Visualization of lidar, camera, and vector data from a railway mobile mapping system

BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

author: Zuzana Miškaňová
supervisor: Ing. Ondřej Klíma, Ph. D.

Excel@FIT 2025

Goal

Create a customizable and user-friendly web application which will display the data from the train operator's perspective, including train movement animation.

Results

A working web application capable of rendering a point cloud which includes 1.5×10^6 points at 46-52 FPS. The new application has a potential to be exploited within the field of mobile mapping system data visualization.

The visualized data

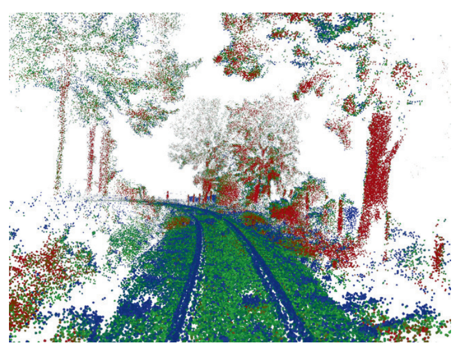


Figure 2 - point cloud

- divided or united

Figure 3 - video

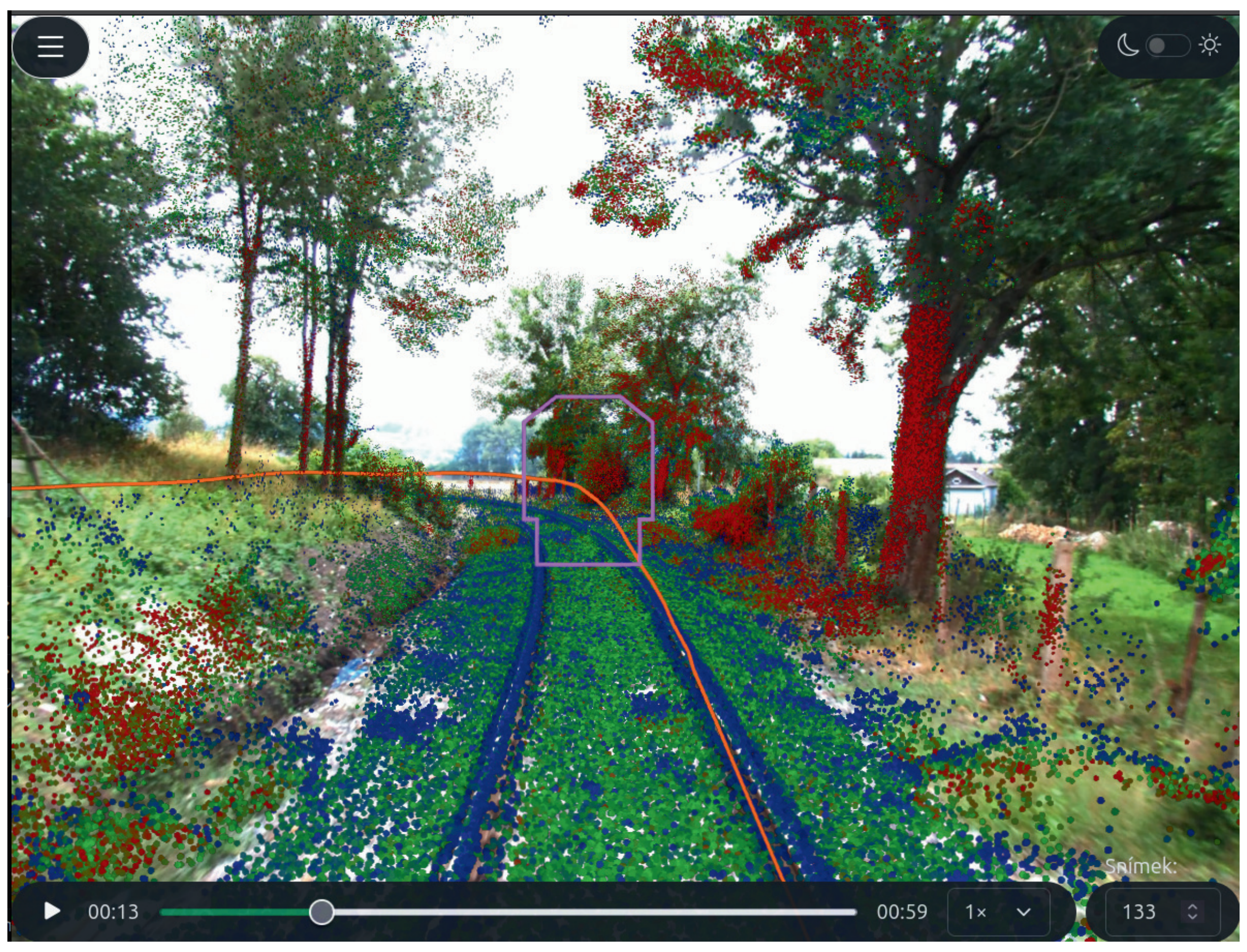


Figure 1 - application main screen

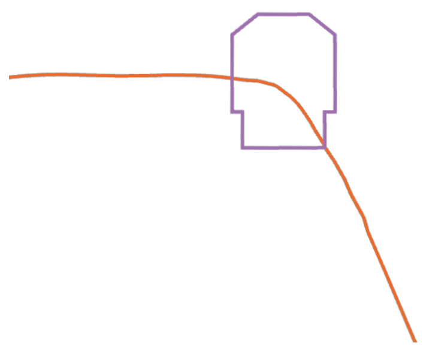


Figure 4 - vector data

- train profile
- line through train profile positions
- other vector data

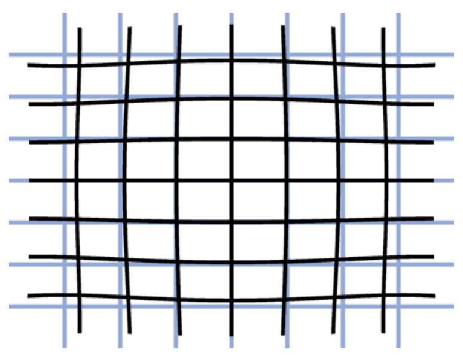


Figure 5 - distortion

The application visualizes many kinds of data, which have to be put together and synchronised according to camera positions (translations + rotations) and two sets of timestamps.

