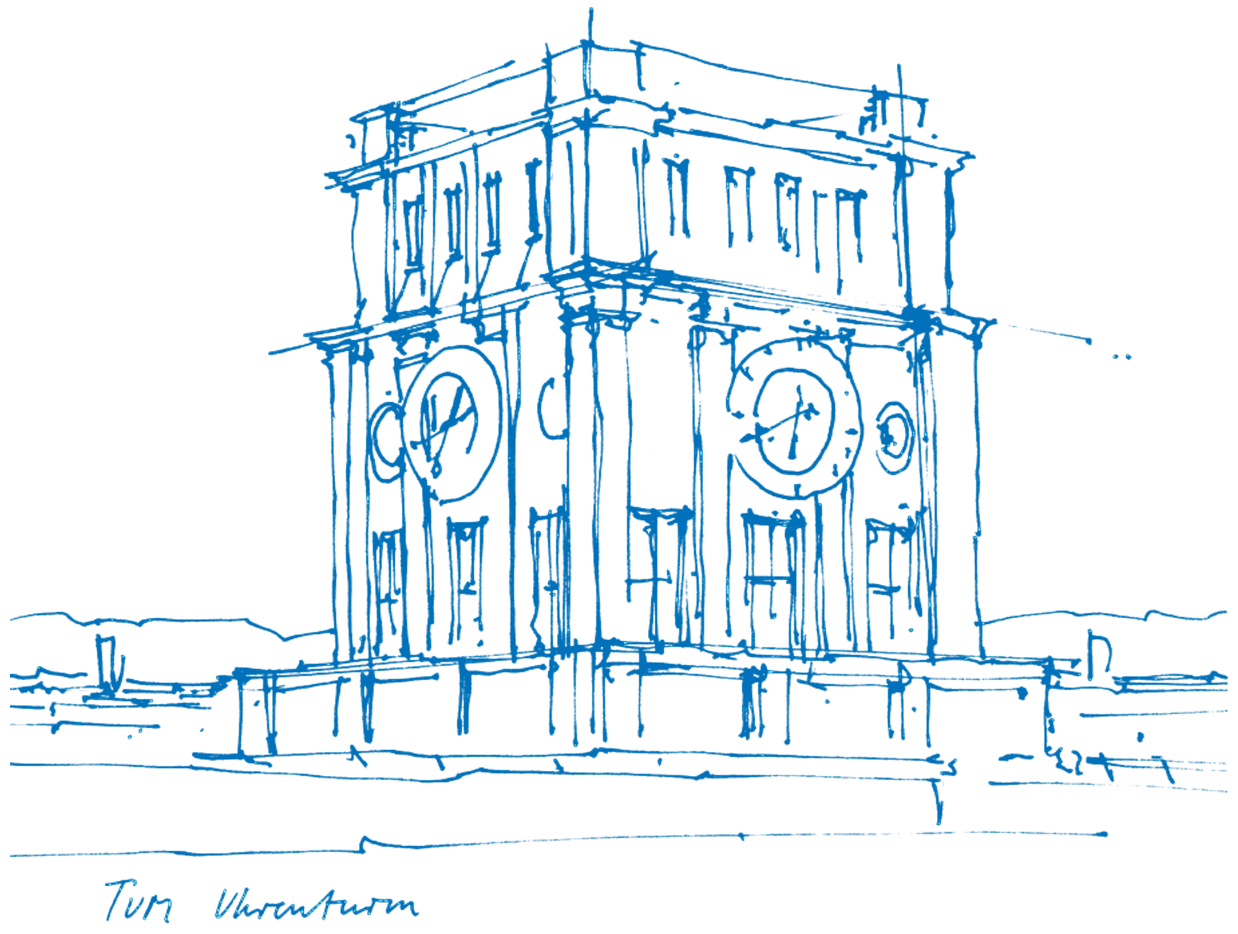
Geometric Scene Understanding

Preliminary meeting

Yan Xia, Linus Härenstam-Nielsen, Weirong Chen, Zhenzhang Ye,
Dominik Schnaus, Nikita Araslanov

