# Visual Odometry Project

Group 11: Martin Kvisvik Larsen

Project Goal

- Implement RGBD-based pose estimation system
- Use TUMs online evaluation tool for:
  - Validation and parameter tuning
  - Performance evaluation for the final system
- Look at some differences from other methods of doing visual odometry

Relevant Literature and Sources

- Books:
  - "Digital Image Processing"
  - "Computer Vision: Algorithms & Applications"
  - "Numerical Optimization"
- Papers:
  - "A Photometrically Calibrated Benchmark For Monocular Visual Odometry"
  - "Trajectory-based Comparison of SLAM Algorithms"
- Other sources:
  - UIO - UNIK4690: Maskinsyn - Lecture Slides
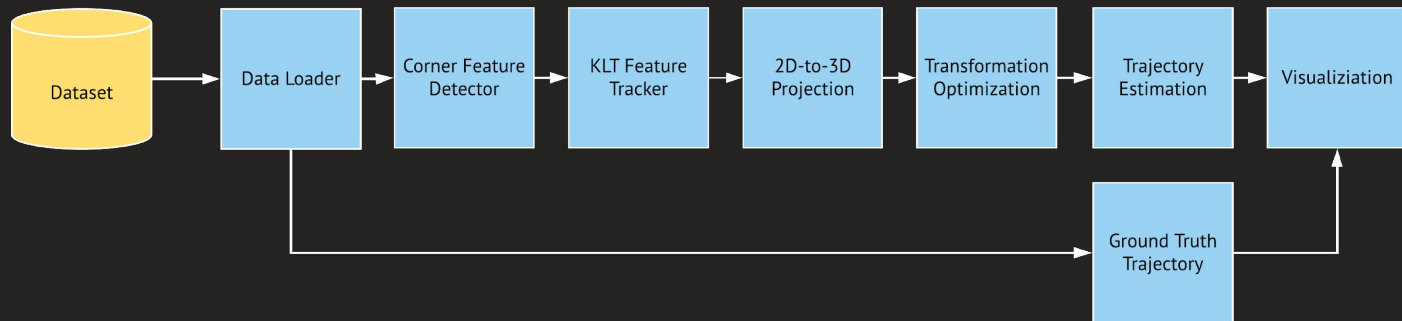 - PSU - CSE486: Computer Vision I - Lecture Slides
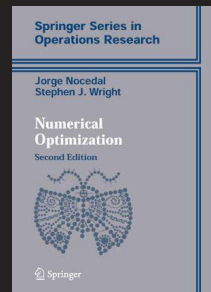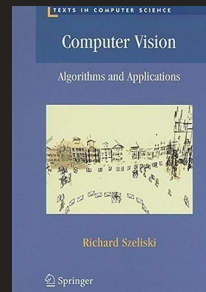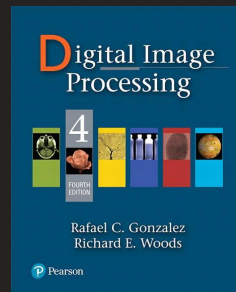
Datasets