Camera movement

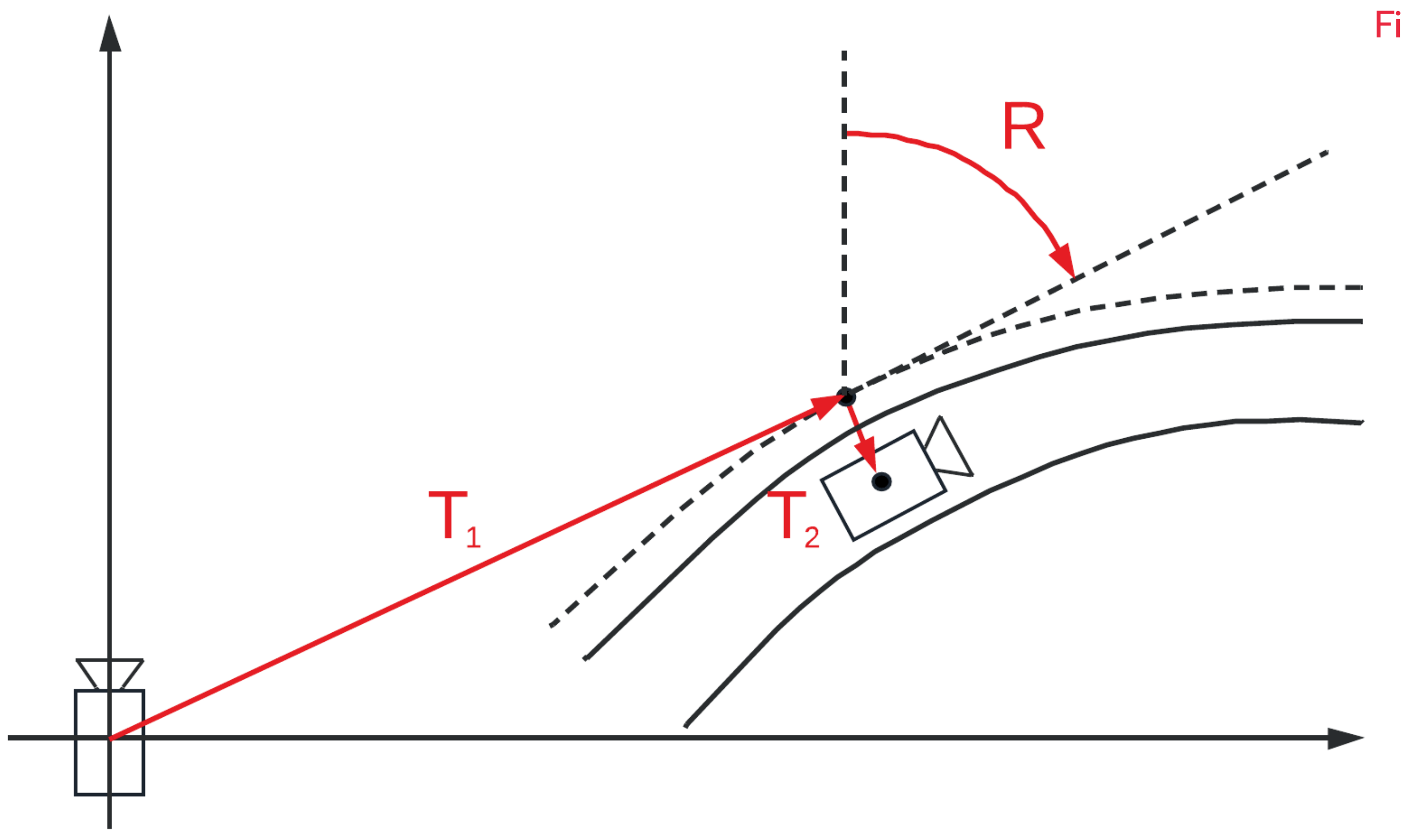


Figure 6

The application enables the user to add custom camera offset and angle to correct potential inaccuracies in the uploaded data.

Deck.gl layers

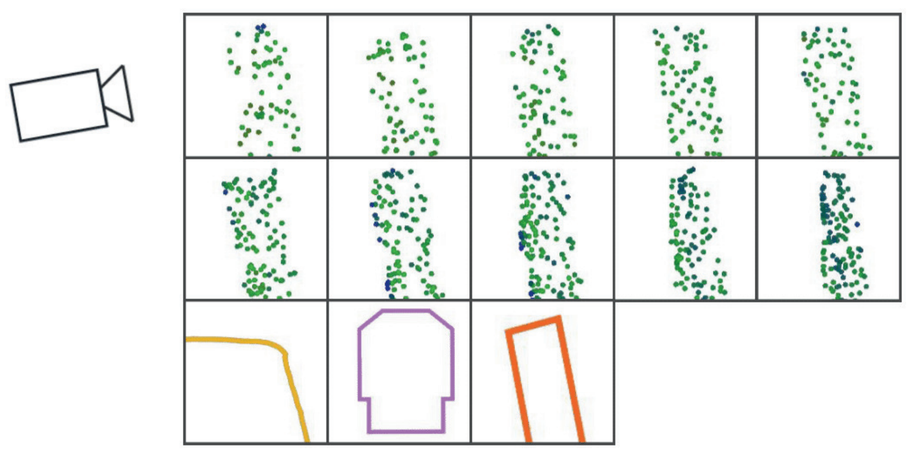


Figure 7

The visualization is composed of multiple point cloud and vector data layers. Some layer's data needs to be changed when the animation of the movement of the train is running.

In the case that the point cloud consists of small, gradually scanned chunks, the visualization displays ten chunks, which are renewed in a circular way when the animation of the movement of the train is running.

