Computer Vision Systems Programming

Multi-Camera Tracking Evaluation

Ewelina Rupnik
Christopher Pramerdorfer

Content

- What we wanted to do
- What we did and how / impressions
- Results
- Conclusion

What we wanted to do

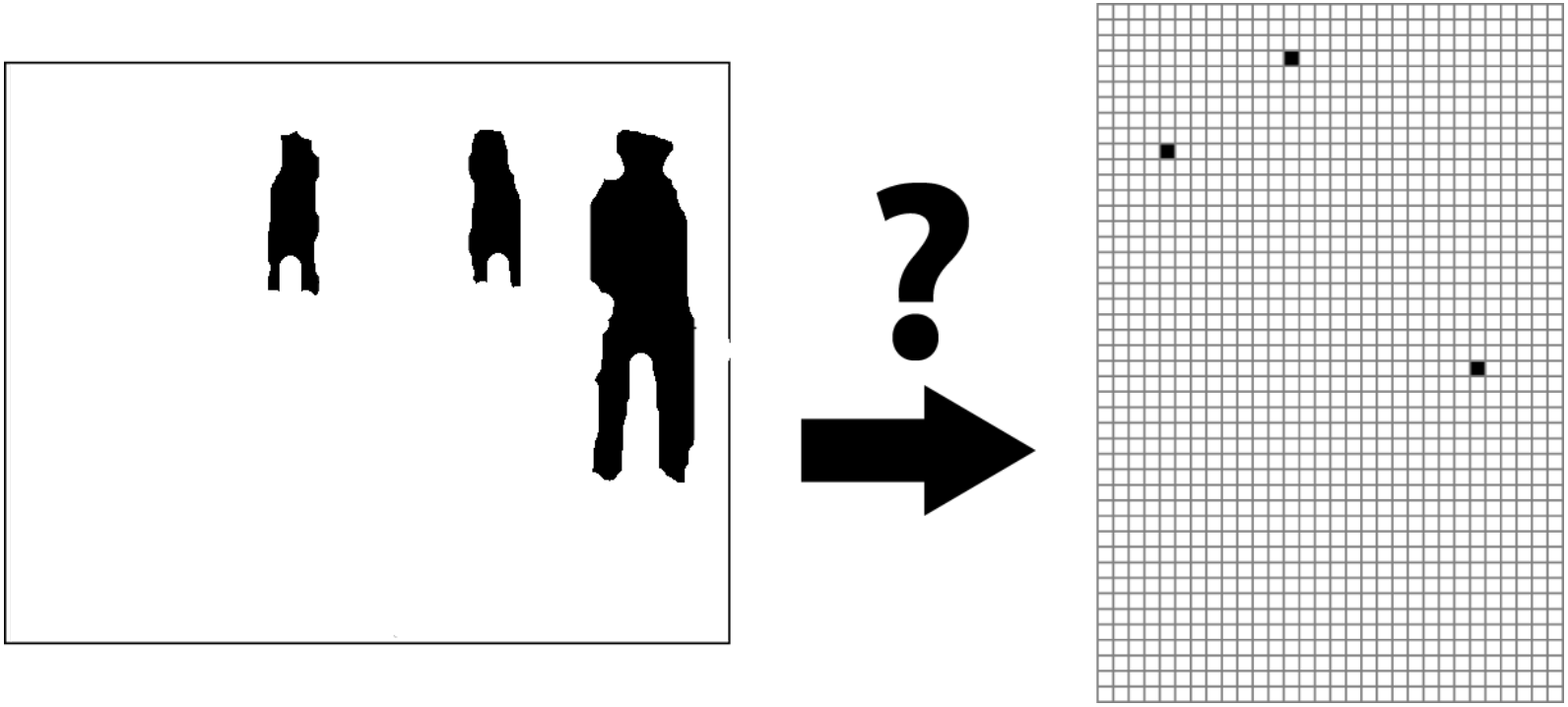
- Goal: person detection / tracking with multiple cameras
  - Chosen algorithm: KSP optimization by Berclaz et al.
  - Questions to answer
    - Single camera performance / limitations
    - Performance scaling with multiple cameras

