SLAM++: Simultaneous Localisation and Mapping at the Level of Objects

Renato F. Salas-Moreno, Richard A. Newcombe, Hauke Strasdat, Paul H. J. Kelly, Andrew J. Davison

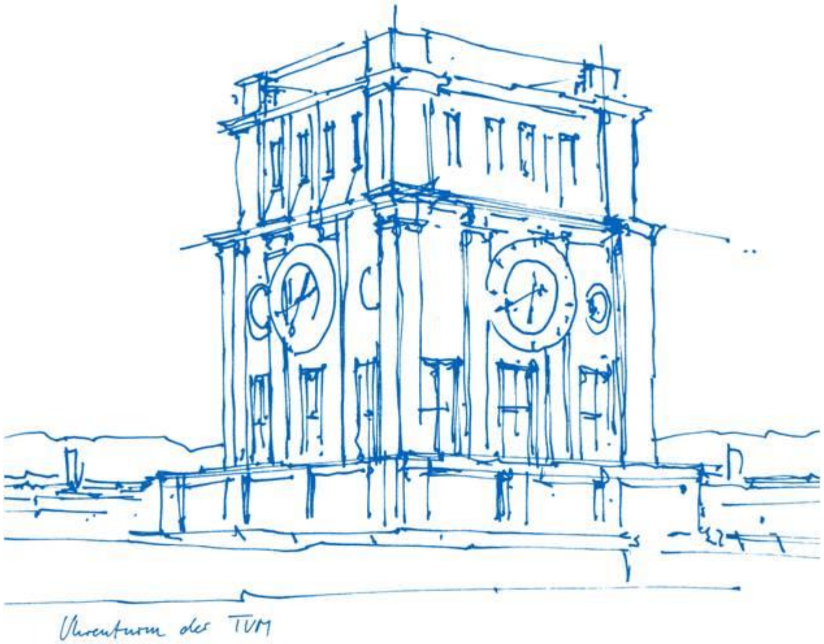
Tim Stricker

Technical University of Munich

Department of Informatics

Chair for Computer Vision and Artificial Intelligence

Munich, 30.05.2018



Contents

1. Problem Motivation

2. Approach
   a. High-Level Architecture
   b. Functionality in Detail

3. Results and Applications

4. Discussion of Scientific Contributions

Problem Motivation

Problem Description

- SLAM: **S**imulteaneous **L**ocalisation **A**nd **M**apping

- Multiple general approaches

- Challenges
  - Real-time operation
  - Scalability and loop closure
  - Dynamic objects
  - Relocalisation („Kidnapped robot problem“)

Main Ideas

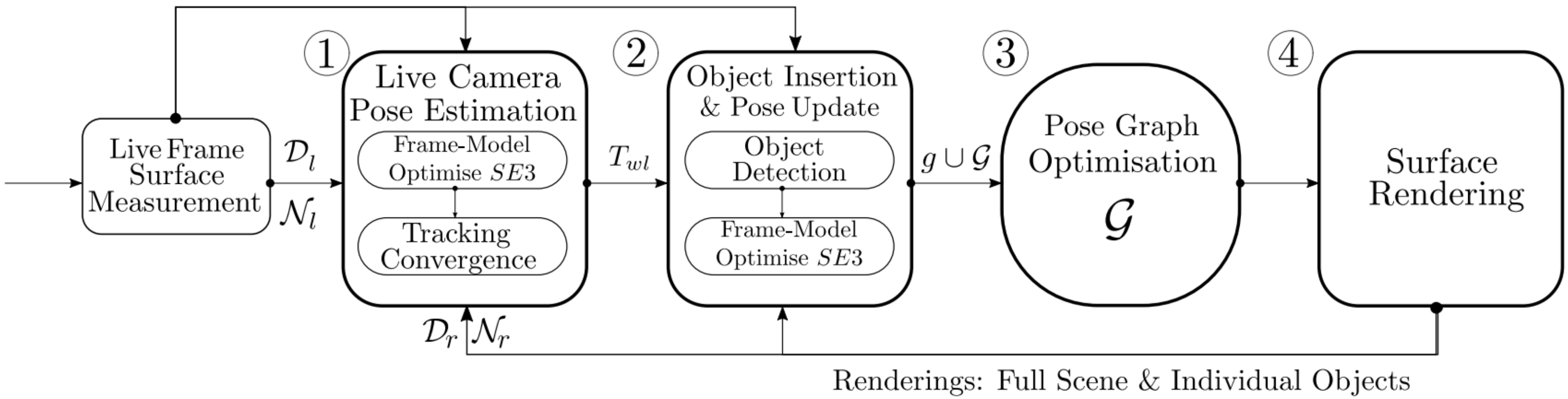
- Prior domain knowledge
  → Many environments consist of repeated objects / structures

