# **Stand von MotionTrace (ehem. „Positions-Tracker“)**

## Neue Features & Erweiterungen seit dem 08.05.2023

* **Allgemeines**
  + Die App startet bei Aufnahme nun einen Foreground-Service, der es erlaubt den Standort-Service auch dann zu nutzen, wenn die App im Hintergrund läuft.
  + Überarbeitung der Benutzeroberfläche (Platzierung von Bedienelementen, Status-Texte, Hinweisnachricht bei fehlenden Berechtigungen)
  + Überarbeitung des Stylings (App-Icon, App-Name, Farben, Icons)
  + **Bugfixes**
* **Informationen & Hilfe** (Seite)
  + Hintergrund
  + Hinweise **ANGEPASST**
  + Bedienungshilfen **ANGEPASST**
* **Einstellungen** (Seite)
  + Steuerung der Ausgabe (Speichern auf dem Gerät, Senden per MQTT)
  + Bearbeitung des MQTT-Brokers (IP, Port, Topic) und -Clients (Identifikation)
  + Links zu wichtigen Geräteeinstellungen, die die Funktion der App direkt beeinflussen: App-Berechtigungen, Standorteinstellungen, Verbindungseinstellungen
  + Alle Einstellungen werden persistent auf dem Gerät gespeichert
* **Gespeicherte Aufnahmen** (Seite)
  + Auflistung aller auf dem Gerät gespeicherten Aufnahmen
  + Zusammenfassende „Statistiken“
  + Erlaubt die Verwaltung der Aufnahmen (Teilen & Löschen)
  + Aufnahmen können geladen und anschließend in der Karte betrachtet werden

## Offene Probleme/Schwierigkeiten

* Aufzeichnung im Hintergrund funktioniert erst zweifelsfrei, wenn der Nutzer manuell (!) gewisse Berechtigungen erteilt. Eine Berechtigungsanfrage aus der App heraus führt zu einem Absturz der App.
  + **GELÖST: Die App fragt nun beim ersten Starten nach allen benötigten Berechtigungen!**
* MQTT-Verbindung kann während einer Aufnahme abbrechen, bspw. wenn ein Wechsel von WLAN auf Mobile-Daten oder umgekehrt passiert. Dies führt aktuell noch zum sofortigen Ende einer laufenden Aufzeichnung.
  + **GELÖST: Die App versucht nun wenn möglich die Verbindung automatisch wieder aufzubauen, wenn diese verloren geht. Eine Aufnahme wird bei einem Verbindungsabbruch nicht mehr beendet (ggf. kommt es zu Datenverlust / einer Lücke der über MQTT versendeten Daten) sondern läuft weiter.**
* Die App läuft aktuell nur auf Android-Geräten. Für iOS-Geräte müssen teile der App umprogrammiert und konfiguriert werden (das ist prinzipiell kein Problem). Außerdem kann die App ausschließlich von einem Mac/MacBook für iOS-Geräte gebaut und kabelgebunden auf diese installiert werden. **Test-Flight, kann nur von Mitgliedern des Apple-Developer-Programms (99€/Jahr) administriert werden.**

**Für Android-Geräte ist die App aus meiner Sicht fürs Erste abgeschlossen.**

# **Datensammlung mit der App: Betaphase**

## Bisher gesammelte Daten (Dauer und Distanz):

**Siehe Diagramme in PyCharm-Projekt.**

## Städte/Orte der Aufnahmen:

*Mengersgereuth-Hämmern* (größeres Dorf)

*Sonneberg (Kleinstadt; primär)* und *Neustadt* *bei Coburg (Kleinstadt)*

*Coburg (Stadt)* und *Rödental (Kleinstadt)*

*Kronach* (Kleinstadt)

Vereinzelte Aufnahmen in *Bamberg* (Stadt; als Fußgänger und Auto) und kleineren Dörfern in der Region.

**Siehe Visualisierungen in PyCharm-Projekt.**

## Verkehrssituationen

Im Wesentlichen wurde bisher „ruhiger Stadtverkehr“ außerhalb von Stoßzeiten aufgezeichnet. Vereinzelt besondere Situationen, wie längere (5min+) Staus.

## Ziele für weitere Datensammlung

* Idealerweise Automatisierung der Datensammlung (z.B. über dauerhaft erreichbaren MQTT-Broker von Seiten der Hochschule & ein 24/7 laufendes Skript zum Abfragen [per Subscription] und Speichern der Daten)
* Ausweitung der Datensammlung:
  + Mehr sammelnde Personen/Nutzer der App werden benötigt (bis dahin sollten letzte Probleme jedoch noch beseitigt werden) um Einflüsse abzudämpfen, die auf vermehrt genutzten Fahrzeugmodellen und individueller Fahrstile basieren
  + Aufnahmen von Motorradfahrern sollten verstärkt in den Fokus rücken
  + Mehr Aufnahmen in anderen Verkehrssituationen (gilt für alle Klassen), insbesondere Aufnahmen zu Stoßzeiten

