ORB-SLAM and Graph Based Optimization

Lu Yu

Senior Algorithm Engineer

iMorpheus.ai

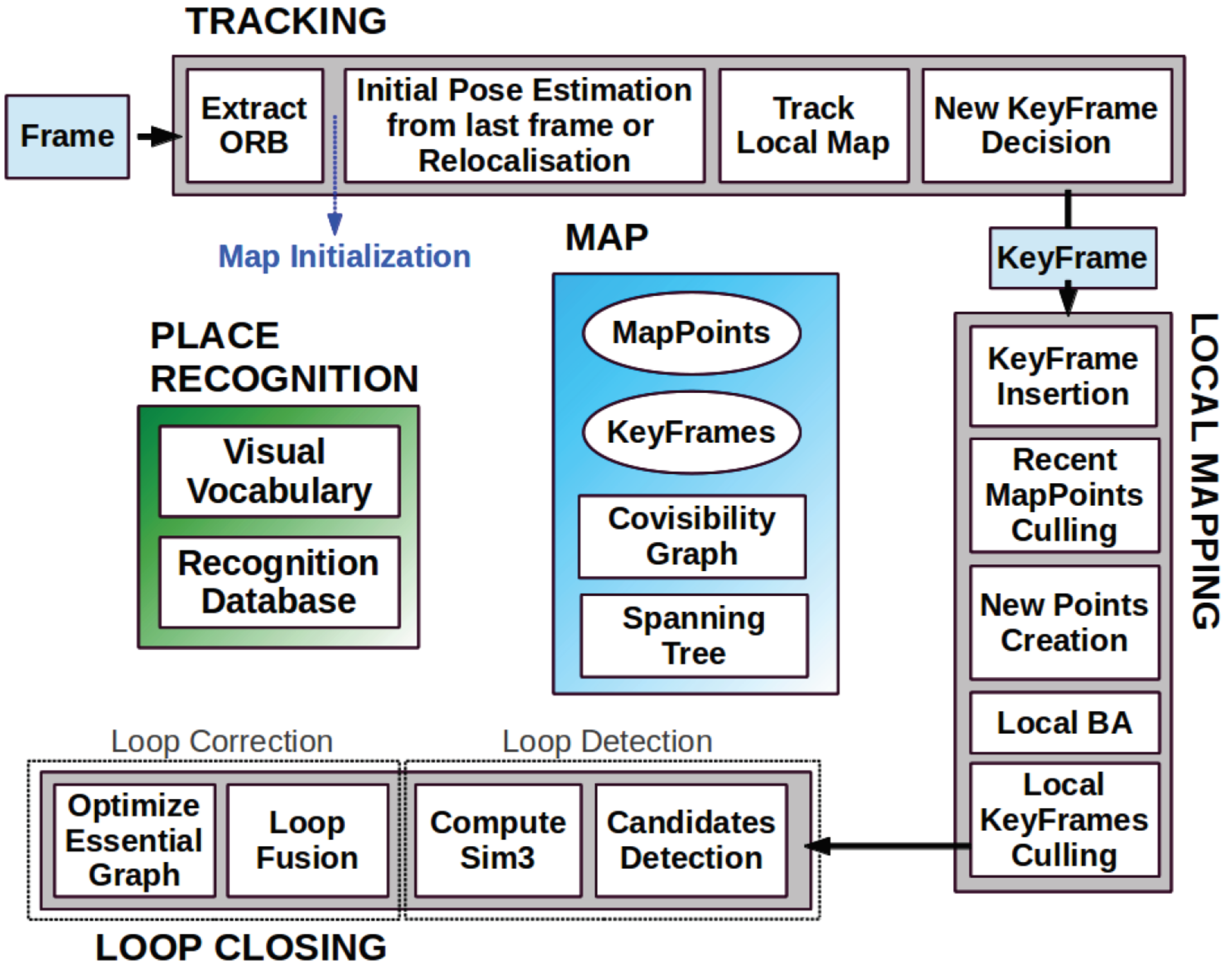
ORB-SLAM: a Versatile and Accurate Monocular SLAM System by Mur-Artal

- ORB-SLAM is a keyframe and feature-based monocular SLAM system that operates in real time, in small and large, indoor and outdoor environments. The system is robust to severe motion clutter, allows wide baseline loop closing and relocalization, and includes full automatic initialization.
- Input: monochrome or color images
- Output: camera trajectory and poses, sparse 3D points, keyframe poses and covisibility graph
- Open source project at GitHub: github.com/raulmur/ORB_SLAM

Main contributions of ORB-SLAM

- Use of the same features (ORB method) for all tasks: tracking, mapping, relocalization and loop closing. Real-time performance without GPU.
- Covisibility graph for local tracking and mapping.
- Essential graph for global optimization (loop closing).
- Real-time camera relocalization.
- An automatic and robust initialization procedure.
- A survival of fittest approach for keyframe selection.

