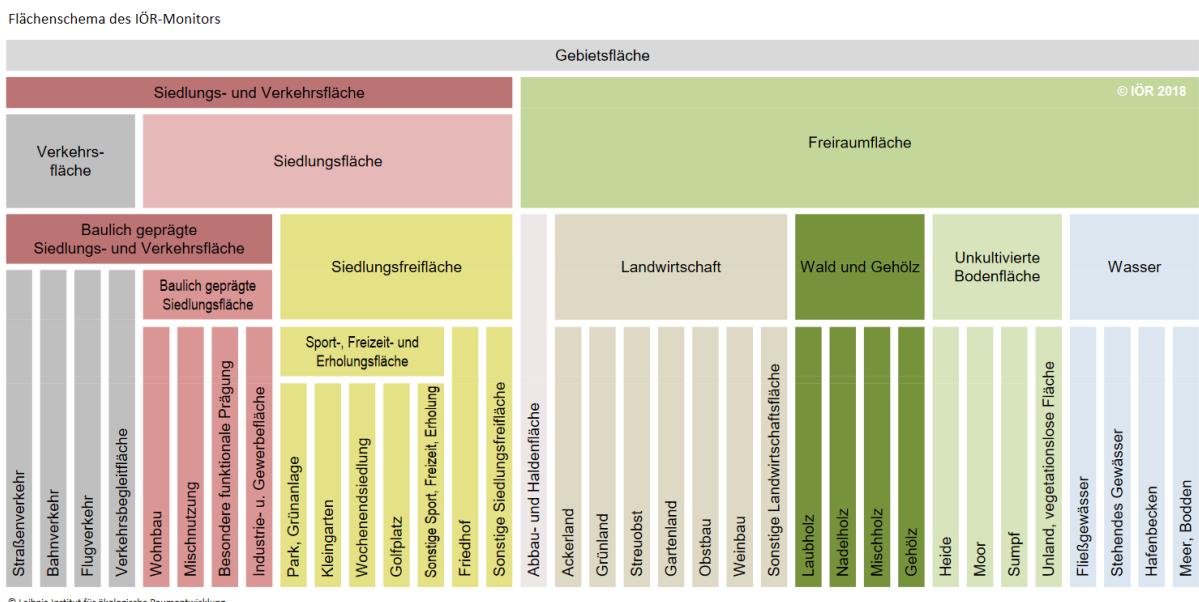


Abbildung 12 Flächenschema des IÖR-Monitors (IÖR 2021)

²² Siehe <https://osmlanduse.org/>

Der abzuleitende Datensatz steht im Spannungsfeld zwischen den Konzepten „Landbedeckung“ und „Landnutzung“. Vereinfacht wird in dieser Arbeit nun der Begriff Landnutzung genutzt. Weitere Schemata wie zum Beispiel von den Projekten LBM-DE, Stadtstrukturtypen der EBK Dresden²³ oder CLC²⁴ wären denkbar, wurden hier nicht weiter untersucht.

Die beiden ausgewählten Zielschemata stellen einen Versuch dar, die komplexe Wirklichkeit in ein vereinfachtes und verallgemeinertes Modell zu überführen. Geht zunächst Detailschärfe der hohen Zahl an Nutzung befindlichen OSM key-value verloren, so erhöht sich die Vergleichbarkeit und Übertragbarkeit. Weiterhin erhöht sich bei diesem methodischen Vorgehen die Anzahl an später zu erfassenden Daten in-situ pro Klasse beziehungsweise Landnutzungswechsel. Das topologisch strukturierte Datenmodell von OSM ist zu berücksichtigen. In Abhängigkeit von gesetzten key-value oder „area“=“yes“ kann bei geschlossenen Linienzügen (als „way“ in OSM erfasst) auf ein flächenhaftes Objekt geschlossen werden. Eine Analyse der Beziehungen der OSM keys untereinander wäre interessant, wurde aus Zeitgründen zurückgestellt²⁵. Das Vorgehen zur Ableitung von überlappungsfreien Landnutzungsdatensätzen erfordert mehrere Verarbeitungsschritte (siehe Abbildung 13).

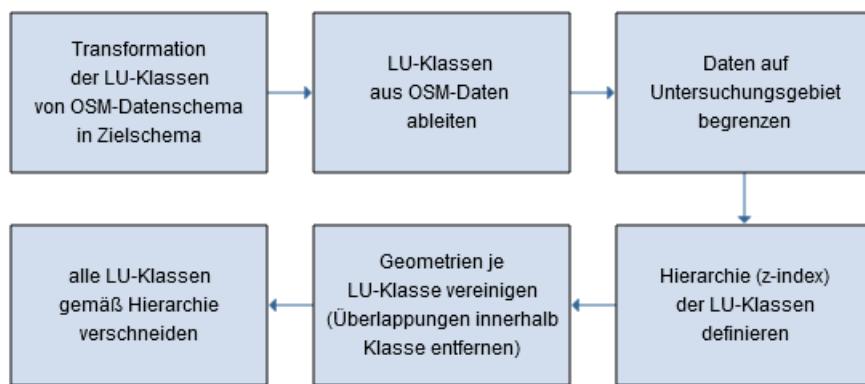


Abbildung 13 Vorgehen zur Ableitung eines überlappungsfreien Landnutzungsdatensatzes

4.4 Ground-Truth

Für eine verbesserte Datenlage werden Daten vor Ort erfasst. Dies erfolgt digital durch die App QField auf einem mobilen Endgerät. Da das verfügbare Wissen zu Landnutzungswechseln und Wegen bezüglich Barrieren unsicher ist, sollen diese Datensätze auf OSM-Basis in-situ überprüft und ergänzt werden. So soll vor Ort eine Einschätzung erfolgen ob eine Barrie-

²³ Erweiterte Blockkartierung

²⁴ Corine Land Cover

²⁵ Die Assoziation bzw. Bindungen zwischen den Keys in einem Graph. Bei welchen vergebenen Key A ist wahrscheinlich, dass auch ein Key B vergeben wurde? Siehe auch (Majic/Winter/Tomko 2017)

re vorliegt oder nicht. Orientierung bietet dabei die Einführung in die Thematik (siehe 3.1). Innerhalb der AOI Dresden werden deutlich verkleinerte Gebiete als AOI für den Ground-Truth definiert. Es ist auf eine möglichst ausgewogene Repräsentation aller benötigten LU-Wechsel und Wegetypen zu achten. So sollen sowohl Grünflächen wie auch bebauten Flächen erfasst werden. Die Kartierung entlang einer festgelegten geraden Linie durch das Untersuchungsgebiet gemäß der Transektmethode wurde angedacht, aus praktischen Aspekten einer solchen Erfassung wieder zurückgestellt (spektrum 2021). Die komplexe Wirklichkeit hinsichtlich dem Nutzerverhalten zur Ausübung von Aktivitäten auf Grünflächen muss in ein vereinfachtes Modell überführt werden. Dies stellt eine Vereinfachung der unterschiedlichen räumlichen Dimensionen eines Aktionsraumes dar. Dies macht eine Einschätzung möglich, ob eine Barriere vorliegt oder nicht. Der Fokus liegt auf Personen, welche ohne größere Einschränkungen mobil und gesund sind. Die Einschätzung soll bezüglich Aktivitäten mit eher kleinerem Aktionsraum erfolgen, wie beispielsweise Fußball spielen oder Lesen.

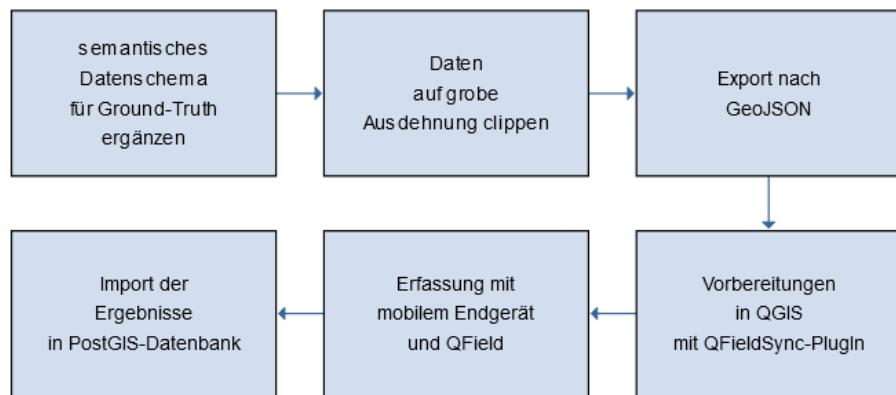


Abbildung 14 Vorgehen zur Durchführung des Ground-Truth

| Bezeichnung | Datentyp | Zulässige Werte |
|----------------|------------------|---------------------------------|
| id / osm_id | INTEGER / SERIAL | |
| geometry / way | GEOMETRY | ST_LineString |
| barriere | VARCHAR | „yes“ – „no“ – „nodata“ |
| comment | VARCHAR | Freies Textfeld für Bemerkungen |
| lastEdit | TIMESTAMP | Aktueller Zeitstempel |
| type_bar | VARCHAR | „alway“ – „temporal“ – „other“ |
| fee_bar | VARCHAR | „yes“ – „no“ |
| fence | VARCHAR | „yes“ – „no“ |
| wall | VARCHAR | „yes“ – „no“ |

Tabelle 3 Datenschema für Ground-Truth

Die Verarbeitung zur Aufbereitung der Ground-Truth-Datensätze ist in Abbildung 14 gezeigt. Zentral bei der Kartierung ist das Attribut „barriere“. Ein „nodata“-Wert bietet die Möglichkeit, nicht zugängliche oder unklare Gegebenheiten vor Ort abzubilden (siehe Tabelle 3). Die Vorbereitungen finden ihren Abschluss mit dem QGIS-Plugin QFieldSync. In QGIS sollen entsprechende Formulare zur vereinfachten und standardisierten Attributeingabe definiert werden. Adäquate Stilregeln zum Visualisieren des Erfassungsstatus sollen die Orientierung während der Kartierung erleichtern. Final wird das Projekt für QField gepackt und auf ein mobiles Endgerät geladen. Anhand der Liniengeometrien der Wege und der LU-Wechsel wird das Untersuchungsgebiet des Ground-Truth erfasst. Für die Auswertung und Analyse erfolgt das Importieren der Resultate in die Datenbank.

4.5 Ermittlung der Barrierenwahrscheinlichkeit

Auf Basis der erfassten Barrieren soll eine Auswertung hinsichtlich der Barrierenwahrscheinlichkeit erfolgen. Für jede Kombination an Landnutzungswechseln ist dabei die Wahrscheinlichkeit einer Barriere zu berechnen. Eine Gewichtung nach kartierten Längen der Liniensegmente erscheint adäquater als nach der Anzahl an Liniensegmenten. So wird eine hohe Anzahl an sehr kurzen Linien nicht überschätzt. Für die Barrierenwahrscheinlichkeit der Wegetypen soll analog vorgegangen werden.

4.6 Prozess der Vermaschung

Aus den abgeleiteten OSM-Barrieren soll gemäß den Annahmen zu sicherem Wissen ein neues Attribut „likelihood“ ergänzt und auf den Wert 1 gesetzt werden. Dies repräsentiert die anzunehmende Barrierenwahrscheinlichkeit von 100%. In einem weiteren Attribut „buffer“ soll zunächst 0 als Standardwert eingetragen werden. Anschließend sollen für ausgewählte Straßen und Bahnstrecken etc. generische Werte für die Breite in Metern gesetzt werden. Orientierung kann die Methodik des IÖR-Monitors zur Linienpufferung wie auch das Vorgehen von Ludwig et al. (2021) geben. Die Ergebnisse des Ground-Truths, gewichtet nach der Länge erfasster Linien, sollen für LU-Wechsel und Wege als Barrierenwahrscheinlichkeit gesetzt werden. Die Vermaschung, also Polygonbildung, kann mit der PostGIS-Funktion ST_Polygonize() erfolgen. Voraussetzung ist ein Liniendatensatz, der an den Schnittpunkten über Stützpunkte beziehungsweise Endpunkte der Linien verfügt (ST_Polygonize 2021; Corti 2014, S. 140). Dies kann entweder durch die Funktion ST_Node() oder ST_Union() umgesetzt werden. Die Vermaschung soll auf ein zu wählende rechteckige AOI für die Analysen begrenzt sein. Final sollen aus Multigeometrien einfache Geometrien erzeugt werden. Ebenfalls muss vom Ergebnis ein gepufferter Datensatz auf Basis der generischen Pufferwerte gebildet und die Differenz berechnet werden. Die Polygone des Resultates sollen über universelle und eindeutige Identifikatoren verfügen.

4.7 Modelle für Zugänglichkeit und Grünheit

Es sind Ansätze zur Ermittlung von Zugänglichkeit (Modell A) oder Grünheit (Modell B) zu erproben. Als Referenz stehen die Datensätze „WFS SN ALKIS Vereinfacht“ (siehe 2.3) und „Stadt Dresden – kommunale Flächen“ (siehe 2.4) zur Verfügung. Beide basieren auf Geometrien von Flurstücken. Als Ausgabe (output) soll entweder die Grünheit oder die öffentliche Zugänglichkeit (ab jetzt nur noch kurz „Zugänglichkeit“) eines Polygons vorhergesagt werden. Es wird für die Ausgabe folgende Festlegung getroffen.

| Wert der Ausgabe | Grünheit | Zugänglichkeit |
|------------------|---|-----------------------------|
| 0 | Keine ausreichende Vegetation vorhanden | Nicht öffentlich zugänglich |
| 1 | Ausreichend Vegetation vorhanden | Öffentlich zugänglich |

Tabelle 4 Festlegungen für Output (Modell „Grünheit“ und Modell „Zugänglichkeit“)

Für das Modell A „Zugänglichkeit“ soll der Datensatz „Stadt Dresden – kommunale Flächen“ als Referenz genutzt werden. Die Fläche des Großen Gartens ist im Sinne einer Datenbereinigung von „Ausreißern“ auszuschließen, da sie als öffentlich zugängliche Fläche in nicht kommunaler Hand zu Fehlern führen würde (sie wird von der Burgen- und Schlösserverwaltung verwaltet). Die ALKIS-Daten können über das Attribut zur Nutzung auf eine Grünheit hinweisen (Modell B).

Als Eingabe-Variablen kommen verschiedene Ansätze in Betracht. Flächeninhalt und Umfang könnten aus der Geometrie abgeleitet werden und treffen eine Aussage zur Dimension und Form des Polygons. Zunächst sollen relevant erscheinende OSM key-value mit den Polygonen verschnitten werden und zu neuen Input-Attributen führen. Intrinsische topologische Beziehungen im OSM-Datensatz sollen so genutzt werden können. Denkbar wäre die Anzahl, oder aber auch die Gesamtlänge eines key-value innerhalb eines Polygons. Ebenso möglich wäre den Anteil am Flächeninhalt als Input zu nutzen. Auch könnte zwischen der originären Polygongeometrie und einer nach Außen gepufferten Geometrie für die Verschneidung variiert werden (um die Nachbarschaft zu inkludieren). Je Polygon sollen folgende Indikatoren abgeleitet und angehängt werden (siehe Tabelle 5, vergleiche auch 6.1).

| Attributname | Datentyp | Input-Indikator | OSM key-value |
|--------------|----------|-----------------------|----------------------|
| „in_1“ | INTEGER | Anzahl Sitzbänke | amenity=bench |
| „in_2“ | INTEGER | Anzahl Picknicktische | leisure=picnic_table |
| „in_3“ | INTEGER | Anzahl Einzelbäume | natural=tree |
| „in_4“ | INTEGER | Anzahl Abfallbehälter | amenity=waste_basket |

| | | | |
|--------|---------|------------------------------|------------------------|
| „in_5“ | INTEGER | Anzahl Trinkwasserstellen | amenity=drinking_water |
| „in_6“ | INTEGER | Anzahl Springbrunnen | amenity=fountain |
| „in_7“ | INTEGER | Anzahl Grillplätze | amenity=bbq |
| „in_8“ | INTEGER | Anzahl Öffentliche Toiletten | amenity=toilets |
| „in_9“ | INTEGER | Anzahl Öffentliches Internet | internet_access=wlan |

Tabelle 5 Mögliche Input-Variablen

Das Modell A (Zugänglichkeit) soll sich auf die Variablen aus Sitzbänken, Abfallbehältern, öffentlichen Toiletten und öffentlichem Internet stützen. Das Modell B (Grünheit) soll auf Sitzbänken, Abfallbehältern, Einzelbäumen und Picknicktischen basieren. Es sind jeweils zufällige Stichproben zu ziehen. Genutzt werden kann die Python-Bibliothek „scikit-learn“, welche unterschiedliche Algorithmen für maschinelles Lernen bereitstellt²⁶. Beide Modelle weisen vier Eingaben und eine erwartbare Ausgabe mit zwei Klassen auf. Es sind Algorithmen des überwachten Lernens zu testen und deren Qualität anzugeben.

²⁶ Siehe <https://scikit-learn.org/stable/>

5 Umsetzung für das Projekt meinGrün

Im Kapitel 4 wurde das methodische Vorgehen beschrieben. Nun soll auf die konkrete technische Umsetzung und die dabei zu lösenden Herausforderungen eingegangen werden.

5.1 Technologie / Arbeitsumgebung

Für die Verarbeitung der Geodaten wurde ein Laptop mit installiertem Betriebssystem Microsoft © Windows 10 Home (64-bit), einer Intel(R) Core(TM) i5-10300H @ 2.50GHz CPU sowie 16 GB RAM genutzt. Die eigentliche Datenhaltung und Datenverarbeitung fand auf einem sogenannten Windows Subsystem für Linux (WSL2) mit installiertem Ubuntu-Betriebssystem Version 20.04 LTS statt. Als Kommandozeilenprogramme wurden das Terminal (Ubuntu) und die Eingabeaufforderung (Windows) verwendet. Grundlage bildet die Technologie von Docker²⁷, welche die Möglichkeit bietet, zum Entwickeln verschiedene portierbare und skalierbare sogenannte Container zu verwenden. Diese verfügen über einen gewissen Grad an Virtualisierung, können selbst frei konfiguriert und mit passenden Programmen und Bibliotheken bestückt werden. Vorteile sind einerseits eine Strukturierung und Übersichtlichkeit in der eigenen Entwicklungsumgebung. Andererseits werden Konflikte bezüglich der installierten Versionen von Programmiersprachen oder Programmen gegenüber anderen Projekten vermieden²⁸. Sie sind kleiner dimensioniert als eine vollwertige virtuelle Maschine.

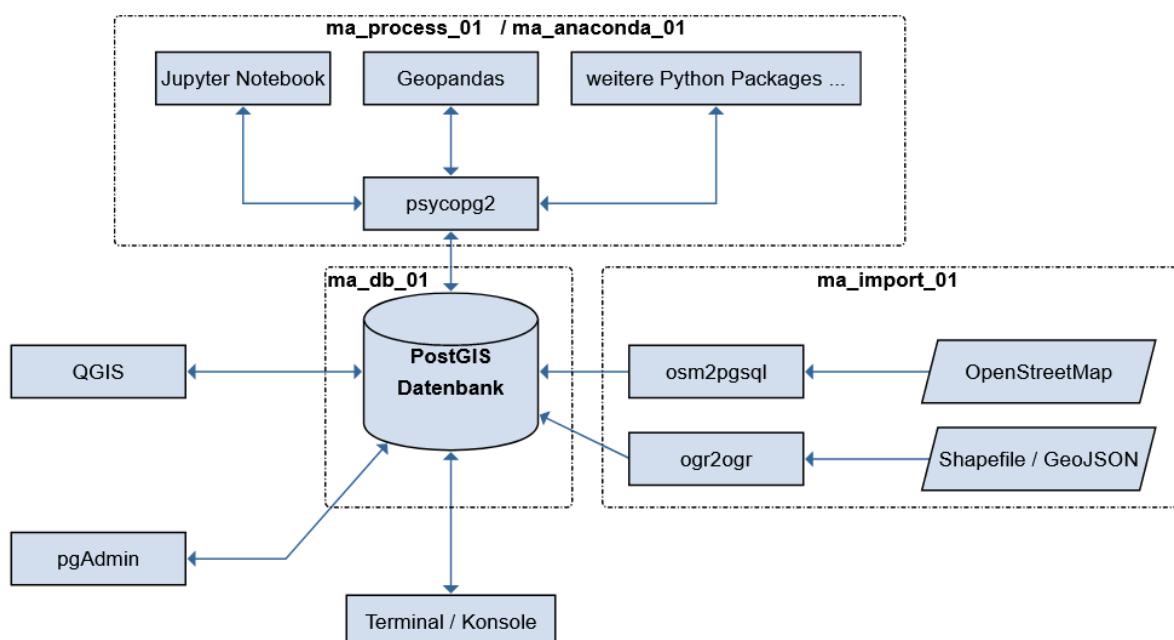


Abbildung 15 Schema Arbeitsumgebung mit Docker Containern und Softwareeinsatz

²⁷ <https://www.docker.com/>

²⁸ Als Beispiel sei hier die Nutzung von Python 2 oder Python 3 in verschiedenen Projekten genannt.

Die Docker Container sind jeweils mit ihren Bezeichnungen und mit gerissenen Linien dargestellt (siehe Abbildung 15). Es existieren eigene Container für die Datenbank, den Import in die DB sowie für Prozessierung, Analysen und Auswertungen. Über entsprechende Datenbankverbindungen und Zugangsdaten kann von QGIS und pgAdmin von außen (vom Windows Betriebssystem aus) auf die DB zugegriffen werden. QGIS überzeugt mit vielfältigen räumlichen Visualisierungsmöglichkeiten, pgAdmin bietet sich für das Ausführen von Abfragen und Funktionen an. Analysen und Auswertungen wurden weiterhin auch auf Basis von Python mit Anaconda, Geopandas und Jupyter Notebooks umgesetzt. Partiell wurden der Editor notepad++ und weitere Programme genutzt. Damit stützt sich der Arbeitsablauf auf Open Source Programme und Technologien.

| Software | Version |
|-------------------------|-------------------------------------|
| Ubuntu | 20.04 LTS |
| Docker | 20.10.7 |
| PostgreSQL | 11.2 |
| • Erweiterung PostGIS | 2.5.2 |
| • Erweiterung hstore | 1.5 |
| • Erweiterung plpgsql | 1.0 |
| • Erweiterung uuid-ossp | 1.1 |
| Python | 3.8.8 |
| Anaconda | 2021.05 Build py38_0 (Conda 4.10.1) |
| QGIS | 3.18.2-Zürich |
| pgAdmin | 5.2 (mit Python 3.8.8) |
| notepad++ | 8.1.1 |

Tabelle 6 Softwareeinsatz mit genutzten Versionen

5.2 Datenhaltung und Datenimport

Die Datenhaltung für diese Arbeit erfolgt vorwiegend zentral in einer PostgreSQL Datenbank mit räumlicher PostGIS Erweiterung. Eine zunächst angedachte zweite DB speziell für importierte OSM-Daten wurde aus technischen Überlegungen nicht separat realisiert²⁹. Die DB ist in einem Docker Container mit der Bezeichnung „ma_db_01“ eingebettet, genutzt wurde das Docker Image von mdillon/postgis³⁰. Zum Ausführen der Datenimporte wurde ein Docker Container „ma_import_01“ auf Basis einer neuen Ubuntu-Umgebung initialisiert.

```
docker run --name ma_db_01 -d -e POSTGRES_PASSWORD=postgres -p 5430:5432 -v
~/ma/ma_db_01:/var/lib/postgresql/data mdillon/postgis

docker run -i -t -d --name ma_import_01 -p 9000:9000 -v
~/ma/ma_import_01:/code ubuntu:latest /bin/bash
```

Quelltext 4 Docker Befehle zum Initialisieren der Container für DB und Datenimporte

Vorliegende Geodatensätze in den Formaten ESRI-Shapefile oder GeoJSON wurden mit dem Tool ogr2ogr in die DB geladen (siehe Anlage 2). Mit curl³¹ konnte der Download des OSM-Extrakts für Sachsen – bereitgestellt durch die Geofabrik (Geofabrik 2021) – mit Stand vom 12.05.2021 realisiert werden³². Der Import der OSM Daten erfolgte mithilfe von osm2pgsql. Dieses Tool ist auf klassische Anwendungsfälle bezüglich der Nutzung von OSM-Daten ausgerichtet, prüft automatisch die Gültigkeit der Geometrien und gewährleistet die Schematransformation aus dem OSM Datenschema³³ hin zu einem GIS-kompatiblen Schema gemäß ISO 19125 / OGC Simple Features Standard³⁴ in einer PostGIS-DB (siehe Abbildung 16).

```
osm2pgsql --create --slim --hstore --multi-geometry --database=osm --
user=postgres --password --host=ma_db_01 --port=5432 --
bbox=13.4,50.9,14.07,51.3 /code/osm_download/sachsen-latest.osm.pbf
```

Quelltext 5 OSM-Daten Import mit osm2pgsql

²⁹ Nötig gewesen wäre die Nutzung von dblink (siehe <https://www.postgresql.org/docs/10/contrib-dblink-function.html>) oder sogenannten „foreign data wrapper“ (siehe <https://www.postgresql.org/docs/9.5/postgres-fdw.html>) für SQL-Abfragen über mehrere Datenbanken hinweg.

³⁰ Siehe <https://hub.docker.com/r/mdillon/postgis/>

³¹ Ein Download mit dem Tool wget war stattdessen nicht erfolgreich.

³² <https://overpass-turbo.eu/> dagegen ist eher zum Download kleinerer Gebiete geeignet

³³ Siehe https://wiki.openstreetmap.org/wiki/Rails_port/Database_schema

³⁴ Siehe <https://www.ogc.org/standards/sfa>

Eine räumliche Eingrenzung gemäß einer definierten Ausdehnung wurde beim DB Import angewandt. Weitere Parameter wurden gesetzt³⁵, mit --slim werden Zwischentabellen generiert. Generell müssen der Bedarf an RAM sowie die Laufzeit eines solchen Imports berücksichtigt werden, besonders bei großen Gebieten.

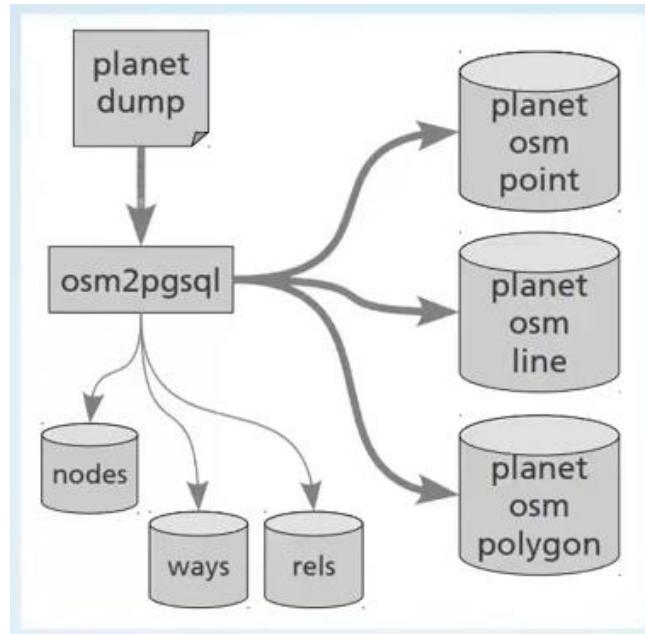


Abbildung 16 Schema zur Arbeitsweise von osm2pgsql (osm2pgsql 2021)

Folgende DB-Tabellen wurden erzeugt:

| | relation text | total_size text |
|---|---------------------------|--------------------|
| 1 | public.planet_osm_ways | 1702 MB |
| 2 | public.planet_osm_nodes | 225 MB |
| 3 | public.planet_osm_polygon | 157 MB |
| 4 | public.planet_osm_line | 101 MB |
| 5 | public.planet_osm_rels | 63 MB |
| 6 | public.planet_osm_point | 52 MB |
| 7 | public.planet_osm_roads | 10 MB |

Abbildung 17 DB-Tabellen nach OSM Import (mit belegtem Speicherplatz)

Aus den OSM Daten wurden 363.718 (*davon 205.315 in der AOI*) Polygone, 233.163 (155.987) Linien und 226.021 (166.956) Punkte abgeleitet. Sie liegen im kartesischen metrischen Koordinatenreferenzsystem WGS 84 / Pseudo-Mercator (EPSG-Code: 3857) vor.

³⁵ Zu möglichen Parameter siehe <https://osm2pgsql.org/doc/manual.html>

5.3 Ableitungen von Barrieren aus OpenStreetMap Daten

Aus den OSM Daten sollen – orientiert am key-value basierten Datenmodell – verschiedene Typen von Barrieren definiert und in separate Datensätze abgeleitet werden. Annahmen wurden getroffen und sind Grundlage dieses Arbeitsschrittes (siehe 4.2).

