Project 1

Detecting Moving Objects in DAS Recordings

Project Goal: Develop an algorithm to detect and track moving objects in acoustic data collected from a Distributed Acoustic Sensing (DAS) system attached to the fiber optic cable at Jana Pawla II Street.

Distributed Acoustic Sensing (DAS) is an advanced technology that transforms optical fiber cables into highly sensitive vibration sensors. By sending pulses of laser light through a fiber optic cable and analyzing the backscattered light, DAS systems can detect subtle strain variations along the fiber, which may be caused by seismic activity, traffic, or small object movements. These variations are recorded as phase shifts in the backscattered light signal, which correlate directly to the fiber's strain. With appropriate calibration, these phase shifts are processed to measure strain or strain rate over time. The resulting time-series data provides a detailed, chronological record of vibrations along the fiber, allowing for further analysis to identify the source, magnitude, and frequency of disturbances.

(Version for 3)
Your task is to:

1. **Load the data.**
   In the report, you must present analyses for **three continuous time segments**, each **2 minutes long**.
   You may also use the entire available dataset for purposes such as calculating overall signal statistics.

2. **Perform a frequency analysis** of the signal.
   Remember that the **FFT should be applied independently to each channel**.

3. **Filter out the noise**, which may include:
   - removing unnecessary frequency components,
   - downsampling,
   - applying appropriate filters,
   - standardization or rescaling of the data,
     and any other preprocessing steps you find useful.
4. **Process the data** so that **only moving objects (slanted lines)** are visible.

5. **Determine the velocity** of these objects (see the figure on the next page for reference).

If your algorithm detects many potential line candidates, you can group these lines and extract the average line from the group.

You may have multiple processing paths responsible for detecting or filtering different elements in an image. However, the approach should be general, and you cannot directly hardcode the position of lines or any elements directly related to specific individual lines.

(Version for 4)

On the images, there are clearly visible **thick, strong lines** (corresponding to the passage of a **tram or a heavy truck**) and **fainter, thinner lines** (corresponding to individual cars).
To receive a **grade of 3**, it is sufficient to detect the thick, clearly visible lines.
To receive a **grade of 4**, you must also detect the thinner, weaker lines.

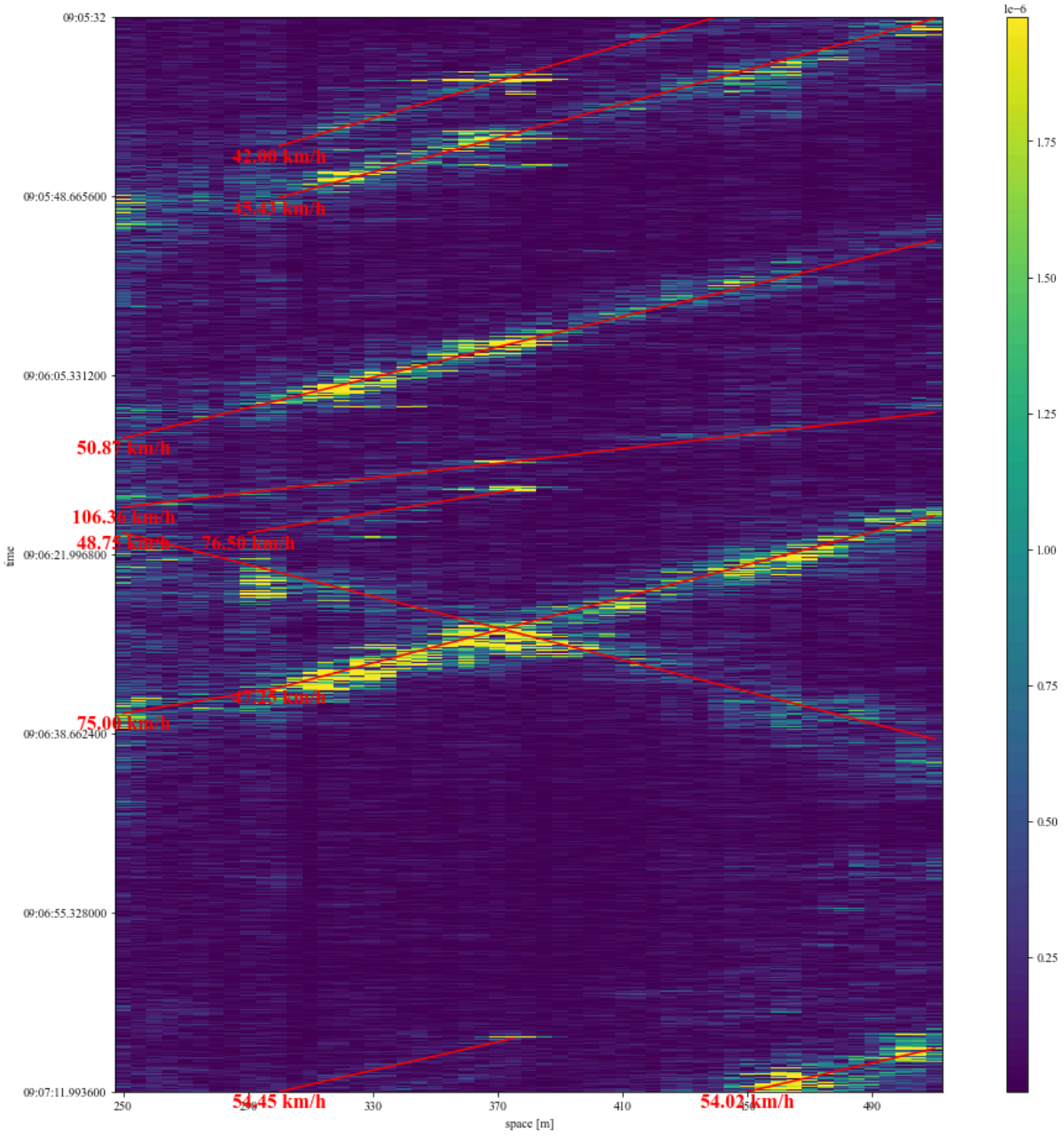
(Version for 5):