System structure



Oriented FAST and Rotated BRIEF (ORB)

- ORB used for feature extraction and description.
- Features from Accelerated Segment Test (FAST) is used to detect corners and extract feature points. Orientation is recorded by displacement vector between center (intensity centroid) and detected corner.
- Binary Robust Independent Elementary Features (BRIEF) is a bit string feature descriptor for image patch. ORB adds rotation invariance to BRIEF.
- Refer to *ORB: an efficient alternative to SIFT or SURF* by Rublee.

Map points and keyframes

- Each map point stores: its 3D position in the world coordinate system; viewing direction, which is the mean unit vector of all its viewing directions; representative ORB descriptor; maximum and minimum distances at which the point can be observed.
- Each keyframe stores: camera pose; camera intrinsics, including focal length and principal point; all ORB features extracted in the frame.
- New map points and keyframes are spawned under a generous policy. But their culling procedure is very strict.
- Keyframes whose 90% of the map points have been seen in at least three other keyframes in the same or finer scale are discarded.

Relocalization

- If tracking is lost, we convert current frame into bag of words and query the recognition database for keyframe candidates for relocalization.
- Compute correspondences with ORB associated to map points in each keyframe. Perform RANSAC iterations for each keyframe and try to find a camera pose using PnP algorithm.
- If a camera pose with enough inliers is found, optimize the pose and perform a guided search of more matches with the map points of the candidate keyframe.
- Finally the camera pose is again optimized, and if supported with enough inliers, tracking procedure continues.

Bag of words place recognition

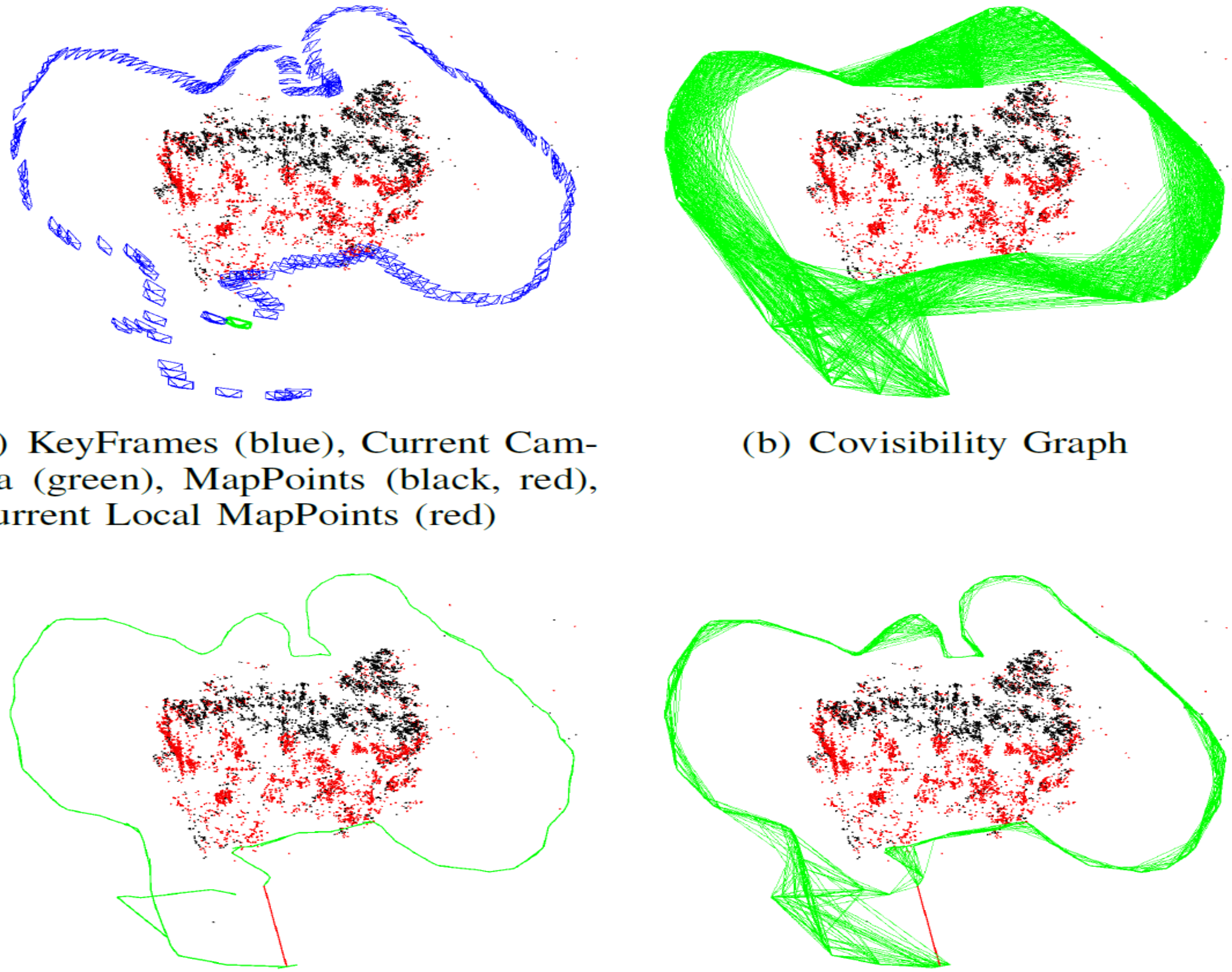
- Bag-of-words model treats image features as words. An image is given a sparse vector of occurrence counts of words. Classification of images can be done by comparing those vectors.
- Used in image matching and loop closing.
- DBoW2 is an open source C++ library for indexing and converting images into a bag-of-word representation on GitHub.