Die Transformation der OSM-Objektarten (key-value) in die finalen Barrierentypen erfolgte durch die Berücksichtigung der OSM-eigenen Dokumentation³⁶, durch stichprobenartiges Objektabfragen in der offiziellen OSM-Karte³⁷ sowie auf Basis von vorhandenem Wissen des Verfassers. Teilweise wurden einzelne key-value in QGIS visualisiert und deren Relevanz als Barriere hinterfragt. In geringem Maße konnten auch falsch erfasste OSM Daten beobachtet werden. Für einen Überblick über vorkommende key-value im Datenbestand kamen simple Abfragen zur Anwendung (als Beispiel siehe Quelltext 6).

```
SELECT DISTINCT railway, COUNT(*) AS "Anzahl" FROM planet_osm_line
GROUP BY railway ORDER BY "Anzahl" DESC;
```

Quelltext 6 Abfrage für schnellen Überblick über existierende OSM key-value

Das Resultat wurde jeweils in QGIS auf Plausibilität und Vollständigkeit visuell überprüft. Durch ein iteratives Vorgehen konnten Korrekturen in der Zuweisung der key-values erfolgen. Die exakten SQL-Anweisungen sind in der digitalen Anlage zu finden, die Zuordnung der key-value in Anlage 3. Es entstanden folgende DB-Tabellen.

| Barrierentyp | DB-Tabelle | Anzahl Objekte (AOI Dresden) | Länge Objekte in Metern (AOI Dresden) | Anzahl key-value („Objektarten“) |
|--------------|--------------|---------------------------------|---|-------------------------------------|
| Bahnverkehr | osm_railway | 3.939 | 1.466.884,19 | 14 |
| Wege | osm_trail | 49.763 | 7.843.608,47 | 6 |
| Straßen | osm_street | 58.328 | 7.800.821,95 | 20 |
| Wasserwege | osm_waterway | 2.338 | 1.156.718,89 | 7 |
| Barrieren | osm_barrier | 18.803 | 1.864.951,13 | 30 |

Tabelle 7 Liste der abgeleiteten Barrierentypen (mit semantischen Metadaten)

Dem Bahnverkehr wurden Objekte des schienengebundenen Verkehrs zugeordnet. Schrägaufzüge (highway=elevator) wurden ebenfalls ergänzt, sofern es sich um einen nicht geschlossenen Linienzug handelte (ST_IsClosed()), in Abgrenzung zu Fahrstühlen.

³⁶ OSM Map Features, siehe https://wiki.openstreetmap.org/wiki/DE:Map_Features

³⁷ Siehe <https://www.openstreetmap.org/>

```
INSERT INTO osm_railway
SELECT * FROM planet_osm_line WHERE highway='elevator' AND ST_IsClosed(way)=false;
```

Quelltext 7 Selektieren und Einfügen von nicht geschlossenen Linienzügen

Für Wege lag der Fokus auf in der Regel nicht versiegelten Objekten, vor allem Wander-, Rad- und Fußwege wie auch Pfade und Wirtschaftswege. Für Straßen wurden asphaltierte bzw. versiegelte Objekte ausgewählt. Die Datensätze für Wege und Straßen konnten gebündelt bearbeitet werden, orientiert am OSM key „highway“. Eine zunächst angedachte Trennung in schmale Bäche und breitere Flüsse wurde verworfen, stattdessen ein Typ „Wasserwege“ gebildet. Schon explizit in OSM erfasste Barrieren wie Mauer, Zäune oder Hecken wurden ebenfalls abgeleitet und als sicheres Wissen zu Barrieren angenommen. (OSM barrier 2021) Hier konnten alle Werte für den Key „barrier“ übernommen werden. Es erfolgte eine Zusammenführung von Polygonen und Linien, erstere wurden zu Linien umgeformt (ST_ExteriorRing()). Eine Prüfung auf Duplikate bezüglich der Geometrie wurde vorgenommen (ST_Equals()). Es konnten 14 Duplikate detektiert werden. Da der Fokus auf der Erdoberfläche liegt wurden in vielen Datensätzen Tunnel und Brücken jeweils wieder entfernt, also nicht als Barrieren definiert. Auf die Vorzeichen-Thematik des Attributs „osm_id“ wurde in Kapitel 3.2 eingegangen.

Technische Metadaten werden in Tabelle 8 gezeigt.

| DB-Tabelle | Vorzeichen „osm_id“ | CRS (EPSG-Code) | Bezeichnung Geometriespalte | Geometriertyp |
|--------------|-----------------------------|-----------------|-----------------------------|--|
| osm_railway | Oft positiv, selten negativ | 3857 | way | ST_Linestring |
| osm_trail | Positiv | 3857 | way | ST_Linestring |
| osm_street | Positiv | 3857 | way | ST_Linestring |
| osm_waterway | Positiv | 3857 | way | ST_Linestring |
| osm_barrier | Oft positiv, selten negativ | 3857 | way | ST_Linestring (<i>inkl. Umwandlung von ST_Polygon</i>) |

Tabelle 8 Liste der abgeleiteten Barrierentypen (mit technischen Metadaten)

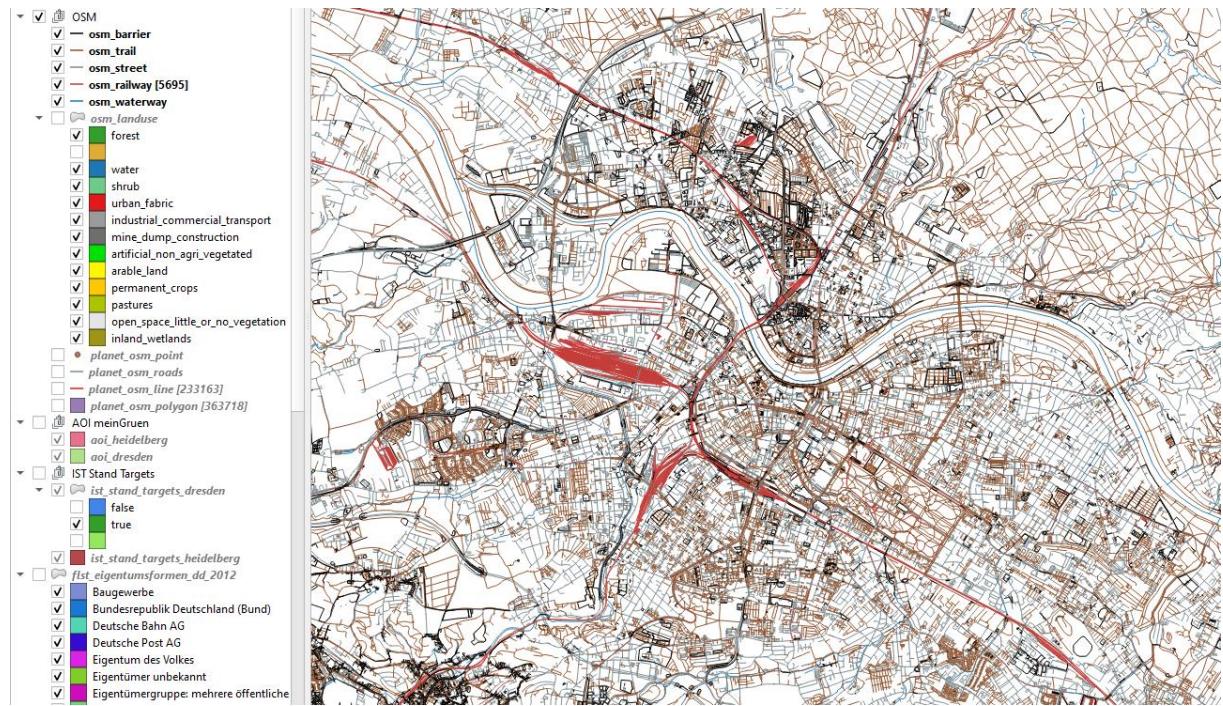


Abbildung 18 Bildschirmabzug in QGIS von abgeleiteten Barrieren (Dresden)

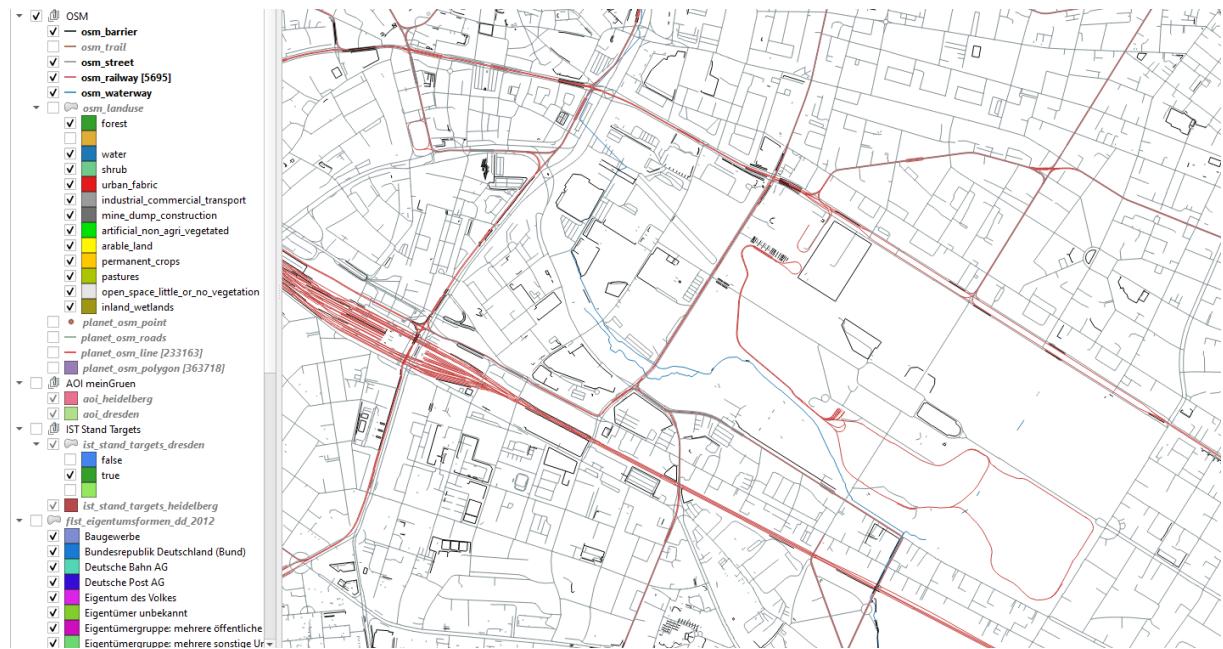


Abbildung 19 Bildschirmabzug in QGIS von abgeleiteten Barrieren (Großer Garten)

Für Barrieren basierend auf Wechsel einer Landnutzung wurde als Zwischenprodukt zunächst ein überlappungs- und lochfreier Landnutzungsdatensatz generiert. Dies wird im folgenden Abschnitt beschrieben.

5.4 Ableitung eines OpenStreetMap Landnutzungsdatensatzes

Zuerst soll der Datensatz gemäß „OSM Landuse/ Landcover“ Schema von GIScience³⁸ abgeleitet werden. Die Zuweisung der key-value wird auf der Webseite³⁹ unter „Info & Contact“ beschrieben. Diese Typisierung ist wie folgt gegeben:

| Nr. | Iu_nr | Bezeichnung (englisch) | Bezeichnung (deutsch) |
|------|-------|---|---|
| 1.1. | 1 | urban fabric | Städtische Bebauung |
| 1.2. | 2 | industrial, commercial and transport units | Industrie, Gewerbe und Verkehr |
| 1.3. | 3 | mine, dump and construction sites | Bergwerke, Deponien und Baustellen |
| 1.4. | 4 | artificial, non-agricultural vegetated areas | Künstlich angelegte, nicht landwirtschaftlich genutzte Vegetationsflächen |
| 2.1. | 5 | arable land | Ackerbau |
| 2.2. | 6 | permanent crops | Dauerkulturen |
| 2.3. | 7 | pastures | Weiden |
| 3.1. | 8 | forests | Wald |
| 3.2. | 9 | shrub and/or herbaceous vegetation assiciations | Sträucher und verkrautete Vegetation |
| 3.3. | 10 | open space with little or no vegetation | Offene Flächen mit wenig oder keiner Vegetation |
| 4.1. | 11 | inland wetlands | Feuchtgebiete im Binnenland |
| 4.2. | 12 | coastal wetlands | Feuchtgebiete an der Küste |
| 5. | 13 | water bodies | Gewässer |

Tabelle 9 LU-Klassen Variante GIScience

Die Klassen wurden basierend auf vorgegebenen key-value zugewiesen. Attribute für eine ganzzahlige Klassenummer (Iu_nr) sowie eine gekürzte Bezeichnung als Text (Iu_type) wurden erstellt und ergänzt. Küstennahe Feuchtgebiete sind im Untersuchungsgebiet Dresden nicht vorhanden. Die Klasse blieb aufgrund der Übertragbarkeit dennoch erhalten. Für eine feingranularere semantische Klassifizierung wurde als zweite Variante das Flächen-

³⁸ GIScience Research Group Institute of Geography Heidelberg University, Germany

³⁹ Siehe <https://osmlanduse.org/>

schema des IÖR-Monitors genutzt, leicht für urbane Grünflächen angepasst und die OSM key-value entsprechend zugeteilt. Einige Klassen des IÖR-Monitor Flächenschemas wurden zusammengefasst, einige wurden bezüglich Grünflächen enger spezifiziert und aufgeteilt (siehe Tabelle 17 und Anlage 4). Wieder wurden die Tabelle „Map Features“ aus dem OSM Wiki genutzt und um stichprobenartige Kontrollen und eigenes Wissen für die Zuweisung ergänzt. Besonders hilfreich in der „Map Features“ Tabelle⁴⁰ sind die in der Spalte „Element“ dargestellten Icons, für welche Geometriertypen dieser key-value genutzt werden kann. Das Icon für „Area“ führt zu flächenhaften OSM key-value (siehe Abbildung 20, ganz rechts).



Abbildung 20 Element-Icons aus OSM Map Features Tabelle

Einige spezifischere Webseiten im OSM Wiki zu einzelnen key-value weisen Zahlen zur globalen Häufigkeit⁴¹ im OSM Datenbestand aus und konnten auch ein Indikator für die Relevanz sein. In den teilweise langen SQL-Anweisungen mussten einige Aspekte beachtet werden: besonders das Attribut „natural“ musste in doppelte Anführungszeichen gesetzt werden (im besten Fall alle Attribute in solch einer Schreibweise nutzen). Das Semikolon zum Terminieren einer Anweisung darf nicht in einer neuen Zeile stehen, sondern am Ende der letzten Zeile. OSM Keys welche durch osm2pgsql keine eigene Spalte erhielten, können und müssen in der Spalte „tags“ im hstore-Datentyp abgefragt werden⁴² (siehe Quelltext 8).

```
INSERT INTO osm_landuse_monitor
SELECT *, 'Unland, vegetationslose Fläche' as "lu_type", 63 as "lu_nr"
FROM planet_osm_polygon WHERE
"natural"='sinkhole' OR
tags -> 'geological'='moraine' OR
landuse='salt_pond';
```

Quelltext 8 (verkürzte) Abfrage zur „hstore“-Nutzung und zu Anführungszeichen bei „natural“

Der Fokus in der Klassentransformation lag auf einer möglichst präzisen Zuweisung im urbanen Raum. Für beide Varianten wurden jeweils selbst gewählte ganzzahlige eindeutige Nummern pro Klasse als Attribut „lu_nr“ vergeben. Es wurden für die Variante GIScience 61, für die Variante IÖR-Monitor 432 flächenhafte OSM key-value insgesamt zugewiesen. Der so generierte (aber auf die AOI Dresden exakt begrenzte) Zwischenstand soll hinsichtlich einiger semantischer, geometrischer und topologische Kriterien kurz analysiert werden (siehe Tabelle 10).

⁴⁰ Ähnlich einem Objektartenkatalog, siehe https://wiki.openstreetmap.org/wiki/DE:Map_Features

⁴¹ Vergleiche auch OSM TagInfo <https://taginfo.openstreetmap.org/> für Statistiken und Häufigkeiten

⁴² Dort befinden sich alle sonstigen – zum Teil sehr speziellen und wenig genutzten – OSM keys. Bedingung ist die Installation der PostgreSQL-Erweiterung hstore.

| Kriterium | Variante GIScience | Variante IÖR-Monitor |
|--|---|--|
| DB-Tabelle | osm_landuse_aoi_dresden | osm_landuse_monitor_aoi_dresden |
| Semantik | | |
| Anzahl LU-Klassen (ohne Restklasse) | 13 (<i>12 in AOI vorkommend</i>) | 33 (<i>28 in AOI vorkommend</i>) |
| Anzahl zugewiesener OSM key-value (Summe) | 61 | 433 |
| Geometrie | | |
| Anzahl Objekte | 25.121 | 96.627 |
| Anzahl Objekte ST_IsValid() ist „true“ | 25.121 | 96.627 |
| Anzahl Objekte ST_IsSimple() ist „true“ | 25.121 | 96.627 |
| Anzahl nach Geometriertyp | ST_MultiPolygon: 77 ST_Polygon: 25.044 | ST_MultiPolygon: 104 ST_Polygon: 96.523 |
| Topologie | | |
| Anzahl geometrischer Duplikate ST_Equals() | 3 | 7 |
| Überlappungen | | |
| Anzahl | 8.445 | 83.080 |
| \sum Fläche [in m ²] | 46.794.834,3316859 | 116.002.079,843137 |
| Anteil Fläche an AOI [%] | 2,48 | 6,16 |
| Löcher (<i>innerhalb AOI</i>) | | |
| Anzahl | 1.971 | 2.073 |
| \sum Fläche [in m ²] | 118.425.936,493064 | 91.973.601,8894801 |
| Anteil Fläche an AOI [%] | 6,29 | 4,88 |

Tabelle 10 Semantische, geometrische und topologische Prüfung – Zwischenstand LU

Es liegen nur gültige Geometrien in der Dimension 2 (flächenhaft) vor. Topologische Fehler sind stark vorhanden und sollen nun entfernt werden (siehe Quelltext 9 und Quelltext 10).

```

SELECT
    COUNT(intersected.*) AS "anzahl_overlaps",
    SUM (ST_Area(ST_Transform(intersected.geometry, 3857))) AS "summe_flaeche"
FROM (
    SELECT
        (ST_Dump(ST_MakeValid(ST_Intersection(a.way, b.way))).geom AS "geometry"
    FROM osm_landuse_aoi_dresden a, osm_landuse_aoi_dresden b
    WHERE ST_Intersects(a.way, b.way) AND a.osm_id > b.osm_id
    ) AS intersected
WHERE ST_Dimension(intersected.geometry) = 2;

```

Quelltext 9 Abfrage zur Ermittlung von Überlappungen (Anzahl und Gesamtfläche)

```

SELECT
    COUNT(intersected.*) AS "anzahl_holes",
    SUM (ST_Area(intersected.geometry)) AS "summe_flaeche"
FROM (
    SELECT
        (ST_Dump(ST_MakeValid(ST_Difference(ST_Transform(a.geometry, 3857), b.geometry))).geom
        AS "geometry"
    FROM aoi_dresden a,
        (SELECT ST_MakeValid(ST_Union(c.way)) AS "geometry" FROM osm_landuse_monitor c
        WHERE c.lu_nr != 0) AS b
    ) AS intersected
WHERE ST_Dimension(intersected.geometry) = 2;

```

Quelltext 10 Abfrage zur Ermittlung von Löchern (Anzahl und Gesamtfläche)

Für einen überlappungsfreien Datensatz bedarf es einer Hierarchie oder Reihenfolge für das Verschneiden der einzelnen LU-Klassen. So kann regelbasiert gewährleistet werden, wie sich die Klassen überlagern⁴³. Auch hier liegt der Fokus auf den Klassen mit Grünflächenbezug. Für die Variante GIScience wurde die Reihenfolge im Original (siehe Nummer in Tabelle 9) übernommen. Mit aufsteigender LU-Nummer soll die Verschneidung durchgeführt werden, sodass die Gewässerkategorie im Vordergrund liegt. Eine Restklasse⁴⁴ mit der niedrigsten LU-Nummer 0 wurde eingeführt, um Löcher schließen zu können. Für die Variante IÖR-Monitor konnte ebenfalls die existierende Abfolge des Schemas übernommen werden (siehe Abbildung 12). Beachtung fanden ebenfalls die neu verfeinerten und adaptierten Klassen. Beide Datensätze wurden in ihren Geometrien exakt auf die AOI Dresden reduziert (ST_Intersection()). Wichtig ist zu beachten, dass bei diesem Schritt aufgrund der AOI Dresden ein Wechsel des CRS von WGS 84 / Pseudo-Mercator (EPSG-Code 3857) hin zu WGS84 (EPSG-Code 4326) erfolgte, und erst später aufgefallen ist. Das Datenschema besteht nun aus den vier Attributen „id“, „lu_nr“, „lu_type“ sowie „geometry“, wobei „lu_nr“ und „lu_type“ voneinander abhängig sind (einmal als Nummer, einmal als Text der LU-Klasse,

⁴³ Aus dem Bereich des Renders sind die Begriffe „z-index“ bzw. „z-order“ im Gebrauch.

⁴⁴ Im Rahmen dieser Arbeit manchmal auch „outer“ genannt.

siehe Abbildung 21). Ebenfalls wurden alle Geometrien pro LU-Klasse vereinigt (ST_Union()), um Überlappungen innerhalb einer Klasse zu eliminieren.

| | id uuid | lu_nr integer | lu_type character varying | geometry geometry |
|----|--------------------------------------|-------------------------|-------------------------------------|-----------------------------|
| 1 | 5098b114-2703-9d95-06c7-131544f14614 | 13 | water | 0106000020E61000... |
| 2 | 643be54d-db50-845a-226e-0fe7e0336862 | 11 | inland_wetlands | 0106000020E61000... |
| 3 | f50629c6-57cd-19f8-0b9f-bd22c74ef198 | 10 | open_space_little_or_no_vegetation | 0106000020E61000... |
| 4 | 8b36e7c4-3190-126a-9e3e-437af965ab0a | 9 | shrub | 0106000020E61000... |
| 5 | 2412e82c-694d-e928-fc3f-41a091ea99d5 | 8 | forest | 0106000020E61000... |
| 6 | 8c4878c1-a984-b5c4-fe1e-c701e1ae7e80 | 7 | pastures | 0106000020E61000... |
| 7 | 2734ce2e-7f22-fa3b-e2cc-f39c1bee4232 | 6 | permanent_crops | 0106000020E61000... |
| 8 | 8236ba1d-615a-b5a3-ce4f-3e51933af388 | 5 | arable_land | 0106000020E61000... |
| 9 | 6b43f948-c2e0-9e43-46f4-40ca3b6a6982 | 4 | artificial_non_agri_vegetated | 0106000020E61000... |
| 10 | 1fedd8a3-ec98-1343-8b63-9e6edb7e83df | 3 | mine_dump_construction | 0106000020E61000... |
| 11 | 84cc0066-cfd4-b5dc-f155-417e2f1798f6 | 2 | industrial_commercial_transport | 0106000020E61000... |
| 12 | e922615a-1497-2f2f-6758-eeb33a220ee8 | 1 | urban_fabric | 0106000020E61000... |
| 13 | f1a1d926-3884-f3dd-5706-3696f0c20b03 | 0 | outer | 0103000020E61000... |

Abbildung 21 Bildschirmabzug Datenschema Variante GIScience nach Vereinigen

Das Finden eines funktionierenden Algorithmus war nicht trivial. Ein erster Ansatz sah wie folgt aus:

1. Umwandlung aller Multipolygone in ihre Außenlinien (ST_ExteriorRing()).
2. Durchführen von Fang-Operation zwischen den Endpunkten der Linien (ST_Snap())
3. Bilden von Polygon anhand der Grenzen des Liniendatensatzes (ST_Polygonize())
4. Je neu gebildeten Polygon (eventuell per Zentroid) die dort in Frage kommenden LU-Nummern aus originalen Polygon-Datensatz auswählen und den größten Wert übernehmen (gemäß Hierarchie)

Aufgrund von Performance-Problemen und Fehlermeldungen bei ST_Polygonize() wurde dieser Ansatz fallengelassen. Ein neuer Ansatz wurde wie folgt entwickelt:

1. Geometrien mit lu_nr=0 (unterste Schichte) in temporäre Tabelle T1 sichern
2. Iteration über Auswahl der im Datensatz inkludierten lu_nr > 0 in aufsteigender Folge
 - a. Von temporärer Tabelle T1 die Geometrien mit lu_nr = i subtrahieren und in T2 sichern
 - b. Entstandene Lücke in T2 durch Geometrien mit lu_nr = i füllen
 - c. Geometrien in T2 entfernen und alle Geometrien aus T2 nach T1 kopieren

Durch dieses iterative Vorgehen werden nacheinander die einzelnen LU-Klassen durchgestanzt. Aufgrund der Schleife wird eine FUNCTION in der Sprache PL/pgSQL notwendig.