# **Weitere Erkenntnisse zu den Forschungsfragen**

## **Formulierte Forschungsfragen:**

1. **Wie können reale Positionsdaten, die ggf. Ungenauigkeiten und Rauschen aufweisen, so vorverarbeitet werden, dass sie sich gut für den Einsatz maschineller Lernverfahren eignen?**
2. **Welche Klassifikationsverfahren sind für die Klassifizierung von Verkehrsteilnehmern auf Basis realer Positionsdaten geeignet?**

## **Zu A) / Datenvorverarbeitung:**

## Identifizierte Probleme bei der Aufnahme:

1. **Rauschen/Ausreißer**
   1. Quer zur Fahrtrichtung

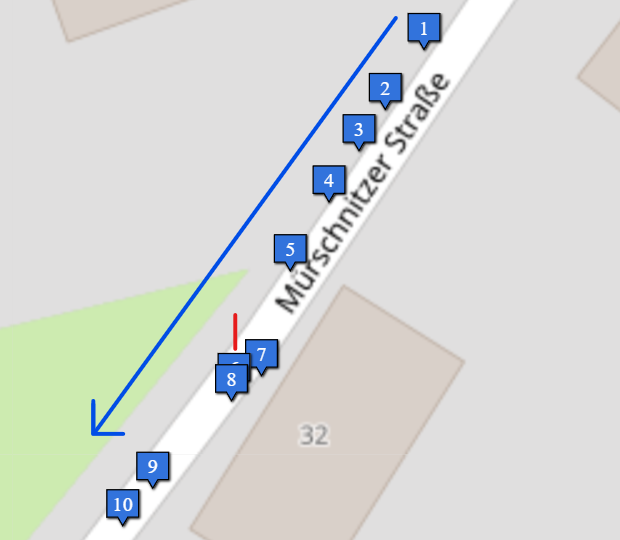
Punkte liegen links oder rechts von der eigentlichen Strecke. 🡪 Gegebenenfalls problematisch für nicht gematchte Daten, **durch Map-Matching jedoch sehr gut behandelbar**.

* 1. Parallel zur Fahrtrichtung

Punkte liegen vor oder hinter der eigentlichen Position, aber auf der gefahrenen Strecke. 🡪 Das ist schlecht, da dies nicht durch reines Map-Matching behandelt werden kann. Außerdem können hierdurch die berechneten Geschwindigkeits- und Beschleunigungswerte erheblich beeinflusst werden.

*Extremfall*: Ein Punkt verrutscht horizontal so stark, dass er hinter dem zuletzt aufgezeichneten Punkt liegt. 🡪 Gefahr vor allem bei geringen Geschwindigkeiten. Ggf. hilft es die Sampling-Rate zu reduzieren, was jedoch auch zu Informationsverlust führen kann.

Beispiel: Horizontales Rauschen führt zur „Überlappung“ (Aufnahme mit Fahrrad bei ca. 20km/h)



1. ***Dopplungen***

Manchmal liefert der Location-Service eines Smartphones trotz Bewegung des Geräts aufeinanderfolgend mehrfach dieselbe Position (gekennzeichnet durch denselben Timestamp). Dies führt in seltenen Fällen dazu, dass erst nach mehreren Sekunden eine neue Position vorliegt.

Aktuell werden in der Datenvorverarbeitung alle doppelten Einträge (Koordinaten inkl. Timestamps) gelöscht. Geschwindigkeiten und Beschleunigungen werden auf Basis der Zeitdifferenz (berechnet aus den zugehörigen Timestamps) zwischen aufeinanderfolgenden Punkten und nicht auf Basis der Sampling-Rate (1 Sample/s) berechnet.

1. ***Schlechtes bzw. unterbrochenes GPS-Signal***

Manchmal kommt es unvermeidbar zum Verlust des GPS-Signals. Dies führt zum mehrfachen „Logging“ des letzten bekannten Standorts bis zu dem Zeitpunkt, an welchem das Signal wieder hergestellt wird. Kurz nachdem das Signal wieder hergestellt wird weisen die Punkte eine Tendenz zum starken Verrauschen auf.

Beispiel: Signalverlust und Dopplungen durch eine Unterführung

Ein Bild, das Karte, Reihe, Text enthält.

Automatisch generierte Beschreibung

1. **Stottern**
2. **Verrutschen**

## Map Matching (kein neuer Stand)

Erste Versuche mit „Valhalla“, einer Open-Source Routing und Map-Matching Engine, die als Instanz lokal in einem Docker-Container installiert wurde. (Basis: [A Practical Guide to an Open-Source Map-Matching Approach for Big GPS Data | SpringerLink](https://link.springer.com/article/10.1007/s42979-022-01340-5)) Valhalla ist sehr mächtig und erlaubt es eine Vielzahl an Zusatzinfromationen zu einzelnen Straßen und Straßensegmenten allein auf Basis von gematchten Positionsdaten abzufragen.