1. ORB feature descriptors are extracted from training scenes.
2. K nearest neighbor clustering of descriptors.
3. Repeat Step 2 and obtain a hierarchical tree. Leaf nodes are descriptors and other nodes are clustering centers.

Graph based optimization

- A graph is an ordered pair G = (V, E) comprising a set V of vertices together with a set E of edges. In weighted graphs, each edge is given a numerical value called weight.
- Each keyframe is treated as a vertex. An edge exists between two vertices if they share observation of common map points (above a certain threshold), with weight equal to the number of shared map points.
- For global optimization, number of edges used should be small. So essential graph is introduced. It contains a spanning tree of covisibility graph and edges with weight greater than a threshold (100 in experiments).

Covisibility graph and essential graph



(a) KeyFrames (blue), Current Camera (green), MapPoints (black, red), Current Local MapPoints (red)

(b) Covisibility Graph

(c) Spanning Tree (green) and Loop Closure (red)

(d) Essential Graph

Graph based optimization Mathematics

- Objective function $\min F(x) = \sum_k e_k^T \Omega_k e_k$, where e_k is error term and Ω_k is covariance matrix. For simplicity, Ω_k can be diagonal matrix.
- e_k can take any form. For example, when camera at pose T_k makes an observation of map point x_k and obtains result z_k, error term $e_k(x_k, T_k, z_k) = z_k - C(Rx_k + t)$, where C is camera intrinsic parameter, R is rotation and t is translation.
- Common optimization techniques, such Gauss-Newton or Levenberg-Marquardt, are applied. The key is that only a small number of vertices share enough common map points (vis covisibility graph), thus matrix involved is sparse.
- Lie group, Lie algebra, manifold.

Loop closing with covisibility and essential graphs

- For a new keyframe K_i, compare bag-of-words vector in its neighborhood in covisibility graph and obtain lowest score s_{min}. Then we discard all those keyframes whose score is lower than s_{min} and those directly connected to K_i.
- To accept a loop candidate three consecutive loop candidates that are consistent must be detected. There can be several loop candidates if there are several places with similar appearance to K_i.
- Essential graph is used in loop closing and graph optimization, which greatly reduces computational complexity by considering only a small number of edges.

