ETH zürich

Autonomous Systems Lab

Extrinsic Camera Calibration for Trains using Maps

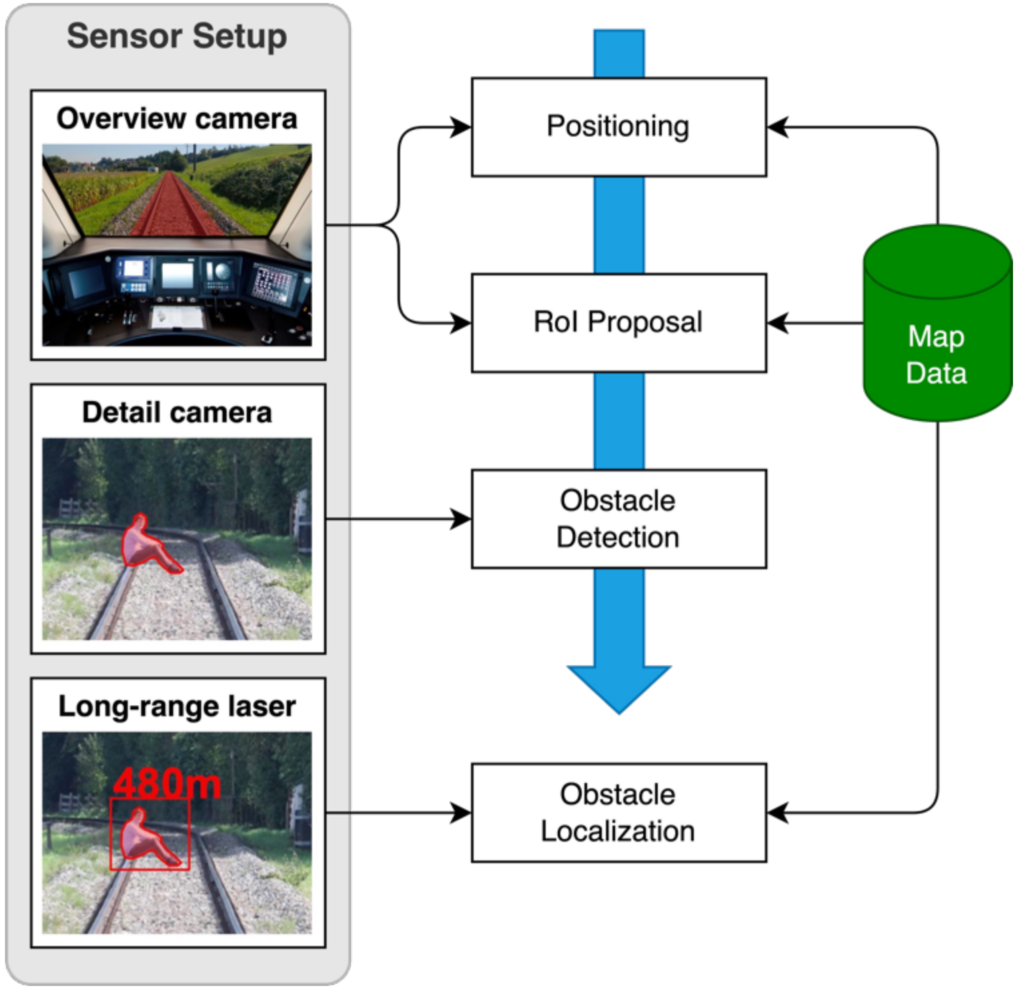
Eric Tüschenbönner, Cornelius Von Einem, David Hug, Prof. Roland Siegwart

Autonomous Systems Lab, ETH Zurich

Motivation

"Long-Range Obstacle Detection for ADAS"

PhD research by Cornelius von Einem



Motivation

"Long-Range Obstacle Detection for ADAS"

PhD research by Cornelius von Einem

Requirements:

- Precise extrinsic camera calibration
- Map overlay for downstream applications

Overview camera

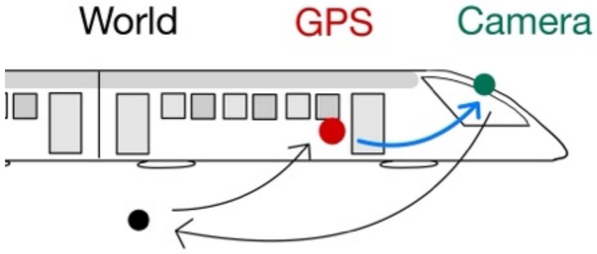
Problem Description

Extrinsic camera calibration & reprojection pipeline based on visual cues and map information.