07.02.2024

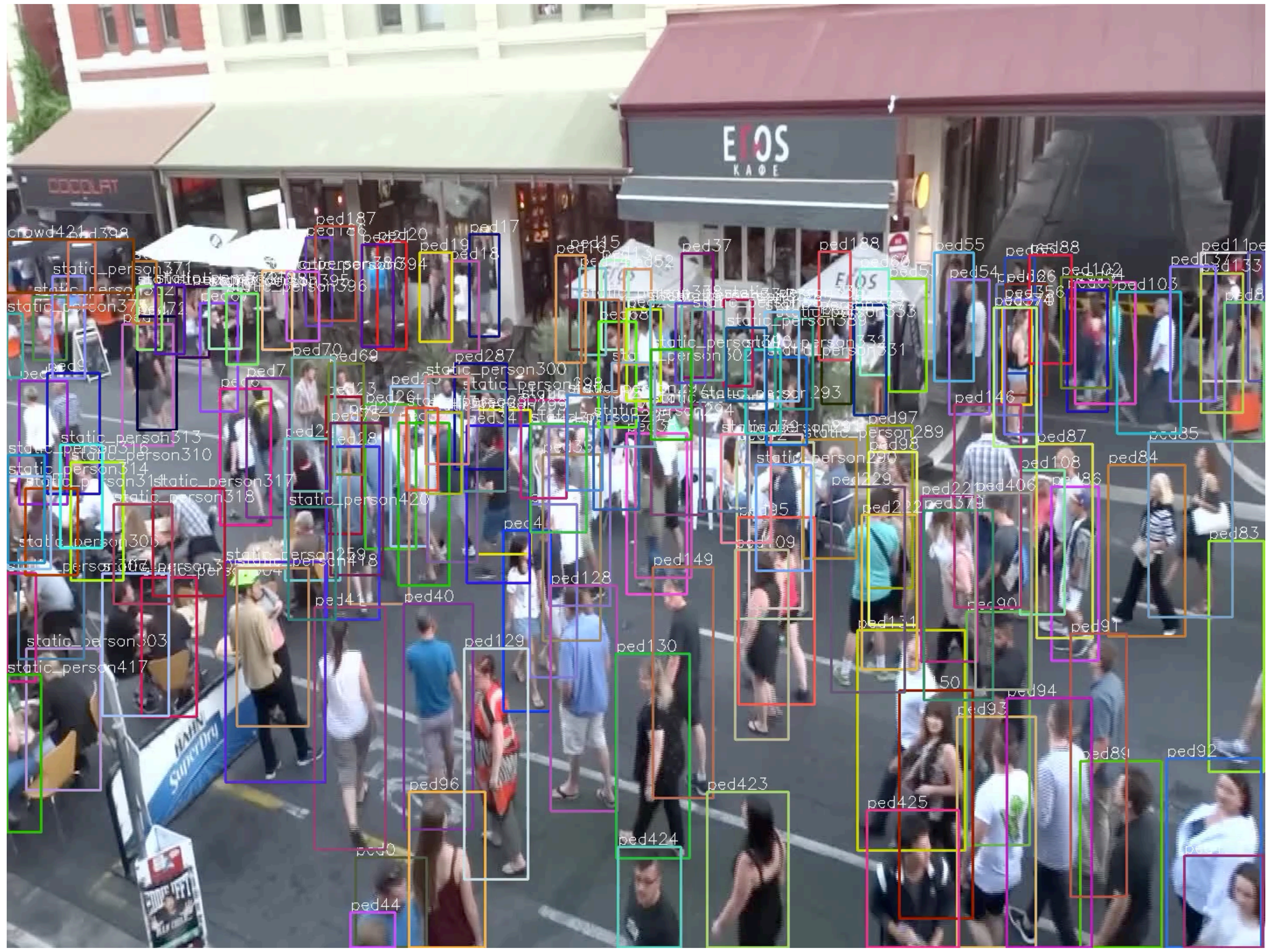


Motivation

- What is scene understanding?
  - Algorithms that describe an image in terms familiar concepts:
    - Content: what is the underlying 3D structure and its properties?
    - Relationship: how are two images related?
    - Temporal evolution: how does the camera/object move?

- What does it mean geometric?
  - Leveraging or inferring geometric (3D) constraints.