Results NewCollege sequence

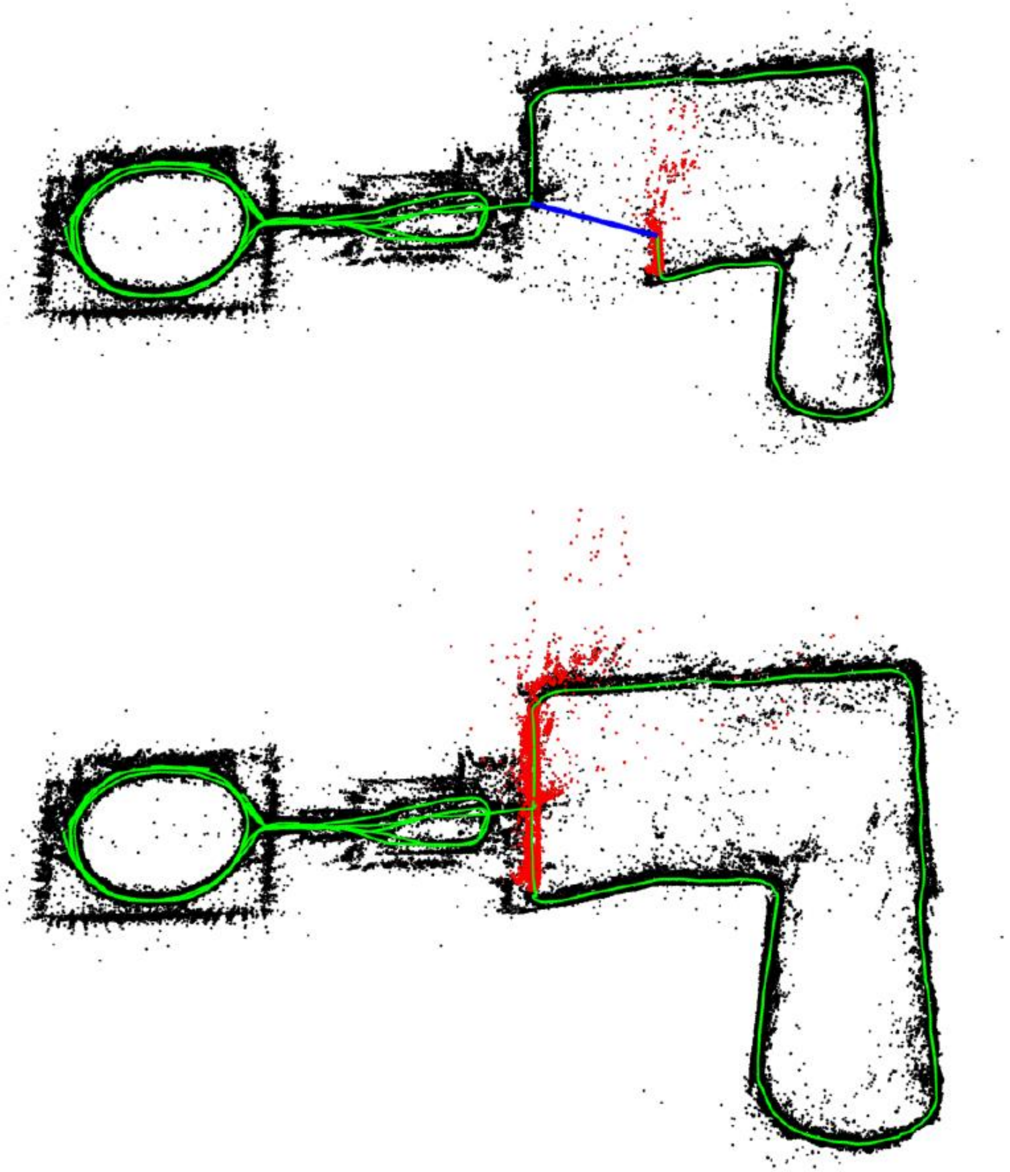
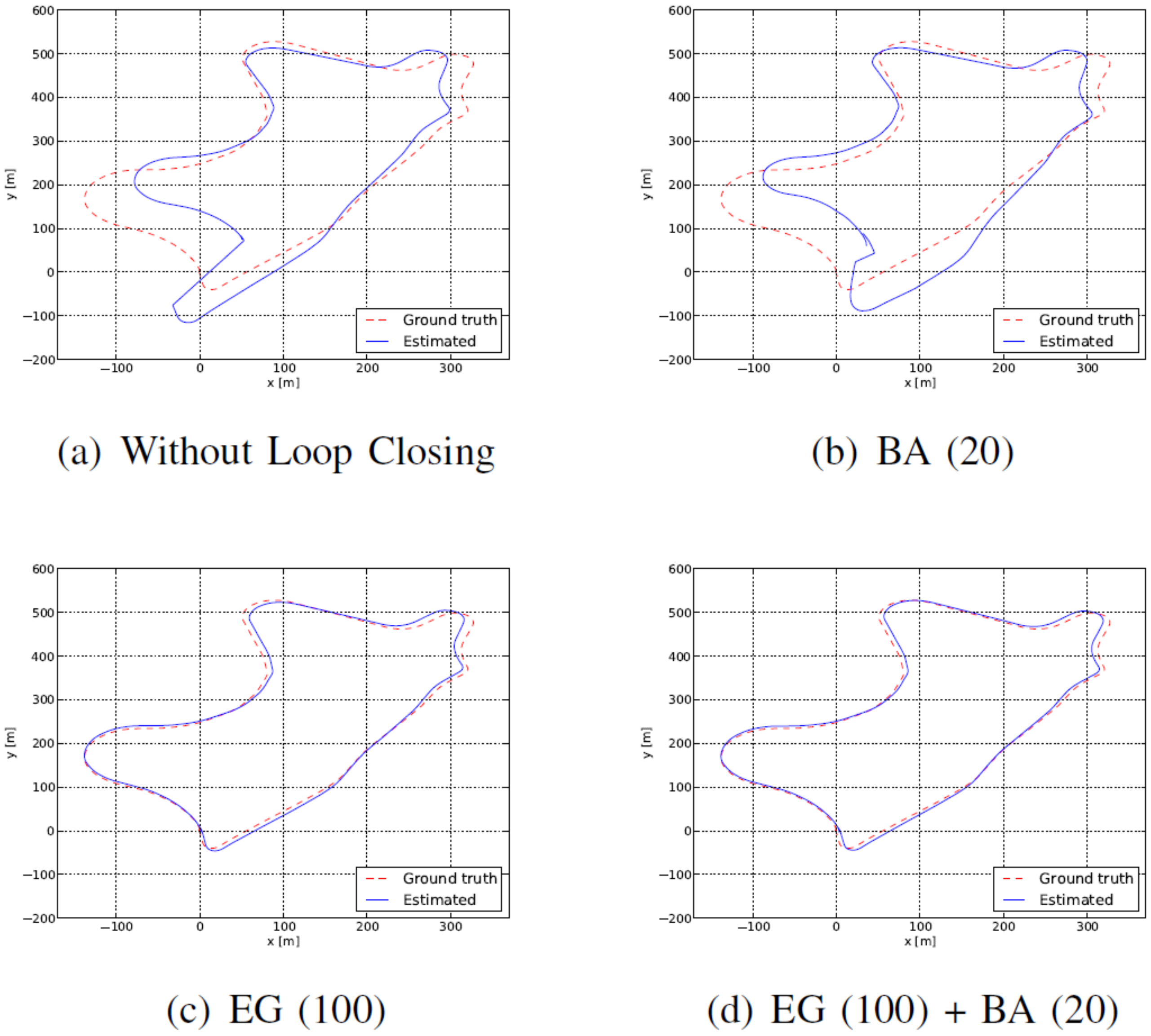