Während ein erster Verschneidungstest zwischen Restpolygon (`lu_nr=0`) und städtischer Bebauung (`lu_nr=1`) in nur circa 5 Sekunden erfolgreich war (siehe Quelltext 11 und Abbildung 22), galt es für den gesamten Ablauf einige Herausforderungen in Bezug auf Geometrie, Topologie und Laufzeit zu lösen.

```
CREATE TABLE temp_clip AS
SELECT uuid_in(md5(random()::text || clock_timestamp()::text)::cstring) AS "id",
       d.lu_nr,
       d.lu_type,
       ST_Difference(d.geometry, b.geometry)

FROM (SELECT * FROM osm_landuse_aoi_dresden_union WHERE lu_nr = 0) AS d,
      (SELECT * FROM osm_landuse_aoi_dresden_union WHERE lu_nr = 1) AS b
UNION
SELECT * FROM osm_landuse_aoi_dresden_union WHERE lu_nr = 1;
```

Quelltext 11 Manuelles Verschneiden von Restpolygon mit städtischer Bebauung

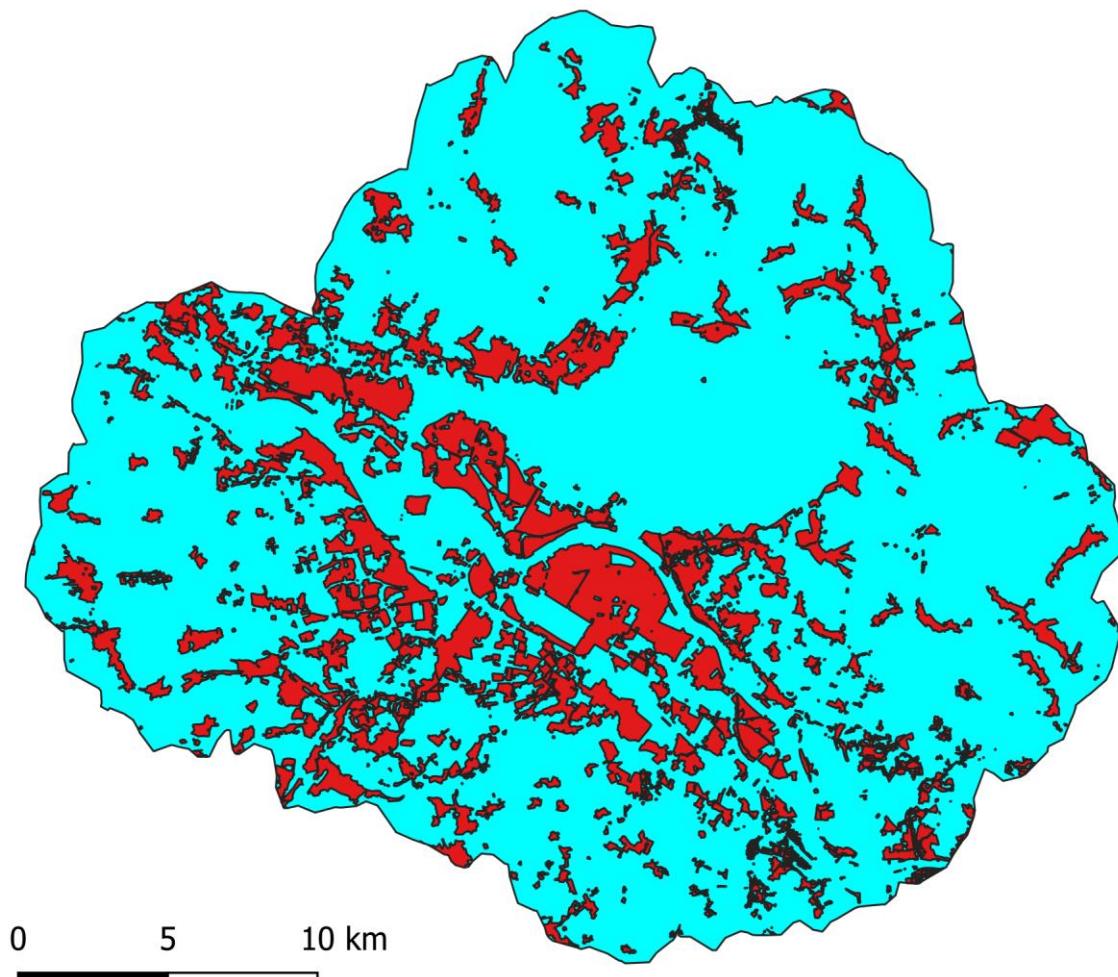


Abbildung 22 Visualisierung eines ersten manuellen Ergebnisses

Entgegen der Erwartung kann eine FUNCTION – zum Beispiel pro Iterationsschritt innerhalb eines LOOP – keine Datenbank-Transaktion ausführen und eine neue starten⁴⁵. Hintergrund ist, dass das Ausführen einer FUNCTION selbst genau eine Transaktion darstellt. In der Konsequenz wird nur einmal am Ende der Schleife bzw. der FUNCTION in die Datenbank geschrieben. Zwischenschritte einer mehrstündigen Prozessierung können so nicht in QGIS geladen werden. Nach erfolgreicher Ausführung des ganzen Algorithmus wurden im Resultat große Löcher beobachtet. Ein manuelles Ausführen und Sichern aller temporären Zwischenschritte in separate Tabellen zeigte, dass ab einem gewissen Iterationsschritt (Nr. 7) nicht mehr nur Multipolygone als Geometriertyp erzeugt wurden, sondern auch ST_GeometryCollection() mit unterschiedlichen Geometriertypen (siehe Abbildung 23). Dort scheiterte das Verschneiden mit nicht vollständig flächenhaft ausgeprägten Geometrien. Die Löcher entstanden und wurden als Fehler in der Iteration fortgeführt.

| | id uuid | lu_nr integer | lu_type character varying | geometry geometry | | st_geometrytype text | |
|---|--------------------------------------|-------------------------|-------------------------------------|-----------------------------|--|--------------------------------|--|
| 1 | 8c4878c1-a984-b5c4-fe1e-c701e1ae7e80 | 7 | pastures | 0106000020E610... | | ST_MultiPolygon | |
| 2 | a8c1bf36-0682-33f8-6ddb-10e620a3df60 | 6 | permanent_crops | 0106000020E610... | | ST_MultiPolygon | |
| 3 | 2d242485-95be-fa5e-11d0-46eb18b220af | 5 | arable_land | 0106000020E610... | | ST_MultiPolygon | |
| 4 | 97b4794b-35c8-47d5-6632-7d06a311a7db | 4 | artificial_non_agri... | 0106000020E610... | | ST_MultiPolygon | |
| 5 | b3a7abc3-c590-294a-8754-04bf9ec3e076 | 3 | mine_dump_constr... | 0106000020E610... | | ST_MultiPolygon | |
| 6 | 1d036008-ec41-f9ca-8b34-127e8f27a444 | 2 | industrial_commer... | 0106000020E610... | | ST_MultiPolygon | |
| 7 | 96b4d304-1709-07be-e1d4-c04bf5a7ea3d | 1 | urban_fabric | 0106000020E610... | | ST_MultiPolygon | |
| 8 | 231a4006-8815-e9d8-c0e2-804cb7d452a0 | 0 | outer | 0107000020E610... | | ST_GeometryCollection | |

Abbildung 23 Bildschirmabzug Zwischenergebnis Variante GIScience Schritt Nr. 7

Im betroffenen Restpolygon (lu_nr=0) konnte nach einer Umwandlung in einfache Geometrien ((ST_Dump()).geom) neben 1.994 Polygonen auch eine Linie aufgezeigt werden. Dies war die Ursache für den Geometriertyp ST_GeometryCollection. Aufgrund des manuellen Ausführens und Sichern in unterschiedliche Tabellen pro Iterationsschritt konnte somit das Problem gefunden werden. Da für die Erstellung dieses Landnutzungsdatensatzes etwaige Linien oder Punkte für die Verschneidung nicht relevant sind, konnten mit dem Befehl ST_CollectionExtract() auftretende ST_GeometryCollection auf nur flächenhafte Geometrien (Dimension = 2) eingeschränkt werden. So konnte dieses Problem gelöst werden. Das Resultat wurde als Tabelle „osm_landuse_aoi_dresden_union_hierarchy“ gespeichert. Während das Ergebnis für die Variante GIScience erfolgreich berechnet werden konnte, erzeugte der selbe Quelltext für die Variante IÖR-Monitor eine kaputte Geometrie für das Restpolygon. Die Form des Multipolygons erscheinen gerastert oder in „Treppchen“-Form (siehe Abbil-

⁴⁵ Der entsprechende Befehl lautet COMMIT. Dieser dient der Kontrolle einer TRANSACTION.

dung 24). Dennoch liefern die Funktionen `ST_IsSimple()` und `ST_IsValid()` für dieses Multipolygon jeweils „true“ zurück.

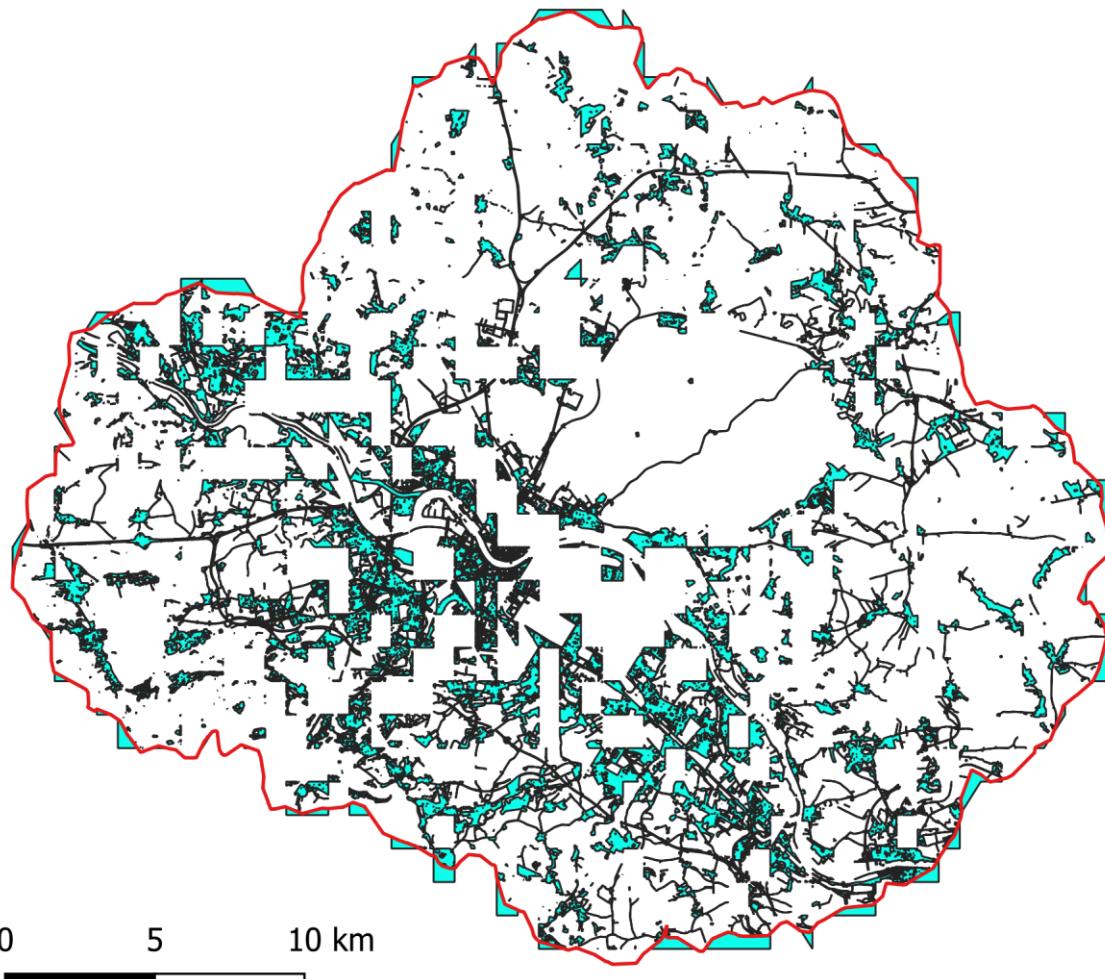


Abbildung 24 Fehlerhaftes Multipolygon der Restklasse in Cyan, AOI in Rot

Vermutet wurden Schwierigkeiten aufgrund der Größe dieses Multipolygons mit einer hohen Anzahl an Teilgeometrien oder durch topologische Verschneidungsfehler bzw. verursacht durch einen räumlichen GIST-Index. Dies konnte jedoch nicht bestätigt werden. Offensichtlich orientiert sich die kaputte Geometrie an einem regelmäßigen Grid bezogen auf das CRS WGS84. Der Versuch wurde unternommen, den Algorithmus anstelle mit je LU-Klasse genau einem Multipolygon mit den zerlegten einfachen Geometrien durchzuführen. Das Ausführen mit vielen kleinen Geometrien erwies sich als deutlich langsamer. Um Daten zu reduzieren, wurden die Ausgangsdaten auf ein verkleinertes Gebiet beschnitten. Nach circa drei Minuten Laufzeit wurde der Fehler „*ERROR: lwgeom_difference: GEOS Error: TopologyException: found non-noded intersection between LINESTRING [...]*“ ausgegeben. Da nun jedoch eine inhaltliche Fehlermeldung existierte, kamen topologische Fehler in der Verarbeitung in Betracht. Die Lösung stellte die gezielte Verwendung von `ST_MakeValid()` wie auch das Snap-

pen aller Stützpunkte auf ein feines Grid dar (`ST_SnapToGrid(geometry, 0.000001)`). Bezo- gen auf das WGS84 und die Lage der AOI in Dresden bei etwa 50° nördliche Breite führt das Snappen zu einem Lagefehler von real etwa bis zu 0,08 Metern. Da dadurch jedoch Run- dungsprobleme und andere Fehler minimiert werden konnten, erschien dies verhältnismäßig.

$$\Delta s [cm] = \frac{\text{Länge 50. Breitenkreis in Zentimetern}}{360^\circ \cdot 10^6}$$

$$\Delta s [cm] = \frac{25.759,17\text{km} \cdot 100.000 \frac{\text{cm}}{\text{km}}}{360^\circ \cdot 10^6} \approx 7,155\text{cm}$$

Formel 1 Berechnung des Lagefehlers durch `ST_SnapToGrid()`

Damit konnte der Algorithmus erfolgreich durchlaufen (Laufzeit für Variante IÖR-Monitor 4 Stunden 29 Minuten). Nach einigen Anpassungen bei `ST_MakeValid()` und `ST_SnapToGrid()` beziehungsweise deren korrekte Platzierung innerhalb der Abfrage konn- ten zuvor noch vorhandene Überlappungen reduziert werden. Nun betrug die Laufzeit für die Variante IÖR-Monitor 2 Stunden 38 Minuten. Das Resultat erhielt den Tabellennamen „osm_landuse_monitor_aoi_dresden_union_hierarchy2“. Eine Analyse der Resultate ist in Kapitel 6.2 zu finden.

5.5 Durchführung eines Ground-Truth

Der Ground-Truth als Datenerfassung in-situ erfordert geeignete Linien-Geometrien, welche zu Trägern der zu erfassenden Attribute werden. Aufbauend auf bisher abgeleiteten Barrie- ren aus OSM wurden diese auf ein grob festgelegtes Gebiet (großzügig um die geplanten Kartierungsgebiete gewählt) reduziert. Zusätzlich zu existierenden Attributen wie `osm_id` oder der Geometrie sollen weitere Attribute für den Ground-Truth ergänzt werden (siehe Ta- belle 3 unter 4.4). Für einen anschließend Export in das Format GeoJSON musste das „tags“ Attribut des Datentyps „hstore“ in den Datentyp „varchar“ konvertiert werden (siehe Quelltext 12 wie auch in 3.2).

```

CREATE TABLE ground_truth_osm_barrier AS
SELECT osm.* FROM osm_barrier osm, aoi_dresden aoi
WHERE st_intersects(aoi.geometry, st_transform(osm.way, 4326));

ALTER TABLE ground_truth_osm_barrier ALTER COLUMN tags TYPE json
USING hstore_to_json(tags);

ALTER TABLE ground_truth_osm_barrier ADD COLUMN tags_text VARCHAR;
UPDATE ground_truth_osm_barrier SET tags_text = tags #>> '{}';

ALTER TABLE ground_truth_osm_barrier DROP COLUMN tags;

```

Quelltext 12 Datenaufbereitung Ground-Truth – Datentyp „hstore“ konvertieren

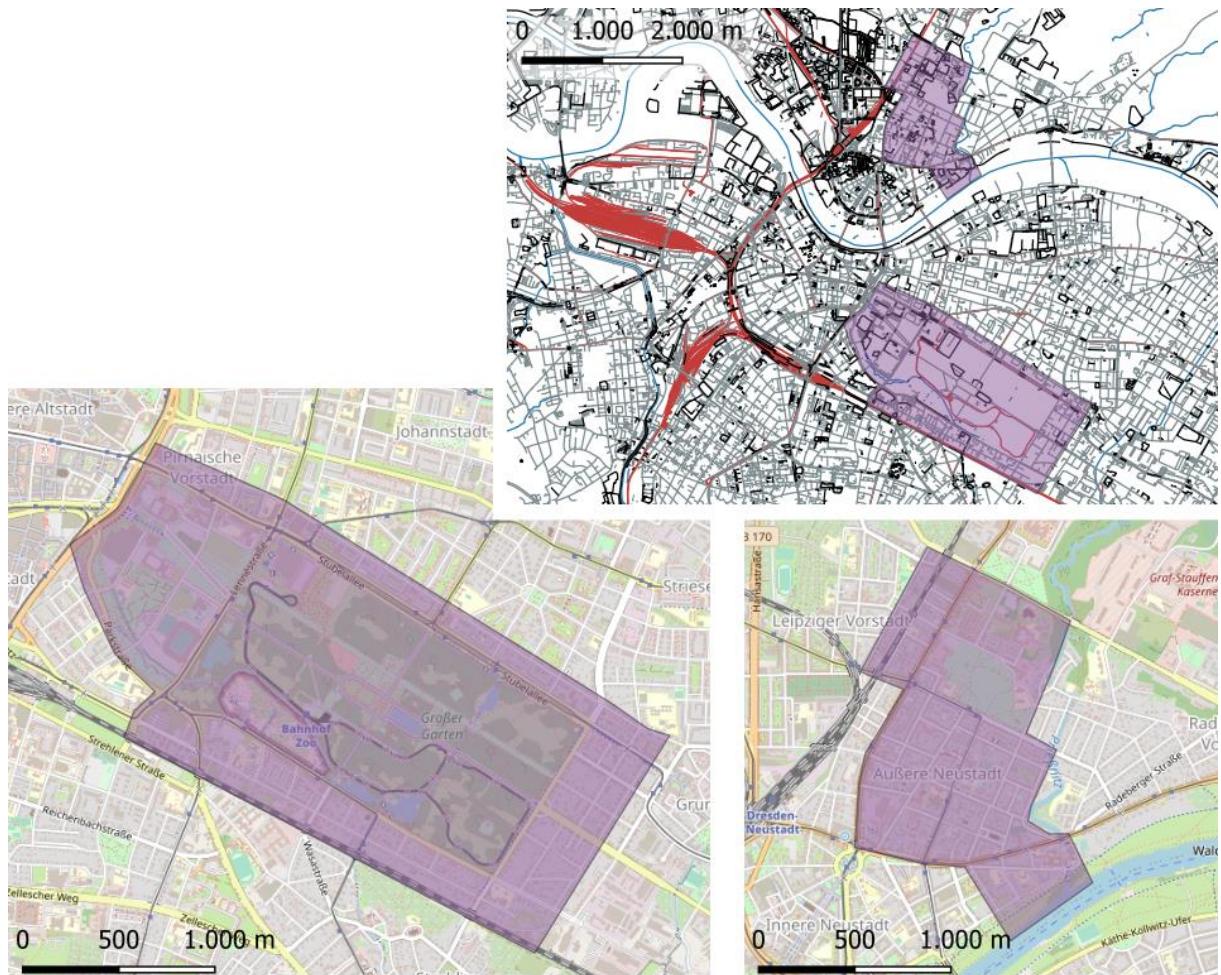


Abbildung 25 Untersuchungsgebiete des Ground-Truth innerhalb der AOI Dresden

Die nun grob ausgeschnittenen Datensätze sind im benötigten Speicherplatz überschaubar (siehe Abbildung 26).

| Name | Änderungsdatum | Typ | Größe |
|------------------|------------------|--------------------|-----------|
| aoi_ground_truth | 27.07.2021 11:31 | OpenStreetMap d... | 3 KB |
| osm_barrier | 27.07.2021 12:27 | OpenStreetMap d... | 7.314 KB |
| osm_railway | 27.07.2021 12:25 | OpenStreetMap d... | 4.452 KB |
| osm_street | 27.07.2021 12:26 | OpenStreetMap d... | 26.979 KB |
| osm_trail | 27.07.2021 12:27 | OpenStreetMap d... | 26.287 KB |
| osm_waterway | 27.07.2021 12:25 | OpenStreetMap d... | 353 KB |

Abbildung 26 Vorbereite GeoJSON-Dateien (außer LU-Wechsel)

Aus dem Landnutzungsdatensatz der Variante IÖR-Monitor wurden passende LU-Wechsel ermittelt. Dies erfolgte durch topologisches Abfragen zur Bestimmung der gemeinsamen Grenzlinie zwischen den Polygonen (siehe Quelltext 13). Mit (ST_Dump()).geom wurden die Multipolygone zuvor in ihre Teilgeometrien zerlegt. Da die Variante IÖR-Monitor noch mini-

malste topologische Fehler enthält, entstanden neben den Linien auch einige Sliver-Polygone. Zweckdienlich erschien eine Lösung, diese Polygone auf ihre längste Seite herunterzubrechen, und diese als Linie zu verwenden. Eine zunächst versuchte Bereinigung und Auflösung der Sliver-Polygone gemäß dem Kriterium der längsten gemeinsamen Seite schlug aufgrund von Fehlermeldungen fehl. Interessant ist die Tatsache, dass die Polygone eventuell eine zu kleine Fläche aufweisen um als Layer in QGIS geladen werden zu können⁴⁶, da für die DB-Tabelle kein Polygon-Layer in QGIS sichtbar war.

```
CREATE TABLE ground_truth_osm_landuse2 AS
SELECT uuid_in(md5(random()::text || clock_timestamp()::text)::cstring) AS "id",
       a.lu_nr AS "a_lu_nr",
       a.lu_type AS "a_lu_type",
       b.lu_nr AS "b_lu_nr",
       b.lu_type AS "b_lu_type",
       ST_Intersection(a.geometry, b.geometry) AS "geometry"
  FROM osm_landuse_monitor_aoi_dresden_union_hierarchy2_ground_dump a,
       osm_landuse_monitor_aoi_dresden_union_hierarchy2_ground_dump b
 WHERE ST_Intersects(a.geometry, b.geometry) AND a.id != b.id;
```

Quelltext 13 Ermittlung der LU-Wechsel

Das Ergebnis enthält Geometrien verschiedener Typen (siehe Abbildung 27). Alle Multigeometrien sollen zu einfachen Geometrien umgewandelt werden. Die Sliver-Polygone – siehe oben – werden mit ihrer längsten Kante ersetzt (Quelltext 14). Alle punkthaften Geometrien können ignoriert werden. So wurde ein Entscheidungsbaum aufgestellt und umgesetzt.

| | Geometry-Type text | anzahl bigint |
|---|-----------------------|------------------|
| 1 | ST_MultiLineString | 23286 |
| 2 | ST_LineString | 3646 |
| 3 | ST_Point | 734 |
| 4 | ST_GeometryCollect... | 572 |
| 5 | ST_Polygon | 154 |
| 6 | ST_MultiPoint | 90 |
| 7 | ST_MultiPolygon | 10 |

Abbildung 27 Enthaltene Geometrietypen mit Anzahl nach ST_Intersection()

Mit diesem Vorgehen wurde allerdings jede Linie doppelt generiert, jeweils von beiden adjazenten Polygonen gesehen. Mit einer ganzzahliger ID könnte in der WHERE-Klausel ein ein-

⁴⁶ Eine Abfrage ergab, dass der Flächeninhalt eines Sliver-Polygons bei $4 \text{ bis } 6 \times 10^{-19}$ Quadratgrad liegt (WGS84)

facher Vergleich `a.id > b.id` hier jeweils nur eine Richtung ermöglichen und die korrekte Anzahl erzeugen. Dies wäre für künftige Verarbeitungen ein adäquater Ansatz.

```

INSERT INTO ground_truth_osm_landuse2_lines
SELECT
    lines.id AS "id",
    lines.a_lu_nr AS "a_lu_nr",
    lines.a_lu_type AS "a_lu_type",
    lines.b_lu_nr AS "b_lu_nr",
    lines.b_lu_type AS "b_lu_type",
    lines.geometry AS "geometry"
FROM (
    SELECT
        t1.id AS "id",
        t1.a_lu_nr AS "a_lu_nr",
        t1.a_lu_type AS "a_lu_type",
        t1.b_lu_nr AS "b_lu_nr",
        t1.b_lu_type AS "b_lu_type",
        (SELECT t3.geometry AS "geometry"
        FROM (SELECT (ST_Dump(ST_Split(ST_Boundary(t2.geometry), (ST_DumpPoints(t2.geometry)).geom))).geom AS "geometry"
              FROM ground_truth_osm_landuse2_polygone t2
              WHERE t2.id=t1.id
            ) AS t3
        ORDER BY ST_Length(t3.geometry) DESC
        LIMIT 1) AS "geometry"
    FROM ground_truth_osm_landuse2_polygone t1
    WHERE ST_NPoints(t1.geometry)= 4
  ) AS lines
  
```

Quelltext 14 Sliver-Polygone ersetzen durch längste Kante

Gemäß den Annahmen zu sicherem Wissen bezüglich Barrieren wurden in der Erfassung nur die Datensätze für Wege und LU-Wechsel genutzt und für die App QField aufbereitet. Diese App ist aus der Vision entstanden, eine mobile Version für das Desktop-QGIS zu sein. Durch die Verwendung identischer Renderer-Bibliotheken im Hintergrund soll die Kartendarstellung der von QGIS gleichen. Da umfangreiche Einstellungen auf einem begrenzten Bildschirm kaum oder nicht vorzunehmen sind, sollen dieser zuvor in QGIS getätigten werden (QField 2021). In QGIS wurde ein entsprechendes Projekt angelegt und Stile in Abhängigkeit der für das Attribut „barriere“ gesetzten Werte definiert. So kann der Status der Kartierung visuell gut erfasst werden (siehe Abbildung 28).

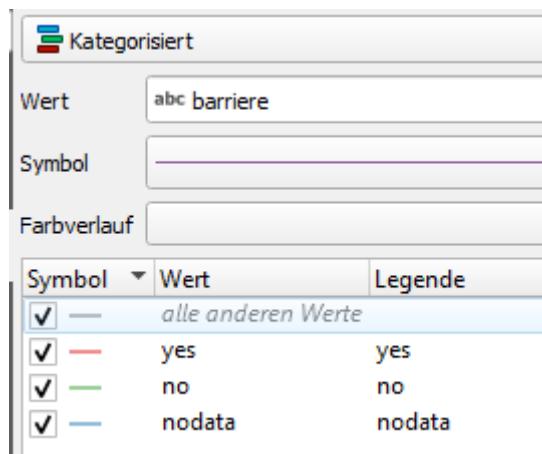


Abbildung 28 Bildschirmabzug QGIS Stil zu Attribut „barriere“

Über die Attributformulare in QGIS können für die Erfassung neuer Objekte klare Regeln definiert werden. Neben Vorgaben für zulässige Werte können verschiedene sogenannte Bedienelementtypen ausgewählt werden. Dazu gehören Eingabefelder („Texteditor“), DropDown-Menüs mit vorgegebenen Werten („Wertabbildung“) oder auch Checkboxen für Attribut vom Datentyp Boolean. Mit dem QFieldSync Plugin kann das Projekt für QField gepackt und auf ein mobiles Endgerät kopiert werden⁴⁷.

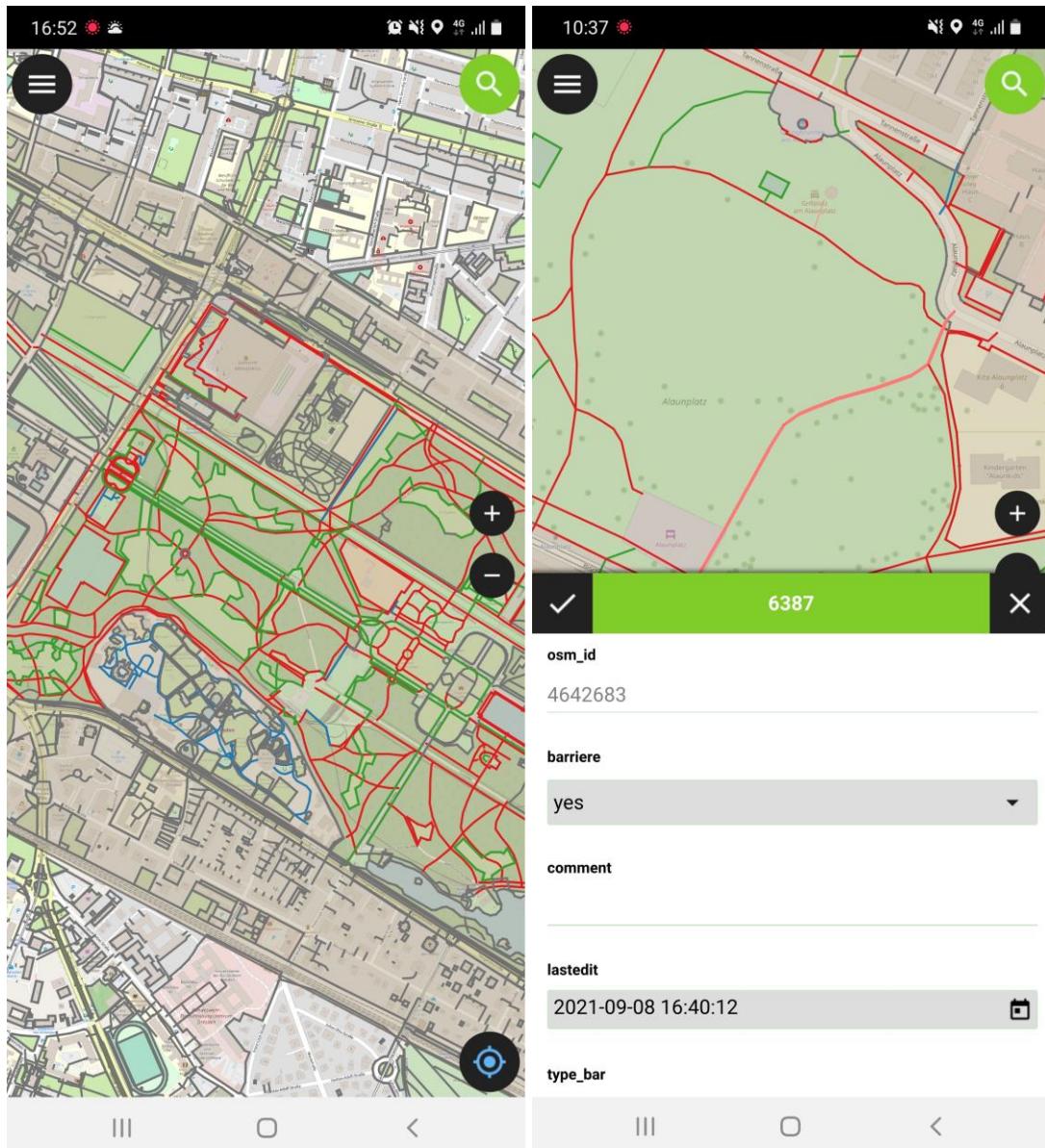


Abbildung 29 Bildschirmabzüge App QField

Es folgen nun einige technische Aspekte und Erfahrungen der Kartierung. Die Aktivierung der Standort-Funktion erleichtert die Orientierung und Navigation. In der Karte kann frei gezoomt werden. So können auch kurze Liniensegmente selektiert und editiert werden. Leider

⁴⁷ Die Layer werden zum Teil in das Format Geopackage konvertiert.

wurde das Feld „lastEdit“ nicht automatisch mit dem aktuellen Zeitstempel gefüllt, sondern erst nach einem Klick auf den rechts positionierten Kalender. So mussten pro Erfassungsvorgang eine Reihe von Klicks getätigt werden. Potenzial zur Vereinfachung ist dadurch noch gegeben. Bei direkter Sonneneinstrahlung war eine schwarze Linie (noch nicht erfasst) kaum von einer grünen Linie (keine Barriere) unterscheidbar. Da alle LU-Wechsel aufgrund des Vorgehens als Duplikate vorhanden waren, gab es Probleme in der Darstellung in der Karte. Die Übersichtlichkeit wurde so erschwert. Während die Liniensegmente der Wege in der Tendenz eher zu lang waren, so bestanden die LU-Wechsel aus vielen kurzen Linien. Letzteres hatte den Vorteil sehr feingranular auf die Begebenheiten vor Ort eingehen zu können. Es entstand allerdings erheblich mehr Aufwand in der Erfassung. Ein wichtiger Aspekt in der Erfassung ist die gebückte Körperhaltung über einen längeren Zeitraum hinweg. Häufigere Pausen wurden so nötig. Die inhaltliche Auseinandersetzung mit der Kartierung der Barrieren ist in Kapitel 6.3 zu finden.

