Filtern der Daten mit Hilfe von R

# Ursprüngliche Zielstellung:

1. Einlesen der Daten aus einer SQlite- Datenbank
2. Snapping der Vibrations-Ereignisse auf OSM-Daten
3. Repositionierung der Daten mithilfe der Laufzeitvariable
4. Filterung der IMU Daten, Generierung eines Qualitätsarguments der Strecke
5. Erzeugung von Vektoren der gefahrenen Strecke mit verknüpften Qualitätsargument des Abschnitts
6. Generierung von KML der erzeugten Vektoren
7. Farbcodierte Darstellung der Vektoren und dessen Qualitätsarguments über OpenLayers

# Einlesen der Daten aus einer SQLite-Datenbank

Die Daten von der SD-Karte wurden vorher mit Hilfe eines Python-Skripts in eine SQlite-Datenbank eingepflegt. Diese Datenbank liegt lokal vor und kann direkt vom angelegten R-Skript aus dem definierten Arbeitsverzeichnis ausgelesen werden. Die Daten werden dabei in ein *SpatialPointsDataframe* gespeichert. Dieses Dataframe enthält neben den Koordinaten der Vibrationsereignisse noch die, zum Zeitpunkt ausgelesenen, IMU-Daten, jeweils in X-, Y- und Z-Richtung. Weiterhin die GPS-Zeit und die Laufzeit des Ereignisses. Zudem hat jeder Eintrag eine ID. Die Koordinaten wurden durch das Python-Skript bereits vom NMEA-Format in geographische Koordinaten umgewandelt. Da R grundsätzlich dafür geeignet ist georeferenzierte Daten zu verarbeiten wurde hier noch der Bezugsellipsoid (*WGS84-global*) hinzugefügt, dadurch ist es später möglich georeferenzierte Daten zu exportieren, hier KML.

# Snapping der Vibrations-Ereignisse auf OSM-Daten

Da die erfassten GPS-Koordinaten, die vom Arduino-Shield erfasst wurden, nicht hinreichend präzise sind um die tatsächliche Fahrtstrecke abzubilden, musste eine Möglichkeit gefunden werden die Daten brauchbar zu interpolieren. Wie sich beim Plotten der ungefilterten Daten herausstellt, liegen die erfassten Koordinaten signifikant häufig außerhalb der gefahrenen Strecke in Häusern, auf Gleisen, oder der Gegenfahrbahn. Um diese Ungenauigkeiten zu kompensieren sollten nun die Daten auf vorhandene Straßen und Radwegdaten aufgebracht werden. Dabei fiel unsere Wahl auf OSM-Daten, da diese, im urbanen Raum, als sehr umfangreich betrachtet werden können und auch kartierte Radwege bieten. Zudem sind diese Daten lizenzrechtlich unbedenklich und frei verfügbar.

Ein kurzer Blick in die zusätzlich erhältlichen Bibliotheken für R ließ es für möglich erscheinen einen direkten Datenbankzugriff auf OSM Daten zu erhalten. Ziel wäre es dann gewesen die nötigen OSM-Daten dann direkt im Skript zu erfragen und für die weitere Verarbeitung der Daten zu nutzen. Leider erfüllte keine der gefunden Bibliotheken diese Anforderungen, sodass die nötigen OSM-Daten extern eingebracht werden mussten. Über die Seite *www.overpass-turbo.eu* wurden die nötigen OSM-Daten in Form eines *GeoJSON* erfragt und exportiert und anschließend in das R-Skript einlesen. Sobald die OSM-Daten, in dem Fall Straßennetzdaten, eingelesen waren wurden die Koordinaten der Daten aus dem Dataframe auf die OSM-Straßendaten projiziert, dabei wurden Daten, die einen zu großen Abstand zu vorhandenen OSM-Daten hatten aus dem Dataframe entfernt.

Diese Lösung ist im Moment kompromissbehaftet. Der Umfang der Daten in der OSM-Datenbank bezüglich kartierten Radwegen ist leider nicht umfangreich genug für diese Anwendung. Eine Beschränkung auf Radweg-Daten führte zu großen Lücken im Datensatz, sodass zusätzlich das Straßennetz hinzugenommen wurde. Weiterhin sollte der grundlegende Datensatz, da dieser extern eingebracht wurde, möglichst umfangreich sein, da er auch für die Verarbeitung von vorher unbekannten Routen einsetzbar sein sollte. Daher ist die *GeoJSON*-Datei recht umfangreich (~180MB) und ein einlesen der Daten benötigt vergleichsweise viel Zeit (ca. 20 min beim genutzten Computer). Weiterhin führt die Umfänglichkeit der OSM-Daten zusammen mit der unzureichenden Präzision der erfassten GPS-Koordinaten zu gelegentlich falschen Zuordnungen einer Koordinate zu einem falschen Wegstück, dies ließe sich durch eine spezifischere Auswahl der Daten für das GeoJSON weitgehend beheben, steht jedoch im Kontrast zur allgemeinen Anwendbarkeit des R-Skripts.

# Repositionierung der Daten mithilfe der Laufzeitvariable

Ursprünglich war es angedacht die erhobenen Daten auf die Streckenabschnitte aus den OSM-Daten zu projizieren. In Folge dessen sollte auch die Nutzung der Geometrie der entsprechenden Streckabschnitte in Kombination mit der erfassten Laufzeitvariable dazu genutzt werden eine feinere zeitliche Auflösung der Daten zu erreichen. Da jedoch eine Bibliothek mit entsprechender Zugriffs-Funktionalität fehlt, wurde ein anderes Verfahren gewählt um die Fahrtvektoren zu erzeugen. Demzufolge wurde vorerst und auch aus zeitlichen Gründen auf eine derartige Repositionierung verzichtet.