KSP optimization - introduction

- Recently published [2011] tracking algorithm
- Prototype code available online
- Two-step approach
  - Per-frame object detection / position estimation
  - Linking and optimization

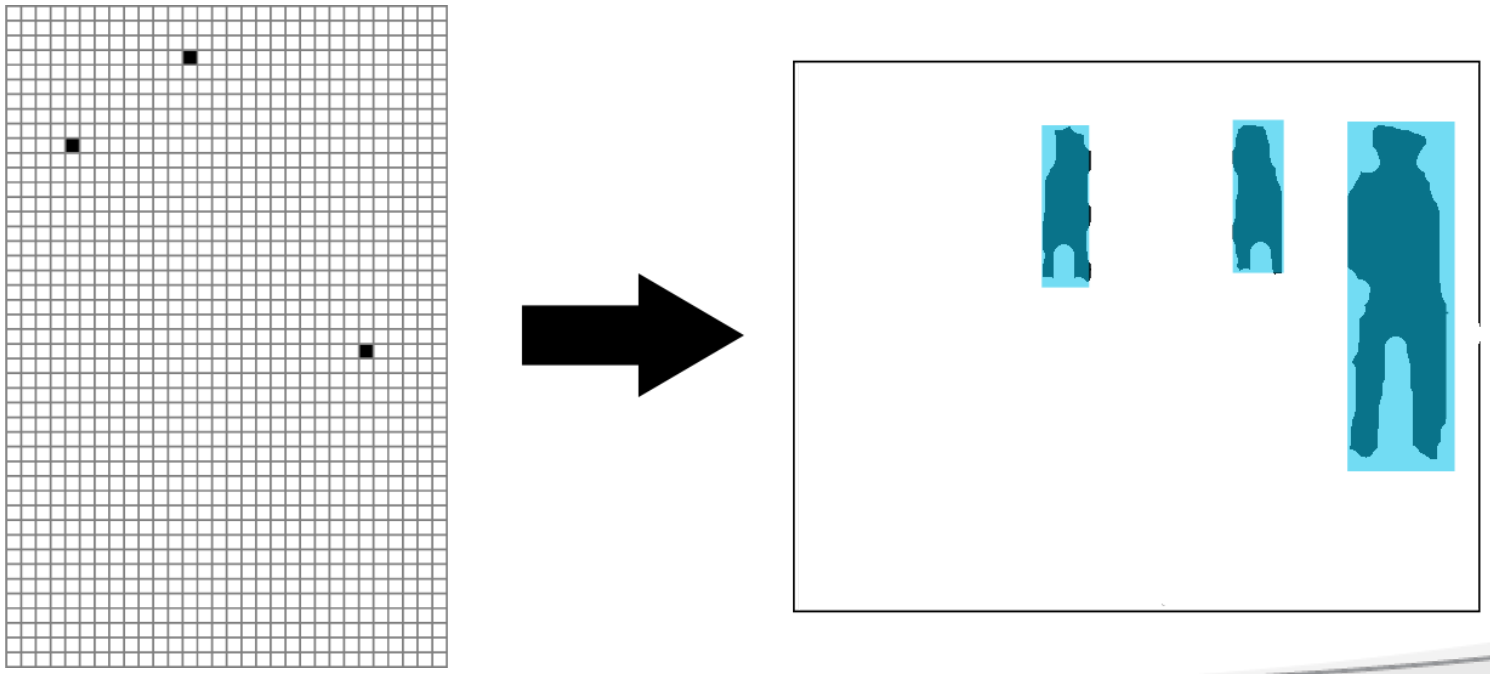
Object detection - approach

- Approach: If we see an object in the camera, what is the most likely position of that object in the scene?