Available data:

- Images
- Railway map
- GPS measurements

Setup:




```
heading: -30.405

p_x:      371443.956
p_y:      5804239.967
p_z:      40.445

q_w:      0.965
q_x:      0.001
q_y:     -0.002
q_z:     -0.262
```

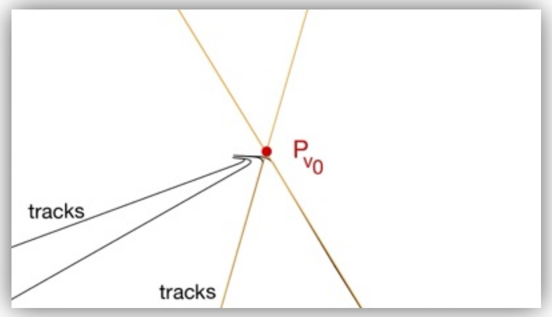
Problem Description

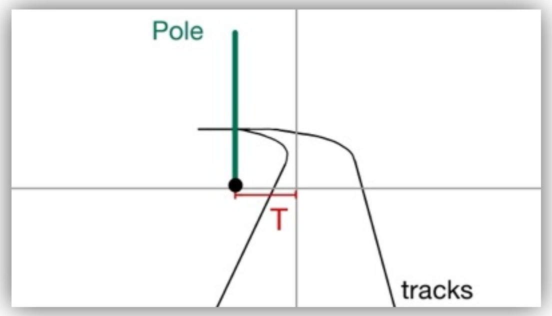
Related work [1]

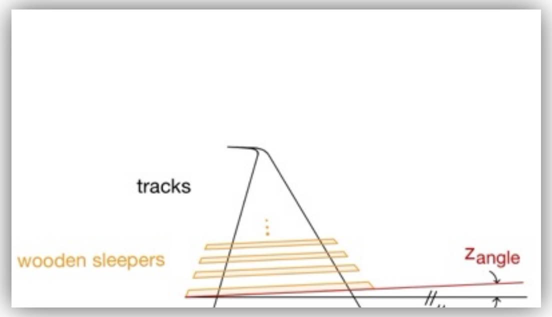
- Geometric approach
- Individual frames, straight track
- Not generalizable / reliable on multiple frames

My solution:

- Reproject 3D map into camera view
- Formulate optimization problem
- Multiple frames