Real-world scene understanding



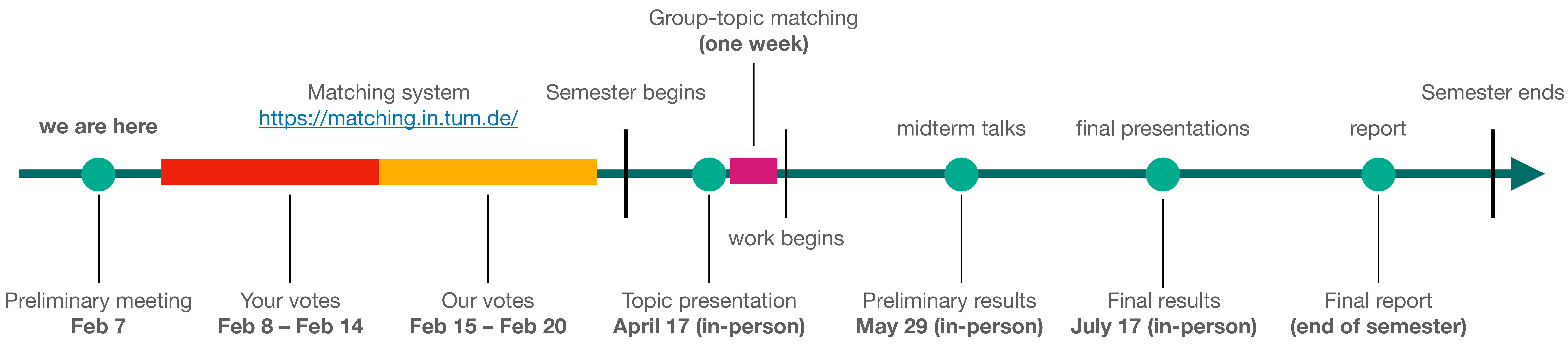
MOTChallenge

Cityscapes

What we expect

- Prerequisites:
  - Introduction to Deep Learning (IN2346)
  - Computer Vision II: Multi-view Geometry (IN2228)
- or any other relevant courses:
  - Computer Vision III (IN2375)
  - Machine Learning for 3D Geometry (IN2392)
- unsure → send us an email (gsu-ss24@vision.in.tum.de)

Timeline



- Send us your application (transcripts) by February 14.
- Projects are assigned to groups of 2-3 people.
- Reports are due by the end of the semester.

Summary

What you get:

- Interesting research problems.
- Teamwork in a group of 2-3 people.
- Regular group meetings (e.g. weekly) with an advisor.
- Access to a GPU cluster.

What you give:

- two presentations (midterm and final);
- written project report.

Your potential advisors

Dr. Yan Xia

Linus Härenstam-Nielsen

Weirong Chen

Dominik Schnaus

Zhenzhang Ye

Dr. Nikita Araslanov

You reach us over the mailing list: gsu-ss23@vision.in.tum.de

Previous projects (1)

- Leverage geometric constraints to infer semantic-level concepts without supervision.

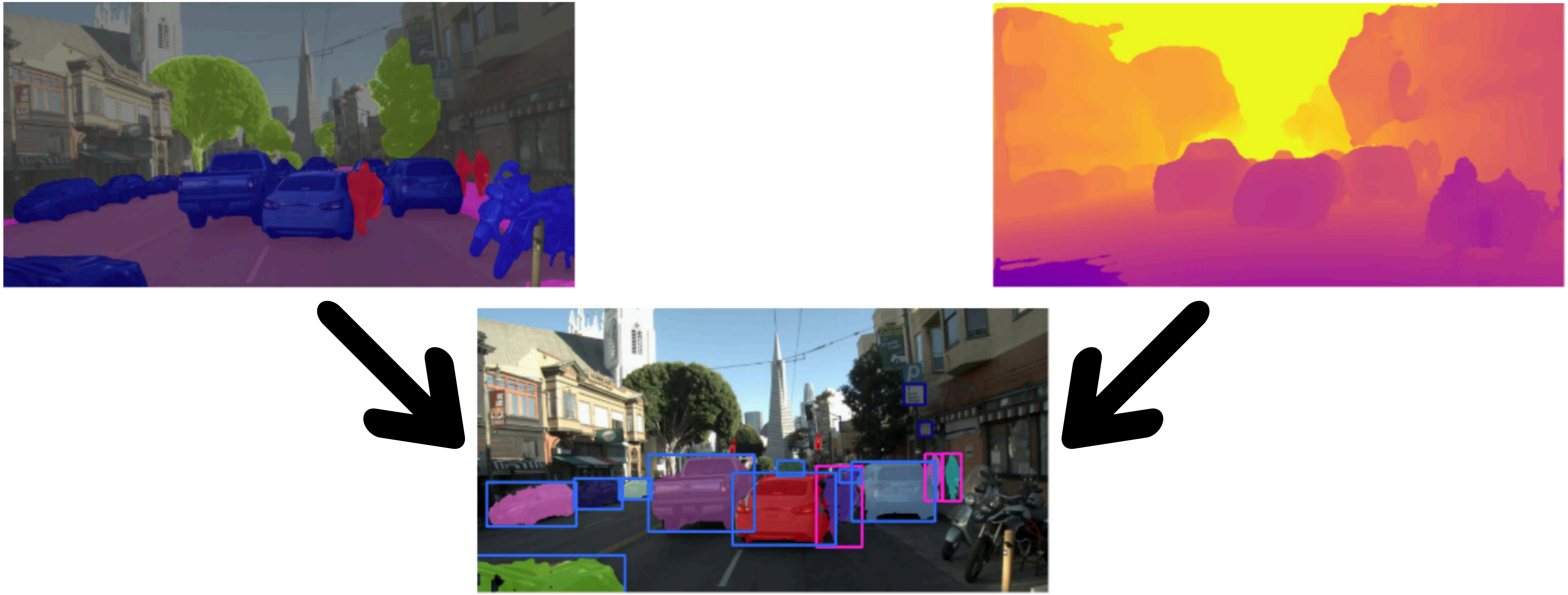


- Example scenario: Can we leverage temporal coherence (in the 3D world) to disentangle dynamic from static objects?