- RGBD SLAM datasets from TUM
  - RGBD Freiburg 1 xyz
  - RGBD Freiburg 1 rpy
  - RGBD Freiburg 1 desk
  - RGBD Freiburg 2 xyz
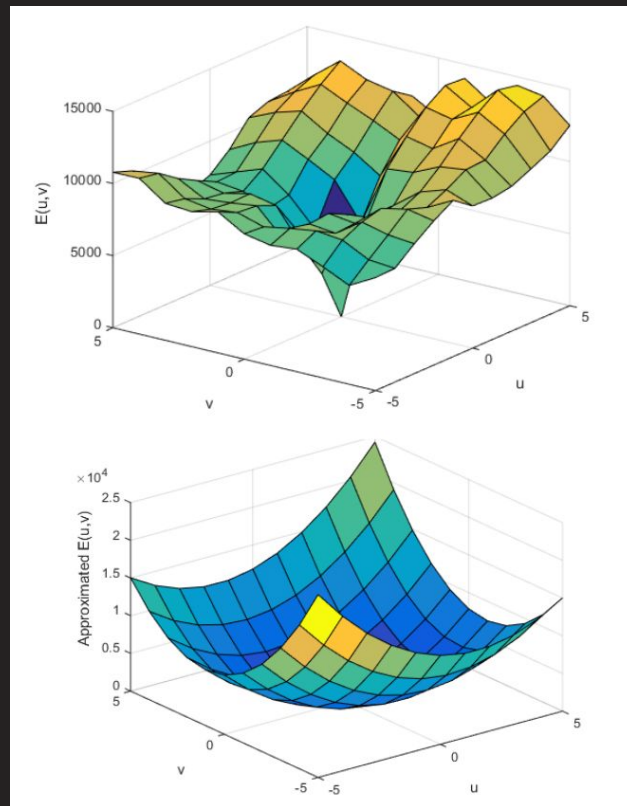 - RGBD Freiburg 2 rpy
  - RGBD Freiburg 2 desk
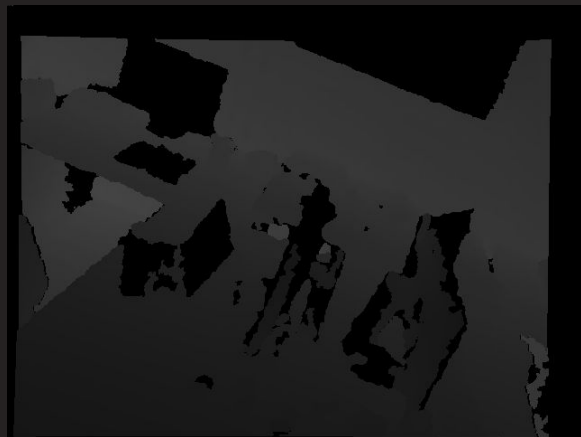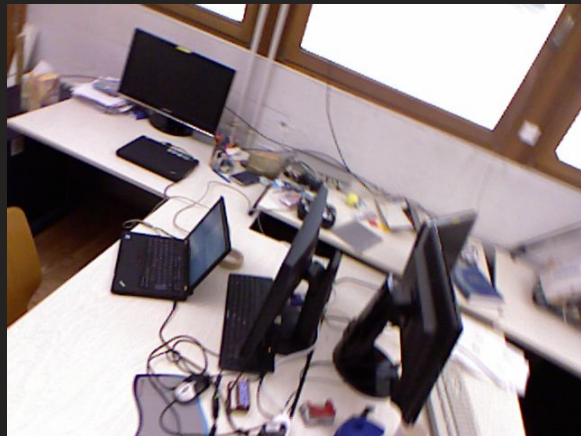
System Architecture
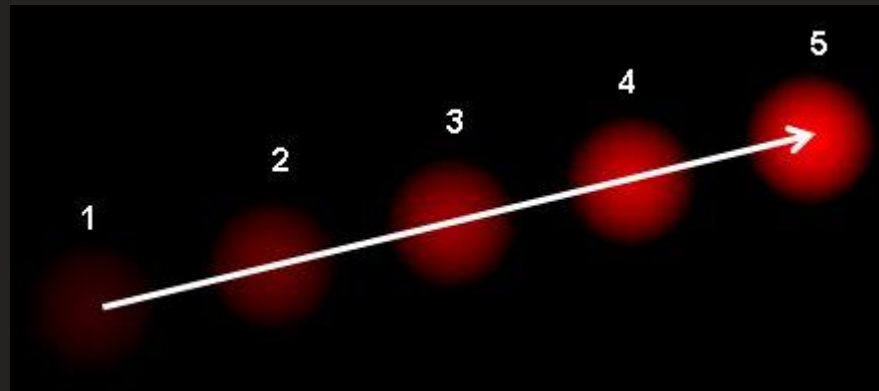
Harris Corner Detector

- First order approximation
- Spatial derivatives
- Harris response
- Non-maximum suppression

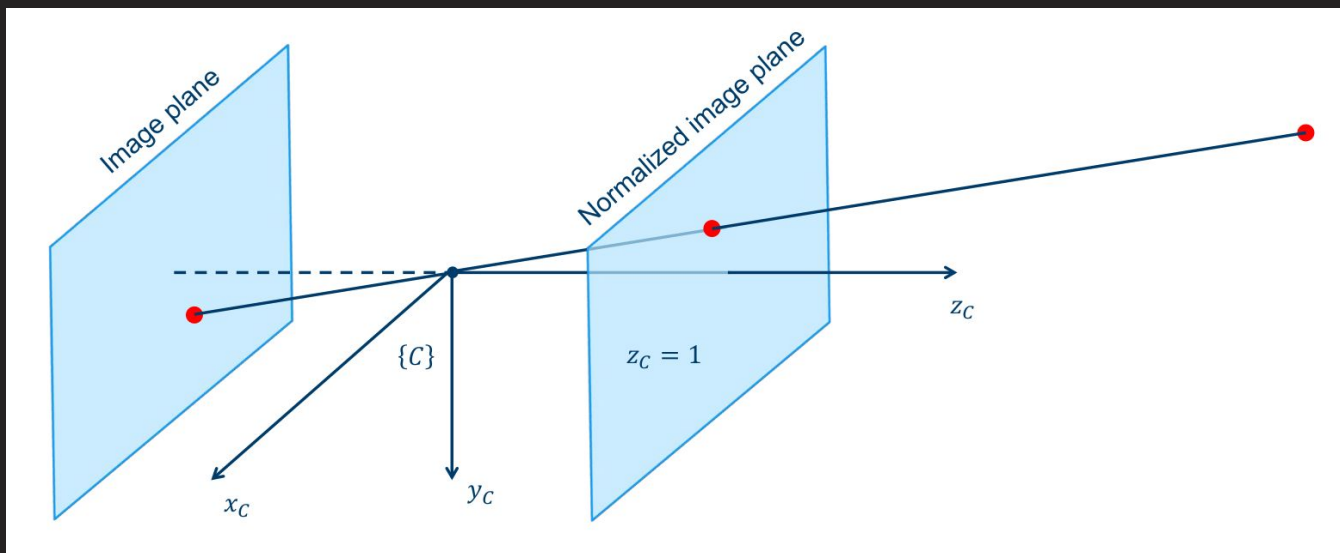| Parameter	Value
Gaussian blur sigma	2.0
Harris response threshold	1.0