5.6 Ermittlung der Barrierenwahrscheinlichkeit

Die Daten des Ground-Truths wurden vom mobilen Endgerät auf den Computer übertragen und in QGIS zunächst in das GeoJSON-Format exportiert. Es folgte der Import in die Datenbank. Duplikate wurden entfernt. Die Daten mussten pro Landnutzungswechsel in der Form zusammengefasst werden, dass jeder LU-Wechsel nicht in beiden Richtungen existiert ($A \rightarrow B$ und $B \rightarrow A$), sondern nur einmal⁴⁸ (siehe Quelltext 15). Dies erleichtert die spätere Analyse.

```
DROP TABLE IF EXISTS gt_landuse_sel_2;
CREATE TABLE gt_landuse_sel_2 AS
SELECT *,
GREATEST(a_lu_nr, b_lu_nr) AS "a2_lu_nr",
LEAST(a_lu_nr, b_lu_nr) AS "b2_lu_nr",
(CASE WHEN a_lu_nr > b_lu_nr THEN a_lu_type ELSE b_lu_type END) AS "a2_lu_type",
(CASE WHEN a_lu_nr < b_lu_nr THEN a_lu_type ELSE b_lu_type END) AS "b2_lu_type"
FROM gt_landuse_sel;
```

Quelltext 15 Zusammenfassen von LU-Wechsel ($A \rightarrow B$ und $B \rightarrow A$) zu $A \rightarrow B$

Für nachfolgende Berechnungen wurden „nodata“-Objekte ausgeschlossen und stattdessen Anzahl und Gesamtlänge basierend auf den gültigen Werten generiert. Die Wahrscheinlichkeit des Existierens einer Barriere wurde nach Anzahl und nach Länge gebildet. Die Gewichtung nach Länge erscheint zweckdienlich zu sein. So werden sehr kurze oder sehr lange Linien korrekt gewichtet beziehungsweise repräsentiert.

⁴⁸ Es wurden Vergleichsoperatoren auf die lu_nr angewandt. Es handelt sich um keine gerichtete Kombination. Welche LU linksseitig oder rechtsseitig ist, ist in diesem Kontext nicht relevant.

```

CREATE TABLE lu_change_likelihood AS
SELECT b.*,
("anz_barriere_yes)::float / "anzahl_gueltige" AS "anteil_yes_anz",
("laenge_barriere_yes)::float / "laenge_gueltige" AS "anteil_yes_laenge"
FROM (
    SELECT
        a.a2_lu_nr,
        a.a2_lu_type,
        a.b2_lu_nr,
        a.b2_lu_type,
        COUNT(a.*) AS "anzahl",
        (COUNT(a.*) - (COUNT (a.*) FILTER (WHERE a.barriere='nodata'))) AS "anzahl_gueltige",
        COUNT (a.*) FILTER (WHERE a.barriere='yes') AS "anz_barriere_yes",
        COUNT (a.*) FILTER (WHERE a.barriere='no') AS "anz_barriere_no",
        SUM(ST_Length(ST_Transform(a.geometry, 3857))) FILTER (WHERE a.barriere='yes') AS "laenge_barriere_yes",
        SUM(ST_Length(ST_Transform(a.geometry, 3857))) FILTER (WHERE a.barriere!='nodata') AS "laenge_gueltige"
    FROM gt_landuse_sel_2 a
    WHERE a.barriere IS NOT NULL
    GROUP BY a.a2_lu_nr, a.b2_lu_nr, a.a2_lu_type, a.b2_lu_type) AS b
WHERE "anzahl_gueltige" > 0
ORDER BY "anteil_yes_laenge" DESC, "anzahl" DESC;

```

Quelltext 16 Berechnung der Barrierenwahrscheinlichkeit nach Anzahl und nach Länge

Die berechneten Wahrscheinlichkeiten stehen nun in der Tabelle „lu_change_likelihood“ bereit. Analog wurde auch die Barrierenwahrscheinlichkeiten für die Wegetypen berechnet und liegen in der Tabelle „trail_likelihood“ vor.

5.7 Prozess der Vermaschung

Das Ziel ist die Generierung der Grünflächengeometrien. Zur Polygonbildung soll eine Vermaschung durchgeführt werden. Voraussetzung dafür ist ein geeigneter Liniendatensatz, dessen Liniensegmente an ihren Schnittpunkten jeweils aufgetrennt sind, also enden oder beginnen. Zunächst wird eine als „Linien-Pool“ bezeichnete Tabelle deklariert, welche alle relevanten abgeleiteten Barrieren mit ihren Barrierenwahrscheinlichkeiten und Pufferweiten aufnehmen soll (siehe Quelltext 17).

```

CREATE TABLE linien_pool (
    id UUID,
    origin VARCHAR,
    likelihood NUMERIC,
    buffer NUMERIC,
    geometry GEOMETRY
);

```

Quelltext 17 Datenschema für Tabelle „linien_pool“

Vor dem Import in die Tabelle „linien_pool“ müssen korrekte Werte für Pufferweite und Barrierenwahrscheinlichkeit gesetzt werden (siehe Tabelle 11). Diese orientieren sich an der Methodik des IÖR-Monitors (IÖR Monitor 2021). Teilweise werden eigene Annahmen und Vereinfachungen vorgenommen. Zu beachten ist, dass die Pufferweiten jeweils die halbe Breite des Realweltoobjektes betragen müssen. Die Modellierung von Bahnstrecken erfolgt in

OSM auf Gleisebene, also pro Gleis eine Linie. Größere Straßen und Autobahnen weisen in der Regel pro Richtung eine eigene Linie auf, aber nicht jeder Fahrstreifen. Für Autobahnen sollen jeweils zwei Fahrbahnen (2 x 3,5 m) und ein Standstreifen (2,0 m) pro Richtung zuzüglich eines Mittelstreifens (3,0 m) angenommen werden. Die resultierende Autobahnbreite von 21,0 Metern wird auf zwei zu puffernde OSM-Linien aufgeteilt. Für andere Straßen (oft durch genau eine Linienachse modelliert) soll eine Fahrspur pro Richtung mit einer Breite von insgesamt 6,0 Metern angenommen werden (Grundmaß eines LKWs, angelehnt an „*Richtlinien für die Anlage von Straßen – Querschnitt*“).

| Bezeichnung | „origin“ | „buffer“ in Metern (halbe Breite des Realweltobjektes) | „likelihood“ mit $0 \leq p \leq 1$ |
|--------------------|-----------------|---|--|
| Straßen | osm_street | Autobahn: 5,25 Andere Straßen: 3 | 1 |
| Bahnstrecken | osm_railway | Allgemein: 3,75 Straßenbahn: 2,25 | 1 |
| Wasserwege | osm_waterway | 1 | 1 |
| Barrieren | osm_barrier | 0 | 1 |
| Wege | osm_trail | 1 | 0 ... 1 |
| LU-Wechsel | lu_change | 0 | 0 ... 1 |

Tabelle 11 Aufbereitung der abgeleiteten Barrieren für Vermaschung

Für eine effizientere Analyse erfolgt das Zuschneiden aller Linien aus der Tabelle „linien_pool“ auf eine BoundingBox⁴⁹. Die daraus resultierenden Linien (`ST_Dimension() = 1`) werden in eine neue Tabelle „linien_pool_bbox“ gesichert (siehe Quelltext 18).

⁴⁹ POLYGON ((13.7131 51.0315, 13.8032 51.0315, 13.8032 51.1, 13.7131 51.1, 13.7131 51.0315))

```
CREATE TABLE linien_pool_bbox AS
SELECT
    dumped.id AS "id",
    dumped.origin AS "origin",
    dumped.buffer AS "buffer",
    dumped.likelihood AS "likelihood",
    dumped.geometry AS "geometry"
FROM (
    SELECT a.id AS "id",
        a.origin AS "origin",
        a.buffer AS "buffer",
        a.likelihood AS "likelihood",
        (ST_Dump(ST_Intersection(ST_Transform(bbox.geometry, 3857), a.geometry))).geom AS "geometry"
    FROM aoi_dresden_bbox bbox, linien_pool a
    WHERE ST_Intersects(ST_Transform(bbox.geometry, 3857), a.geometry)) AS dumped
WHERE ST_Dimension(dumped.geometry) = 1;
```

Quelltext 18 ST_Intersection und ST_Dump auf „linien_pool“ zu BoundingBox

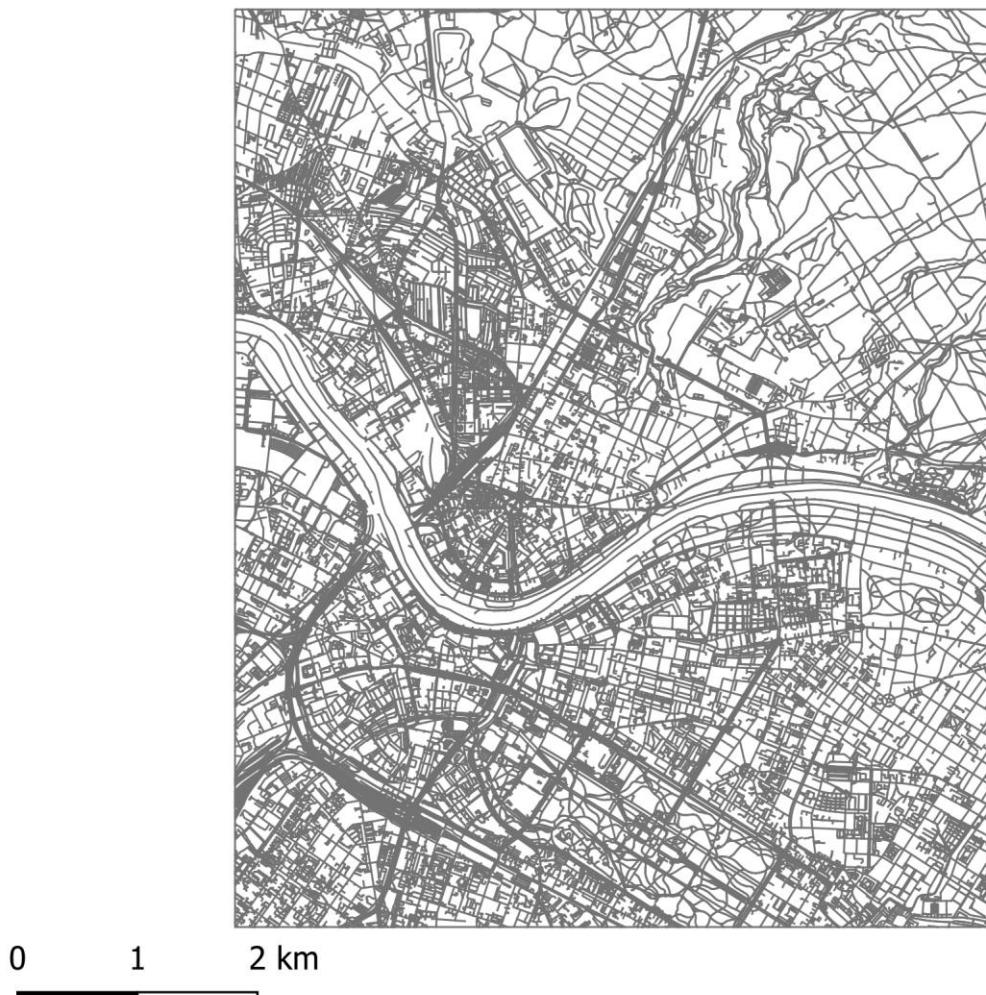


Abbildung 30 Datensatz „linien_pool_bbox“ mit BoundingBox

Zuzüglich der Begrenzungsline der BoundingBox⁵⁰ liegt folgender Datensatz vor (siehe Abbildung 30). Er umfasst 101.208 Linien mit einer Gesamtlänge von 4.751.944,81 Metern (ohne Begrenzungsline der BoundingBox). Für die Vermaschung werden alle Linien zunächst vereinigt. Es folgt die Polygonbildung und eine Reduktion auf einfache Polygone (siehe Quelltext 19). Für die BoundingBox dauert dieser Vorgang circa 13 Minuten.

```
CREATE TABLE polygonize_all AS
SELECT uuid_in(md5(random()::text || clock_timestamp()::text)::cstring) AS "id",
       dumped.geometry AS "geometry"
  FROM (
    SELECT (ST_Dump(polygonized.geometry)).geom AS "geometry"
  FROM (
    SELECT ST_Polygonize(unified.geometry) AS "geometry"
  FROM (
    SELECT ST_Union(src.geometry) AS "geometry" FROM linien_pool_bbox src
  ) AS unified
  ) AS polygonized
  ) AS dumped
 WHERE ST_Dimension(dumped.geometry) = 2;
```

Quelltext 19 Vermaschung mit ST_Polygonize()

Ein zu bildener Pufferlayer wird von dem Resultat subtrahiert⁵¹. Die letzten beiden Schritte können für unterschiedliche Grenzwerte bezüglich der Barrierenwahrscheinlichkeit generiert werden.

5.8 Modelle für Zugänglichkeit und Grünheit

Zu Beginn erfolgt eine Datenaufbereitung. Der ALKIS WFS-Layer „nutzung_flurstueck“ wurde via QGIS lokal als GeoJSON gespeichert und in die PostGIS-Datenbank importiert. Die Daten wurden in das CRS WGS 84 / Pseudo-Mercator (EPSG-Code: 3857) transformiert und auf die Bounding Box der Vermaschung eingegrenzt. Ebenso wurde mit Daten zu „Stadt Dresden kommunale Flächen“ verfahren. Ein neues Attribut „out“ wird pro Polygon jeweils auf den korrekten Wert gesetzt (siehe 4.7). Die angegebene Nutzung der ALKIS-Flurstücke muss dabei in „enthält Vegetation“ (1) oder „enthält keine Vegetation“ (0) konvertiert werden. Es wird die Annahme getroffen, dass die Nutzungsarten „Wald“, „Gehölz“, „Friedhof“ und „Sport-, Freizeit- und Erholungsfläche“ als Grün anzusehen sind, daher den Wert eins als „out“ erhalten. Neue Eingabe-Variablen wurden als „Features“ pro Polygon berechnet (auf Basis der Tabelle 5 siehe 4.7). Grundlage bildete die Anzahl des OSM key-value pro Polygon. Nach der Datenaufbereitung wurde ein neues Jupyter Notebook im Docker Container „ma_anaconda_01“ angelegt. Die eben aufbereiteten Daten wurden geladen und in zwei DataFrames „data“ (die vier Input-Variablen) und „target“ (Output mit zwei Klassen, 0 oder 1)

⁵⁰ ST_ExteriorRing() mit „buffer“=0 und „likelihood“=1

⁵¹ Dieser Schritt benötigt circa 46 Minuten für die BoundingBox.

aufgeteilt. Es wurden verschiedene Algorithmen auf den Datensatz angewandt. Der Datensatz kann mit der Funktion „train_test_split()“ simple nach dem Zufallsprinzip in einen Trainings (70%) - und einen Testdatensatz (30%) aufgeteilt werden ((Raschka/Mirjalili 2021, S. 147)) (siehe Quelltext 20).

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(data,
                                                    target.values.ravel(),
                                                    test_size=0.30,
                                                    random_state=42)
```

Quelltext 20 Aufteilung in Trainings- und Testdatensatz