Previous projects (2)

Dominik Schnaus

Self-supervised instance segmentation using depth Information

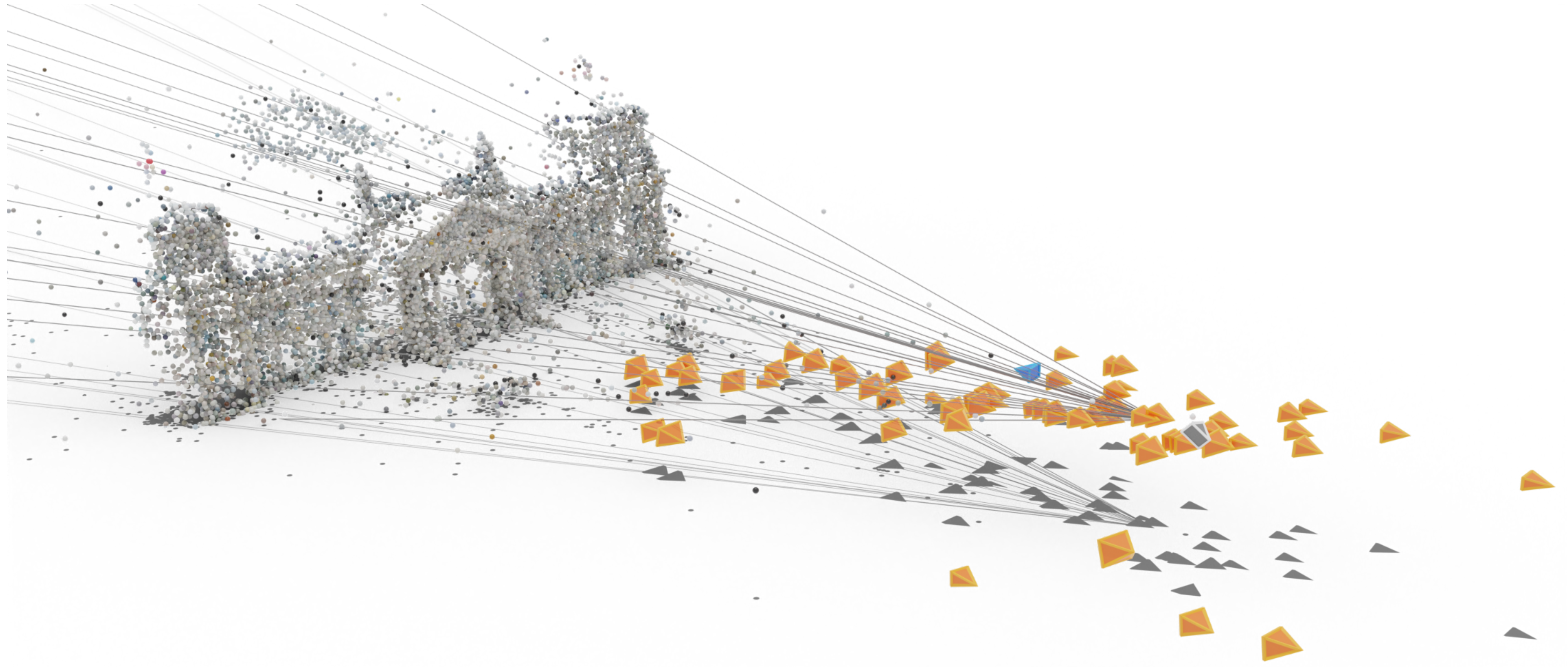


Previous projects (3)

Linus Härenstam-Nielsen

- **This project**: Implement pose estimation using JAX
- DL library from Google, similar to PyTorch but with several advantages:
  - **Fully JIT-compiled** → very fast runtime
  - **JAXopt** → high quality optimizers, just need to define loss functions
  - **End-to-end differentiable** → the results can be combined with network training as a future project

$$\min_{R \in SO(3), t \in \mathbb{R}^3} \sum_{i=1}^{n} \|x_i - \pi(R, t, K, X_i)\|^2$$



Questions?

Contact: gsu-ss24@vision.in.tum.de

Course webpage: https://cvg.cit.tum.de/teaching/ss2024/gsu