KLT Feature Tracker

- Feature tracking method
  - Adjacent frames
- First order approximation
- Iterative optical flow calculation

2D to 3D Projection

- The perspective camera model
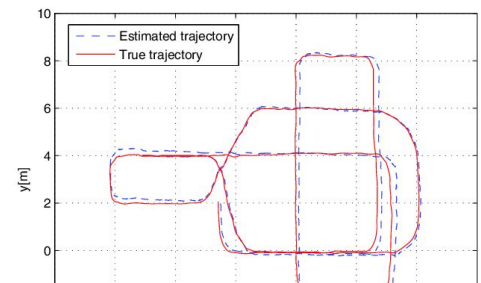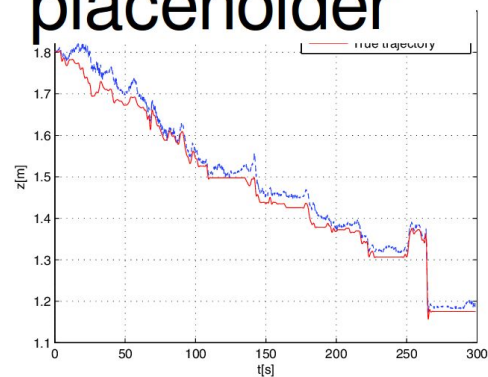- Intrinsic camera parameters

Trajectory Estimation

Results: Freiburg 1 xyz

Translational RMSE	-	[m]
Rotational drift	-	[deg]
Alignment error	-	[m]

Trajectory plot placeholder

Results: Freiburg 2 xyz

Translational RMSE	- [m]
Rotational drift	- [deg]
Alignment error	- [m]
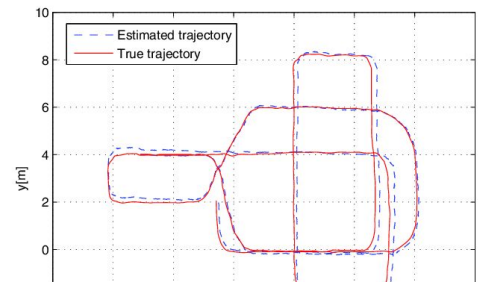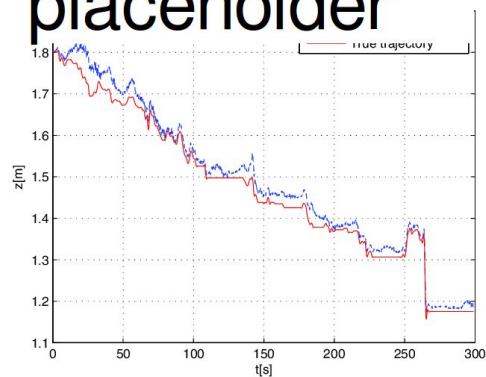
Results: Freiburg 1 rpy

Translational RMSE	- [m]
Rotational drift	- [deg]
Alignment error	- [m]
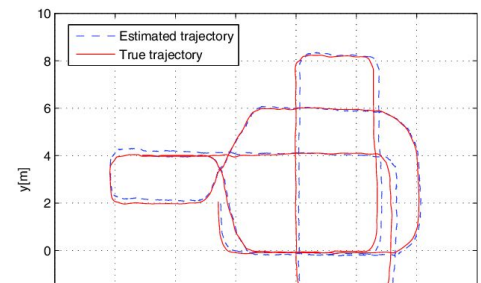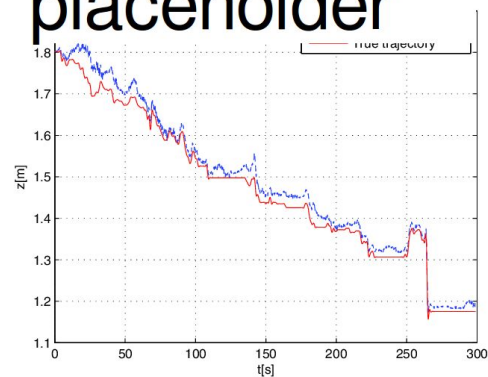
Results: Freiburg 2 rpy

