Loop-closure for global consistency in SLAM

Shobhit Srivastava, Sandeep Konam

Robotics Institute, School of Computer Science, Carnegie Mellon University

Overview

The goal of this project is to develop a system capable of generating globally consistent maps. Components include:

- Stereo visual-inertial perception head as the sensor
- Pose estimation based on stereo-visual odometry
- Feature appearance based place-matching for loop-closure detection
- A pose-graph formulation for error minimization in pose estimates

The Perception Head

- Exynos 5 based ARM system serves as an embedded CPU
- Two Matrix Vision BlueFox Cameras mounted to a custom carbon-fiber crossbar, creating a lightweight yet torsionally-rigid stereo rig.
- YEI inertial measurement unit (IMU). Provides IMU pose measurements at a frequency of 200Hz
- An IMU-Camera USB hub to multiplex the cameras and the IMU

Our contribution

- Wrote the IMU-Camera USB hub firmware to trigger the cameras from the IMU at the desired frequency
- This ensures that the images from the two cameras are in sync
- Modified the ROS driver for Bluefox cameras so as to support external trigger.

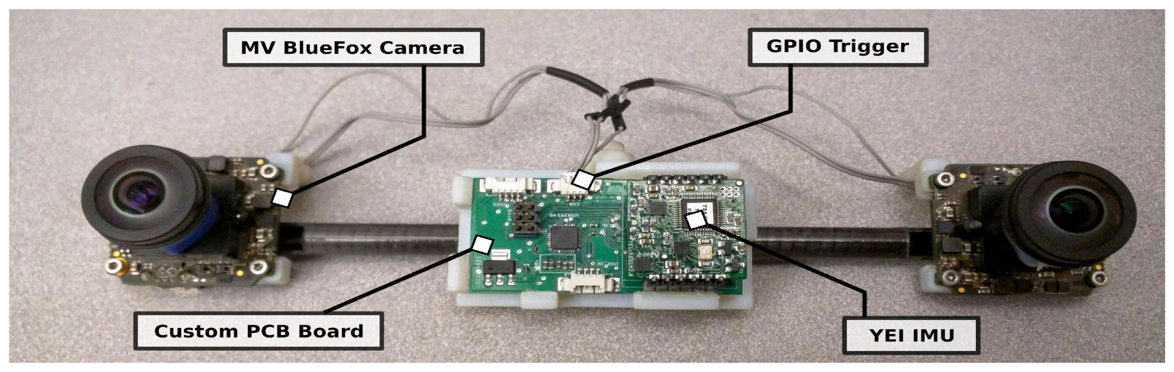


Figure 1: The custom perception sensor package include three components: (1) two off-the-shelf cameras, an embedded IMU module, and a custom PCB that includes an MPU that aligns cameras images and IMU observations.

Vision-based State Estimation

- Raw image undistortion and stereo-rectification
- Shi-Tomasi Feature detector
- KLT feature tracker
- Stereo-matching to find epipolar correspondences
- PNP to calculate the fundamental matrix

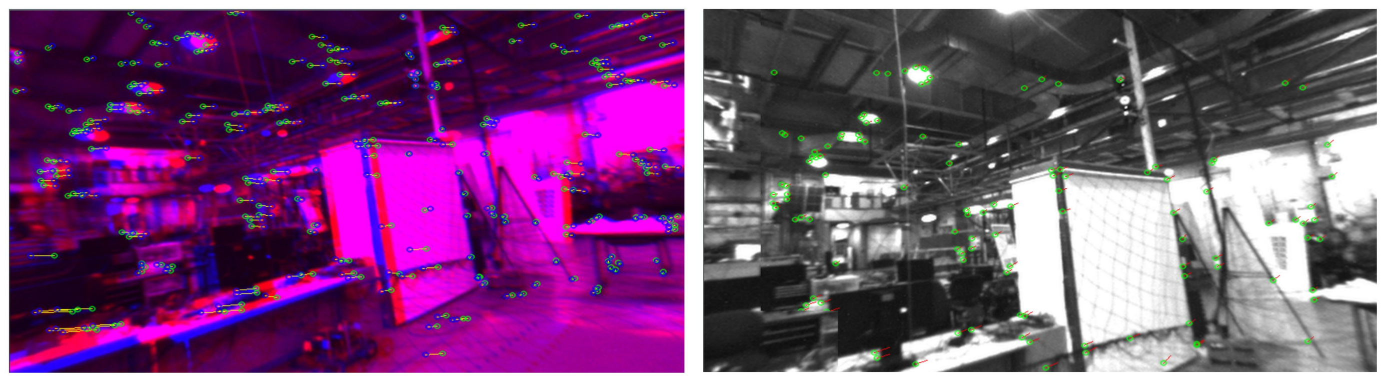


Figure 2: Stereo and single-camera images showing tracked features while moving, in an indoor environment