1. Instead of providing only a single value for each object's speed, **plot a graph showing how the velocity changes over time**.
2. Additionally, **perform clustering** of the detected objects into several groups that differ in the characteristics of their signals, for example:
    - thick lines,
    - thin lines,
    - objects moving at a constant speed,
    - objects with varying speed.
    - shape of the lines

Input Data:
The data is stored in numpy format as a 2D matrix. The particular value represents the strain rate of a fiber optic cable located on a busy street (Jana Pawla II). The data shows the passage of trams, trucks or cars along this street.
The first dimension is time, the second is space, in order to calculate the correct units use the following metadata:

- dx: 5.106500953873407 [m]
- dt: 0.0016 [s]
- file duration: 10s
- number of time samples in file: 6250
- file name format: HHMMSS
- files date: 2024-05-07

- Link: https://drive.google.com/file/d/1IJKLz3LsQmnAf9q5GGi6arEYBz-3CMhx/view?usp=sharing
- There are 259 files available in the file (note they are not continuous) that can be used as a learning set.
- Example of loading and displaying input data: https://drive.google.com/file/d/1TTCqmtQ2ddT5l401RGq3ngQa0NtJsfYD/view?usp=sharing



Requirements

- The implementation of the algorithm must be created in Python with use the following image processing libraries: OpenCV, NumPy, Matplotlib, Pillow, and Skimage. If You want to use other libraries, consult by e-mail (do not use neural networks or external tools).
- The solution must contain the **source code** and a **report** describing the algorithm along with a description of intermediate steps and the experiments performed (if you have tested several successful/unsuccessful solutions, describe them)

- The solution can be presented in the form of source code (.py) and an attached report (.pdf) or a jupyter notebook file (.ipynyb and one of .pdf or .html) containing the code and description of the next steps
- Images in the report can be scaled down so that the report is not too large.
- All sources of data, code and inspiration must be reported with a description of what was used.

The project can be done in pairs.
Deadline for sending the code, and report: 26.11.2025 23:59
The project will be presented in class: 24.11.2025 / 26.11.2025

Evaluation

The project will be assessed based on the following criteria:
Source code:

1. correctness of implementation (e.g. is the code executable and does not contain warnings and errors),
2. correct use of known image processing techniques,
3. the complexity of the solution, conducting additional tests, analyses performed
4. processing optimization (e.g. frequent use of loops to iterate over pixels instead of broadcast operations),
5. completeness of the source code for running the application (e.g. downloading input data),
6. number and accuracy of detected objects
7. code aesthetics and comments

Report:

- The way of presenting intermediate results (visualizations, descriptions),
- **Data set analysis** including several statistics or plots for images
- The report should include the algorithm presented in the form of a **block diagram or flowchart** with a description of the subsequent processing steps.
- Each person has been assigned a range of test files in which 2 minutes of data are covered https://docs.google.com/spreadsheets/d/1QgKmiEHVqAeIR-amm2fypDkoUoC8Y1IRxvsp8C-dH4c/edit?gid=0#gid=0 In the report, show detailed results for 3 time ranges of 2 minutes each. If you are working in a pair then you need to select another arbitrary area that does not overlap the two resulting from the assignment, and if the project is done alone then you need to select 2 time ranges.
- analysis and conclusions of the obtained results,
- factual and methodological errors
- potential references to literature,