Object detection - position estimation

- Approach: Opposite direction / optimize shape similarities
- Requirements
  - Known object shape
  - Calibrated cameras

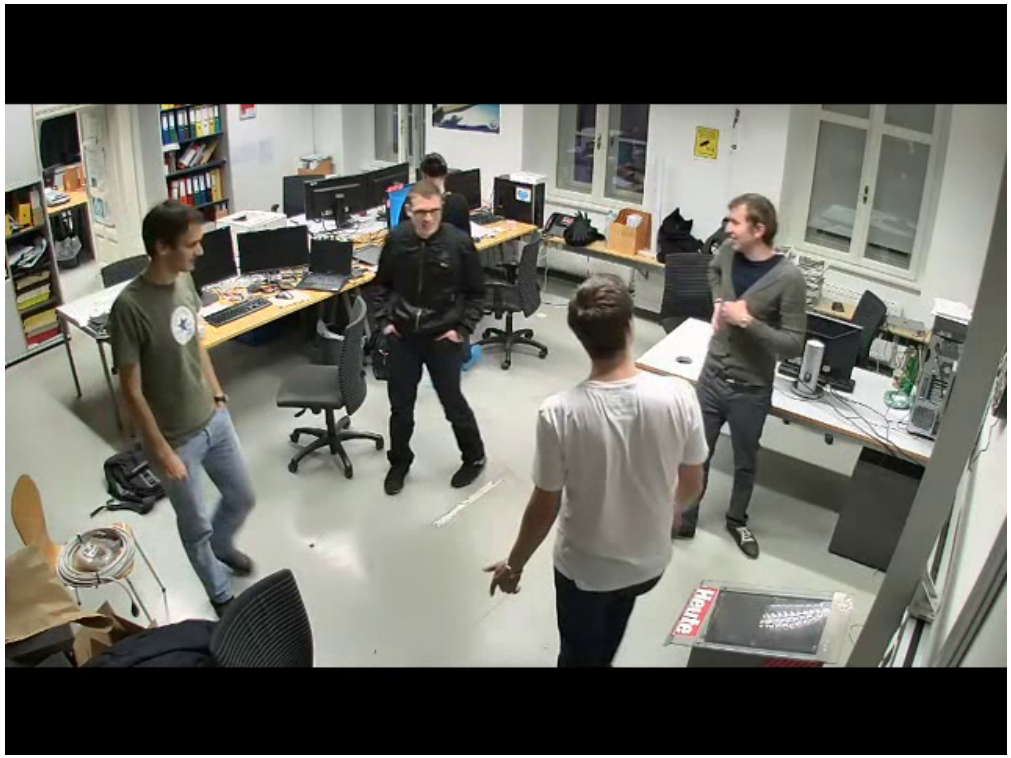


What we did and how

- Pre-processing
  - Camera alignment and configuration
  - Camera calibration (intrinsic + extrinsic)
  - Tracker configuration
  - Test scene recording
- Post-processing
  - Video normalization
  - Foreground detection (background subtraction)
  - Tracker application
- Result visualization and comparison