- Usage of previously detected objects
  → Camera Tracking and active search

- Representing the world as a pose-graph

Approach

High-Level Architecture



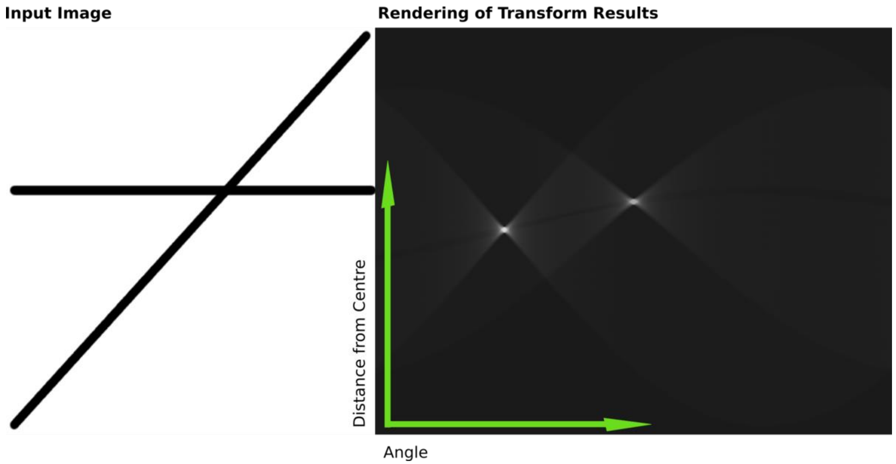
1. Live Camera Pose Tracking (ICP)
2. Object detection
3. Adding succesfully detected objects to graph
4. Rendering objects

Main Functionalities

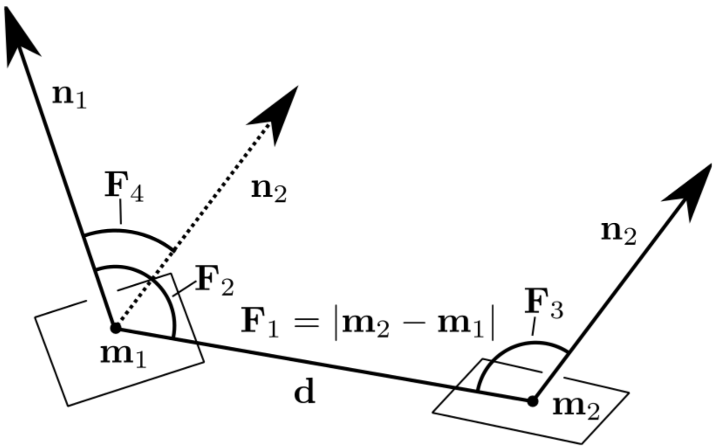
- Creating a database of 3D object models
  - *KinectFusion* for mapping
  - *Marching Cubes* for extracting the mesh
- Real-time object recognition
- Camera tracking and object pose estimation
- Graph optimisation
- Relocalisation

Real-Time Object Recognition

- General approach derived from *Drost et al.*
- Main Idea: Accumulation of votes
  - Generalised Hough Transform
  - Point-Pair Features (PFFs)



https://commons.wikimedia.org/wiki/File:Hough-example-result-en.png#/media/File:Hough-example-result-en.png