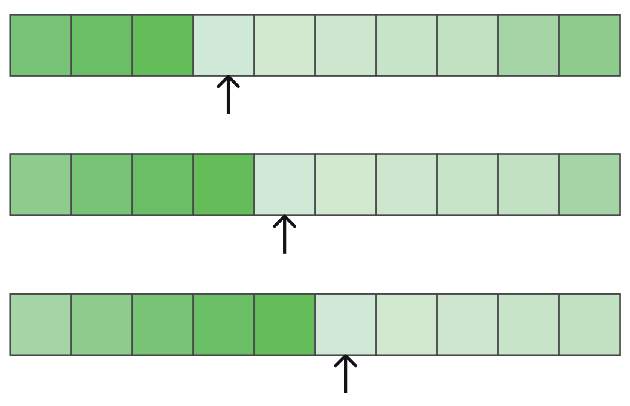
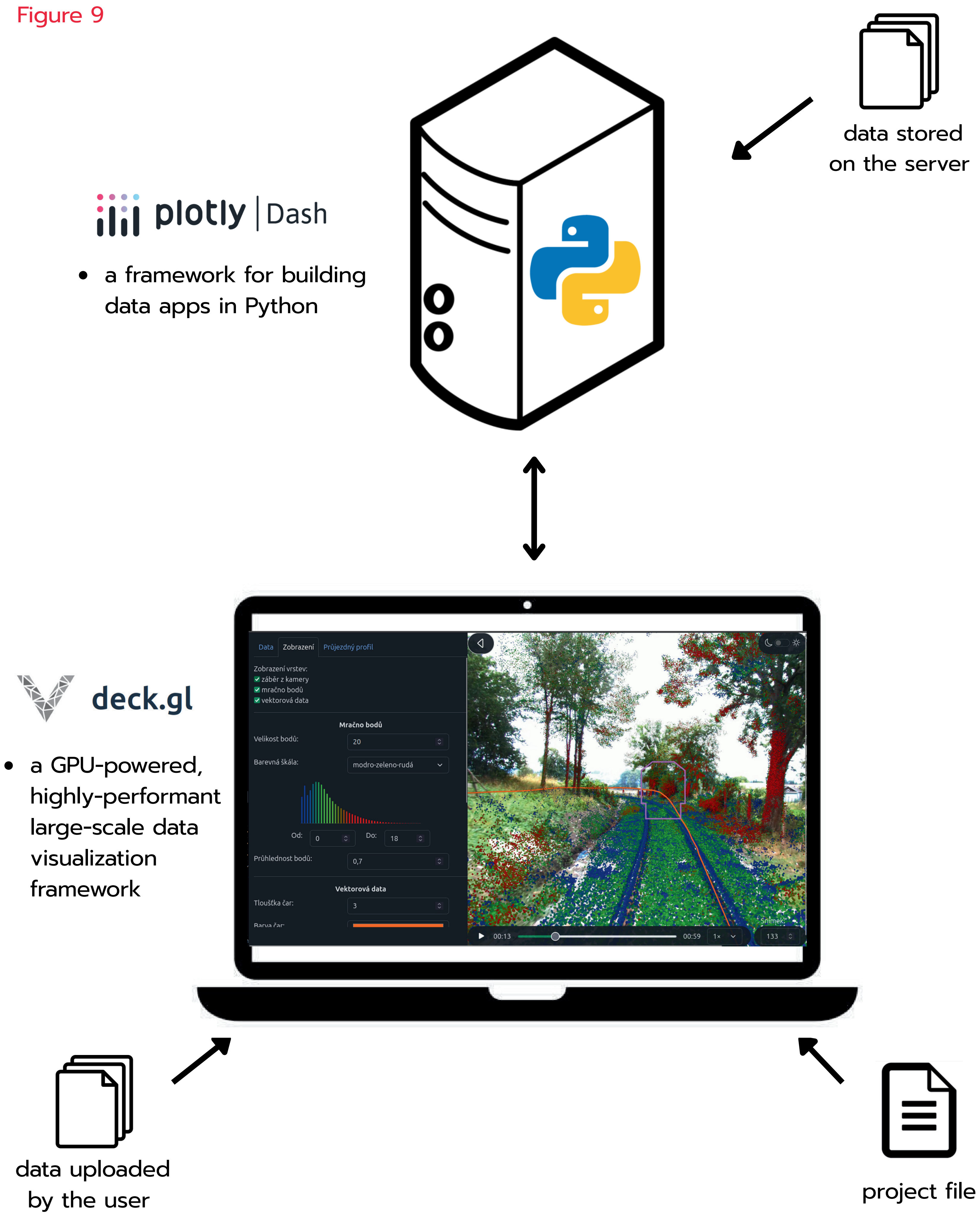


Figure 8

Implementation

Figure 9



data stored on the server

- a framework for building data apps in Python

- a GPU-powered, highly-performant large-scale data visualization framework

data uploaded by the user

project file

The application is written mainly in Python using framework Dash, but for optimization it includes a script written in JavaScript which runs on the client and controls the visualization. In case that the user runs the application on localhost or has access to the server, there is a possiblity to load a project file to avoid loading all the data files separately.