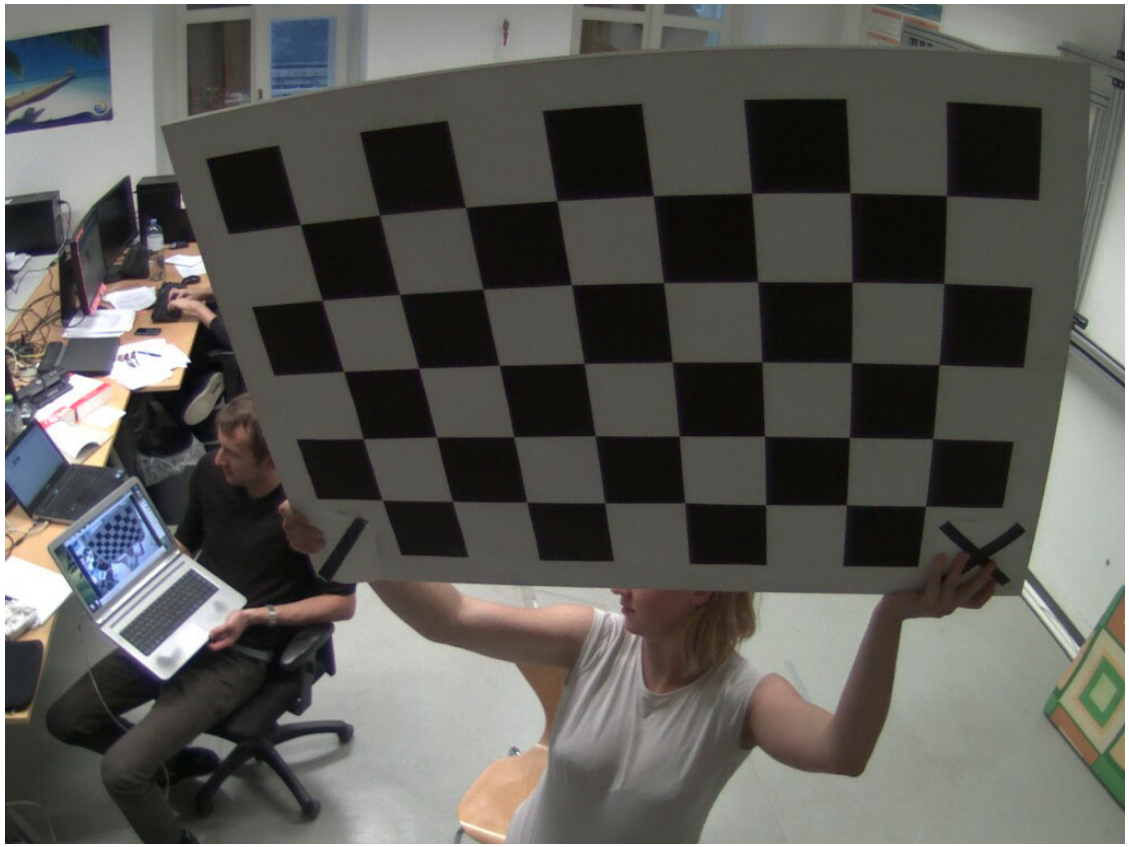
What we did and how - camera setup

- We used the cameras at the CVL
  - 2 * Axis 211w, 2* Axis P1346, 2 * Sony CH140
- Ali□gned to produce partially overlapping views
- Configuration: resolution, FPS, focal length, ...



What we did and how - camera calibration

- Interior orientation for modelling camera parameters
- Matlab camera calibration toolbox