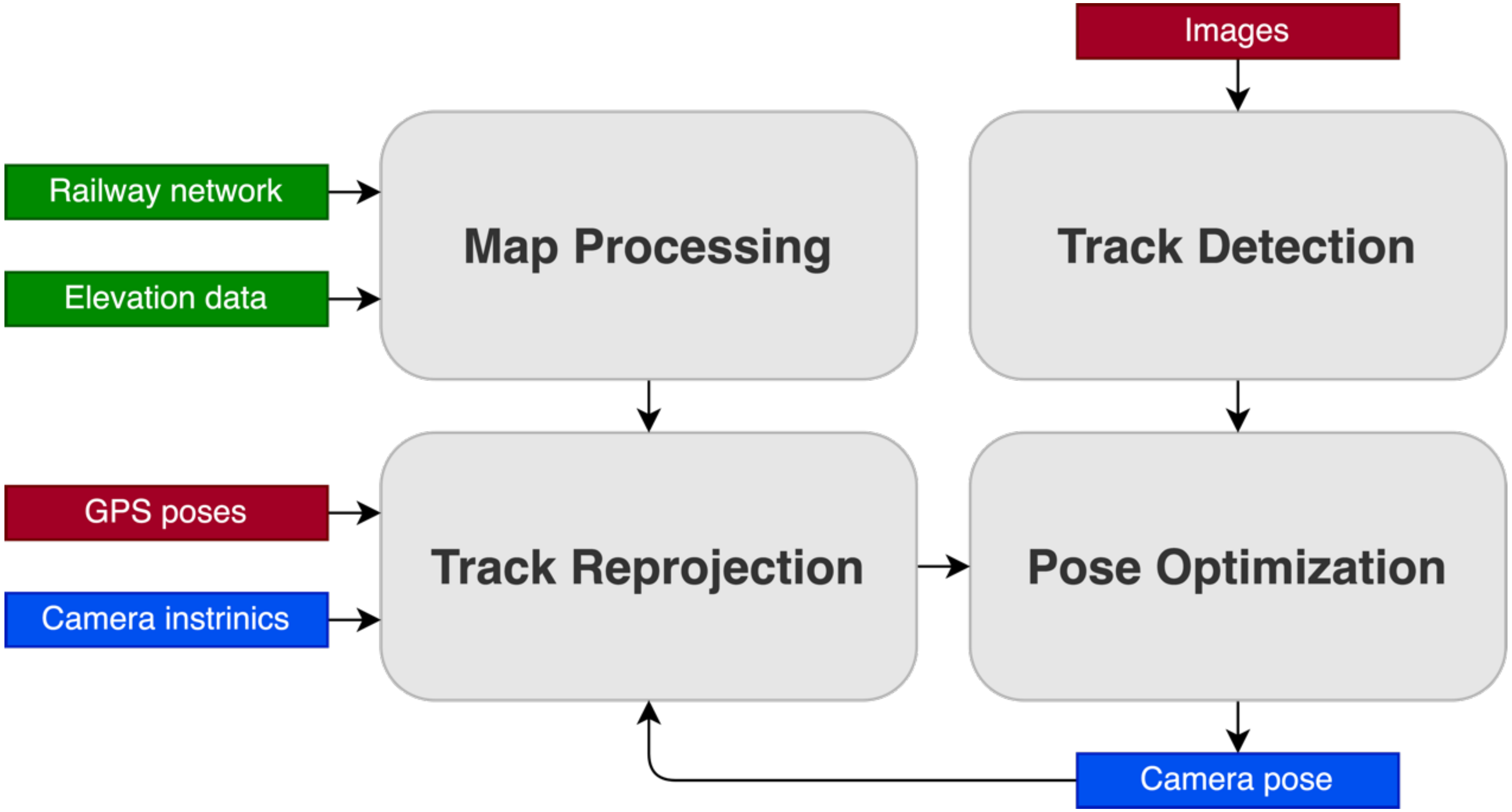






1. N. Spiegelhalter, C. Von Einem, and D. Hug, "Online estimation of camera extrinsics using map information," 2023.

Method | Overview

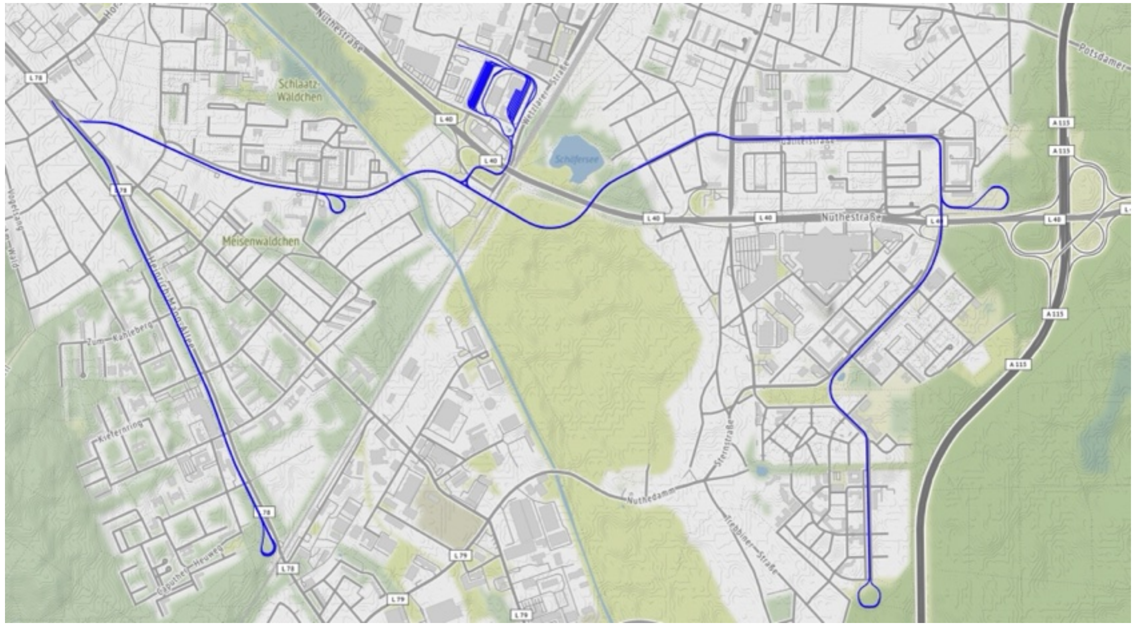


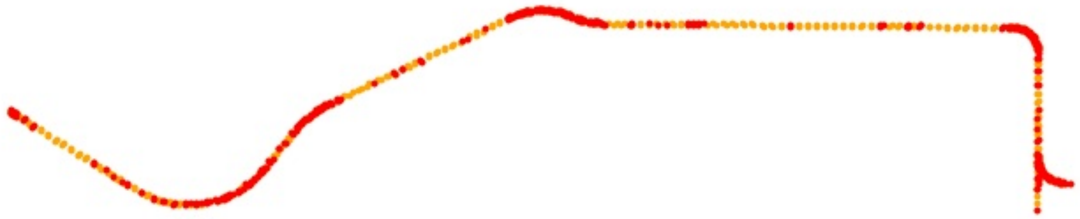
Method | Map Processing

Input:
- Railway network
- Elevation data

Process:
1. Extract nodes from railway network