```
from sklearn.svm import SVC
modell = SVC(kernel='linear', C=1)
modell.fit(X_train, y_train)
modell.score(X_test, y_test)
```

Quelltext 21 Anwenden eines linearen Support Vector Classifier (SVC)

6 Evaluation der Ergebnisse

6.1 Explorative Datenanalyse OpenStreetMap

Zu Beginn erfolgen einige Untersuchungen des originären OSM-Datenbestandes. Sie sind bezogen auf die AOI Dresden. So soll der Datensatz bezüglich relevanter Objekte mit Bezug zu Grünflächen besser quantitativ und qualitativ erfasst werden können.

| Bezeichnung | OSM key-value | Anzahl |
|-----------------------|------------------------|--------|
| Sitzbänke | amenity=bench | 5.965 |
| Picknicktische | leisure=picnic_table | 90 |
| Abfallbehälter | amenity=waste_basket | 1.591 |
| Einzelbäume | natural=tree | 22.933 |
| Trinkwasserstellen | amenity=drinking_water | 17 |
| Springbrunnen | amenity=fountain | 171 |
| Grillplätze | amenity=bbq | 24 |
| Öffentliche Toiletten | amenity=toilets | 257 |
| Öffentliches Internet | internet_access=wlan | 259 |

Tabelle 12 Analyse zu punkthaften OSM-Objekten mit Grünflächen-Bezug (POIs)

| Bezeichnung | key-value | Anzahl | Fläche | | | |
|-------------|--------------------|--------|------------------------|------------------------|--------------------------|--------------------------|
| | | | Ø in m ² | σ in m ² | Max in m ² | Min in m ² |
| Wald | natural=wood | 595 | 37.095,34 | 137.616,24 | 2.143.829,70 | 32,60 |
| | landuse=forest | 1.674 | 340.253,85 | 4.006.902,09 | 134.519.530,00 | 141,30 |
| Parks | leisure=park | 551 | 30.254,83 | 174.423,89 | 3.834.848,22 | 27,89 |
| Kleingärten | landuse=allotments | 1.014 | 39.357,21 | 62.747,61 | 1.429.995,89 | 296,75 |
| Friedhöfe | landuse=cemetery | 89 | 61.154,96 | 112.532,33 | 794.249,87 | 2.963,97 |
| | amenity=grave_yard | 27 | 8.867,96 | 9.273,05 | 41.563,35 | 147,21 |
| Spielplätze | leisure=playground | 733 | 1.316,08 | 2.211,14 | 29.059,54 | 2,49 |
| Sportplatz | leisure=pitch | 852 | 4.857,83 | 6.655,22 | 50.515,02 | 5,47 |

Tabelle 13 Analyse zu flächenhaften OSM-Objekten mit Grünflächen-Bezug

Grünflächen-bezogene Objekte variieren quantitativ bezüglich ihrer Erfassung in OSM, sind dennoch Teil der OSM-Datenbank und werden von der Community prinzipiell miterfasst (siehe Tabelle 12 und Tabelle 13). Eine Analyse hinsichtlich eines Referenzdatensatzes⁵² könnte die Qualität von Vollständigkeit, Lagetreue und Semantik prüfen. Wie in Tabelle 13 gezeigt, sind in OSM pro Objektart zum Teil mehrere key-value zu berücksichtigen.

| Geometriertyp | Anzahl Gesamtanzahl | Anzahl barrier IS NOT NULL | Anteil Barrier IS NOT NULL zu Gesamtanzahl | Top 5 Values nach Anzahl |
|----------------------|-------------------------------|---|--|--|
| Punkt | 166.956 | 8.611 | 5,16 % | ,gate' – 5.398 ,bollard' – 1.479 ,lift_gate' – 789 ,cycle_barrier' – 235 ,block' – 191 |
| Linie | 155.987 | 18.371 | 11,78 % | ,fence' – 8.020 ,hedge' – 4.950 ,wall' – 2.643 ,retaining_wall' – 1.821 ,handrail' - 601 |
| Polygon | 205.315 | 447 | 0,22 % | ,fence' – 350 ,wall' – 71 ,hedge' – 20 ,city_wall' – 3 , retaining_wall ' – 2 |

Tabelle 14 Analyse Key „barrier“ nach Anzahl (bezogen auf AOI Dresden)

Die Analyse in Tabelle 15 zeigt, dass Linien (gemäß Simple Features Access) mit 11,78% den höchsten Anteil an vergebenen „barrier“-Key aufweisen. Der geringste Anteil liegt mit nur 0,22 % bei den Polygonen. Values sind Geometrie-Typ spezifisch vergeben und erscheinen plausibel. Während für punkthafte Objekt vorzugsweise Tore oder Poller kartiert werden, verfügen Linien oder Flächen stattdessen über Zäune, Mauern oder Hecken.

⁵² Wie offene amtlichen Daten.

6.2 Landnutzungsdatensatz auf OpenStreetMap Basis

Unter Nutzung einer selbst gewählten Symbolisierung nach Landnutzungsklasse sollen zunächst beide Varianten in ihrer finalen Version visualisiert werden. Die Restklasse ist jeweils Magenta dargestellt.

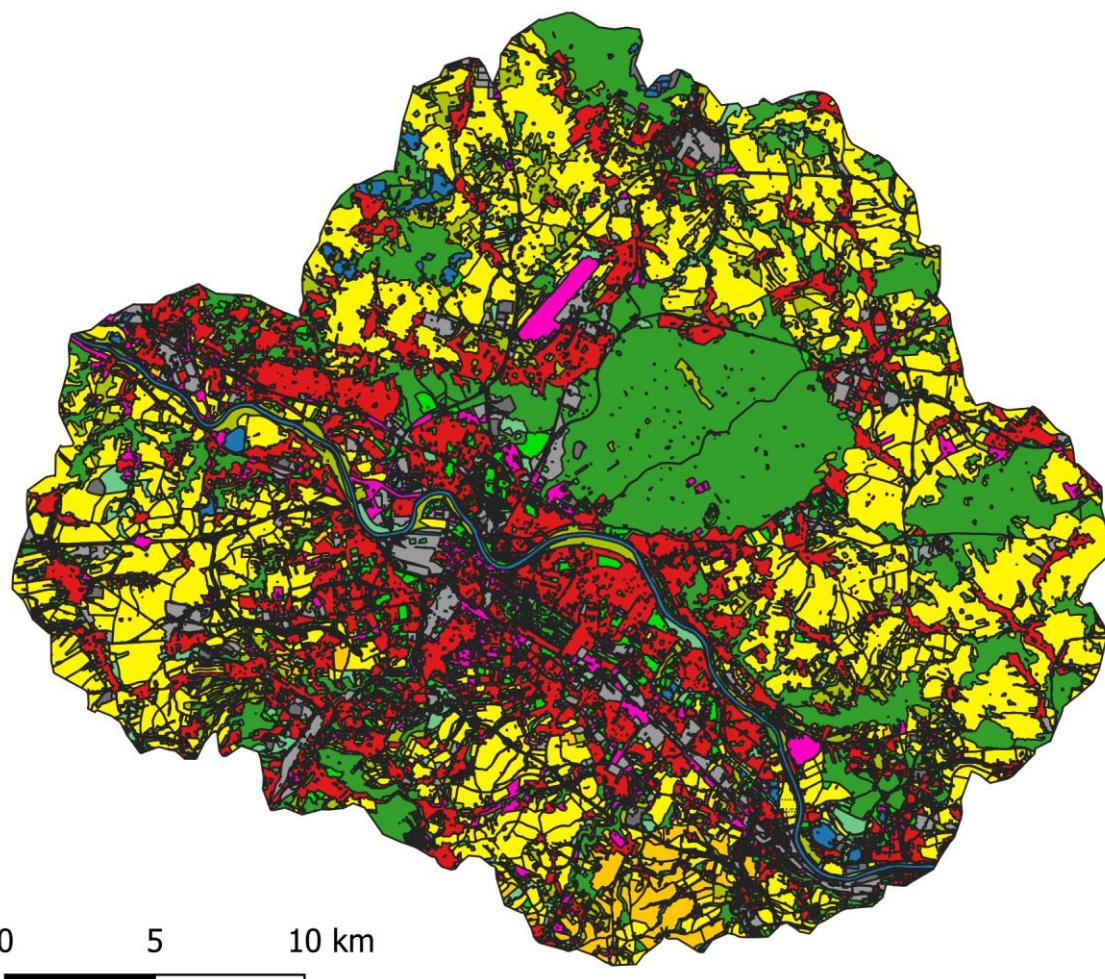


Abbildung 31 Visualisierung der finalen Version – LU-Datensatz Variante GIScience

In Variante GIScience (siehe Abbildung 31) sind deutlich einige größere Flächen in Magenta zu erkennen. Diese wurden zu Beginn durch die OSM-Daten nicht abgedeckt. Dennoch ergibt sich ein stimmiges Bild der der Beschaffenheit von Topographie und Stadtstruktur. Dies gilt ebenso für die Variante IÖR-Monitor (siehe Abbildung 32). Es sind nahezu alle Flächen einer der LU-Klassen zugewiesen. Doch auch hier ergibt sich Bedarf für Verbesserungen, siehe Flächen in Magenta in peripherer Lage im Süden und Osten der AOI.

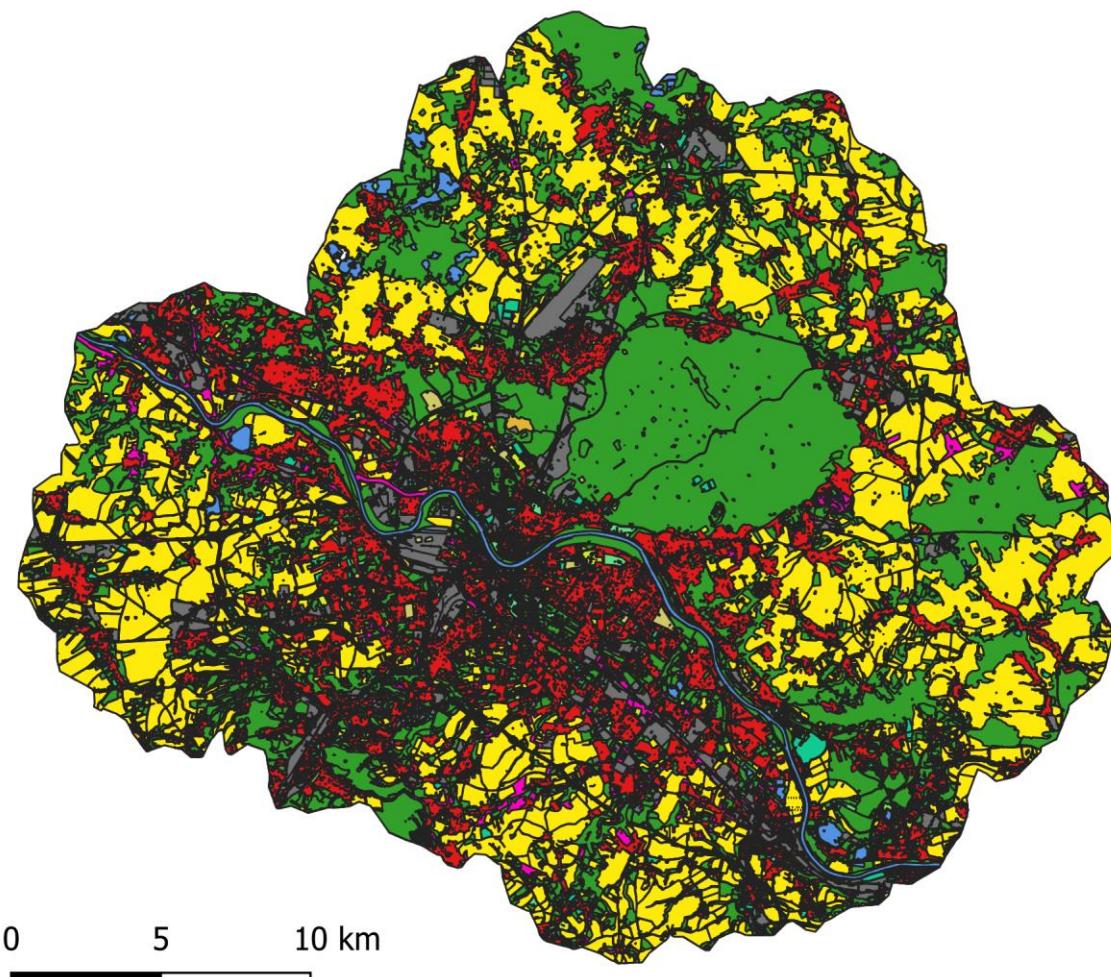


Abbildung 32 Visualisierung der finalen Version – LU-Datensatz Variante IÖR-Monitor

Die Analyse soll sich auf beschreibende Fakten zu den Landnutzungsklassen beziehen. Während für die Variante GIScience alle Klassen gelistet werden, wurde für die Variante IÖR-Monitor eine Auswahl der ersten 14 von 29 Klassen inklusive der Restklasse getroffen. Die Sortierung soll absteigend nach dem Flächenanteil relativ zur Gesamtfläche erfolgen. Es wird die Anzahl der Teilgeometrien der Multipolygone genutzt. Die zugehörige Bezeichnung (*lu_type*) für die Nummer einer LU-Klasse ist in Tabelle 9 (Variante GIScience) auf Seite 36 sowie in Tabelle 17 (Variante IÖR-Monitor) zu finden.

| lu_nr | Anz. Objekte | Anteil Objekte | Summe Fläche | Anteil Fläche |
|--------------|---------------------|-----------------------|-------------------------------|----------------------|
| 5 | 1.080 | 4,93 % | 534.949.464,55 m ² | 28,40 % |
| 8 | 2.033 | 9,28 % | 410.706.783,96 m ² | 21,81 % |
| 1 | 1.667 | 7,61 % | 353.128.909,81 m ² | 18,75 % |
| 7 | 2.667 | 12,17 % | 145.345.555,86 m ² | 7,72 % |
| 0 | 1.963 | 8,96 % | 118.425.936,50 m ² | 6,29 % |
| 9 | 6.037 | 27,55 % | 87.169.064,84 m ² | 4,63 % |
| 2 | 940 | 4,29 % | 83.678.692,47 m ² | 4,44 % |
| 4 | 3.206 | 14,63 % | 69.223.964,73 m ² | 3,68 % |
| 6 | 251 | 1,15 % | 34.390.633,59 m ² | 1,83 % |
| 13 | 1.478 | 6,75 % | 30.863.189,72 m ² | 1,64 % |
| 3 | 343 | 1,57 % | 13.912.989,47 m ² | 0,74 % |
| 11 | 94 | 0,43 % | 1.295.289,52 m ² | 0,07 % |
| 10 | 153 | 0,70 % | 294.794,61 m ² | 0,02 % |

Tabelle 15 Analyse LU-Klassen von Variante GIScience

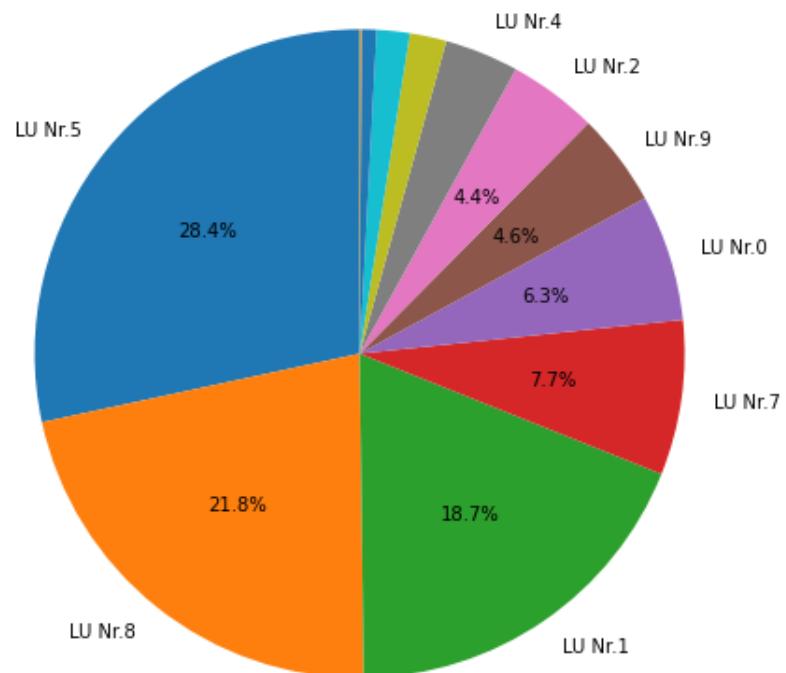


Abbildung 33 Relativer Flächenanteil pro Landnutzungsklasse – Variante GIScience

| lu_nr | Anz. Objekte | Anteil Objekte | Summe Fläche | Anteil Fläche |
|--------------|---------------------|-----------------------|-------------------------------|----------------------|
| 40 | 984 | 2,17 % | 528.427.103,66 m ² | 28,06 % |
| 50 | 3.970 | 8,76 % | 439.439.319,10 m ² | 23,33 % |
| 10 | 1.958 | 4,32 % | 339.306.182,69 m ² | 18,02 % |
| 41 | 2.907 | 6,41 % | 149.363.621,55 m ² | 7,93 % |
| 0 | 2.528 | 5,58 % | 91.976.965,70 m ² | 4,88 % |
| 13 | 4.820 | 10,63 % | 84.591.702,08 m ² | 4,49 % |
| 26 | 4.375 | 9,65 % | 52.139.526,28 m ² | 2,77 % |
| 21 | 879 | 1,94 % | 38.779.865,00 m ² | 2,06 % |
| 43 | 145 | 0,32 % | 30.898.258,17 m ² | 1,64 % |
| 70 | 1.482 | 3,27 % | 30.863.976,33 m ² | 1,64 % |
| 12 | 1.633 | 3,60 % | 17.704.230,20 m ² | 0,94 % |
| 20 | 980 | 2,16 % | 12.474.773,33 m ² | 0,66 % |
| 46 | 634 | 1,40 % | 11.094.953,26 m ² | 0,59 % |
| 30 | 39 | 0,09 % | 8.348.882,93 m ² | 0,44 % |

Tabelle 16 Analyse LU-Klassen von Variante IÖR-Monitor (Auswahl)

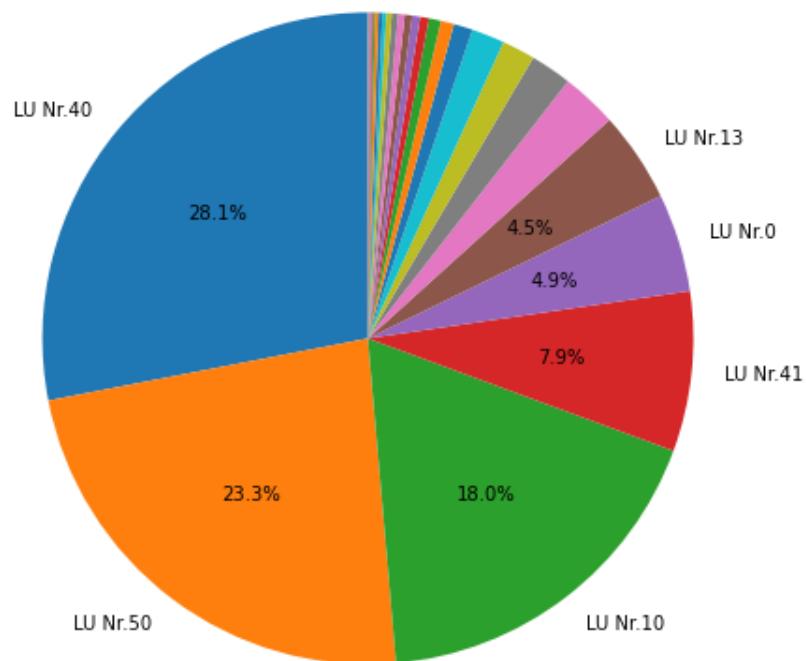


Abbildung 34 Relativer Flächenanteil pro Landnutzungsklasse – Variante IÖR-Monitor

| lu_nr | Bezeichnung |
|--------------|--|
| 0 | Restklasse |
| 1 | Straßenverkehr |
| 2 | Bahnverkehr |
| 3 | Flugverkehr |
| 10 | Wohnbau |
| 11 | Mischnutzung |
| 12 | Besondere funktionale Prägung |
| 13 | Industrie- und Gewerbeflächen |
| 20 | Park, Grünanlage |
| 21 | Kleingarten |
| 22 | Wochenendsiedlung |
| 23 | Golfplatz |
| 24 | Sonstige Sport, Freizeit, Erholung: Spielplatz, Spielbereich |
| 25 | Sonstige Sport, Freizeit, Erholung: Sportplatz, Sportanlage |
| 26 | Sonstige Sport, Freizeit, Erholung: Rasen, Gras (ohne Agrar) |
| 27 | Sonstige Sport, Freizeit, Erholung: sonstige |
| 28 | Friedhof |
| 29 | Sonstige Siedlungsfläche |
| 30 | Abbau- und Haldenfläche |
| 40 | Ackerland |
| 41 | Grünland |
| 43 | Streuobst / Obstbau |
| 45 | Weinbau |
| 46 | Sonstige Landwirtschaftsfläche |
| 50 | Laubholz / Nadelholz / Mischholz / Gehölz |
| 60 | Heide |
| 61 | Moor |
| 62 | Sumpf |
| 63 | Unland, vegetationslose Fläche |
| 70 | Fließgewässer / Stehendes Gewässer |
| 72 | Hafenbecken |
| 73 | Meer, Bodden |

Tabelle 17 LU-Klassen Variante IÖR-Monitor

| Kriterium | Variante GIScience | Variante IÖR-Monitor |
|--|---|--|
| DB-Tabelle | osm_landuse_aoi_dresden_union_hierarchy | osm_landuse_monitor_aoi_dresden_union_hierarchy2 |
| Semantik | | |
| Anzahl LU-Klassen (ohne Restklasse) | 13 (<i>12 in AOI vorkommend</i>) | 33 (<i>28 in AOI vorkommend</i>) |
| Anzahl zugewiesener OSM key-value (Summe) | 61 | 433 |
| Geometrie | | |
| Anzahl Objekte | 13 | 29 |
| \sum über alle Teilgeometrien | 21.912 | 45.336 |
| Anzahl Objekte ST_IsValid() ist „true“ | 13 | 29 |
| Anzahl Objekte ST_IsSimple() ist „true“ | 13 | 29 |
| Anzahl nach Geometriertyp | ST_MultiPolygon: 13 | ST_MultiPolygon: 29 |
| Topologie | | |
| Anzahl geometrischer Duplikate ST_Equals() | 0 | 0 |
| Überlappungen | | |
| Anzahl | 163 | 287 |
| \sum Fläche [in m ²] | $1,9755 \times 10^{-6}$ | $2,4241 \times 10^{-6}$ |
| Anteil Fläche an AOI [%] | $1,0489 \times 10^{-13}$ | $1,2871 \times 10^{-13}$ |
| Löcher (<i>innerhalb AOI</i>) | | |
| Anzahl | 0 | 0 |
| \sum Fläche [in m ²] | 0 | 0 |
| Anteil Fläche an AOI [%] | 0 | 0 |

Tabelle 18 Semantische, geometrische und topologische Prüfung – Endstand LU

Da während der topologischen Prüfung auf Löcher eine Fehlermeldung bezüglich „non-noded intersection“ geworfen wurde, musste ein Puffer von rund 8 cm (0.000001°) vor dem Vereinigen gebildet werden. Dadurch ist das Resultat unter Umständen verfälscht! Alternative Ansätze mit ST_MakeValid() oder ST_SnapToGrid() konnten dies leider nicht lösen. Geeignet erschien auch eine Bereinigung des Randes durch eine Differenzbildung zwischen dem LU-Datensatz und einer „negativen AOI“ zuzüglich eines Puffers entlang der AOI-Grenze von 0,000002° (ca. 16 cm). Topologische Fehler können so ebenfalls minimiert werden. Dennoch kann festgestellt werden, dass jeweils annähernd topologisch korrekte Landnutzungsdatensätze aus OSM-Daten abgeleitet wurden. Das Ergänzen einer Restklasse war zielführend gewesen. Der Flächenanteil der Restklasse ist jeweils plausibel, repräsentiert er doch die Fläche der Löcher im Datensatz vor der Verarbeitung. Die höhere Anzahl an insgesamt zugewiesenen OSM key-value in Variante IÖR-Monitor ermöglichen eine deutlich höhere Anzahl an Polygonen (vergleiche Summe aller Teilgeometrien), und damit eine detailliertere Repräsentation der realen Welt. Es verbleibt Verbesserungspotenzial in der semantischen Zuweisung der OSM key value. Die technischen Abläufe im Hintergrund einer PostGIS in Bezug auf gerundete Koordinaten oder Verschneidungsmechanismen können noch effektiver verstanden und realisiert werden.

6.3 Ground-Truth

Das Ergebnis der Kartierung wird in den folgenden Karten visualisiert. Als Barrieren klassifizierte Linien sind in Rot, keine Barrieren in Grün und Linien ohne Daten („nodata“) in Blau dargestellt.

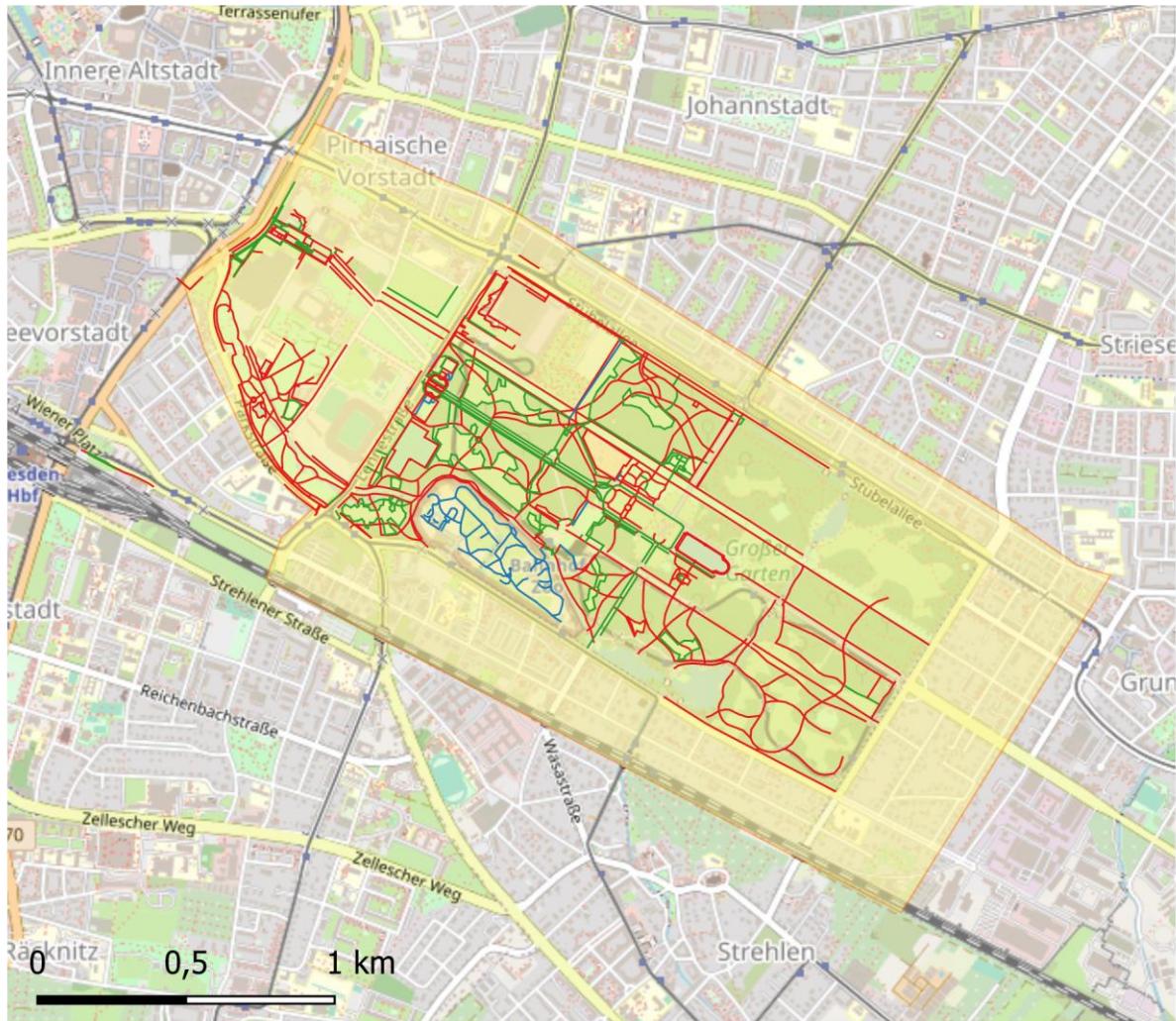


Abbildung 35 Erfassungsstand Ground-Truth im Großen Garten

Nicht erreichbare Objekte oder unklare Fälle wurden als „nodata“ gekennzeichnet. Die beiden Gebiete des Ground-Truths wurden nicht vollständig kartiert. Der hier gezeigte Stand entspricht der Analysegrundlage für diese Arbeit. Hauptsächlich wurde innerhalb der beiden Gebiete kartiert. In seltenen Fällen wurde davon abgewichen, um gezielt noch LU-Wechsel ergänzen zu können (Friedhöfe, Bahnverkehr, ...).

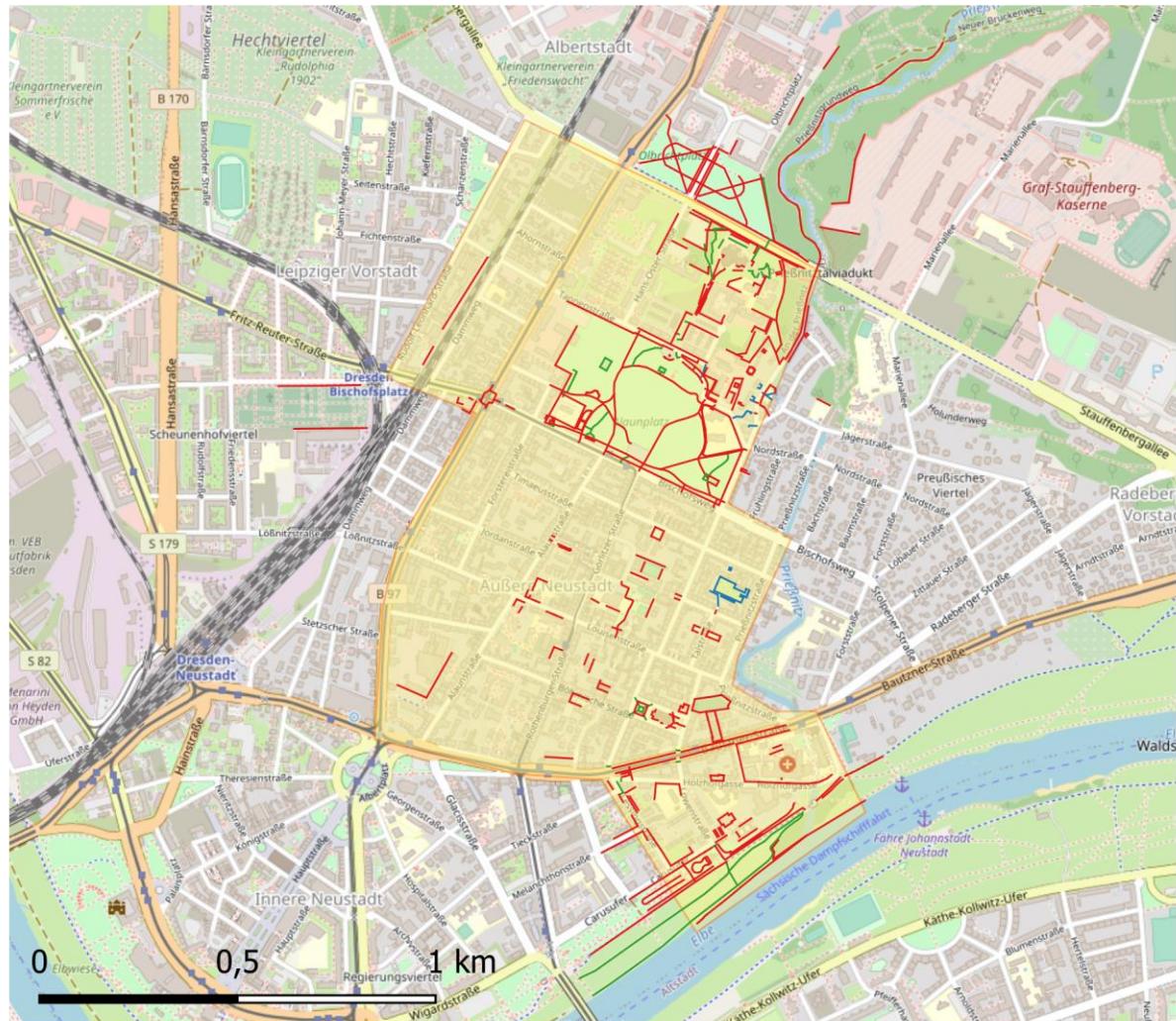


Abbildung 36 Erfassungsstand Ground-Truth in Dresden-Neustadt

| Gebiet | Wege | | LU-Wechsel | |
|---------------|---------------|----------------------------------|-------------------|----------------------------------|
| | Anzahl | Σ Länge | Anzahl | Σ Länge |
| Großer Garten | 297 | 64.682,28 m | 1.145 | 41.429,90 m |
| Neustadt | 96 | 12.802,35 m | 548 | 19.718,66 m |
| Sonstige | 15 | 4.816,81 m | 27 | 3.057,84 m |
| Summe | 408 | 82.301,44 m | 1.720 | 64.206,40 m |

Tabelle 19 quantitative Analyse des Ground-Truths (inklusive „nodata“)

Die Erfahrungen zeigten, dass das Vorgehen prinzipiell geeignet ist Barrieren zu kartieren. Die Interpretation ist immer auch subjektiven Einflüssen vor Ort unterworfen. Die genutzte Definition einer Barriere bezüglich der Zielgruppe und der Aktivitäten (siehe 4.4) stellte eine

wichtige Einschränkung und Vereinfachung für die Erfassung dar. Bei Wegen wurden oft die geschätzte Wegebreite und die Oberflächenbeschaffenheit für die Bewertung herangezogen. Diese hätten auch als zusätzliche Attribute erfasst werden können. Bei Wechsel der Landnutzung sorgten existierende Mauern, Zäune oder Hecken für Klarheit. Barrieren können auch niedrige Zäune, Stufen, Reihen aus Pollern oder Hänge sein. So könnte ein Digitales Oberflächenmodell ebenfalls zur Ableitung von Barrieren genutzt werden⁵³. Blumenreihen und Beete wurden ebenfalls als Barriere kartiert (siehe Abbildung 37). Dennoch gab es auch unklare Begebenheiten vor Ort. Stellt der Übergang von einer Rasenfläche auf einen breiten und wenig genutzten Gehweg eine Barriere dar? Könnte die Frequentierung von Wegen durch die Bevölkerung ein weiterer Indikator bezüglich der Bewertung sein? Der Übergang von Wald auf Wiese wurde im Falle von freistehenden Bäumen nicht als Barriere gewertet. War eine dichte Vegetation zu erkennen, wurde dies als Barriere erfasst. Das zusätzlich erfasste Attribut „type_bar“ zur temporalen Dimension einer Barriere wurde oft auf „always“ gesetzt. Es kam eher für Eingänge und Durchlässe mit Öffnungszeiten zum Einsatz, da ein Zaun eines Spielplatzes oder Zoos auch während der Öffnungszeiten eine Barriere darstellt. Ein Spannungsfeld existiert zu flächenhaft ausgeprägten Konzepten wie Zugänglichkeit oder Eigentumsstatus⁵⁴. Im Ground-Truth erfasst wurden stattdessen linienhafte Objekte.



Abbildung 37 Barrieren in der Realität: Beete (links), Holzpoller (rechts)

⁵³ Eventuell können Barrieren aus DOM ableiten durch Berechnen von Linien mit maximaler Hangneigung?

⁵⁴ Bei Flächen könnte ein zeitweise geöffneter Zugang über die Zugänglichkeit der ganzen Fläche entscheiden.

6.4 Barrieren-Wahrscheinlichkeit

Die Clustermap (siehe Abbildung 38) zeigt die Anzahl an erfassten Objekten pro LU-Wechsel. Es liegen Daten zu 61 der insgesamt bis zu 406 ($(29 \times 29 - 29) / 2$) theoretisch möglichen Landnutzungswechsel vor. Weitere Daten sind in der Anlage 5 zu finden.

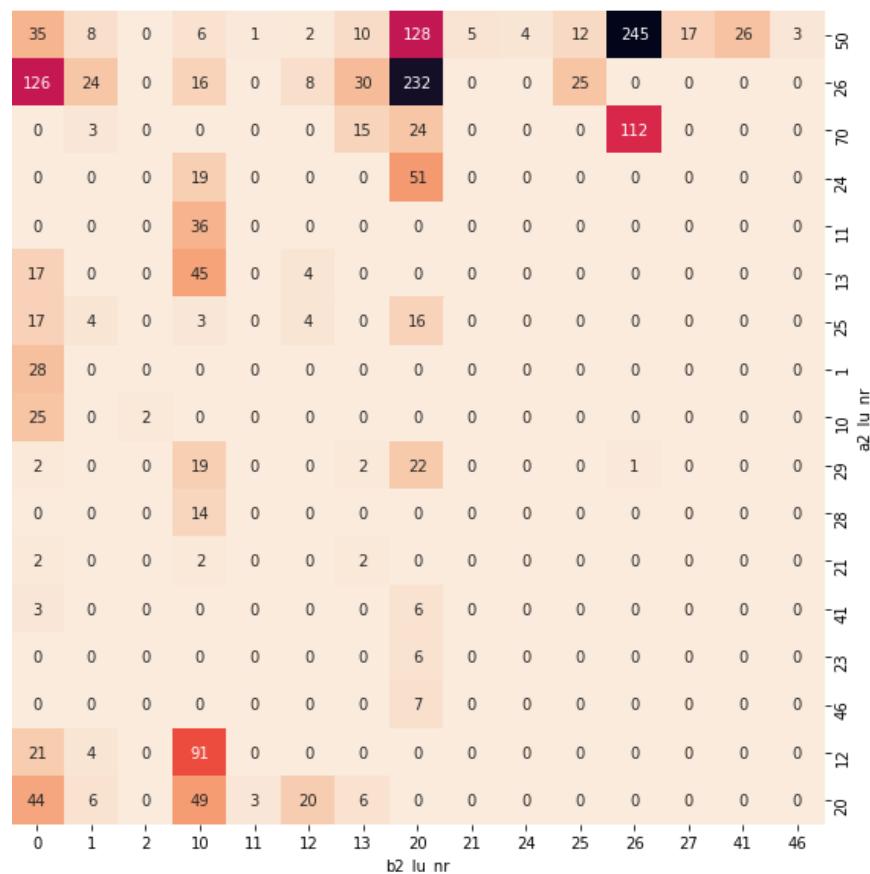


Abbildung 38 Clustermap LU-Wechsel nach erfasster Anzahl an Objekten (Ground-Truth)

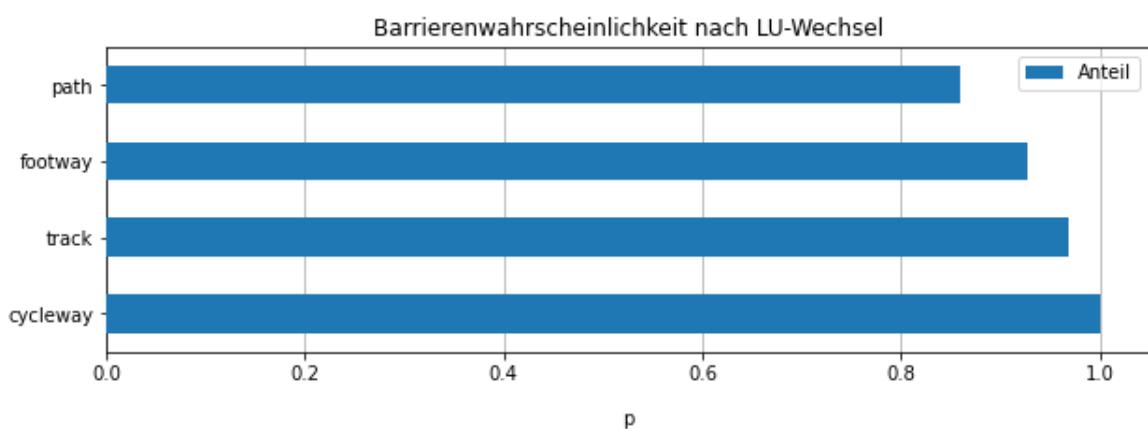


Abbildung 39 Barrierenwahrscheinlichkeit nach Wegetyp

Wege verfügen über eine hohe Barrierenwahrscheinlichkeit von mindestens 0,8.

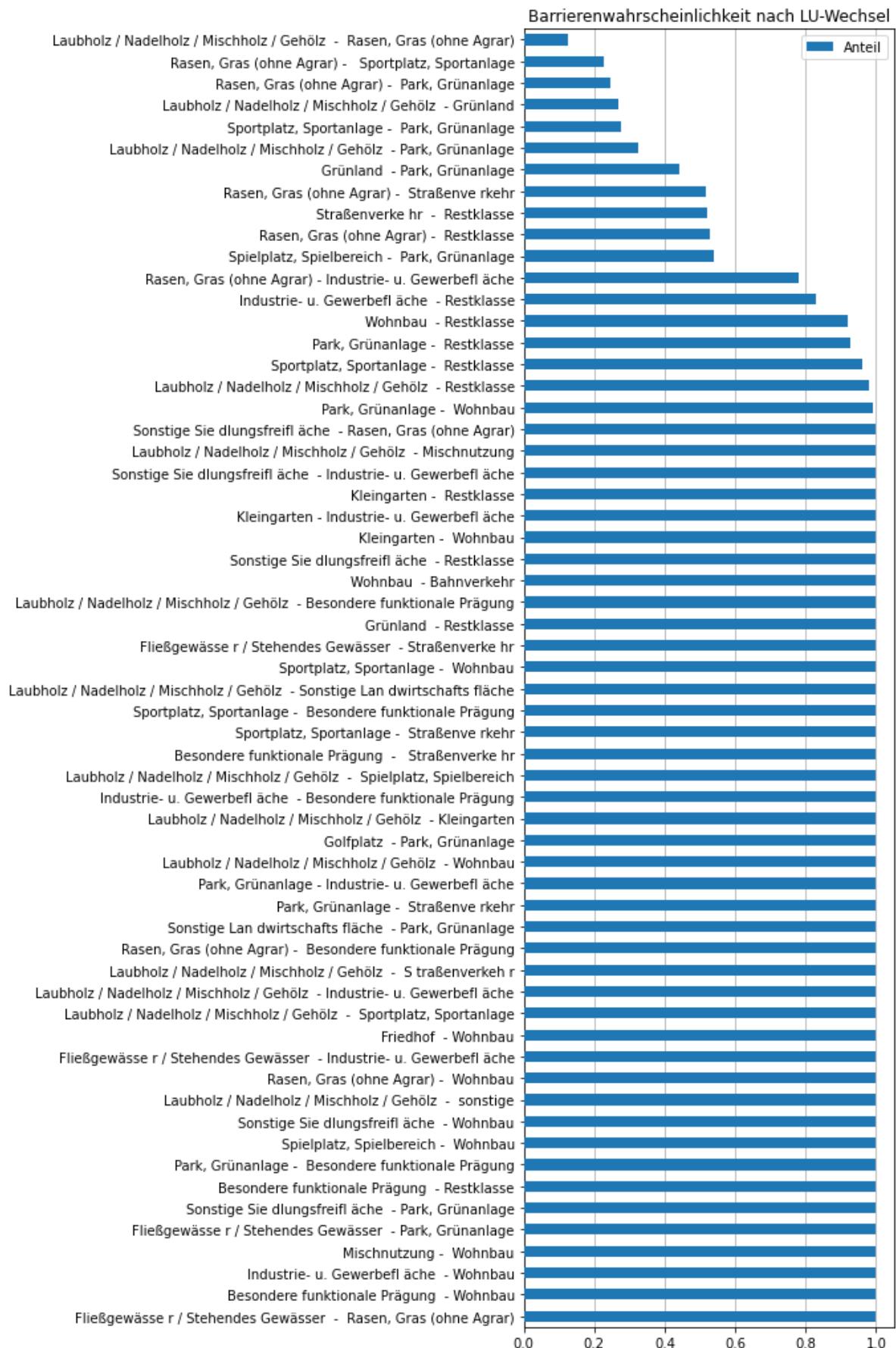


Abbildung 40 Barrierenwahrscheinlichkeit nach LU-Wechsel

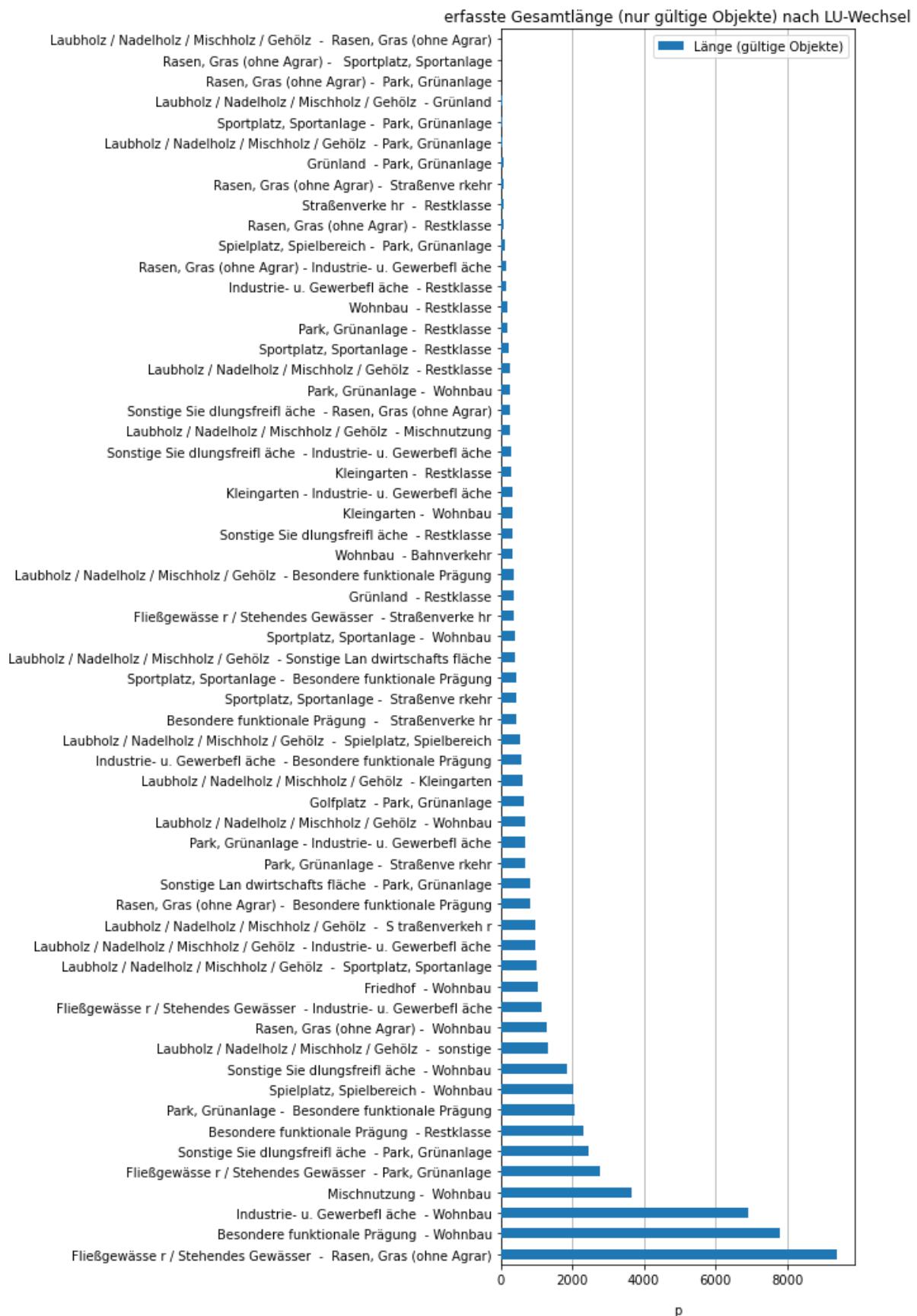


Abbildung 41 LU-Wechsel nach erfasster Gesamtlänge in Metern

6.5 Prozess der Vermaschung

Um die Funktion in Abhängigkeit der Barrierenwahrscheinlichkeit sichtbar zu machen, sollen in Tabelle 20 die Liniensegmente gruppiert nach ihrer Wahrscheinlichkeit in absteigender Folge aufgezeigt werden (bezogen auf das Analysegebiet⁵⁵).

| „likelihood“ (p) | Anzahl Linien | Gesamtlänge in Metern | „origin“ |
|-------------------|---------------|-----------------------|--|
| 1 | 64.526 | 3.098.520,55 | bbox, lu_change, osm_barrier, osm_railway, osm_street, osm_trail, osm_waterway |
| 0,990255033057788 | 3.356 | 67.108,81 | lu_change |
| 0,979001611534504 | 1.607 | 65.560,92 | lu_change |
| 0,9685078727345 | 387 | 120.348,38 | osm_trail |
| 0,95929959488236 | 259 | 9.907,07 | lu_change |
| 0,926919307890808 | 7.913 | 585.503,63 | osm_trail |
| 0,926621176395434 | 924 | 29.970,63 | lu_change |
| 0,919942497713395 | 2.441 | 93.754,53 | lu_change |
| 0,860162319296291 | 2.253 | 327.381,89 | osm_trail |
| 0,827298874079449 | 2.369 | 70.418,16 | lu_change |
| 0,779414439059279 | 1.524 | 26.231,78 | lu_change |