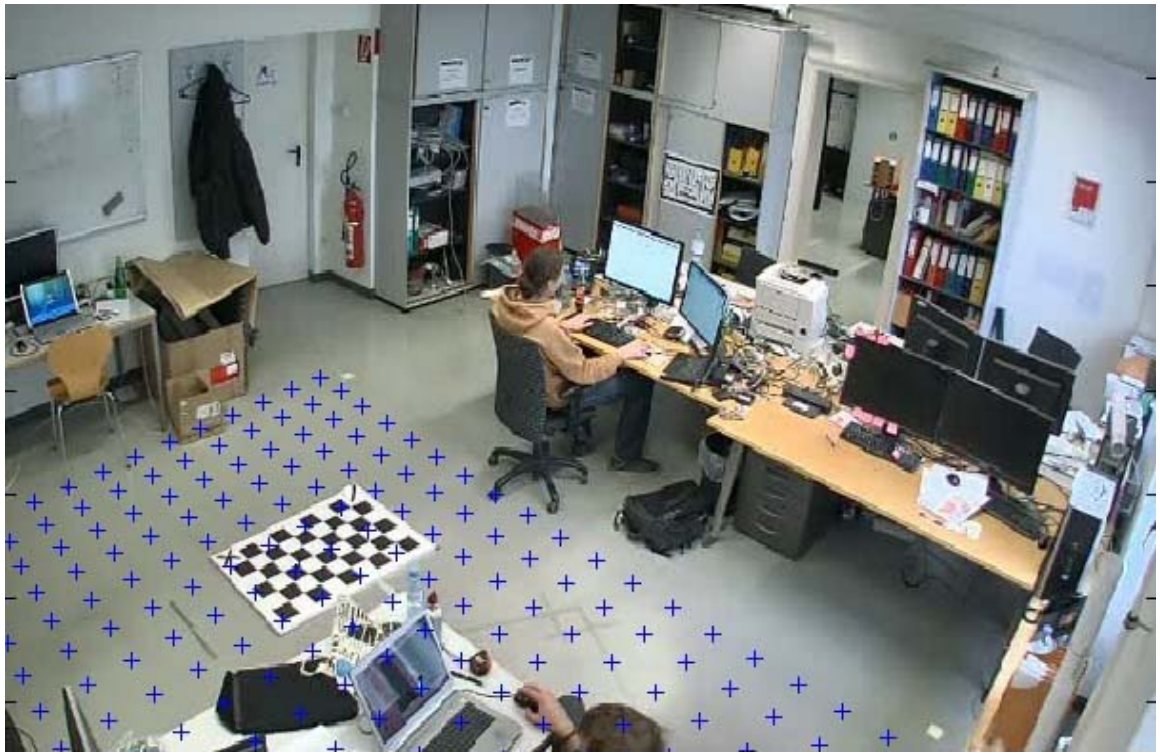


What we did and how - camera calibration

- Exterior orientation for mapping object space with image space
  - initial EO - plane2plane homography
  - final EO - minimization of reprojection error

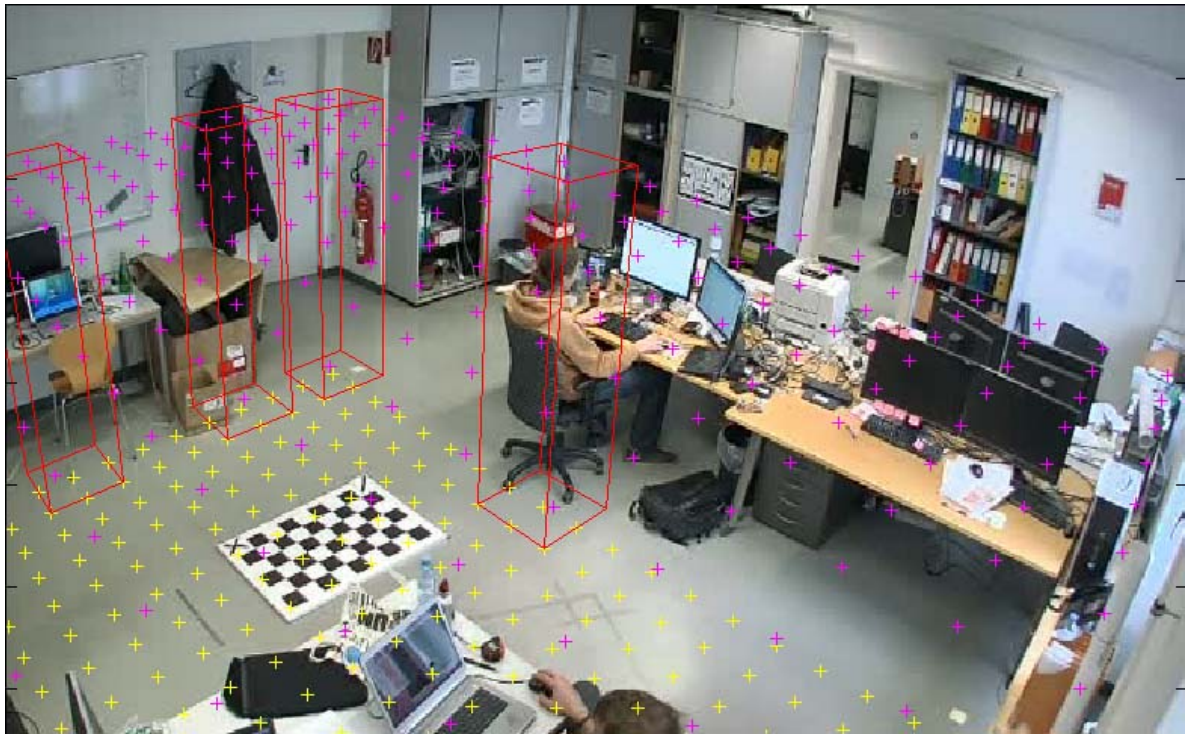
What we did and how -ROI setup (1)