2. Convert tracks to 2D splines
3. Interpolate to fill gaps
4. Add elevation data to points

Output:
- Railway tracks as 3D point clouds



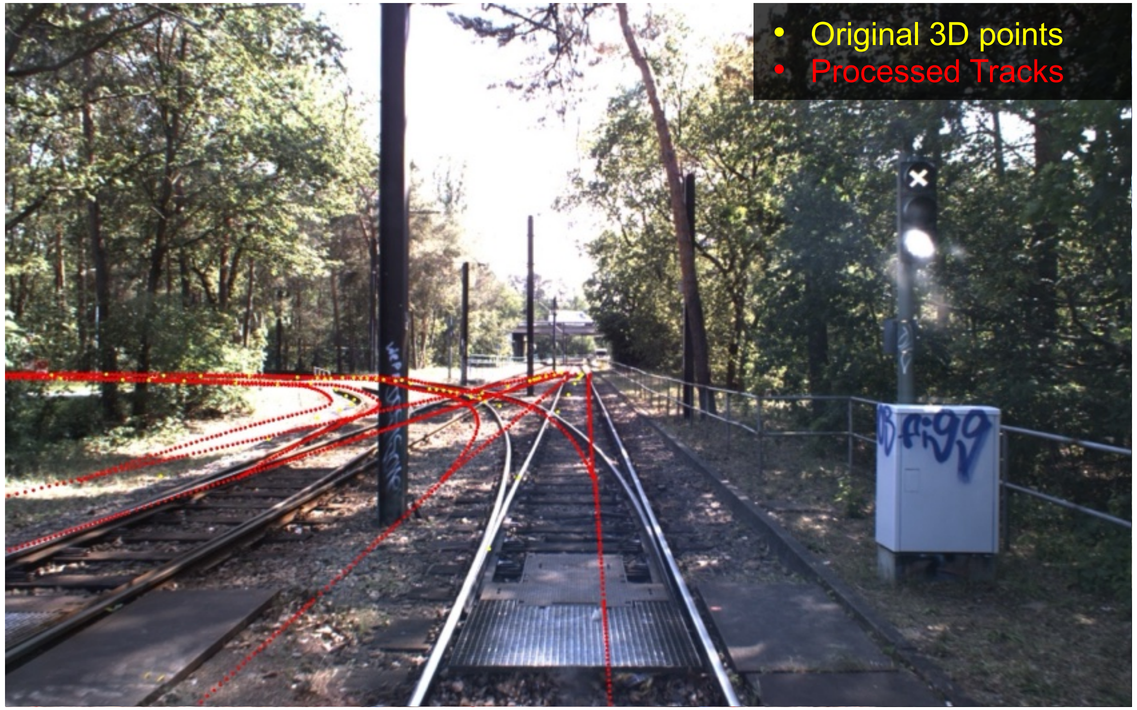


Method | Track Reprojection

Input:
- Railway tracks as 3D point clouds
- GPS pose
- Camera pose estimate

Process:
1. Find local railway tracks
2. Adjust & interpolate points
3. Reproject points onto image