| 0,540461262830253 | 565 | 4.815,49 | lu_change |
| 0,528455791566701 | 7.433 | 137.501,85 | lu_change |
| 0,518710018612371 | 1.797 | 41.884,13 | lu_change |
| 0,515109565292657 | 520 | 10.617,19 | lu_change |
| 0,440413747338458 | 41 | 2.602,18 | lu_change |
| 0,325306714095088 | 447 | 21.839,25 | lu_change |

⁵⁵ POLYGON ((13.7131 51.0315, 13.8032 51.0315, 13.8032 51.1, 13.7131 51.1, 13.7131 51.0315))

| | | | |
|-------------------|-------|-----------|-----------|
| 0,274621122886677 | 163 | 4.044,33 | lu_change |
| 0,266586080582736 | 559 | 15.119,47 | lu_change |
| 0,245770704222081 | 841 | 19.612,93 | lu_change |
| 0,227521791867359 | 218 | 5.719,63 | lu_change |
| 0,125297859170249 | 1.066 | 37.809,91 | lu_change |

Tabelle 20 Anzahl und Länge der Linien in Abhängigkeit der Barrierenwahrscheinlichkeit

Die Konzentration vieler Barrierentypen am Wert $p=1$ ist plausibel (gemäß den Annahmen in 4.2). Interessant ist, dass beinahe das vollständige Intervall $[0, 1]$ genutzt wird. Abbildung 42 visualisiert zunächst Straßen, Wege, Bahnstrecken und Wasserläufe, während in Abbildung 43 Linien vom Typ Barriere und LU-Wechsel gezeigt werden. Es folgt eine statistische Auswertung bezüglich verschiedener erzeugter Varianten.

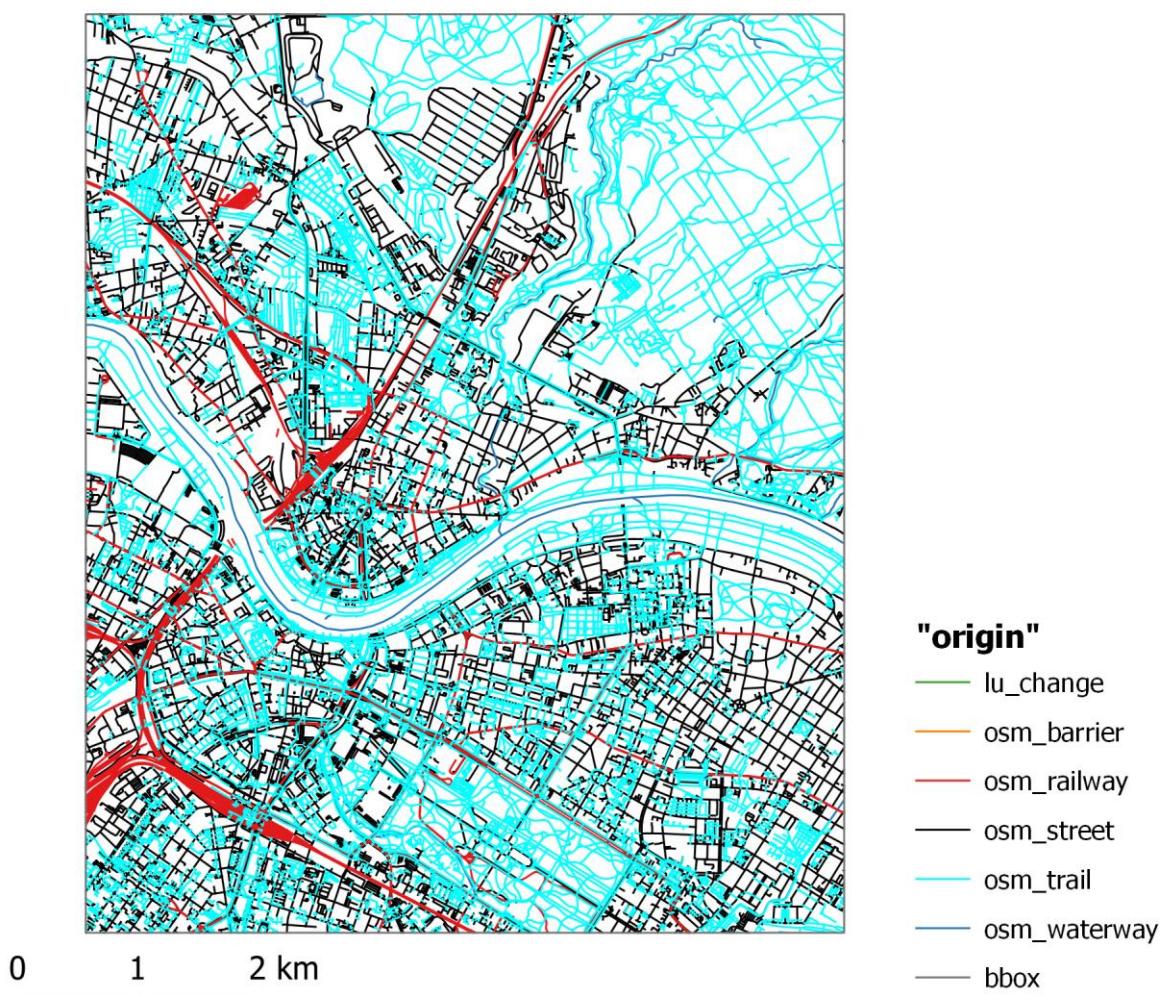


Abbildung 42 Linien im Linien-Pool (Straßen, Wege, Bahnverkehr, Wasserwege)

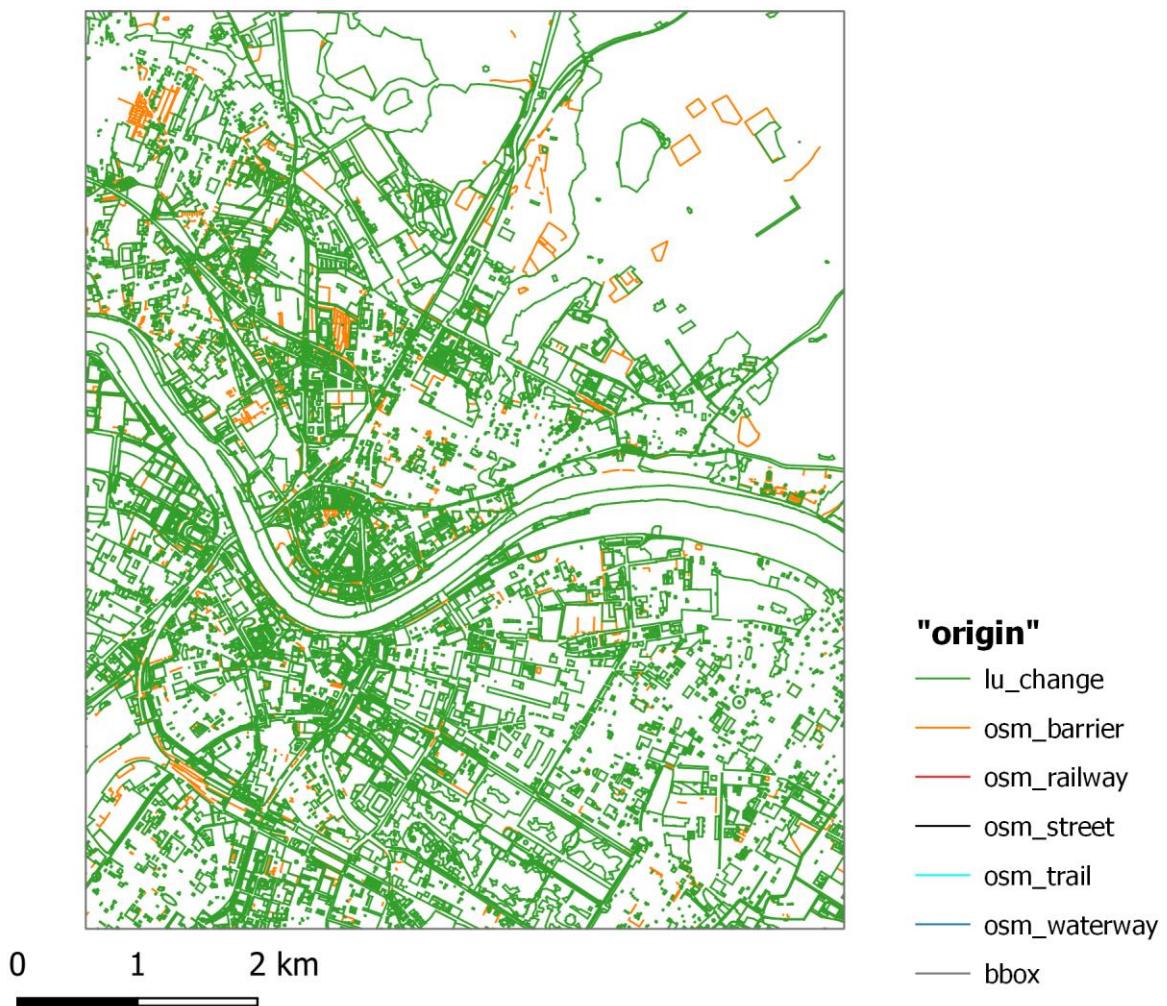


Abbildung 43 Linien im Linien-Pool (Barrieren und LU-Wechsel)

| Variante mit Bedingung | Anzahl Polygone | Fläche | | | |
|------------------------|-----------------|-----------------------------|-------------------------------|--------------------------|--------------------------|
| | | ϕ in m ² | σ in m ² | Max in m ² | Min in m ² |
| $p \geq 0,0$ | 30.371 | 3.654,89 | 16.351,58 | 1.114.481,59 | 0,00 |
| $p \geq 0,2$ | 30.009 | 3.698,98 | 16.492,86 | 1.114.481,59 | 0,00 |
| $p \geq 0,4$ | 29.331 | 3.784,48 | 16.807,43 | 1.114.481,59 | 0,00 |
| $p \geq 0,6$ | 27.162 | 4.086,69 | 17.501,43 | 1.114.481,59 | 0,00 |
| $p \geq 0,8$ | 26.821 | 4.138,64 | 17.641,26 | 1.114.481,59 | 0,00 |
| $p = 1,0$ | 13.416 | 8.421,20 | 134.651,63 | 14.499.997,92 | 0,00 |

Tabelle 21 Auswertung Vermaschung nach Polygonen

Ab einer Barrierenwahrscheinlichkeit gleich eins entfallen viele Barrieren (vergleiche Tabelle 20 und Tabelle 21). Die Folge ist eine deutlich niedrigere Anzahl an gebildeten Polygonen.

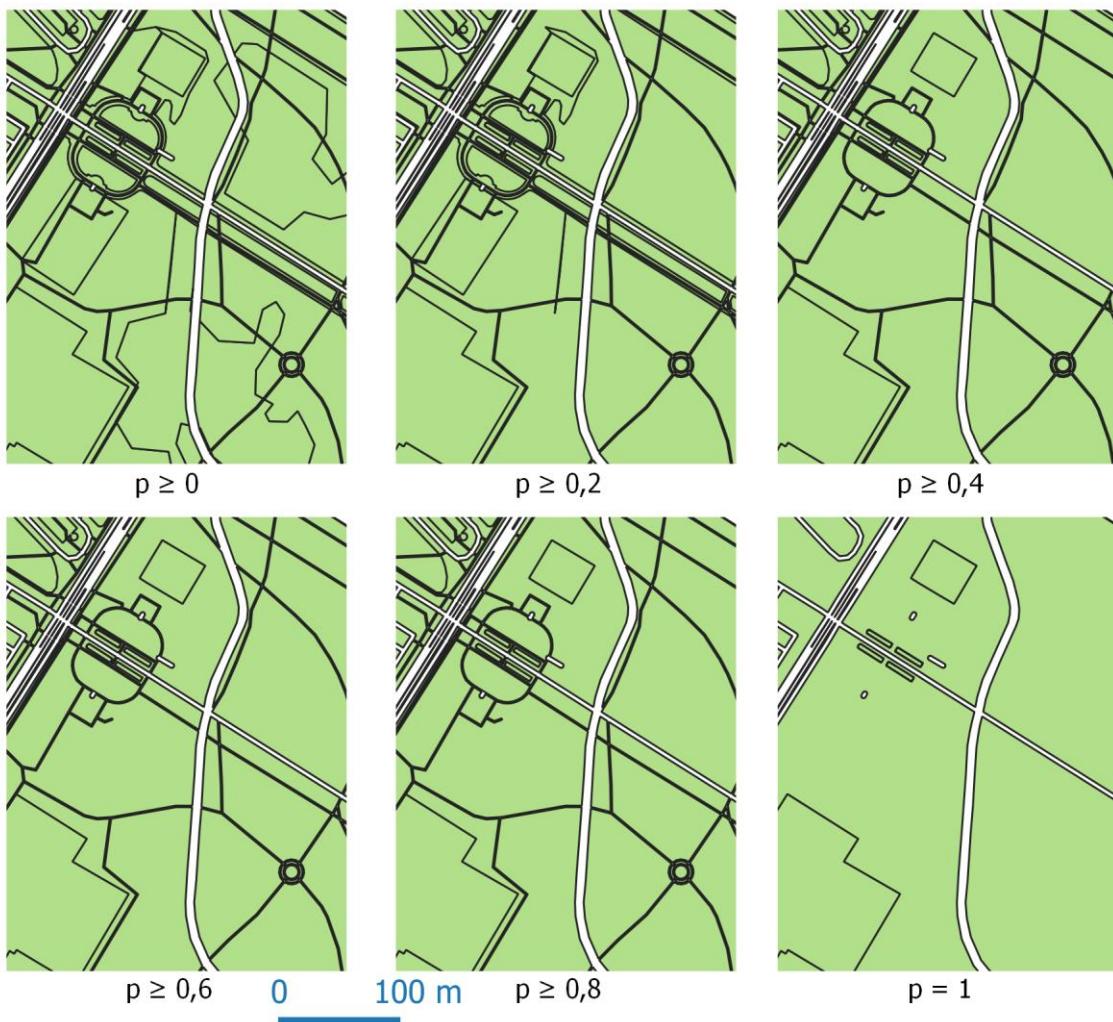


Abbildung 44 Ergebnis der Vermaschung nach Barrierenwahrscheinlichkeit – Großer Garten Dresden, westlicher Eingang

Die Kartenserie (siehe Abbildung 44) verdeutlicht das Prinzip unterschiedlich dimensionierter Zielgeometrien (vergleiche 3.1). Werden bei einer Barrierenwahrscheinlichkeit größer gleich 0 auch Wechsel zwischen Wald und Wiese als Barrieren genutzt, so nimmt die Anzahl an gebildeten Polygonen mit steigender Barrierenwahrscheinlichkeit stetig ab. Auffällig ist eine zu groß angesetzte Pufferweite für sonstige Bahnstrecken bzw. Gartenbahnen (von Norden nach Süden verlaufend), welche dadurch im Vergleich zur Realität zu breit ist. Dies bietet Potenzial für Verbesserungen. Das Prinzip wird ebenfalls in Abbildung 45 verdeutlicht.

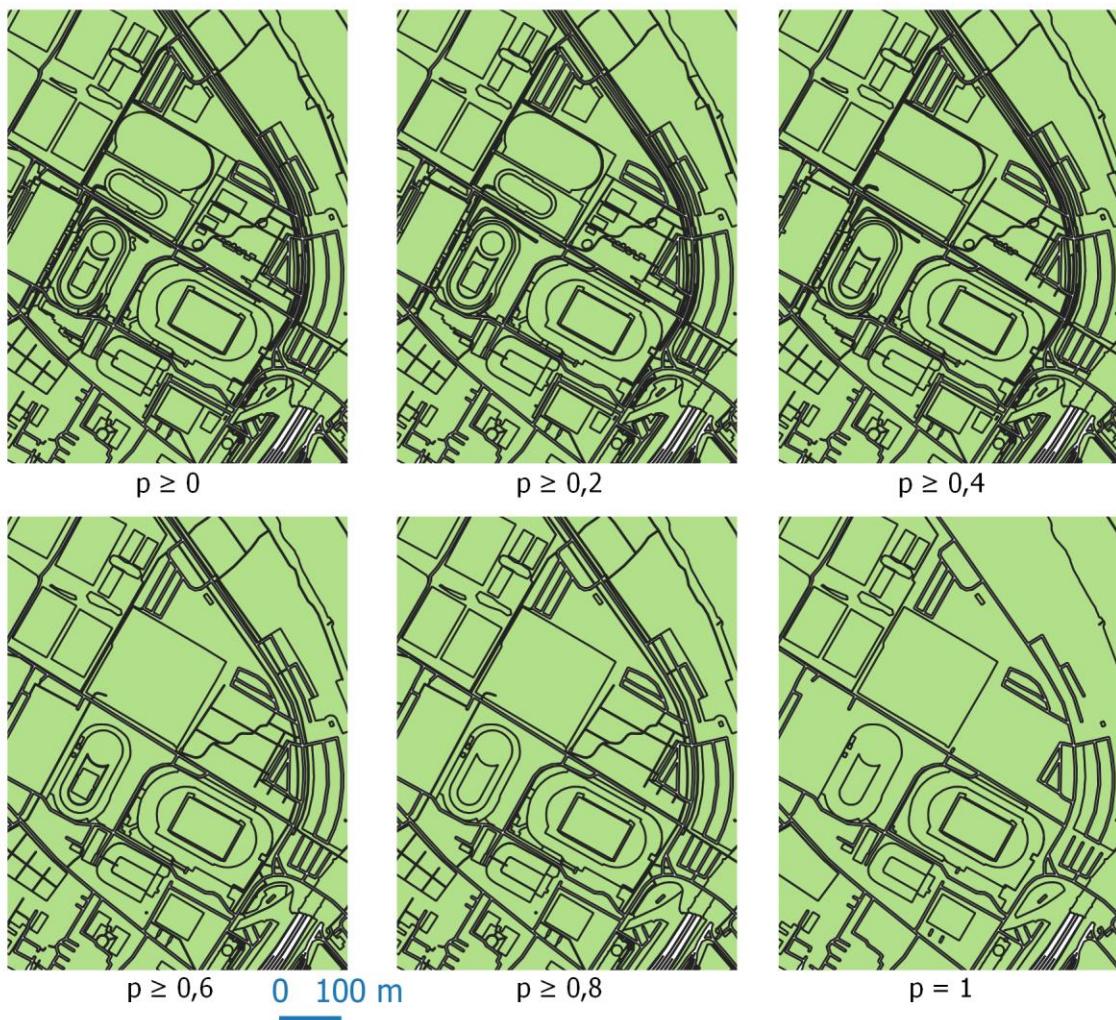


Abbildung 45 Ergebnis der Vermischung nach Barrierenwahrscheinlichkeit – Sportanlagen Ostragehege, Dresden

6.6 Modelle für Zugänglichkeit und Grünheit

Das Modell „Zugänglichkeit“ beruht auf Basis der kommunalen Flächen. Für das Modell „Grünheit“ wurden ALKIS-Daten genutzt. Gezeigt werden die Scores verschiedener Ansätze des maschinellen Lernens hinsichtlich unterschiedlicher Kombinationen an genutzten Input-Variablen. Pro Verfahren ist jeweils die optimalste Variante fett hinterlegt. Ein Score von 1,0 wäre das Optimum.

| Sitzbänke (in_1) | Abfallbehälter (in_4) | Öffentliche Toiletten (in_8) | Öffentliches Internet (in_9) | Score |
|--|--------------------------|------------------------------------|------------------------------------|-----------------|
| Logistische Regression | | | | |
| X | X | X | X | 0,767433 |
| X | X | X | | 0,767433 |
| X | X | | X | 0,767096 |
| X | | X | X | 0,766424 |
| | X | X | X | 0,767321 |
| Support Vector Classifier (SVC) | | | | |
| X | X | X | X | 0,767545 |
| X | X | X | | 0,767545 |
| X | X | | X | 0,767321 |
| X | | X | X | 0,766200 |
| | X | X | X | 0,767321 |

Tabelle 22 Modell Zugänglichkeit – Scores linearer Verfahren

Die logistische Regression wie auch SVC bieten als lineare Trennungsverfahren noch keine gute Qualität für das Modell „Zugänglichkeit“. Die jeweils optimalste Variante wäre durch Weglassen von „in_9“ erreicht, da dies keine Verschlechterung des Ergebnisses bedeutet. Das Modell zur Grünheit erscheint deutlich präziser für die Vorhersage zu sein (Score von über 0,9), als das Modell zur Zugänglichkeit (Score etwas um 0,77). Gründe könnten eine heterogene Struktur von den kommunalen Flächen sein.

| Sitzbänke (in_1) | Abfallbehälter (in_4) | Öffentliche Toiletten (in_8) | Öffentliches Internet (in_9) | Score |
|--|--------------------------|------------------------------------|------------------------------------|-----------------|
| SVC mit Radial Basis Function (C=1E6, gamma=1.) | | | | |
| X | X | X | X | 0,772197 |
| X | X | X | | 0,771804 |
| X | X | | X | 0,771637 |
| X | | X | X | 0,770123 |
| | X | X | X | 0,769170 |

Tabelle 23 Modell Zugänglichkeit – Scores nicht-linearer Verfahren

Für das Verfahren „Radial Basis Function“ wurden verschiedene Werte für C und gamma eingesetzt. Über das Maximum von rund 0,77 führte dies aber nicht hinaus.

| Sitzbänke (in_1) | Picknicktische (in_2) | Einzelbäume (in_3) | Abfallbehälter (in_4) | Score |
|--|--------------------------|-----------------------|--------------------------|-----------------|
| Logistische Regression | | | | |
| X | X | X | X | 0,922867 |
| X | X | X | | 0,922779 |
| X | X | | X | 0,922867 |
| X | | X | X | 0,922867 |
| | X | X | X | 0,922691 |
| Support Vector Classifier (SVC) mit linearem Kernel und C=1 | | | | |
| X | X | X | X | 0,926018 |
| X | X | X | | 0,925432 |
| X | X | | X | 0,926018 |
| X | | X | X | 0,926018 |
| | X | X | X | 0,926018 |

Tabelle 24 Modell Grünheit – Scores linearer Verfahren

6.7 Laufzeit-Analyse

Eine grobe Abschätzung der Laufzeit soll erfolgen. Dafür wird zunächst die Laufzeit bei der Generierung der Vermaschung (in ihren Varianten) dargestellt (siehe Tabelle 25).

| Bedingung | Anzahl Linien | \sum Länge Linien in Metern | Anzahl Polygone generiert | Gesamlaufzeit in Sekunden (Polygonbildung + Differenz mit Puffer) |
|------------------|----------------------|---|----------------------------------|---|
| $p \geq 0,0$ | 101.209 | 4.796.272,71 | 30.371 | 3.536 (774 + 2.762) |
| $p \geq 0,2$ | 100.143 | 4.758.462,80 | 30.009 | 3.510 (771 + 2.739) |
| $p \geq 0,4$ | 97.915 | 4.692.127,20 | 29.331 | 3.458 (760 + 2.698) |
| $p \geq 0,6$ | 87.559 | 4.494.706,35 | 27.162 | 3.333 (625 + 2.708) |
| $p \geq 0,8$ | 86.035 | 4.468.474,57 | 26.821 | 3.281 (719 + 2.562) |
| $p = 1,0$ | 64.526 | 3.098.520,55 | 13.416 | 1.028 (326 + 702) |

Tabelle 25 Auswertung Vermaschung nach Laufzeit

Der Flächeninhalt des Analysegebietes (Bounding Box) beträgt 121.703.271,37 Quadratmeter. Die AOI Dresden weist 1.883.385.333,37 m² auf. Es ist um den Faktor 15,48 größer. Unter Annahme einer gleichen Dichte an Linien (in der Realität wird sie vermutlich in der Peripherie abnehmen) und eines linearen Laufzeitverhaltens⁵⁶ der Verarbeitungsschritte könnte für den Fall $p \geq 0,0$ eine Abschätzung der maximalen Laufzeit von rund 54.720 Sekunden berechnet werden. Dies sind in 18,24 Stunden. Zu beachten sind außerdem die technischen Leistungsparameter des genutzten Computers.

⁵⁶ Es müsste geprüft werden, um nicht auch logarithmische Funktionen oder Potenzfunktionen das Verhalten der Laufzeit besser beschreiben können.

6.8 Übertragbarkeit

Die gezeigten Ergebnisse wurden anhand des Untersuchungsgebietes Dresden oder von Teilbereichen erhoben. Es soll die Übertragbarkeit auf andere Gebiete diskutiert werden.

Prinzipiell ist es möglich, mit den Geodaten von OpenStreetMap global zu arbeiten. Sie werden weltweit von Freiwilligen erfasst und haben das key-value-basierte Datenschema (fast immer mit derselben Semantik) als gemeinsame Grundlage. Dennoch ist der Datensatz heterogen. Klimazonen und unterschiedliche Landbedeckungen bringen automatisch eine daran angepasste Verwendung von key-value mit sich. So werden sich regenreiche Gebiete von trockenen Gebieten auch bezüglich enthaltener Grünflächen unterscheiden⁵⁷. Dazu kommen kulturelle Einflüsse, Unterschiede in der Struktur einer Stadt und deren Bebauung und Verteilung von POIs etc. Auch kann das Herangehen beim Kartieren abweichend sein. Lokale Communities des OSM-Projektes setzen sich ihre eigene Agenda und beeinflussen auch damit den Datensatz für ihr Gebiet. Für diese Arbeit wurde der Versuch unternommen die Datenextrakte möglichst weitreichend für andere urbane Gebiete mitzudenken, wie zum Beispiel küstennahe Städte und den nur dort auftretenden key-value. Dennoch steht das in dieser Arbeit gesammelte Wissen eher für eine mitteleuropäische Stadt im Hinterland. Zu in der AOI Dresden fehlenden Objektarten kann unter Umständen keine Aussage zur Barrirenwahrscheinlichkeit getroffen werden. Eine Anpassung der Eingabe-Variablen für die Modelle ist eventuell nötig, um sie für andere Gebiete zu optimieren. Das Verständnis für die kulturellen Besonderheiten würde Erleichterung bringen. Bezüglich der Koordinatenreferenzsystem erleichtert das weltweite WGS84 (EPSG-Code 4326) ein globales Arbeiten. Für genaue Berechnungen von metrischen Maßen wie Längen oder Flächeninhalten ist allerdings in Abhängigkeit der Lage beziehungsweise des Landes oder Kontinentes ein passendes Koordinatenreferenzsystem für die Transformation auszuwählen. Sollten aufgrund der kulturellen Unterschiede die Referenzdatensätze aus Dresden nicht geeignet sein, müssten andere Daten – auch vor Ort – akquiriert werden. Globale Datensätze wie WikiData, Daten aus SocialMedia oder Fernerkundungsdaten könnten hier eine Grundlage bieten. Beim Import mit osm2pgsql ist auf eine zielführende Größe (Speicherplatz) des Datenextraktes zu achten. Gegebenenfalls bietet sich eine vorherige Auswahl gemäß tatsächlich benötigter key-value an, um eine erneute Reduktion zu realisieren. Für große Polygone können sich beim Import unter Nutzung einer Bounding Box falsche gerade Linien durch das automatische Entfernen von (außerhalb der Bounding Box gelegenen) Stützpunkten ergeben.

⁵⁷ Ob eine Stadt von Wüste oder Regenwald umgeben ist, kann sich auch in der Kartierung des Stadtgebietes wiederfinden.

7 Zusammenfassung und Ausblick

Die Fragestellungen konnten in verschiedenen Teilkomplexen bearbeitet werden. Es sind mehrere Zwischenprodukte und Ergebnisse in dieser Arbeit erstellt worden. Prinzipiell sind Daten aus OpenStreetMap für ein Ableiten von Barrieren und Grünflächen gut geeignet. Die erzeugten Landnutzungsdatensätze können die Grundlage für weitere Analysen sein. Der Ground-Truth lieferte erste Ergebnisse zum Vorliegen einer Barriere. Eine quantitativ breiter aufgestellte Kartierung könnte das empirische Fundament vergrößern und die Barrierenwahrscheinlichkeiten belastbarer machen. Dennoch ließen sich plausible und wichtige Erkenntnisse generieren. Sie ermöglichen das Demonstrieren der Polygonbildung in Abhängigkeit einer gewählten Barrierenwahrscheinlichkeit. Die dabei entstandenen Varianten können auch mit den individuellen Aktionsräumen bei der Nutzung von Grünflächen in Verbindung gebracht werden. Eine Nachbearbeitung der Vermischung hinsichtlich von Minimaldimensionen wie zu kleine Polygone oder zu schmale Polygone würde eine Verbesserung erzielen. Zum Einsatz könnten das Kriterium der längsten gemeinsamen Grenze (zur Auflösung kleiner Polygone), oder morphologische Operationen (zur Reduktion von schmalen „Schläuchen“ in Polygonen) kommen. Die aufgestellten Modelle zur Grünheit und zur öffentlichen Zugänglichkeit gehen in eine richtige Richtung. Verbesserung könnte eine Ausweitung in Frage kommender OSM key-value auch auf nicht Grünflächen-bezogene Objekte wie weitere POIs sein. Eine Nutzung einer Pufferfläche als Berechnungsgrundlage der Anzahl könnte die angrenzende Nachbarschaft inkludieren und würde sich für weitere Erprobungen anbieten („in der Nähe eines Restaurants oder einer öffentlichen Toilette könnten Flächen öffentlich zugänglich sein“). Zu berücksichtigen ist auch die quantitative Erfassung Grünflächen-bezogener key-value im Untersuchungsgebiet. Eine geringe Vollständigkeit bringt eine größere Unsicherheit im Modell mit sich. In dem Suchen nach Mustern in den topologischen Beziehungen der OSM key-value bieten sich Chancen für einen weiteren Erkenntnisgewinn. Im Sinne von freiwillig kartierten Daten (VGI) und „Citizen Sciene“-Projekten kann auch die Bevölkerung mit einbezogen werden und ein Teil von wissenschaftlichen Projekten werden.

Glossar

Ground-Truth Eine Kartierung in-situ, um die Realität vor Ort zu erfassen.

key-value Wird im OpenStreetMap (OSM) Datenschema ähnlich dem Begriff „Objektart“ verwendet. Zum Beispiel „highway“ = „path“ (für einen Pfad) oder „amenity“ = „bench“ (Sitzbank).

pgAdmin Benutzerfreundliche Web-Anwendung zum Verwalten von und Zugreifen auf eine PostgreSQL Datenbank

PostGIS Eine Erweiterung für Postgresql, welche unter anderem Geometriertypen und räumliche Funktionen bereitstellt.

PostgreSQL Ein quelloffenes und weit verbreitetes Datenbankmanagementsystem auf Basis der SQL-Syntax.

QField Eine App für mobile Endgeräte, welche für Datenerfassung im Gelände geeignet ist. Es fungiert als mobile Version von QGIS und verwendet identische Renderer-Bibliotheken. Einstellungen werden weitestgehend am Desktop-Computer vorgenommen und als gepacktes Projekt auf das mobile Endgerät geladen.

QGIS Ein quelloffenes Geoinformationssystem. Wird durch eine breite Community und Firmen getragen und weiterentwickelt. Bietet zahlreiche Geowerkzeuge und Erweiterungsmöglichkeiten.

Simple Features Ist ein Standard, der durch das Open Geospatial Consortium definiert wurde. Es werden Anforderungen an universelle Geometriertypen aufgestellt, um den Austausch und die Weiternutzung in der Geoinformatik-Branche zu vereinheitlichen. (siehe (Andrae/Zipf 2013))

WebApp Eine Untergattung einer App, welche leichtgewichtig ohne Installation auf dem Endgerät genutzt werden kann. Sie wird direkt im Internetbrowser ausgeführt. Die App kann dabei nicht auf alle Gerätesensoren zugreifen (wie das bei einer nativen App möglich wäre).

Quellenverzeichnis

- Andrae, Christine/Zipf, Alexander (2013): Simple features: praxisnahe Standards für einfache Geoobjekte in Datenbanken und GIS. (= OpenGIS essentials / Christine Andrae ... (Hrsg.)). Berlin Offenbach: Wichmann.
- ArcGIS (2021): Überlegungen zur Geoverarbeitung für die Shapefile-Ausgabe—Hilfe | ArcGIS for Desktop. <https://desktop.arcgis.com/de/arcmap/10.3/manage-data/shapefiles/geoprocessing-considerations-for-shapefile-output.htm> (letzter Zugriff 10.09.2021).
- Arsanjani, Jamal Jokar/Zipf, Alexander/Mooney, Peter/Helbich, Marco (2015): Openstreetmap in GIScience. New York, NY: Springer Berlin Heidelberg.
- BMUB (Hg.) (2017): Weißbuch Stadtgrün - Grün in der Stadt – Für eine lebenswerte Zukunft.
- Bolund, Per/Hunhammar, Sven (1999): Ecosystem services in urban areas. In: Ecological Economics 29 (2), S. 293–301. doi:10.1016/S0921-8009(99)00013-0.
- Corti, Paolo (Hg.) (2014): PostGIS cookbook: over 80 task-based recipes to store, organize, manipulate, and analyze spatial data in a PostGIS database; [quick answers to common problems]. 1. publ. Birmingham: Packt Publ.
- FOSSGIS (2021): Vortrag - Vorteile einer Datenhaltung in PostgreSQL/PostGIS, die ein Shapefile nicht bietet – FOSSGIS 2021. <https://www.fossgis-konferenz.de/2021/sessions/7ZW3YE.php> (letzter Zugriff 07.09.2021).
- Geofabrik (2021): Geofabrik Download Server. <https://download.geofabrik.de/europe/germany/sachsen.html> (letzter Zugriff 07.09.2021).
- GeoSN (2021): WFS SN ALKIS Vereinfacht - smart.finder SDI. <https://geomis.sachsen.de/geomis-client/?lang=de#/datasets/iso/19ca457b-97d6-439d-a3ea-fd060f053bc4> (letzter Zugriff 11.09.2021).