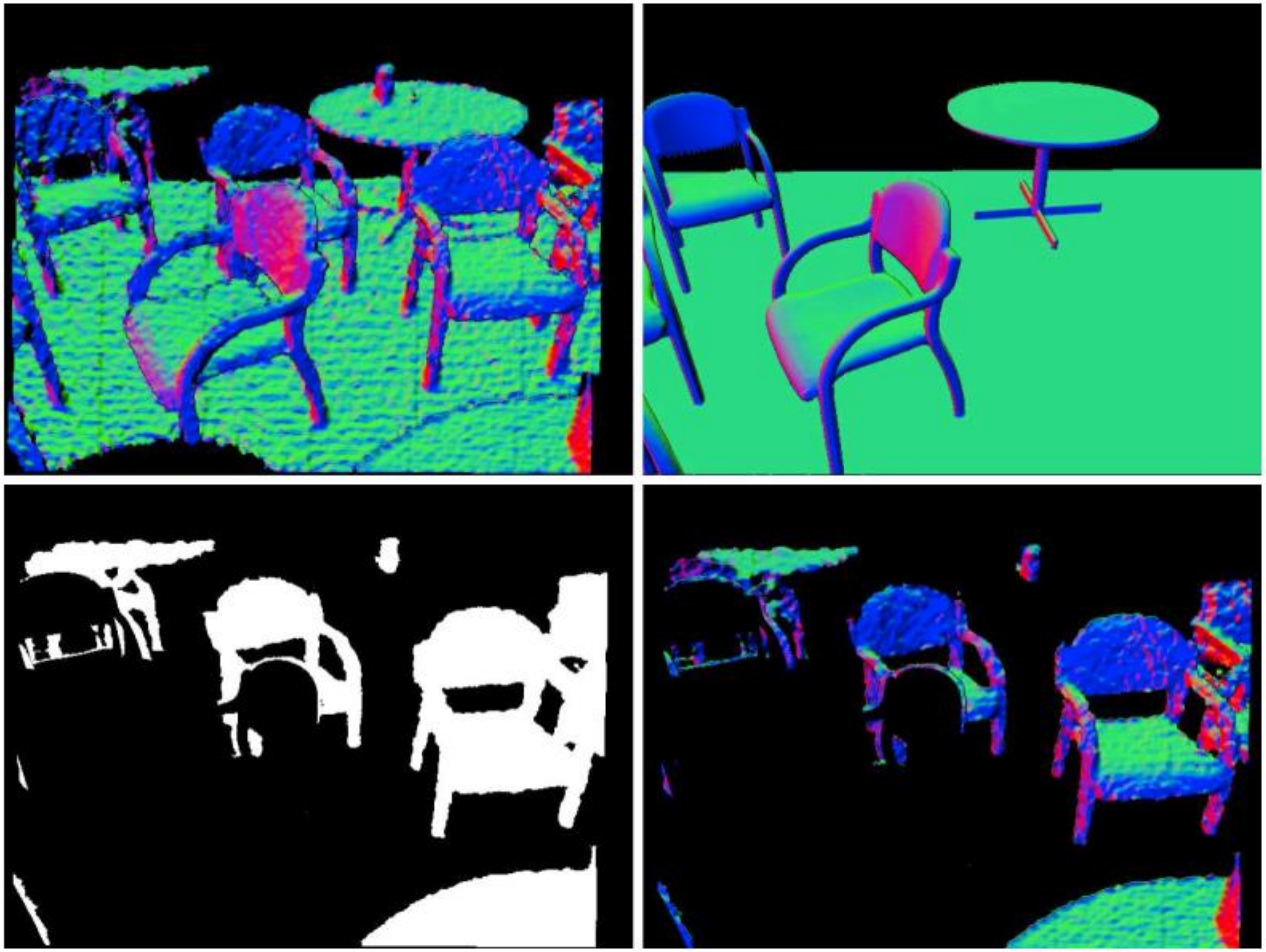
https://www.researchgate.net/figure/Two-surface-points-m-i-and-their-normals-n-i-determine-a-point-pair-feature_fig3_307934827

Real-Time Object Recognition

- Usage of GPU
  - Generation of global object descriptions
  - Matching, Voting, Vote Accumulation

- Active Object Search:
  - Mask Generation
  - Major advantage of SLAM++

Active Object Search



Camera Tracking and Object Pose Estimation

- Camera tracking: KinectFusion
  - ICP on basis of mapped model
  - Model incomplete at early stages

- Camera tracking: SLAM++
  - High quality multi-object prediction
  - Minimizing the point-to-plane metric (depth image)

$$E_c(\xi) = \sum_{u \in \Omega} \psi(e(u, \xi))$$

$\xi \in \mathbb{R}^6$: Twist in SE(3)