Beispiel: Fahrt mit Motorrad durch Kreisel (grün = map-matched)

Ein Bild, das Karte enthält.

Automatisch generierte Beschreibung

Ergebnisse dieser ersten Versuche sind zum Teil im Großen und Ganzen vielversprechend, allerdings wurden auch bereits einige Schwächen offenbart.

Negativbeispiel 1: Verzerrung auf Grund ungenauer Karte

Ein Bild, das Karte enthält.

Automatisch generierte Beschreibung

Negativbeispiel 2: Verzerrung durch falsches Matching

Ein Bild, das Text, Im Haus enthält.

Automatisch generierte Beschreibung

Negativbeispiel 3: Starke Verzerrung durch falschen Matching-Modus

Ein Bild, das Diagramm enthält.

Automatisch generierte Beschreibung

## **Zu B) / Machine Learning Verfahren**

## Verfahren und Methodik (kein neuer Stand)

# Bereits eingesetzte Verfahren umfassen einfachere (K-Nearest-Neighbours, Decision-Tree und SVMs) und komplexere Klassifikatoren (Deep Q-Learning- & Feed-Forward-Netz), die darauf trainiert wurden die Klasse eines Verkehrsteilnehmers auf Basis einer festen Anzahl sequenziell aufgenommener Positionsdaten zu bestimmen.

**Am Beispiel der Arbeit von Dietmar Fischer:**

Aus einer Sequenz von 100 Koordinaten, die in einem zeitlichen Abstand von 0,5s aufgenommen wurden, werden zusammenfassende Merkmale errechnet (z.B. die minimale, durchschnittliche und maximale Geschwindigkeit oder Beschleunigung), die zur Klassifikation des Verkehrsteilnehmers genutzt werden. 🡪 Basis für die Klassifikation bildet also ein statisches 50 Sekunden Aufnahmeintervall.

Nur 3 Klassen: Fußgänger, Fahrradfahrer und motorisiertes Fahrzeug (keine weitere Differenzierung zwischen den motorisierten Fahrzeugen).

**Probleme bei bisherigen Ansätzen:**

* Bisher keine hinreichend guten Ergebnisse bei der Klassifizierung verschiedener Typen motorisierter Fahrzeuge.
* Aufgenommene Sequenzen, werden durch die Zusammenfassung aller 100 Koordinaten nicht als Sequenzen betrachtet, aber ggf. liegt genau hier der Schlüssel zum Erfolg.
* Es gibt im Straßenverkehr häufig Szenarien, in welchen eine klare Unterscheidung verschiedener Fahrzeugtypen anhand eines festen Intervalls nicht möglich sein wird („Motorrad beim Beschleunigen durch vorausfahrendes Auto begrenzt“, „Auto auf kurviger Landstraße hinter Fahrradfahrer“, Ampeln, Staus…). Bisherige Ansätze kommen in solchen Szenarien an ihre Grenzen.

**Idee für besseres Verfahren:**

1. Aufnahme in Intervalle fester Größe aufteilen z.B. in Zeitspannen von 10 Sekunden.
2. Die einzelnen Intervalle durch Merkmale, ähnlich wie bei Dietmar, zusammenfassen.
3. Alle Intervalle einer Aufnahme als Sequenz betrachten.

Eine Klassifizierung soll also nicht von den Merkmalen eines einzigen Aufnahmeintervalls, sondern auch von der Klassifizierung des vorherigen Intervalls abhängen.

Hierfür eignen sich insbesondere ***Rekurrente Neuronale Netzwerke***, die ihre Ausgabe zum Zeitpunkt t als zusätzliche Eingabe im Zeitpunkt t+1 zur Verfügung stellen.

Erste Recherchen zu RNNs in diesem Kontext erweisen sich als vielversprechend, allerdings sind evtl. gegebene Nachteile aktuell noch nicht absehbar (z.B. wäre es möglich, dass mehr Trainingsdaten als bei anderen Ansätzen benötigt werden).

Eine Arbeit, welche einen solchen Ansatz verfolgt ist: [Vehicle classification from low-frequency GPS data with recurrent neural networks - ScienceDirect](https://www.sciencedirect.com/science/article/abs/pii/S0968090X18304017?via%3Dihub).

## Welche Merkmale eignen sich zur (sequentiellen) Klassifikation?

**Rohdaten pro Punkt:** Reine Koordinaten mit Timestamp und geschätzter Genauigkeit.

**Abgeleitete Merkmale pro Punkt**: Geschwindigkeit, Beschleunigung, Orientierung und Winkelgeschwindigkeit.

**Lagemaße über „Fenster“**: Minimum, Maximum, Median, Arith. Mittel, Quantile

**Streuungsmaße über „Fenster“**: Varianz od. Standardabweichung