Output:
- Reprojected local tracks



Method | Track Detection

Input:
- Image

Process:
1. Manual annotation
2. Convert to splines
3. Increase point density

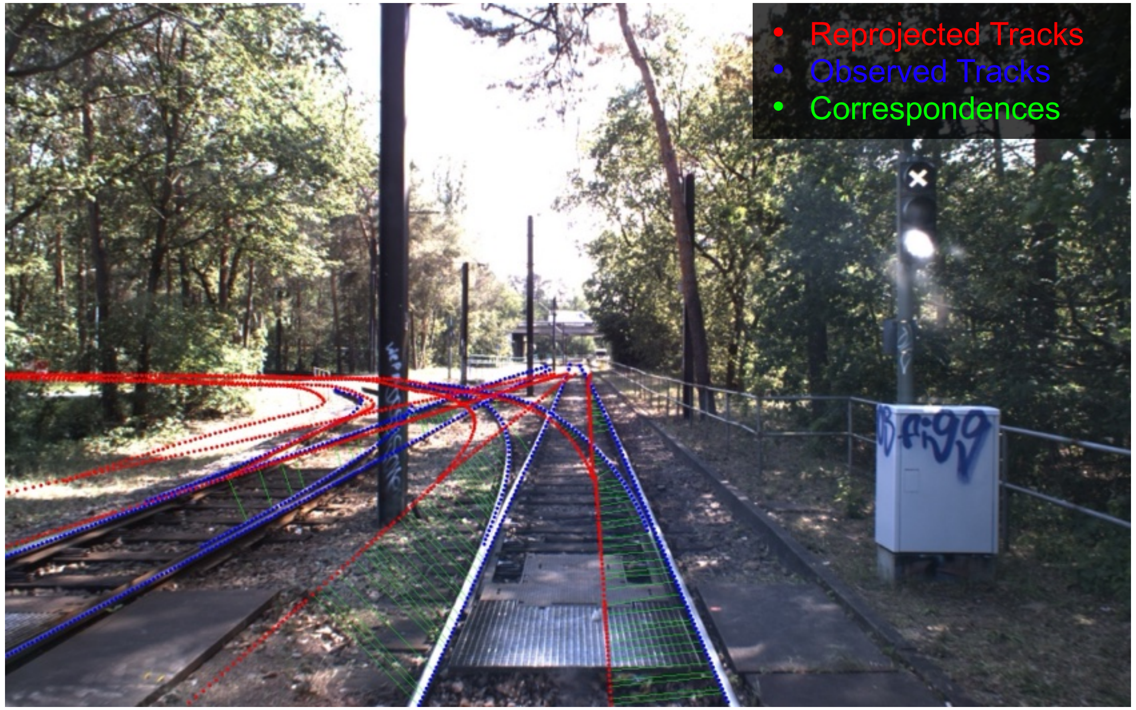
Output:
- Observed tracks

Method | Pose Optimization

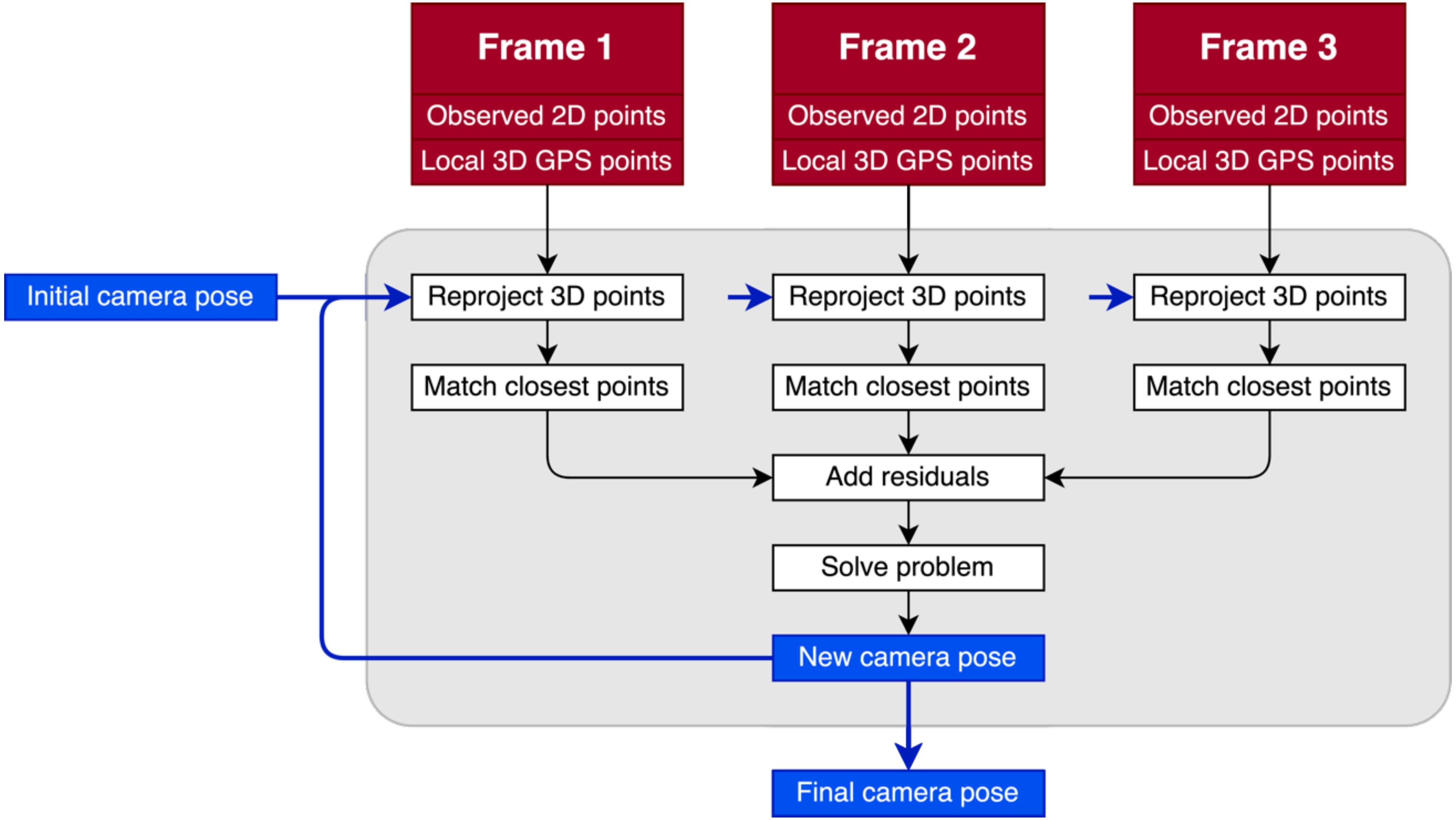
Input:
- Observed tracks
- Reprojected local tracks

Process:
1. Find one-to-one correspondences
2. Compute residuals
3. Solve optimization problem
4. Update camera pose

Output:
- New camera pose estimate



Method | Pose Optimization



Results | Single-frame

Results | Single-frame

Results | Multi-frame

Results | Multi-frame

Results | Multi-frame vs. Single-frame

Evaluation | Single-frame Performance

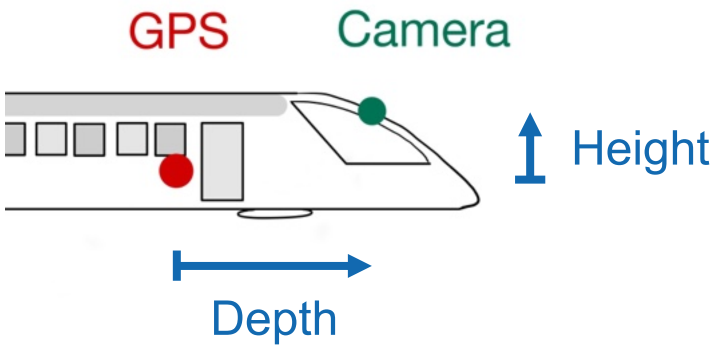
Experimentation on variety of frames

Convergence

- Always after 10-50 ICP iterations
- Given initial height, depth estimates

Robustness & Generalization

- ICP correspondences adjust themselves
- Best to use keyframes: curves / intersections