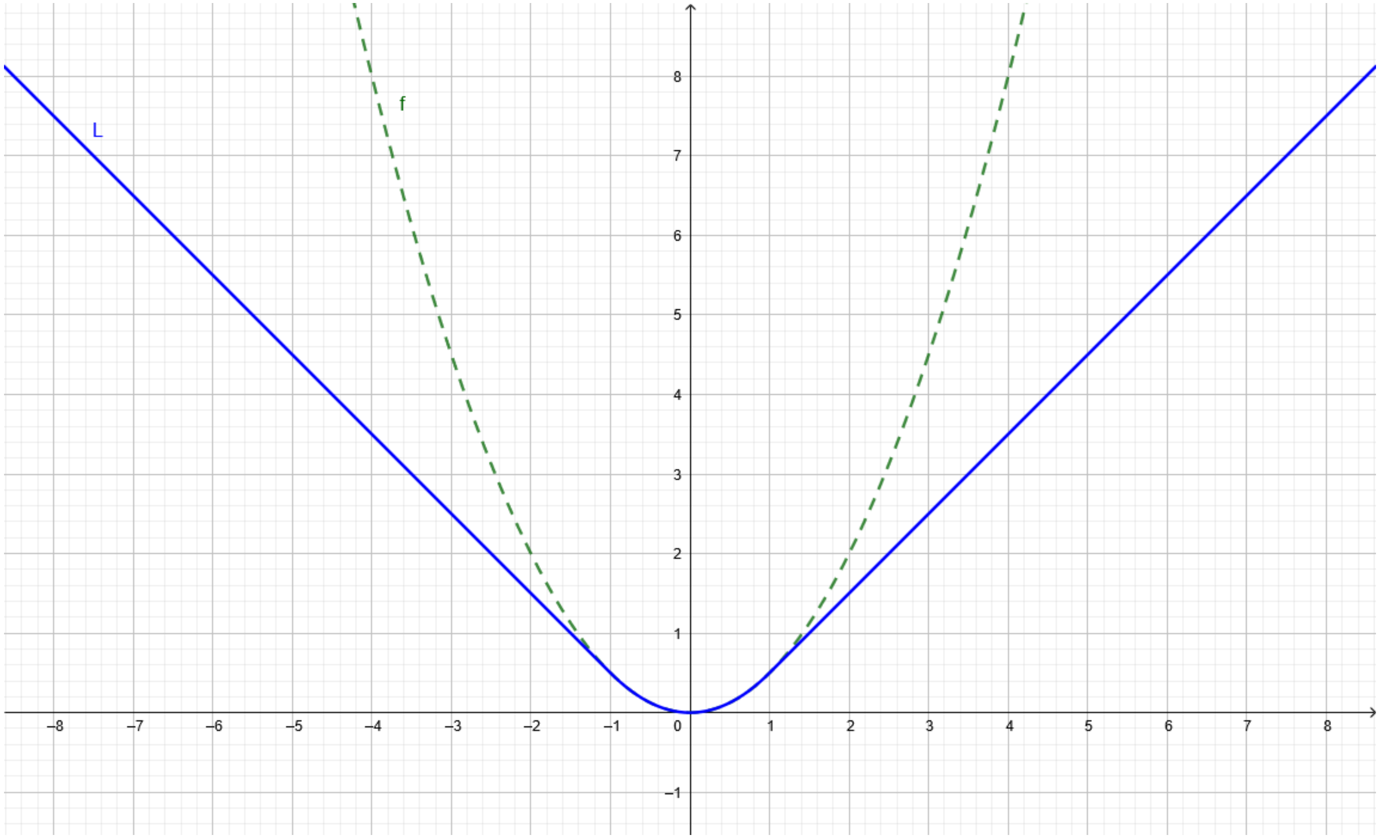
$u \in \Omega$: Valid pixels

ψ: Huber penalty function

Camera Tracking and Object Pose Estimation

Huber penalty function

→ Softer outlier weighting



Camera Tracking and Object Pose Estimation

- Tracking convergence
  - Maximum: 10 iterations
  - Check for poor convergence

- Additional usage of ICP:
  - Model initialisation
  - Camera-object pose constraints