- ROI - points represent cells' grid
  - floor plane Z=0.00m
  - cell size = 0.25m



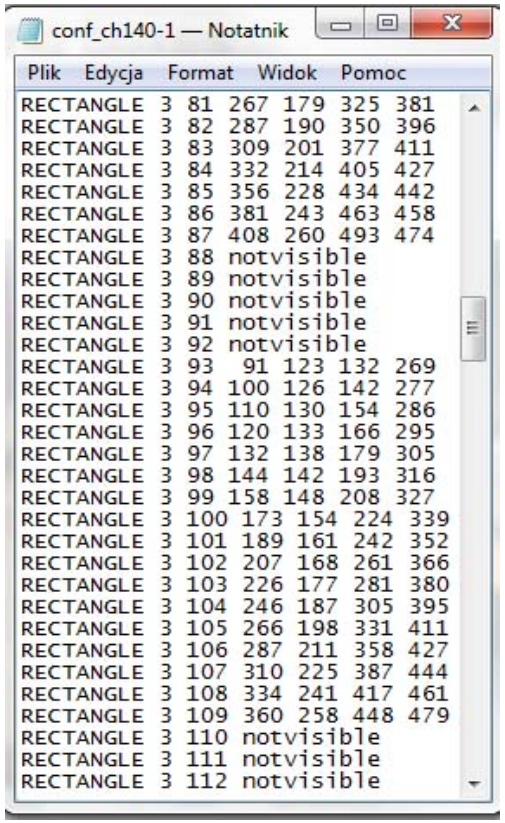
What we did and how - Bounding box

- Examplar bounding boxes
  - bounding box size = 0.50 x 0.50 x 1.75m



What we did and how - configuration file

- RECTANGLE <camera number> <location number> notvisible|<xmin> <ymin> <xmax> <ymax>



What we did and how - test scene recording

- Recorded scenes with 2-5 people standing / moving around

What we did and how - video normalization

- All video streams need to have
  - Identical resolution
  - Identical framerate (!)
- All streams were normalized to 640*480 @ 20fps

What we did and how - foreground detection

- Tracker requires fg/bg segmented frames as input
- Foreground detection via background subtraction
  - Codebook algorithm (OpenCV implementation)