Fig. 5. Map before and after a loop closure in the NewCollege sequence. The loop closure match is drawn in blue, the trajectory in green, and the local map for the tracking at that moment in red. The local map is extended along both sides of the loop after it is closed.

Results KITTI data set



(a) Without Loop Closing

(b) BA (20)

(c) EG (100)

(d) EG (100) + BA (20)

Fig. 13. Comparison of different loop closing strategies in KITTI 09.

ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras

- Open-source SLAM system for monocular, stereo and RGB-D cameras, including loop closing, relocalization and map reuse.
- RGB-D results show that by using Bundle Adjustment, ORB-SLAM 2 achieves more accuracy than state-of-art methods based on ICP or photometric and depth error minimization.
- By using close and far stereo points and monocular observations, the stereo results are more accurate than the state-of-art direct stereo SLAM.
- A lightweight localization mode that can effectively reuse the map with mapping disabled.

Summary
Advantages of ORB-SLAM

- Uses mainstream methods that have been proved to be good in visual SLAM. A complete study of ORB-SLAM helps to learn many key ideas in visual SLAM.
- Nice results. And they are repeatable.
- Very strong loop closing performance.
- Open source project on GitHub. Good code quality.

Summary
Problems of ORB-SLAM

- Monochrome and monocular camera yields (literally) sparse 3D points. If main interest is the track, this is not a problem. Otherwise, there are better choices for SLAM.
- Can be implemented real-time on laptop PC with quad-core CPU. But not real-time on embedded system or mobile phone.
- Preloading of ORB vocabulary tree (currently 138 MB) takes time, especially on light systems such as mobile phone.
- Lots of magic numbers (parameters, thresholds) that are potentially sub-optimal.

Thank you!
Questions?

Lu Yu

Senior Algorithm Engineer

iMorpheus.ai