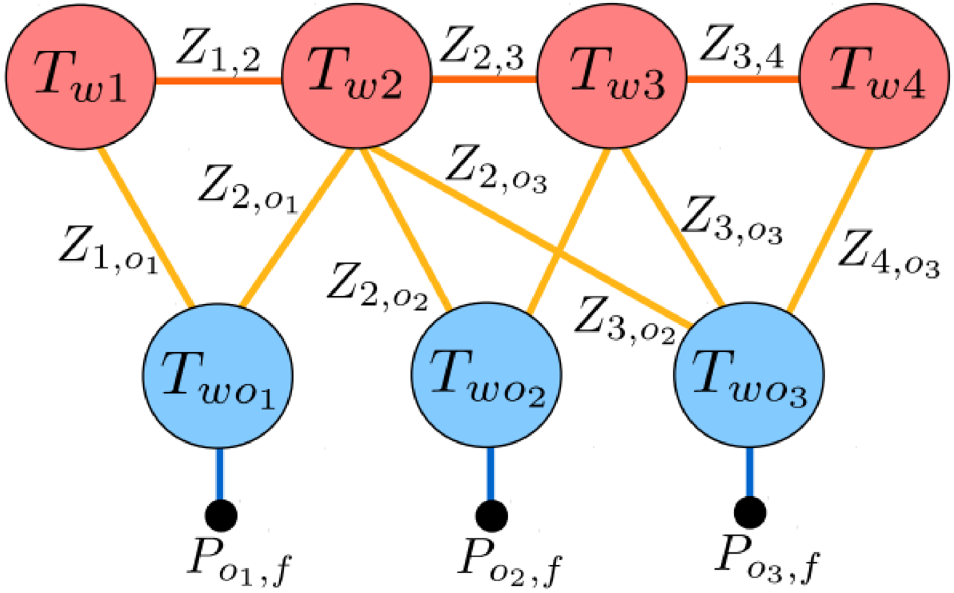
SLAM Map Representation

- Representation of the world as a graph
  - Nodes:
    - Estimated poses of objects: T_{wo_j}
    - Historical poses of the camera: T_{wi}
  - Edges (Constraints):
    - Measurements of an object pose: Z_{i,o_j}
  - $T_{wo_j}, T_{wi}, Z_{i,o_j} \in \mathrm{SE}(3)$

- Additional constraints (optionally)
  - Camera-Camera motion: $Z_{i,i+1}$
  - Common ground plane: $P_{o_j,f}$

SLAM Map Representation

Example Graph:



Graph Optimisation

- Variables and constants:
  - Object and camera poses
  - Object measurements and camera-camera motions

- Minimization: Sum of all measurement constraints:

$$E_m = \sum_{Z_{i,o_j}} \left\| \log\left(Z_{i,o_j}^{-1} \cdot T_{wi}^{-1} \cdot T_{wo_j} \right) \right\|_{\Sigma_{i,o_j}} + \sum_{Z_{i,i+1}} \left\| \log\left(Z_{i,i+1}^{-1} \cdot T_{wi}^{-1} \cdot T_{wi+1} \right) \right\|_{\Sigma_{i,i+1}}$$

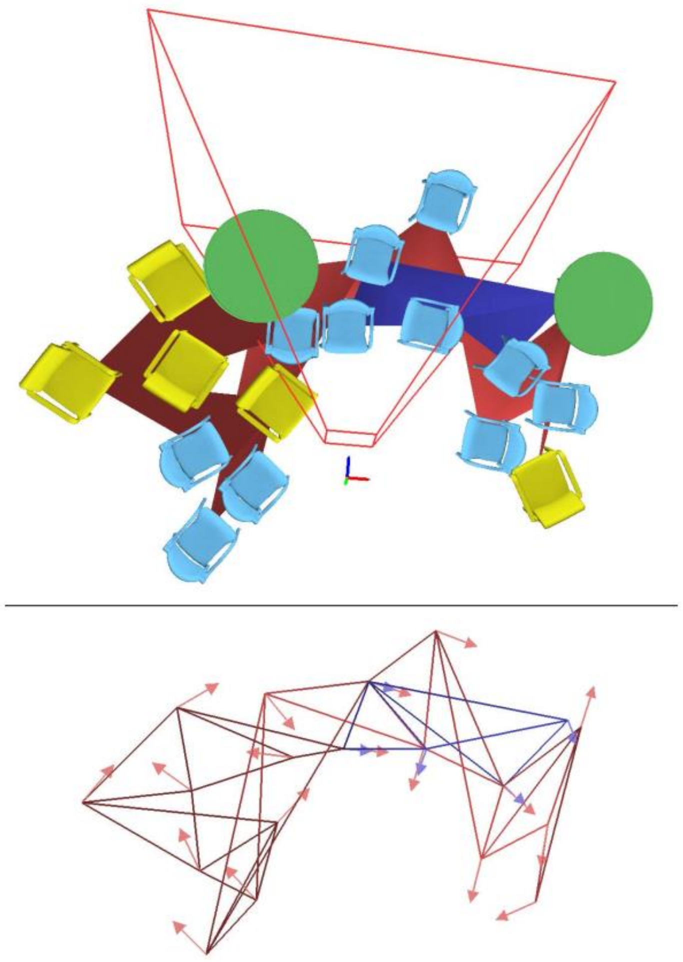
$\|x\|_\Sigma \coloneqq x^T \Sigma^{-1} x$ Mahalanobis distance