Translational RMSE	-	[m]
Rotational drift	-	[deg]
Alignment error	-	[m]

Trajectory plot placeholder
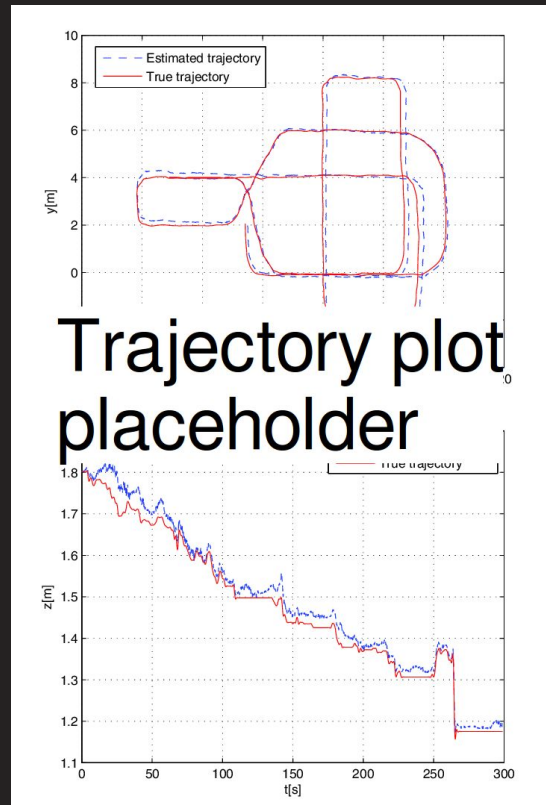
Results: Freiburg 1 desk

Translational RMSE	- [m]
Rotational drift	- [deg]
Alignment error	- [m]
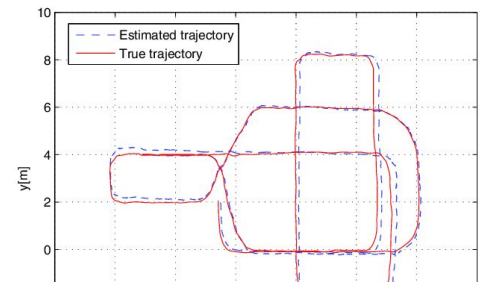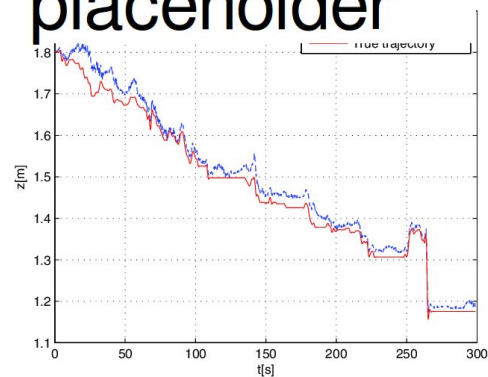
Results: Freiburg 2 desk

Translational RMSE	- [m]
Rotational drift	- [deg]
Alignment error	- [m]
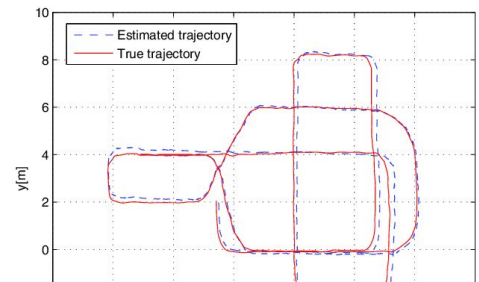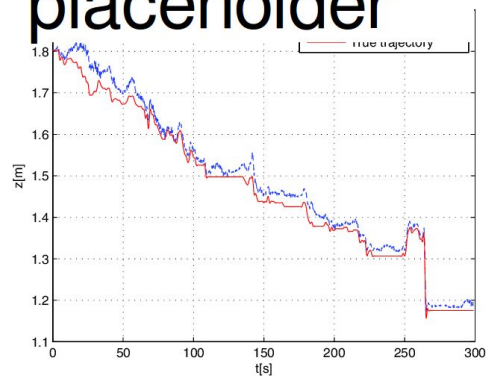
Comparison with other techniques

-

Remaining Work

- Get transformation optimization to work
- Get trajectory estimates from the transformations
- Tune parameters
- Evaluate the performance of the final system
- Find results from algorithms using other methods, ideally:
  - An algorithm another feature detector / feature pyramids
  - An algorithm using feature matching
  - Possible compare to state-of-the-art algorithms like DSO or ORB-SLAM