- Handbuch (2021): Handbuch des meinGrün-Projektes (Entwurf).
- Hardy, Stephen (2000): Placiality: The renewal of the significance of place in moden cultural theory. In: SBORNIK PRACI FILOZOFICKE FAKULTY BRNENSKE UNIVERZITY STUDIA MINORA FACULTATIS PHILOSOPHICAE UNIVERSITATIS BRUNENSIS.,
- IÖR (2021): IÖR Monitor. In: IÖR Monitor. <https://www.ioer-monitor.de/>.
- IÖR Monitor (2021): Methodik - IÖR Monitor. <https://www.ioer-monitor.de/methodik/#c239> (letzter Zugriff 08.09.2021).
- Janakiev, Nikolai (2018): Data Science with OpenStreetMap. Maptime Salzburg. <https://av.tib.eu/media/38510>.
- Jones, Lizzie/Holland, Robert A./Ball, Jennifer/Sykes, Tim/Taylor, Gail/Ingwall-King, Li-sa/Snaddon, Jake L./Peh, Kelvin S.-H. (2020): A place-based participatory mapping approach for assessing cultural ecosystem services in urban green space. In: People and Nature 2 (1), S. 123–137. doi:10.1002/pan3.10057.
- Kirschner, Nina/Bruns, Julian/Abecker, Andreas/Kutterer, Johannes/Günther-Diringer, Detlef (2020): Framework zur Erstellung komplexer, räumlicher Vergleichsindizes auf Basis offener Daten – Anwendung am Beispiel der Fahrradfreundlichkeit in Städten Baden-Württembergs. In: GIS-Zeitschrift für Geoinformatik 33, S. 59–71.

- Krellenberg, Kerstin/Artmann, Martina/Stanley, Celina/Hecht, Robert (2021): What to do in, and what to expect from, urban green spaces – Indicator-based approach to assess cultural ecosystem services. In: *Urban Forestry & Urban Greening* 59, S. 126986. doi:10.1016/j.ufug.2021.126986.
- lucasmation (2016): osm_cnefe_import.
- Ludwig, Christina/Hecht, Robert/Lautenbach, Sven/Schorcht, Martin/Zipf, Alexander (2021): Mapping Public Urban Green Spaces Based on OpenStreetMap and Sentinel-2 Imagery Using Belief Functions. In: *ISPRS International Journal of Geo-Information* 10 (4), S. 251. doi:10.3390/ijgi10040251.
- Majic, Ivan/Winter, Stephan/Tomko, Martin (2017): Finding equivalent keys in OpenStreetMap: semantic similarity computation based on extensional definitions.
- meinGrün (2021): meinGrün. In: meinGrün - Entdecke deine Grüne Stadt. <https://meingruen.org/> (letzter Zugriff 06.09.2021).
- mg (2021): Projektbeschreibung - meinGrün. <http://meingruen.ioer.info/projekt/projektbeschreibung/> (letzter Zugriff 06.09.2021).
- natural earth (2021): Natural Earth » Downloads - Free vector and raster map data at 1:10m, 1:50m, and 1:110m scales. <https://www.naturalearthdata.com/downloads/> (letzter Zugriff 07.09.2021).
- Ng, Annalyn/Soo, Kenneth (2018): Data Science - was ist das eigentlich?! Algorithmen des maschinellen Lernens verständlich erklärt. Berlin [Heidelberg]: Springer.
- Nguyen, Trung (2021): Kedro hands-on: Build your own demographics atlas. Pt. 1: data gathering and cleaning. In: Medium. <https://towardsdatascience.com/kedro-hands-on-build-your-own-demographics-atlas-pt-1-16f2cb6676d5> (letzter Zugriff 14.09.2021).
- Nguyen, Chi Nhan/Zeigermann, Oliver (2021): Machine Learning - kurz & gut. (= kurz & gut). 2. Auflage. Heidelberg: O'Reilly.
- Obe, Regina/Hsu, Leo (2011): PostGIS in action. Greenwich, Conn. : London: Manning ; Pearson Education [distributor].
- OSM barrier (2021): Key:barrier – OpenStreetMap Wiki. <https://wiki.openstreetmap.org/wiki/Key:barrier> (letzter Zugriff 08.09.2021).
- OSM history (2021): History of OpenStreetMap – OpenStreetMap Wiki. https://wiki.openstreetmap.org/wiki/History_of_OpenStreetMap (letzter Zugriff 11.09.2021).
- OSM schema (2021): OSM DB Schema. https://wiki.openstreetmap.org/w/images/5/58/OSM_DB_Schema_2016-12-13.svg (letzter Zugriff 11.09.2021).
- OSM stats (2021): Stats – OpenStreetMap Wiki. https://wiki.openstreetmap.org/wiki/Stats#Registered_users_statistics_reports (letzter Zugriff 11.09.2021).
- osm2pgsql (2021): Osm2pgsql/schema – OpenStreetMap Wiki. <https://wiki.openstreetmap.org/wiki/Osm2pgsql/schema> (letzter Zugriff 11.05.2021).
- osm2pgsql-Manual (2021): Osm2pgsql Manual - osm2pgsql. <https://osm2pgsql.org/doc/manual.html> (letzter Zugriff 10.09.2021).
- Patten, Bernard C./Auble, Gregor T. (1981): System Theory of the Ecological Niche. In: *The American Naturalist* 117 (6), S. 893–922. doi:10.1086/283777.
- Perrings, C. et al. (2010): Ecosystem Services for 2020. In: *Science* 330 (6002), S. 323–324. doi:10.1126/science.1196431.

- Piraten-Dresden (2021): Daten aller kommunalen Grundstücke sind frei! In: Piraten Dresden. <https://www.piraten-dresden.de/daten-aller-kommunalen-grundstuecke-sind-frei/> (letzter Zugriff 13.09.2021).
- PostGIS (2021): PostGIS — Spatial and Geographic Objects for PostgreSQL. <https://postgis.net/> (letzter Zugriff 07.09.2021).
- QField (2021): QField - Efficient field work built for QGIS. <https://qfield.org/> (letzter Zugriff 11.09.2021).
- Raschka, Sebastian/Mirjalili, Vahid (2021): Machine learning mit Python und Keras, TensorFlow 2 und Scikit-learn: das umfassende Praxis-Handbuch für data science, deep learning und predictive analytics. 3., aktualisierte und erweiterte Auflage. Frechen: mitp.
- Schultz, Michael/Voss, Janek/Auer, Michael/Carter, Sarah/Zipf, Alexander (2017): Open land cover from OpenStreetMap and remote sensing. In: International Journal of Applied Earth Observation and Geoinformation 63, S. 206–213.
doi:10.1016/j.jag.2017.07.014.
- Sk53-osm (2015): Maps Matter: Urban Areas 3 : derivation from OSM using residential blocks. In: Maps Matter. <http://sk53-osm.blogspot.com/2015/11/urban-areas-3-derivation-from-osm-using.html> (letzter Zugriff 14.09.2021).
- spektrum (2021): Transektmethode. <https://www.spektrum.de/lexikon/geographie/transektmethode/8216> (letzter Zugriff 11.09.2021).
- statista (2021): Grad der Urbanisierung in Deutschland. In: Statista. <https://de.statista.com/statistik/daten/studie/662560/umfrage/urbanisierung-in-deutschland/> (letzter Zugriff 09.09.2021).
- ST_Polygonize (2021): ST_Polygonize. https://postgis.net/docs/ST_Polygonize.html (letzter Zugriff 11.09.2021).
- TC (2018): Big Data fürs Stadtgrün. In: Transforming Cities. <https://www.transforming-cities.de/big-data-fuers-stadtgruen/> (letzter Zugriff 10.09.2021).
- TC (2020): Werden Großstädte in Deutschland wirklich immer grüner? In: Transforming Cities. <https://www.transforming-cities.de/gruenanlagen-in-deutschen-grossstaedten/> (letzter Zugriff 10.09.2021).
- Thomas, Daniel L. J. (2020): A Guide: Turning OpenStreetMap Location Data into ML Features. In: Medium. <https://towardsdatascience.com/a-guide-turning-openstreetmap-location-data-into-ml-features-e687b66db210> (letzter Zugriff 14.09.2021).
- uuid-ossp (2021): F.45. uuid-ossp. In: PostgreSQL Documentation. <https://www.postgresql.org/docs/10/uuid-ossp.html> (letzter Zugriff 09.09.2021).
- Vargas-Munoz, John E./Srivastava, Shivangi/Tuia, Devis/Falcão, Alexandre X. (2021): OpenStreetMap: Challenges and Opportunities in Machine Learning and Remote Sensing. In: IEEE Geoscience and Remote Sensing Magazine 9 (1), S. 184–199.
doi:10.1109/MGRS.2020.2994107.
- Young, Peter T. (2014): Ho'okuleana: Placial vs Spatial. In: Ho'okuleana. <http://totakeresponsibility.blogspot.com/2014/03/placial-vs-spatial.html> (letzter Zugriff 13.09.2021).

Abbildungsverzeichnis

| | |
|---|----|
| Abbildung 1 Urbanisierungsgrad: Anteil der Stadtbewohner an der Gesamtbevölkerung in Deutschland in den Jahren von 2000 bis 2020 (statista 2021) | 1 |
| Abbildung 2 Bildschirmabzug der meinGrün-App nach Aktivitätensuche (meinGrün 2021) | 2 |
| Abbildung 3 Grünflächentypen in Dresden (Handbuch 2021)..... | 4 |
| Abbildung 4 Untersuchungsgebiet Dresden mit Umlandsaum (natural earth 2021; mg 2021)..... | 7 |
| Abbildung 5 Akkumulierte OSM-Nutzerzahlen nach Monat (OSM stats 2021) | 8 |
| Abbildung 6 Generierung von OSM Geometrien nach Zeit (OSM stats 2021)..... | 9 |
| Abbildung 7 Bildschirmabzug ALKIS WFS – Datenschema / Attribute | 10 |
| Abbildung 8 Datensatz kommunale Flächen Dresden..... | 11 |
| Abbildung 9 Fehlermeldung in PostgreSQL zu fehlendem Speicherplatz..... | 20 |
| Abbildung 10 Wahl der Limitierung des Speicherplatzes pro Container | 20 |
| Abbildung 11 Methodisches Vorgehen..... | 22 |
| Abbildung 12 Flächenschema des IÖR-Monitors (IÖR 2021)..... | 23 |
| Abbildung 13 Vorgehen zur Ableitung eines überlappungsfreien Landnutzungsdatensatzes..... | 24 |
| Abbildung 14 Vorgehen zur Durchführung des Ground-Truth | 25 |
| Abbildung 15 Schema Arbeitsumgebung mit Docker Containern und Softwareeinsatz | 29 |
| Abbildung 16 Schema zur Arbeitsweise von osm2pgsql (osm2pgsql 2021) | 32 |
| Abbildung 17 DB-Tabellen nach OSM Import (mit belegtem Speicherplatz)..... | 32 |
| Abbildung 18 Bildschirmabzug in QGIS von abgeleiteten Barrieren (Dresden) | 35 |
| Abbildung 19 Bildschirmabzug in QGIS von abgeleiteten Barrieren (Großer Garten).. | 35 |
| Abbildung 20 Element-Icons aus OSM Map Features Tabelle | 37 |
| Abbildung 21 Bildschirmabzug Datenschema Variante GIScience nach Vereinigen ... | 40 |
| Abbildung 22 Visualisierung eines ersten manuellen Ergebnisses | 41 |
| Abbildung 23 Bildschirmabzug Zwischenergebnis Variante GIScience Schritt Nr. 7 ... | 42 |
| Abbildung 24 Fehlerhaftes Multipolygon der Restklasse in Cyan, AOI in Rot..... | 43 |
| Abbildung 25 Untersuchungsgebiete des Ground-Truth innerhalb der AOI Dresden... | 45 |
| Abbildung 26 Vorbereite GeoJSON-Dateien (außer LU-Wechsel) | 45 |
| Abbildung 27 Enthaltene Geometriertypen mit Anzahl nach ST_Intersection()..... | 46 |
| Abbildung 28 Bildschirmabzug QGIS Stil zu Attribut „barriere“ | 47 |
| Abbildung 29 Bildschirmabzüge App QField | 48 |
| Abbildung 30 Datensatz „linien_pool_bbox“ mit BoundingBox | 52 |
| Abbildung 31 Visualisierung der finalen Version – LU-Datensatz Variante GIScience. | 57 |
| Abbildung 32 Visualisierung der finalen Version – LU-Datensatz Variante IÖR-Monitor | 58 |
| Abbildung 33 Relativer Flächenanteil pro Landnutzungsklasse – Variante GIScience | 59 |
| Abbildung 34 Relativer Flächenanteil pro Landnutzungsklasse – Variante IÖR-Monitor | 60 |
| Abbildung 35 Erfassungsstand Ground-Truth im Großen Garten | 64 |
| Abbildung 36 Erfassungsstand Ground-Truth in Dresden-Neustadt | 65 |
| Abbildung 37 Barrieren in der Realität: Beete (links), Holzpoller (rechts) | 66 |
| Abbildung 38 Clustermap LU-Wechsel nach erfasster Anzahl an Objekten (Ground- Truth)..... | 67 |

| | |
|---|----|
| Abbildung 39 Barrierenwahrscheinlichkeit nach Wegetyp | 67 |
| Abbildung 40 Barrierenwahrscheinlichkeit nach LU-Wechsel..... | 68 |
| Abbildung 41 LU-Wechsel nach erfasster Gesamtlänge in Metern..... | 69 |
| Abbildung 42 Linien im Linien-Pool (Straßen, Wege, Bahnverkehr, Wasserwege)..... | 71 |
| Abbildung 43 Linien im Linien-Pool (Barrieren und LU-Wechsel) | 72 |
| Abbildung 44 Ergebnis der Vermaschung nach Barrierenwahrscheinlichkeit – Großer Garten Dresden, westlicher Eingang | 73 |
| Abbildung 45 Ergebnis der Vermaschung nach Barrierenwahrscheinlichkeit – Sportanlagen Ostragehege, Dresden | 74 |

Quelltextverzeichnis

| | |
|---|----|
| Quelltext 1 Generieren einer UUID | 17 |
| Quelltext 2 Struktur einer FUNCTION mit PL/pgSQL | 17 |
| Quelltext 3 möglicher „Workaround“ mit mehreren Funktionen und COMMIT | 18 |
| Quelltext 4 Docker Befehle zum Initialisieren der Container für DB und Datenimporte | 31 |
| Quelltext 5 OSM-Daten Import mit osm2pgsql | 31 |
| Quelltext 6 Abfrage für schnellen Überblick über existierende OSM key-value | 33 |
| Quelltext 7 Selektieren und Einfügen von nicht geschlossenen Linienzügen | 34 |
| Quelltext 8 (verkürzte) Abfrage zur „hstore“-Nutzung und zu Anführungszeichen bei „natural“..... | 37 |
| Quelltext 9 Abfrage zur Ermittlung von Überlappungen (Anzahl und Gesamtfläche) ... | 39 |
| Quelltext 10 Abfrage zur Ermittlung von Löchern (Anzahl und Gesamtfläche) | 39 |
| Quelltext 11 Manuelles Verschneiden von Restpolygon mit städtischer Bebauung.... | 41 |
| Quelltext 12 Datenaufbereitung Ground-Truth – Datentyp „hstore“ konvertieren..... | 44 |
| Quelltext 13 Ermittlung der LU-Wechsel | 46 |
| Quelltext 14 Sliver-Polygone ersetzen durch längste Kante | 47 |
| Quelltext 15 Zusammenfassen von LU-Wechsel (A→B und B→A) zu A→B | 49 |
| Quelltext 16 Berechnung der Barrierenwahrscheinlichkeit nach Anzahl und nach Länge | 50 |
| Quelltext 17 Datenschema für Tabelle „linien_pool“ | 50 |
| Quelltext 18 ST_Intersection und ST_Dump auf „linien_pool“ zu BoundingBox | 52 |
| Quelltext 19 Vermaschung mit ST_Polygonize() | 53 |
| Quelltext 20 Aufteilung in Trainings- und Testdatensatz | 54 |
| Quelltext 21 Anwenden eines linearen Support Vector Classifier (SVC) | 54 |

Formelverzeichnis

| | |
|--|----|
| Formel 1 Berechnung des Lagefehlers durch ST_SnapToGrid()..... | 44 |
|--|----|

Tabellenverzeichnis

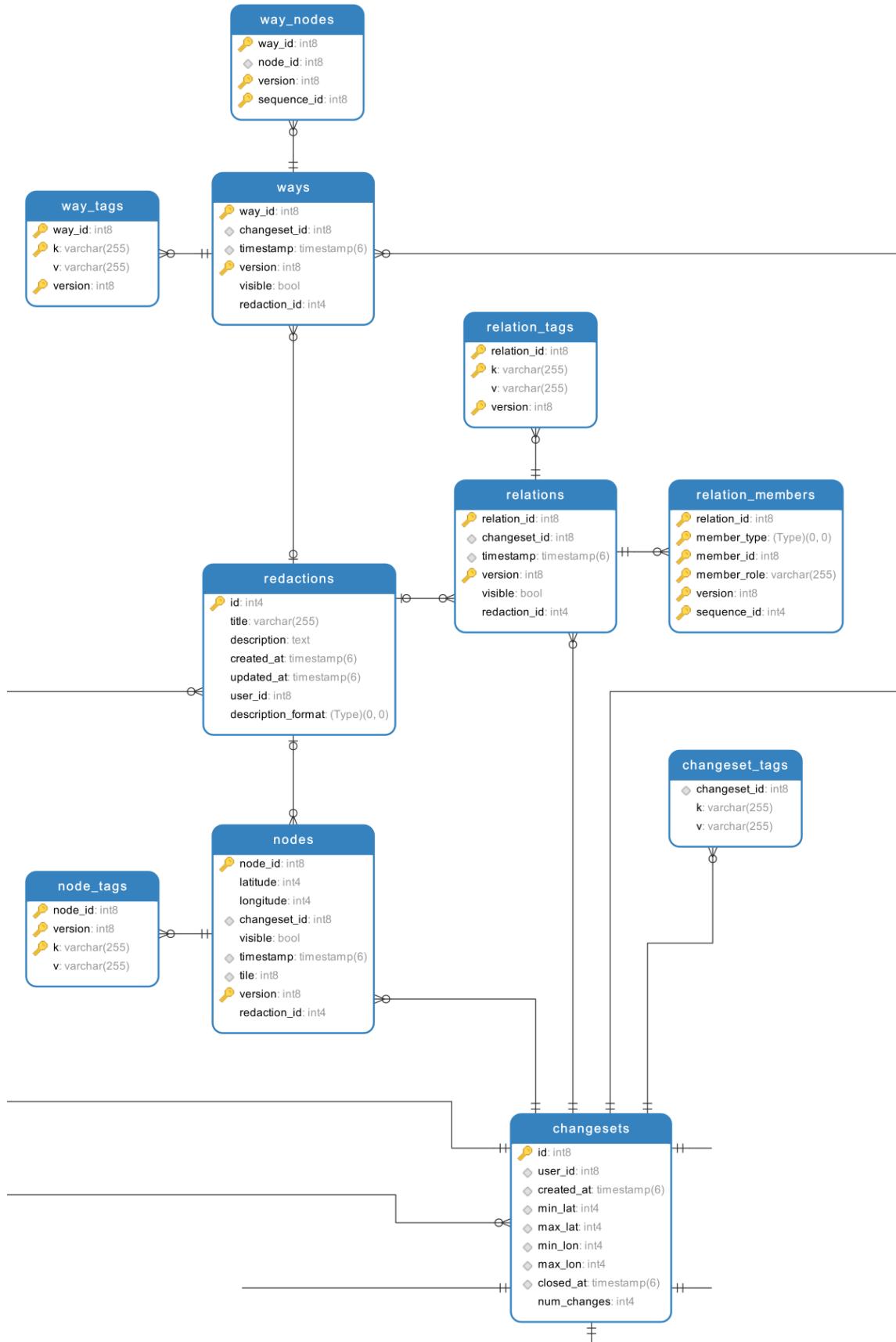
| | |
|--|----|
| Tabelle 1 Definierte Aktivitäten im Projekt meinGrün (mit Aktionsraum) | 14 |
| Tabelle 2 Verwendung spezieller Datentypen (Auswahl) | 18 |

| | |
|--|----|
| Tabelle 3 Datenschema für Ground-Truth..... | 25 |
| Tabelle 4 Festlegungen für Output (Modell „Grünheit“ und Modell „Zugänglichkeit“) ... | 27 |
| Tabelle 5 Mögliche Input-Variablen..... | 28 |
| Tabelle 6 Softwareeinsatz mit genutzten Versionen..... | 30 |
| Tabelle 7 Liste der abgeleiteten Barrierentypen (mit semantischen Metadaten)..... | 33 |
| Tabelle 8 Liste der abgeleiteten Barrierentypen (mit technischen Metadaten) | 34 |
| Tabelle 9 LU-Klassen Variante GIScience | 36 |
| Tabelle 10 Semantische, geometrische und topologische Prüfung – Zwischenstand LU | 38 |
| Tabelle 11 Aufbereitung der abgeleiteten Barrieren für Vermaschung | 51 |
| Tabelle 12 Analyse zu punkthaften OSM-Objekten mit Grünflächen-Bezug (POIs).... | 55 |
| Tabelle 13 Analyse zu flächenhaften OSM-Objekten mit Grünflächen-Bezug | 55 |
| Tabelle 14 Analyse Key „barrier“ nach Anzahl (bezogen auf AOI Dresden) | 56 |
| Tabelle 15 Analyse LU-Klassen von Variante GIScience | 59 |
| Tabelle 16 Analyse LU-Klassen von Variante IÖR-Monitor (Auswahl) | 60 |
| Tabelle 17 LU-Klassen Variante IÖR-Monitor | 61 |
| Tabelle 18 Semantische, geometrische und topologische Prüfung – Endstand LU | 62 |
| Tabelle 19 quantitative Analyse des Ground-Truths (inklusive „nodata“)..... | 65 |
| Tabelle 20 Anzahl und Länge der Linien in Abhängigkeit der Barrierenwahrscheinlichkeit | 71 |
| Tabelle 21 Auswertung Vermaschung nach Polygonen | 72 |
| Tabelle 22 Modell Zugänglichkeit – Scores linearer Verfahren..... | 75 |
| Tabelle 23 Modell Zugänglichkeit – Scores nicht-linearer Verfahren | 76 |
| Tabelle 24 Modell Grünheit – Scores linearer Verfahren..... | 76 |
| Tabelle 25 Auswertung Vermaschung nach Laufzeit | 77 |

Anlagenverzeichnis

- Digitaler Datenträger
 - Masterarbeit
 - Quelltexte
 - Docker
 - PostGIS
 - Python – Jupyter Notebooks
 - Daten

Anlage 1 OSM Datenbankschema Geometrien (Auszug) (OSM schema 2021)



Anlage 2 Zuweisung OSM key-value zur Ableitung von Barrieren

| osm_railway | osm_trail | osm_street | osm_waterway | osm_barrier |
|-------------------------|-----------------------|----------------------------|------------------------|----------------------------|
| railway='construction' | highway = 'bridleway' | highway = 'construction' | waterway = 'canal' | barrier=* |
| railway='disused' | highway = 'cycleway' | highway = 'living_street' | waterway = 'dam' | (alle Value sind relevant) |
| railway='facility' | highway = 'footway' | highway = 'motorway' | waterway = 'ditch' | |
| railway='funicular' | highway = 'no' | highway = 'motorway_link' | waterway = 'drain' | |
| railway='miniature' | highway = 'path' | highway = 'pedestrian' | waterway = 'fish_pass' | |
| railway='narrow_gauge' | highway = 'track' | highway = 'platform' | waterway = 'river' | |
| railway='platform' | | highway = 'primary' | waterway = 'stream' | |
| railway='platform_edge' | | highway = 'primary_link' | | |
| railway='preserved' | | highway = 'raceway' | | |
| railway='rail' | | highway = 'residential' | | |
| railway='tram' | | highway = 'road' | | |
| railway='tram_stop' | | highway = 'secondary' | | |
| railway='turntable' | | highway = 'secondary_link' | | |
| | | highway = 'service' | | |
| | | highway = 'steps' | | |
| | | highway = 'tertiary' | | |
| | | highway = 'tertiary_link' | | |
| | | highway = 'trunk' | | |
| | | highway = 'trunk_link' | | |
| | | highway = 'unclassified' | | |

Anlage 3: Docker Befehle zum Erstellen der Docker Container

