

Fachpraktikum Algorithms on OpenStreetMap Data: eMaps

Tobias Mathony

March 16, 2020

Abstract

Electric vehicles, such as eBikes or eCars, play an important role in today's

1 Introduction

Climate change and its consequences heavily impacted the engineering of means of transport. As a result, vehicles powered by electric batteries emerged, such as electrically-powered cars, bikes or scooters. By using regenerative energy sources, electrically-powered vehicles have the potential to significantly reduce the dependency of fossil fuel reserves [AHLS10]. Furthermore, electric vehicles emit no emissions, hence, being more eco-friendly than vehicles running on gasoline. However, electric vehicles have characteristics that currently hinder its wide-spread adaption, i.e. (i) limited cruising range, and (ii) long recharge times[AHLS10]. Especially the limited cruising range require an adaption of existing navigation and routing systems. We need to determine routes for electric vehicles considering the availability of charging stations and the range of electric vehicles to avoid running out of power.

Using OpenStreetMap¹ data, we demonstrate the implementation of a route planner based on the range of an electric vehicle and the availability of charging stations.

2 Foundations

This section briefly describes the foundations of the implementation such as OpenStreetMap and Dijkstra.

2.1 OpenStreetMap

This project was implemented on OpenStreetMap data. OpenStreetMap (OSM) is a community-driven open-source project that provides user-generated map data [?]. Besides that, also further geographical information like charging stations, bars, clubs or restaurants are available. OpenStreetMap data is available in different formats with parsers for the most popular programming languages.

2.2 Dijkstra

3 Implementation

This section outlines the architecture of the application, the used technologies and the main features of the implementation.

¹<https://www.openstreetmap.org>

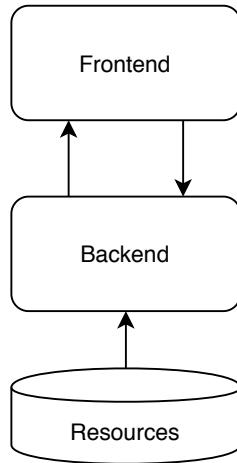


Figure 1: Conceptual overview of the application

3.1 Architecture and Technologies

For the architecture of the application, the common **Model-View-Controller** paradigm was used for a strict separation of functionality and an increased re-usability of the components as depicted in Figure 1.

The **Resources** represent the raw OpenStreetMap data which was provided in the *Protocol-buffer Binary Format (PBF)*².

The **Backend** provides the core functionality and implements algorithms on the OpenStreetMap data. It is implemented as Web Server in *Rust*³. The core functionality of the Backend includes (i) a parser to map the OpenStreetMap data to a *Domain Object Model (DOM)* from Rust, (ii) a shortest path algorithm, and (iii) an *Application Programming Interface (API)* to expose the functionality for access from the **Frontend**.

The Frontend is built using the *React*⁴ Web Framework and *Leaflet*⁵. Leaflet is an open-source JavaScript library for interactive maps based on OpenStreetMap. Leaflet is used to provide a user-friendly map and allows to e.g. set markers or draw routes. Figure 2 shows the basic look of the Frontend. Furthermore, the *Nominatim*⁶ API is used to allow the user to search for places, cities or Point-Of-Interests (POIs). The Frontend interacts with the Backend via *HTTP* requests.

3.2 Features

4 Conclusion and Future Work

References

- [AHLS10] Andreas Artmeier, Julian Haselmayr, Martin Leucker, and Martin Sachenbacher. The shortest path problem revisited: Optimal routing for electric vehicles. In Rüdiger Dillmann, Jürgen Beyerer, Uwe D. Hanebeck, and Tanja Schultz, editors, *KI 2010: Advances in Artificial Intelligence*, pages 309–316, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

²https://wiki.openstreetmap.org/wiki/PBF_Format

³<https://www.rust-lang.org>

⁴<https://reactjs.org>

⁵<https://leafletjs.com>

⁶<https://nominatim.openstreetmap.org>

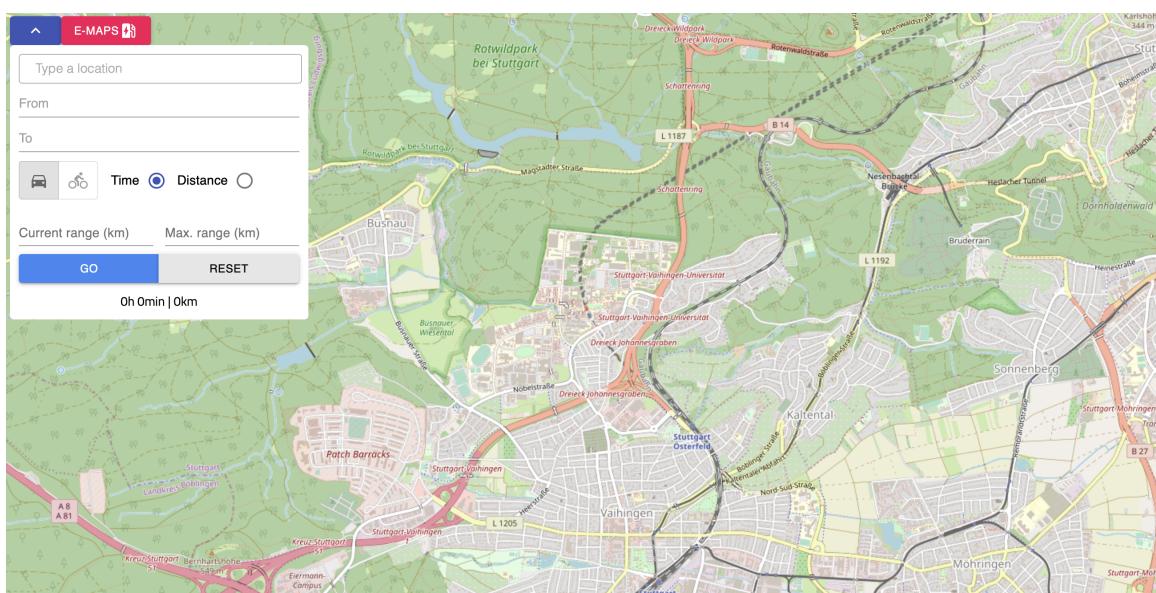


Figure 2: Screenshot of the Frontend