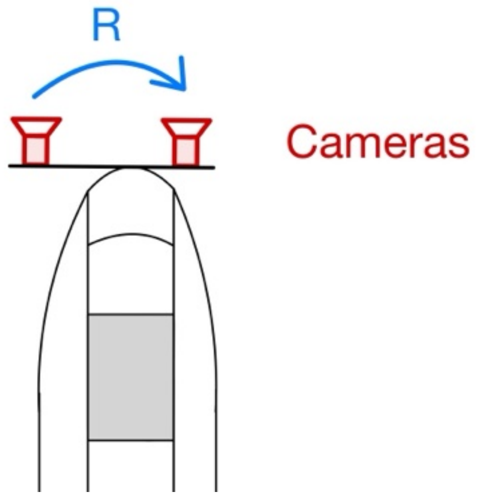
Evaluation | Single-frame Accuracy

Stereo Camera Setup

1. Optimize separately
2. Compute relative pose
3. Compare to calibration

Position

- Horizontal distance (ΔX)
- Height difference (ΔY)
- Depth (ΔZ)
  - Least critical

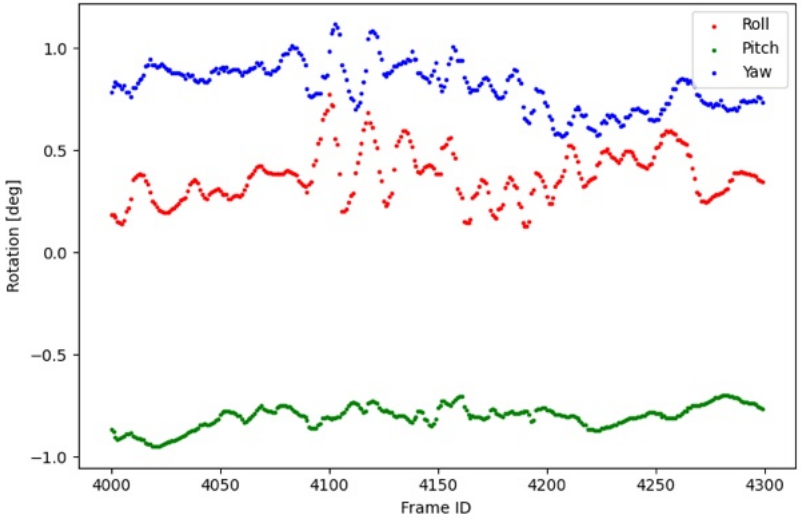


	ΔX	ΔY	ΔZ
Calibration	0.307	0.002	0.010
Frame 1	0.289	-0.006	0.163
Frame 2	0.261	-0.047	0.128
Frame 3	0.300	0.010	0.133

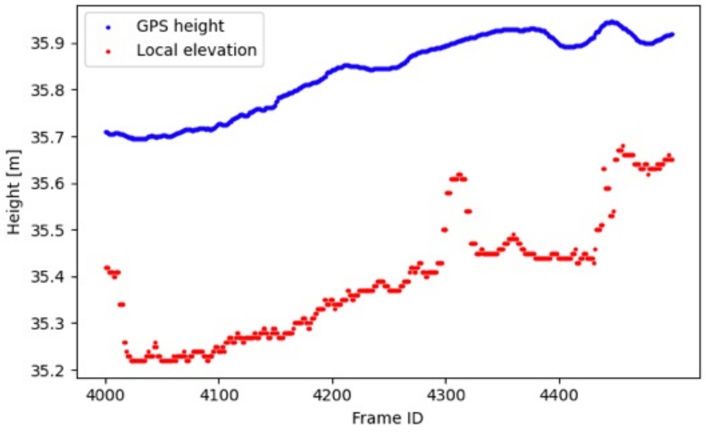
Evaluation | Multi-frame Accuracy

Limited by data precision

- RTK-GPS pose
- Elevation
- Railway nodes



Same camera pose, but inconsistent reprojections



Conclusion

Outcome

- Robust reprojection and optimization pipeline
- Multi-frame limited by data accuracy
- Modular, expandable codebase

Extensions

- Improve state estimate with IMU & Odometry

Questions?

Eric Tüschenbönner
etueschenboe@student.ethz.ch