```
docker run --name ma_db_01 -d -e POSTGRES_PASSWORD=postgres -p 5430:5432 -v  
~/ma/ma_db_01:/var/lib/postgresql/data mdillon/postgis
```



```
docker run -i -t -d --name ma_import_01 -p 9000:9000 -v  
~/ma/ma_import_01:/code ubuntu:latest /bin/bash
```



```
docker run -i -t -d --name ma_process_01 -p 9001:9001 -v  
~/ma/ma_process_01:/code ubuntu:latest /bin/bash
```



```
docker run -i -t -d --name ma_anaconda_01 -p 8888:8888 -v  
~/ma/ma_anaconda_01:/opt/notebooks continuumio/anaconda3 /bin/bash -c  
"/opt/conda/bin/conda install jupyter -y --quiet && mkdir /opt/notebooks &&  
/opt/conda/bin/jupyter notebook --notebook-dir=/opt/notebooks --ip='*' --  
port=8888 --no-browser --allow-root"
```

Anlage 4: Schematransformation Variante IÖR-Monitor

| Zielschema | Ausgangsschema |
|----------------|---|
| Straßenverkehr | <p>highway='service'</p> <p>highway='pedestrian'</p> <p>highway='footway'</p> <p>highway='platform'</p> <p>highway='rest_area'</p> <p>highway='services'</p> <p>barrier='toll_booth'</p> <p>public_transport='stop_area'</p> <p>amenity='bicycle_parking'</p> <p>amenity='bicycle_repair_station'</p> <p>amenity='bicycle_rental'</p> <p>amenity='bus_station'</p> <p>amenity='charging_station'</p> <p>amenity='motorcycle_parking'</p> <p>amenity='parking'</p> <p>amenity='parking_space'</p> <p>amenity='taxi'</p> <p>building='transportation'</p> |
| Bahnverkehr | <p>landuse='railway'</p> <p>public_transport='platform'</p> <p>public_transport='station'</p> <p>railway='platform'</p> <p>railway='station'</p> <p>railway='turntable'</p> <p>railway='roundhouse'</p> <p>railway='traverser'</p> |

| | |
|-----------------------|--|
| | railway='wash' building='train_station' |
| Flugverkehr | aeroway='aerodrome' aeroway='apron' aeroway='hangar' aeroway='helipad' aeroway='heliport' aeroway='spaceport' aeroway='terminal' (gleich aeroway=*) building='hangar' |
| Verkehrsbegleitfläche | |
| Wohnbau | landuse='residential' building='apartments' building='cabin' building='detached' building='dormitory' building='farm' building='ger' building='house' building='houseboat' building='residential' building='semidetached_house' building='static_caravan' building='terrace' |
| Mischnutzung | landuse='garages' building='hut' building='shed' building='carport' |

| | |
|-------------------------------|---|
| | building='garage' building='garages' building='bridge' building='container' building='marquee' |
| Besondere funktionale Prägung | emergency='ambulance_station' emergency='landing_site' emergency='lifeguard' emergency='assembly_point' historic='building' historic='castle' historic='church' historic='city_gate' historic='aqueduct' historic='fort' historic='manor' historic='memorial' historic='monastery' historic='monument' historic='ruins' historic='ship' landuse='education' landuse='depot' landuse='religious' |

| | |
|--|----------------------------------|
| | aerialway='station' |
| | barrier='city_wall' |
| | historic='city_gate' |
| | historic='citywalls' |
| | healthcare=* |
| | man_made='lighthouse' |
| | man_made='observatory' |
| | man_made='telescope' |
| | man_made='pier' |
| | man_made='tower' |
| | man_made='water_well' |
| | man_made='water_tap' |
| | office='diplomatic' |
| | office='educational_institution' |
| | office='employment_agency' |
| | office='forestry' |
| | office='government' |
| | office='political_party' |
| | office='religion' |
| | office='research' |
| | office='visa' |
| | tourism='attraction' |
| | tourism='artwork' |

| | |
|--|---------------------------|
| | tourism='information' |
| | amenity='college' |
| | amenity='driving_school' |
| | amenity='kindergarten' |
| | amenity='language_school' |
| | amenity='library' |
| | amenity='toy_library' |
| | amenity='music_school' |
| | amenity='school' |
| | amenity='university' |
| | amenity='baby_hatch' |
| | amenity='clinic' |
| | amenity='dentist' |
| | amenity='doctors' |
| | amenity='hospital' |
| | amenity='nursing_home' |
| | amenity='pharmacy' |
| | amenity='social_facility' |
| | amenity='veterinary' |
| | amenity='arts_centre' |
| | amenity='planetarium' |
| | amenity='public_bookcase' |
| | amenity='social_centre' |
| | amenity='courthouse' |
| | amenity='embassy' |
| | amenity='fire_station' |
| | amenity='police' |
| | amenity='post_depot' |

| | |
|--|--|
| | amenity='post_office' amenity='prison' amenity='ranger_station' amenity='townhall' amenity='give_box' amenity='shelter' amenity='shower' amenity='toilets' amenity='childcare' amenity='funeral_hall' amenity='kitchen' amenity='monastery' amenity='place_of_mourning' amenity='place_of_worship' amenity='refugee_site' leisure='bandstand' leisure='bird_hide' leisure='common' leisure='sports_centre' leisure='stadium' leisure='summer_camp' building='cathedral' building='chapel' building='church' building='monastery' building='mosque' building='presbytery' |
|--|--|

| | |
|-----------------------------|--|
| | <p>building='religious'</p> <p>building='shrine'</p> <p>building='synagogue'</p> <p>building='temple'</p> <p>building='bakehouse'</p> <p>building='civic' (→ amenity=, leisure=...)</p> <p>building='fire_station'</p> <p>building='government'</p> <p>building='hospital'</p> <p>building='public'</p> <p>building='toilets'</p> <p>building='kindergarten'</p> <p>building='school'</p> <p>building='university'</p> <p>building='college'</p> <p>building='grandstand'</p> <p>building='pavilion'</p> <p>building='riding_hall'</p> <p>building='sports_hall'</p> <p>building='stadium'</p> <p>building='parking'</p> <p>building='gatehouse'</p> <p>building='ruins'</p> |
| Industrie- u. Gewerbefläche | <p>landuse='commercial'</p> <p>landuse='industrial'</p> <p>landuse='retail'</p> <p>landuse='port'</p> <p>man_made='chimney'</p> <p>man_made='communications_tower'</p> |

| | |
|--|---|
| | <p>man_made='crane'</p> <p>man_made='mineshift'</p> <p>man_made='monitoring_station'</p> <p>man_made='pumping_station'</p> <p>man_made='silo'</p> <p>man_made='storage_tank'</p> <p>man_made='street_cabinet'</p> <p>man_made='tailings_pond'</p> <p>man_made='wastewater_plant'</p> <p>man_made='watermill'</p> <p>man_made='water_tower'</p> <p>man_made='water_works'</p> <p>man_made='windmill'</p> <p>man_made='works'</p> |
| | <p>craft=*</p> <p>landuse='military'</p> <p>military=* → eigentlich funktionale Prägung, aber bei Industrie und Gewerbe besser eingeordnet.</p> <p>building='military'</p> <p>office='accountant'</p> <p>office='advertising_agency'</p> <p>office='architect'</p> <p>office='association'</p> <p>office='charity'</p> |

office='company'
office='consulting'
office='courier'
office='coworking'
office='energy_supplier'
office='engineer'
office='estate_agent'
office='financial'
office='financial_advisor'
office='foundation'
office='graphic_design'
office='guide'
office='harbour_master'
office='insurance'
office='it'
office='lawyer'
office='logistics'
office='moving_company'
office='newspaper'
office='ngo'
office='notary'
office='property_management'
office='quango'
office='surveyor'
office='tax_advisor'
office='telecommunication'
office='water_utility'
office='yes'

| | |
|--|--------------------------|
| | power=* |
| | shop=* |
| | telecom=* |
| | |
| | tourism='alpine_hut' |
| | tourism='apartment' |
| | tourism='aquarium' |
| | tourism='chalet' |
| | tourism='gallery' |
| | tourism='guest_house' |
| | tourism='hostel' |
| | tourism='hotel' |
| | tourism='motel' |
| | tourism='museum' |
| | tourism='wilderness_hut' |
| | |
| | waterway='dam' |
| | waterway='weir' |
| | waterway='fuel' |
| | |
| | amenity='bar' |
| | amenity='biergarten' |
| | amenity='cafe' |
| | amenity='fast_food' |
| | amenity='food_court' |
| | amenity='ice_cream' |
| | amenity='pub' |
| | amenity='restaurant' |
| | amenity='car_rental' |

| | |
|--|--|
| | amenity='car_sharing' amenity='car_wash' amenity='vehicle_inspection' amenity='ferry_terminal' amenity='fuel' amenity='bank' amenity='bureau_de_change' amenity='brothel' amenity='casino' amenity='cinema' amenity='community_centre' amenity='conference_centre' amenity='events_venue' amenity='gambling' amenity='love_hotel' amenity='nightclub' amenity='stripclub' amenity='studio' amenity='swingerclub' amenity='theatre' amenity='recycling' amenity='waste_transfer_station' amenity='animal_boarding' amenity='animal_shelter' amenity='crematorium' amenity='dive_centre' amenity='hunting_stand' amenity='internet_cafe' amenity='marketplace' |
|--|--|

| | |
|------------------|---|
| | leisure='adult_gaming_centre' leisure='amusement_arcade' leisure='beach_resort' leisure='dance' leisure='disc_golf_course' leisure='escape_game' leisure='fitness_centre' leisure='hackerspace' leisure='horse_riding' leisure='ice_rink' leisure='water_park' |
| | building='hotel' building='commercial' building='industrial' building='kiosk' building='office' (aber office=* ist wichtiger) building='retail' building='supermarket' building='warehouse' building='digester' building='service' building='transformer_tower' building='water_tower' building='bunker' building='roof' |
| Park, Grünanlage | landuse='flowerbed' amenity='fountain' |

| | |
|--|---|
| | leisure='garden' leisure='park' |
| Kleingarten | landuse='allotments' |
| Wochenendsiedlung | building='bungalow' |
| Golfplatz | sport='golf' leisure='miniature_golf' |
| Sonstige Sport, Freizeit, Erholung: Spielplatz, Spielbereich | amenity='kneipp_water_cure' leisure='fitness_station' leisure='playground' |
| Sonstige Sport, Freizeit, Erholung: Sportplatz, Sportanlage | landuse='recreation_ground' leisure='pitch' sport='*' (außer 'golf', und nur wenn kein building='yes' vorhanden ist! Oder nur bei pitch!) amenity='public_bath' leisure='swimming_pool' leisure='swimming_area' leisure='track' |
| Sonstige Sport, Freizeit, Erholung: Rasen, Gras (ohne Agrarnutzung?) | landuse='grass' landuse='village_green' |
| Sonstige Sport, Freizeit, Erholung: sonstige | tourism='camp_site' tourism='camp_pitch' tourism='caravan_site' tourism='picnic_site' tourism='zoo' tourism='theme_park' tourism='viewpoint' amenity='bbq' amenity='dog_toilet' |

| | |
|--------------------------------|---|
| | leisure='dog_park' leisure='firepit' leisure='fishing' |
| Friedhof | landuse='cemetery' amenity='grave_yard' |
| Sonstige Siedlungsfläche | landuse='construction' landuse='greenfield' landuse='brownfield' barrier='hedge' barrier='wall' man_made='breakwater' man_made='bunker_silo' man_made='gasometer' building='construction' |
| Abbau- und Haldenfläche | landuse='landfill' landuse='quarry' |
| Ackerland | landuse='farmland' |
| Grünland | landuse='meadow' natural='grassland' |
| Gartenland | |
| Streuobst | landuse='orchard' |
| Obstbau | |
| Weinbau | landuse='vineyard' |
| Sonstige Landwirtschaftsfläche | landuse='farmyard' landuse='greenhouse_horticulture' landuse='plant_nursery' historic='farm' building='barn' building='conservatory' |

| | |
|--------------------------------|---|
| | building='cowshed' building='farm_auxiliary' building='greenhouse' building='slurry_tank' building='stable' building='sty' |
| Laubholz | natural='wood' |
| Nadelholz | landuse='forest' |
| Mischholz | natural='scrub' |
| Gehölz | |
| Heide | natural='heath' |
| Moor | natural='moor' (veraltet) |
| Sumpf | natural='wetland' natural='mud' |
| Unland, vegetationslose Fläche | natural='bare_rock' natural='scree' natural='shingle' natural='sand' natural='glacier' natural='beach' natural='cliff' natural='dune' natural='rock' natural='stone' natural='sinkhole' geological='moraine' geological='outcrop' geological='volcanic_vent' geological='volcanic_lava_field' |

| | |
|--------------------|---|
| | <p>landuse='salt_pond'</p> <p>man_made='clearcut'</p> |
| Fließgewässer | <p>natural='water'</p> <p>natural='spring'</p> <p>natural='hot_spring'</p> <p>emergency='water_tank'</p> <p>landuse='basin'</p> <p>landuse='reservoir'</p> <p>water='river'</p> <p>water='oxbow'</p> <p>water='canal'</p> <p>water='ditch'</p> <p>water='fish_pass'</p> <p>water='lake'</p> <p>water='reservoir'</p> <p>water='pond'</p> <p>water='basin'</p> <p>water='stream_pool'</p> <p>water='reflecting_pool'</p> <p>water='moat'</p> <p>water='wastewater'</p> <p>waterway='riverbank'</p> |
| Stehendes Gewässer | |
| Hafenbecken | <p>water='lock'</p> <p>waterway='dock'</p> <p>waterway='boatyard'</p> <p>amenity='boat_rental'</p> |

| | |
|--------------|---|
| | amenity='boat_sharing' leisure='marina' leisure='slipway' |
| Meer, Bodden | natural='bay' natural='strait' natural='reef' water='lagoon' |

Anlage 5: Barrierenwahrscheinlichkeit LU-Wechsel (nach Länge)

| lu_nr A | lu_nr B | Anzahl Obj. | Anzahl gültige Obj. | Anzahl Barriere | Barrierenwahrscheinlichkeit nach Länge |
|----------------|----------------|--------------------|----------------------------|------------------------|---|
| 50 | 26 | 245 | 245 | 27 | 0,125297859 |
| 26 | 25 | 25 | 25 | 2 | 0,227521792 |
| 26 | 20 | 232 | 232 | 127 | 0,245770704 |
| 50 | 41 | 26 | 26 | 6 | 0,266586081 |
| 25 | 20 | 16 | 16 | 3 | 0,274621123 |
| 50 | 20 | 128 | 117 | 43 | 0,325306714 |
| 41 | 20 | 6 | 5 | 1 | 0,440413747 |
| 26 | 1 | 24 | 24 | 10 | 0,515109565 |
| 1 | 0 | 28 | 28 | 6 | 0,518710019 |
| 26 | 0 | 126 | 122 | 49 | 0,528455792 |
| 24 | 20 | 51 | 51 | 35 | 0,540461263 |
| 26 | 13 | 30 | 30 | 26 | 0,779414439 |
| 13 | 0 | 17 | 17 | 7 | 0,827298874 |
| 10 | 0 | 25 | 25 | 17 | 0,919942498 |
| 20 | 0 | 44 | 44 | 38 | 0,926621176 |
| 25 | 0 | 17 | 17 | 13 | 0,959299595 |
| 50 | 0 | 35 | 35 | 32 | 0,979001612 |
| 20 | 10 | 49 | 35 | 34 | 0,990255033 |

| | | | | | |
|----|----|-----|-----|-----|---|
| 46 | 20 | 7 | 7 | 7 | 1 |
| 20 | 1 | 6 | 6 | 6 | 1 |
| 20 | 13 | 6 | 6 | 6 | 1 |
| 50 | 10 | 6 | 6 | 6 | 1 |
| 23 | 20 | 6 | 6 | 6 | 1 |
| 50 | 21 | 5 | 5 | 5 | 1 |
| 13 | 12 | 4 | 4 | 4 | 1 |
| 50 | 24 | 4 | 4 | 4 | 1 |
| 12 | 1 | 4 | 4 | 4 | 1 |
| 25 | 1 | 4 | 4 | 4 | 1 |
| 25 | 12 | 4 | 4 | 4 | 1 |
| 50 | 46 | 3 | 3 | 3 | 1 |
| 25 | 10 | 3 | 3 | 3 | 1 |
| 70 | 1 | 3 | 3 | 3 | 1 |
| 41 | 0 | 3 | 3 | 3 | 1 |
| 50 | 12 | 2 | 2 | 2 | 1 |
| 10 | 2 | 2 | 2 | 2 | 1 |
| 29 | 0 | 2 | 2 | 2 | 1 |
| 21 | 10 | 2 | 2 | 2 | 1 |
| 70 | 26 | 112 | 112 | 112 | 1 |
| 21 | 0 | 2 | 2 | 2 | 1 |
| 29 | 13 | 2 | 2 | 2 | 1 |
| 50 | 11 | 1 | 1 | 1 | 1 |
| 29 | 26 | 1 | 1 | 1 | 1 |
| 21 | 13 | 2 | 2 | 2 | 1 |
| 12 | 10 | 91 | 90 | 90 | 1 |
| 13 | 10 | 45 | 45 | 45 | 1 |
| 11 | 10 | 36 | 21 | 21 | 1 |
| 70 | 20 | 24 | 24 | 24 | 1 |
| 29 | 20 | 22 | 22 | 22 | 1 |
| 12 | 0 | 21 | 21 | 21 | 1 |
| 20 | 12 | 20 | 20 | 20 | 1 |
| 24 | 10 | 19 | 19 | 19 | 1 |
| 29 | 10 | 19 | 19 | 19 | 1 |
| 50 | 27 | 17 | 17 | 17 | 1 |
| 26 | 10 | 16 | 16 | 16 | 1 |

| | | | | | |
|----|----|----|----|----|---|
| 70 | 13 | 15 | 15 | 15 | 1 |
| 28 | 10 | 14 | 14 | 14 | 1 |
| 50 | 25 | 12 | 12 | 12 | 1 |
| 50 | 13 | 10 | 10 | 10 | 1 |
| 50 | 1 | 8 | 8 | 8 | 1 |
| 26 | 12 | 8 | 8 | 8 | 1 |

Erklärung über die eigenständige Erstellung der Arbeit

Hiermit erkläre ich, dass ich die vorgelegte Arbeit mit dem Titel

**Untersuchungen zur automatisierten Ableitung von Grünflächen im urbanen Raum
basierend auf offenen Daten**

selbständig verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie alle wörtlich oder sinngemäß übernommenen Stellen in der Arbeit als solche und durch Angabe der Quelle gekennzeichnet habe. Dies gilt auch für Zeichnungen, Skizzen, bildliche Darstellungen sowie für Quellen aus dem Internet.

Mir ist bewusst, dass die Hochschule für Technik und Wirtschaft Dresden Prüfungsarbeiten stichprobenartig mittels der Verwendung von Software zur Erkennung von Plagiaten überprüft.

Ort, Datum

Unterschrift Student