$\log(\cdot)$ Logarithmic map of SE(3)

- Generalised least squares problem
- Incorporation of other constraints additively

Relocalisation

- Lost camera tracking: Relocalisation
- Matching of local graph against long-term graph
- Matching procedure adapted from object recognition



Results and Applications

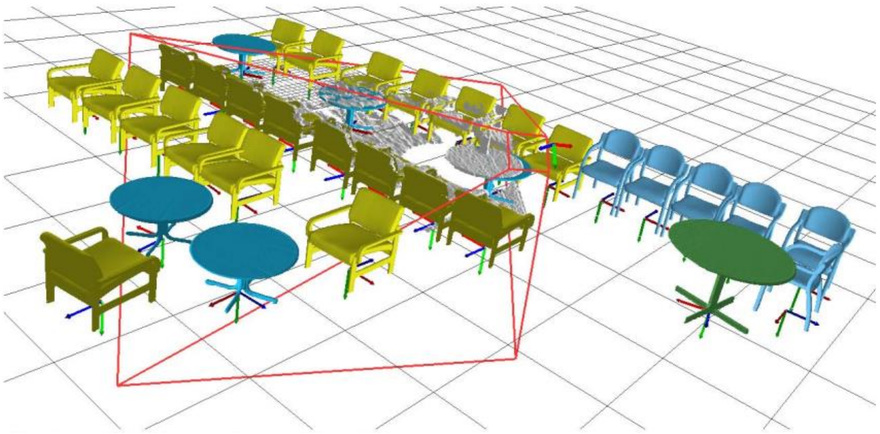
Statistics

- Room sizes mapped: 15 x 10 x 3 meters
- Runtime of real-time process: 10 minutes

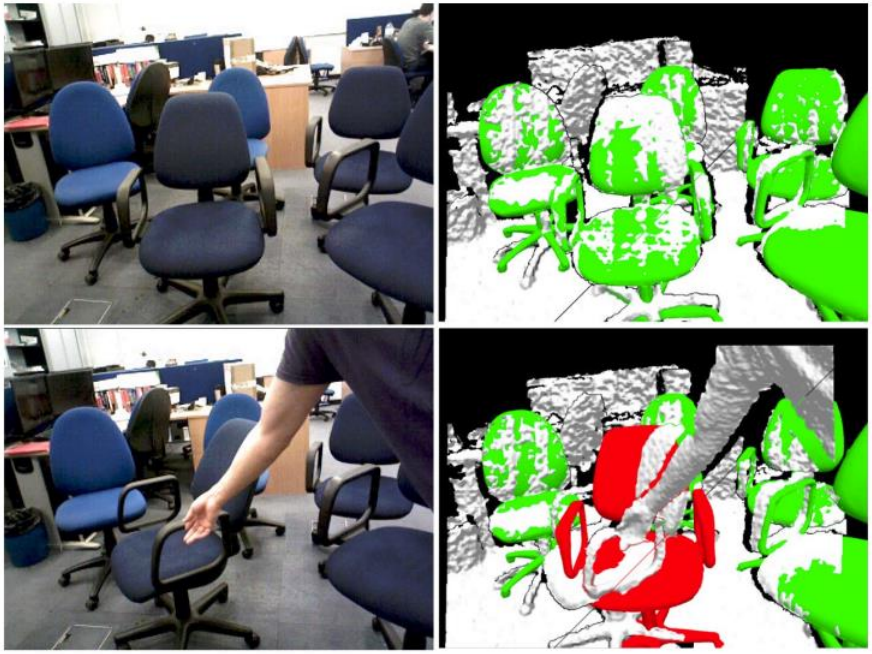
- System settings for mapping in a room of size 10 x 6 x 3 meters
  - Framerate: 20 fps
  - Object count: 35 objects in 5 classes
    Edge count: 338
  - Memory: 350 kB (Graph), 20 MB (Database)
    Comparison with KinectFusion: 1.4 GB

