# Evaluation of the Modern Visual SLAM Methods

Artur Huletski, Dmitriy Kartashov, Kirill Krinkin

The Academic University

2015

# Introduction to SLAM

## Definitions

- **SLAM** (simultaneous localization and mapping) – is the computational problem of constructing a map of an unknown environment while simultaneously keeping track of a robot position within it.
- **Visual SLAMs** – is a set of SLAM methods that use visual information (e.g. from monocular camera) to solve SLAM problem.

## Things that make SLAM difficult

- Position estimate that based on raw odometry (e.g. IMU, shaft encoders) accumulates error very fast.
- Chicken-Egg nature of SLAM – accurate localization is essential for accurate mapping and vice versa.

# Research Aims

## Goals

- find out which of recently introduced visual SLAM methods can be used to navigate mobile ground robot;
- determine how visual SLAM methods should be improved in order to be useful for autonomous ground robot.

## Criteria to be evaluated

- Open-source implementation
- Recent publication (2013-mid.2015)
- Suitable for autonomous ground robot

# Candidates for Evaluation

## Rejected

- DV-SLAM (proprietary robot is used)
- StructSLAM (source code is not available; no response from the authors)
- Robust Large Scale Monocular VSLAM (source code is supposed to be published by the end of 2015)
- Quaternion Based EKF-SLAM (source code is not available; author suggested to use other optimization techniques)

## Accepted

- L-SLAM
- OpenRatSLAM
- ORB-SLAM
- LSD-SLAM

# L-SLAM

| | |
|---|---|
| Origin | an extension of FastSLAM 2.0 |
| Year | 2014 |
| Authors | Zikos N., Petridis V. |
| Input | Odometry, features from any detector |
| Implementation | Matlab, open-source |
| Robot orientation | Estimated by Rao-Blackwellized Particle Filter |
| Robot coordinates | Estimated by Kalman Filter |
| Loop closure | Feature matching |
| Map type | Cloud of features |
| Feature position | Estimated by Kalman Filter |
| Extra Refinement | Kalman Smoothing is used to refine previously estimated trajectory |

# OpenRatSLAM

| | |
|---|---|
| Origin | Model of hippocampus |
| Year | 2013 |
| Authors | Ball D., Wyeth G., Corke P., Milford M. |
| Input | Odometry (optional), RGB image |
| Implementation | C++, open-source, ROS package |
| Robot pose | Pose Cells Network |
| Loop closure | *Data structure*: Local View Cells<br>*Matching algorithm*: sliding window with SAD |
| Map type | *Graph-based*<br>*Edge*: distance and travel time<br>*Vertex*: position |

# ORB-SLAM

| | |
|---|---|
| Year | 2015 |
| Authors | Mur-Artal P., Montiel J., Tardos J. |
| Input | RGB image (ORB-features) |
| Implementation | C++, open-source, ROS package |
| Robot pose | Brute-force feature correspondence search |
| Loop closure | *Data structure*: Bag of words database<br>*Matching algorithm*: Multiple hypotheses |
| Map type | Cloud of features<br>Covisibility graph |
| Extra Refinement | Background graph optimization with g2o |

## Covisibility graph

- *Edge*: exists if two keyframes have $\theta$ common points
- *Vertex*: keyframe - camera position and detected ORB-features

# LSD-SLAM

| | |
|---|---|
| Origin | Monocular visual odometry method |
| Year | 2014 |
| Authors | Engel J., Schöps T., Cremers D. |
| Input | RGB image |
| Implementation | C++, open-source, ROS package |
| Robot pose | *Source*: Implicitly estimated by temporal stereo<br>*Method*: Photometric error minimization |
| Loop closure | *Candidates*: 10 most closest keyframes<br>*Algorithm*: Estimation of fwd and bwd transforms |
| Map type | *Graph-based*<br>*Edge*: scale-aware transformation<br>*Vertex*: Keyframe (image, inverse depth map) |
| Inverse depth map | *Estimator*: per-pixel Kalman filter<br>*Tracking requirement*: high pixel's gradient |
| Extra Refinement | Background graph optimization with g2o. |

# Comparison Approach

## Theoretical

Compare assumptions made during an algorithm derivation.

## Practical

Measure root-mean-square-error on estimated trajectory with respect to ground truth.

- RGB-D dataset from TUM
- Number runs for each sequence: 15
- RMSE was estimated by TUM's evaluation tool

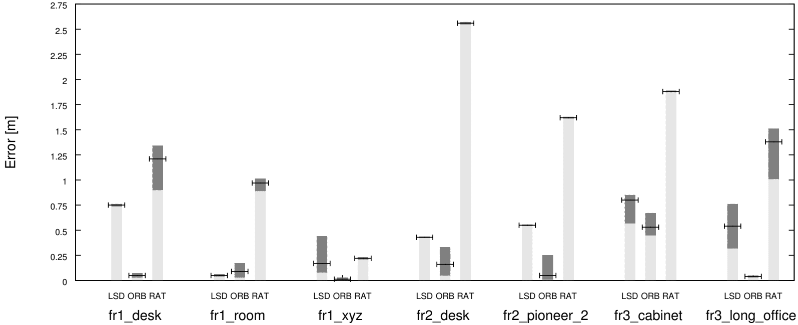
# Theoretical Comparison

| **Algorithm** | **Assumption** |
|---|---|
| OpenRatSLAM | - weights of links between PCN and LVC are Gaussian |
| L-SLAM | - detector noise is low and Gaussian |
| ORB-SLAM | - used BoW vocabulary is relevant<br>- ORB-features can be reliably detected |
| LSD-SLAM | - noises and inverse depth are Gaussians<br>- some pixels have high gradient values |

## Common assumption for visual odometry

Robot movement between two consecutive frames is relatively small.

# Practical Comparison. RMSE




## Notes

- L-SLAM hasn't been tested because it doesn't have visual odometry and ROS implementation right now.
- ORB-SLAM output trajectories were "manually" scaled.

# Practical Comparison. Tracking robustness

| Seq. name | LSD | ORB | Rat |
|---|---|---|---|
| fr1_desk | 7% | 100% | 100% |
| fr1_room | 0% | 87% | 93% |
| fr1_xyz | 93% | 100% | 100% |
| fr2_desk | 7% | 100% | 93% |
| fr2_pioneer_slam2 | 7% | 100% | 100% |
| fr3_large_cabinet | 20% | 40% | 100% |
| fr3_long_office_household | 0% | 100% | 100% |

## Notes

- OpenRatSLAM was provided with data at 15Hz rate instead of 30Hz to make tracking robust.
- Suggested parameters were tweaked for LSD-SLAM in order to achieve better tracking rate.
- ORB-SLAM was run using vocabulary provided out of the box.

# Conclusion

Several modern SLAM methods (L-, LSD-, ORB- and OpenRat-) were analyzed and evaluated with RGB-D dataset provided by TUM. Our tests showed that estimated trajectory has significant error or requires manual post-processing.

The following requirements for autonomous visual SLAM can be formulated based on our experience:

- an algorithm should be scale-aware
- an algorithm should do some recovery when tracking is lost

# Q & A

# Paper References

## L-SLAM

N. Zikos, V. Petridis, “6-DoF Low Dimensionality SLAM (L-SLAM)”;

## OpenRatSLAM

D. Ball et al., “OpenRatSLAM: an open source brain-based SLAM system”;

## ORB-SLAM

R. Mur-Artal, J. D. Tardos, “Fast relocalisation and loop closing in keyframe-based SLAM”;

## LSD-SLAM

J. Engel et al., “LSD-SLAM: Large-Scale Direct Monocular SLAM”.