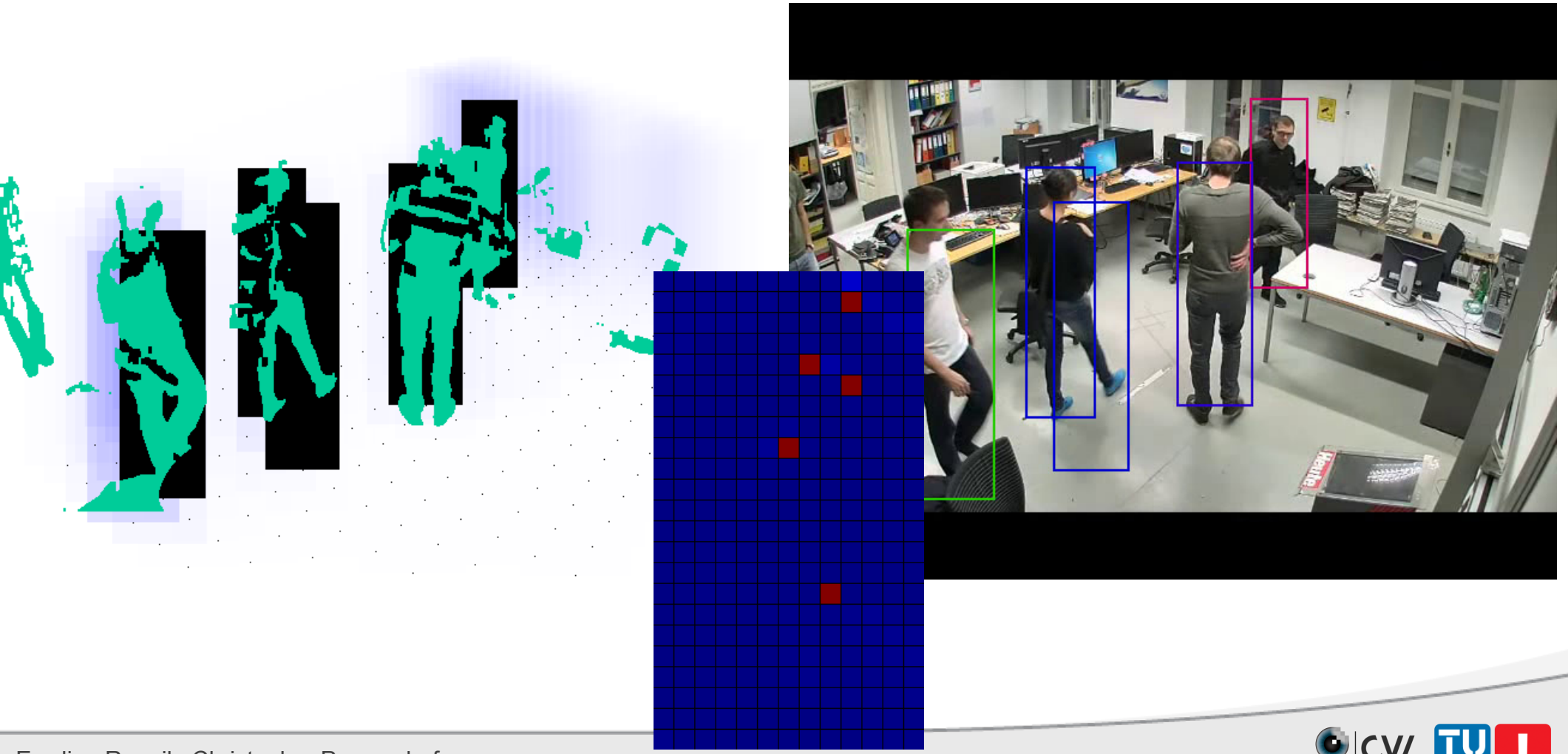
What we did and how - tracker application

- Input
  - Segmented frames for all camera views
  - Config: object appearances for all positions and views
- Output
  - Occupancy probability for every position and frame
    - Input to the second step of the tracker

0 0.000135422
1 0.000141084
2 0.00264496
3 0.0214263
4 0.999974
5 0.030795
6 0.0290455
7 0.02753
...

What we did and how - result visualization

- Visual evaluation - proper result visualization required



TU WIEN

CVL Computer Vision Lab

Results ...

Conclusion

- Using multiple cameras improves performance noticeably
- Three cameras seems a good number in our case
- Performance varies greatly
  - Good performance with 1 or 2 people
  - 4 and more people are problematic
  - Precise and complicated configuration required
  - Artefacts in segmented frames can cause false positives

TU WIEN

CVL Computer Vision Lab

Thank you ... Questions?