Applications

Large scale mapping

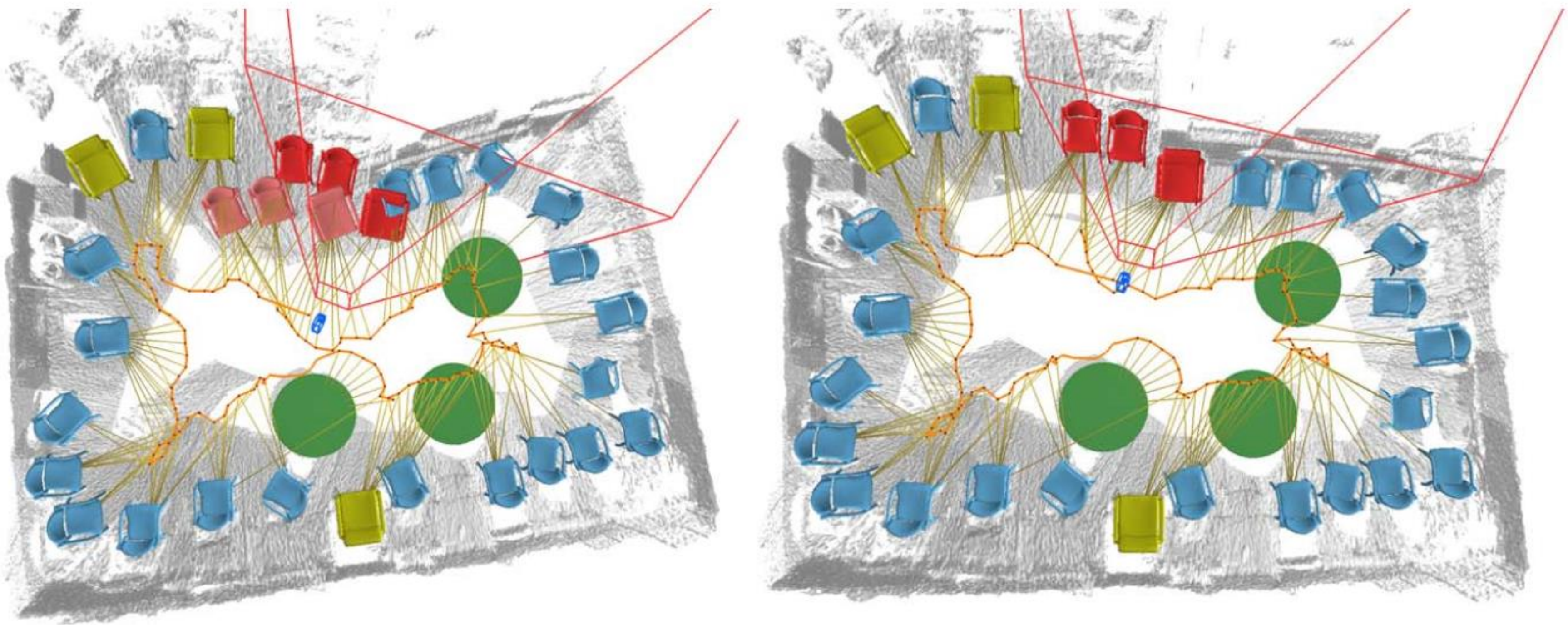


Detection of moved objects



Applications

Loop closure



Discussion of Scientific Contributions

Positives

- Conceptually new approach to the SLAM problem

- Advantages over dense SLAM:
  - Better scalability
  - Easier loop closure capability

- Advantages over feature-based SLAM:
  - Less features needed
  - More efficient and robust

Criticism

- Missing evaluation of accuracy
- Very structured environments needed
  → Repeated objects
- Environment cannot be completely unknown
  → Pre-defined database of objects