# Filterung der IMU Daten, Generierung eines Qualitätsarguments der Strecke

In der Testphase stellte sich heraus, dass der Vibrationssensor für die schwachen Vibrationen sehr empfindlich ist, wohingegen der Vibrationssensor für starke Erschütterungen auf Ereignisse während der Fahrt nicht regierte. Demzufolge sorgt der schwache Sensor für eine Registrierung einer Vielzahl an unerheblichen Erschütterungsereignissen. Mit Hilfe der zugehörigen IMU-Daten lässt sich eine Auswahl der relevanten Ereignisse jedoch durch die Auswahl eines Betrachtungsfensters jedoch leicht realisieren. Die Definition des Betrachtungsfensters geschieht im Moment noch großzügig auf der Annahme, dass Ereignisse im Betrag unter 4G keine Relevanz für die Betrachtung haben.

Die Generierung des Qualitätsmerkmals eines Streckenanschnitts beruht im Moment darauf, dass Ereignisse gleicher GPS-Koordinaten zusammengefasst werden und eine Aussage über die Qualität der Strecke geben, die sich zu den nächsten Punkten mit anderer GPS-Koordinate ergeben. Demzufolge ergibt sich die Qualität der Strecke abschnittsweise aus der Anzahl der Erschütterungsereignisse und der Summe der individuellen IMU-Ausschläge. Künftig ist es nötig das Betrachtungsfenster durch weitere Datenauswertung anzupassen.

# Erzeugung von Vektoren der gefahrenen Strecke mit verknüpften Qualitätsargument des Abschnitts

Im vorherigen Schritt wurden die Erschütterungsereignisse mit gleichen GPS-Koordinaten zusammengefasst und die Anzahl der Ereignisse, sowie die zugehörigen IMU-Daten wurden zur Generierung eines Qualitätsarguments verwendet. Eine Vektordarstellung wird nun dadurch erreicht, dass aufeinander folgende Koordinatensätze durch die *line*-Methode verbunden werden und die erzeugte Line erhält das Qualitätsmerkmal des zugehörigen Abschnitts. Diese Daten werden in einem *SpatialLinesDataframe* zusammengefasst, sodass der Export von georeferenzierten Daten möglich wird. Die vorangegangene Nutzung von ausschließlich Radwegdaten führte zu großen Lücken im Datensatz und zeigte, dass es nötig ist die Erzeugung von bestimmten Verknüpfungen von Koordinatenpaaren zu unterbinden. Hierfür wurde die Distanz die zwischen zwei Koordinatenpaaren möglich ist beschränkt. Die Datenfülle und die Verwendung von Straßendaten haben diesen Filtermechanismus weitgehend unnötig gemacht, da die Datenlücken nicht mehr so groß sind. Dennoch ist der Filter noch implementiert um eine Nutzung von spezifischeren Datensätzen/OSM-Daten zu ermöglichen.

# Generierung von KML der erzeugten Vektoren

Zur Visualisierung der gewonnen Erkenntnisse ist es nötig die Daten außerhalb von R auch nutzen zu können. Das vorher erzeugte *SpatialLinesDataframe* lässt sich recht komfortabel in eine KML-Datei umwandeln. Diese Datei enthält zurzeit die generierten Vektoren nebst ihres zugehörigen Qualitätsmerkmals. KML bildet auch die Grundlage der folgenden Visualisierung. Die Nutzung des KML Standards beruht somit auf der generellen Verwendbarkeit des Formats und resultiert aus dem angeschlossenen Schritt zur Visualisierung mit Hilfe von OpenLayers.

# Farbcodierte Darstellung der Vektoren und dessen Qualitätsarguments über OpenLayers

Ziel war es hier das Qualitätsargument der einzelnen Streckenanschnitte mit Hilfe von Farben zu kodieren. Innerhalb von R wird nun der gesamte Datensatz in drei Abschnitte in Bezug auf das Qualitätsargument geteilt. Die Festlegung auf drei Teile liegt vorerst in der Darstellungswahl der Daten auf der Karte begründet. Rot soll schlechte Qualität symbolisieren, Grün gute Qualität und Gelb für den Übergang. Da hier empirische Daten fehlen, beruht die Teilung noch auf dem Dritteln des Maximalwerts (schlechteste Qualität) des Qualitätsarguments.

# Perspektive

Primäres Ziel war es ein System zu entwerfen, welches die gewonnen Daten sinnvoll visualisieren kann. In Folge dessen sind einige Aspekte der Datenauswertung noch kompromissbehaftet.

* Zunächst sollten weitere Datensätze analysiert werden und validiert werden, damit das Betrachtungsfenster sinnvoller gewählt werden kann.
* Dazu ist es nötig den genauen Streckenverlauf als kml/geojson vorher zu recherchieren, damit das „Snapping“ der Daten auch durchgehend repräsentative Daten/Vektoren liefert.
* Die Teilung des Qualitätsarguments zur Visualisierung ist stark davon abhängig welche Aussage getätigt werden soll.
* Ist man sich sicher, dass die gewonnenen Vektoren über die Fahrtstrecke zureichend repräsentativ sind, kann eine Repositionierung der einzelnen Erschütterungsereignisse erfolgen.
* Künftig wäre es so möglich eine weitere Validierung der einzelnen Erschütterungen, deren Koordinaten und deren Repräsentation anhand der IMU-Daten zu ermöglichen.