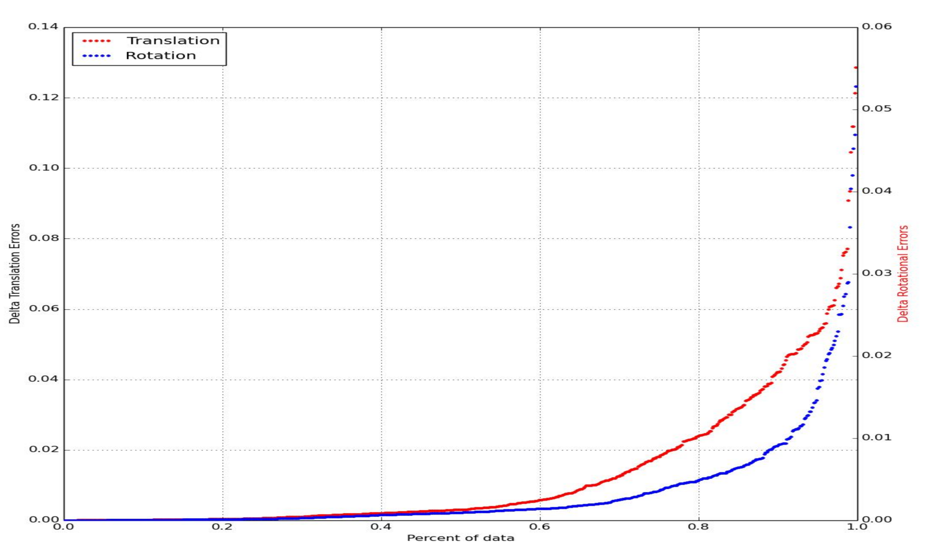


Figure 3. Percent Error (Translation Error is in meters; Angular error is in radians)

Loop Closure

- Pose-estimates obtained from Visual Odometry have an inherent drift due to the accumulation of error overtime
- To correct this drift, a loop-closure system that is independent of odometry is required
- Components
  - Place-matching system : Detects if the robot is re-visiting a particular place
  - Bundle-Adjustment : Performs a least squared error minimization over the pose estimates and the landmark-measurements

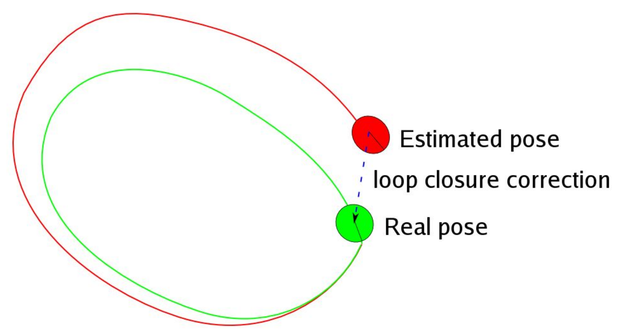


Fig 4. Illustration of loop-closure [6]

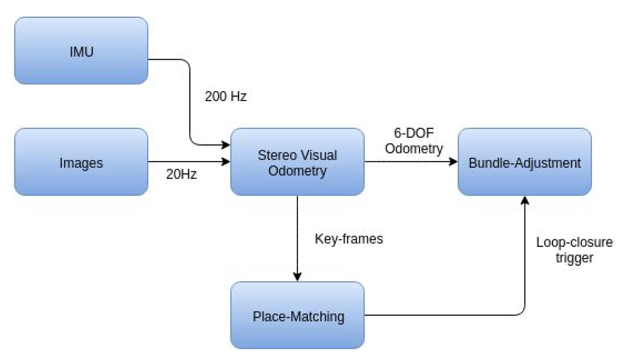


Fig 5. High-level system diagram for the loop-closure system

The Place-Matching System

- An adaptation of FABMAP 2.0 [1][2] to work in a lab environment
- A feature-appearance based matching approach.
- A Chow-Liu tree is trained to distinguish between objects that appear similar.
- Collected data-sets of the Gates Highbay to train a vocabulary and the Chow-Liu tree
- Experimented with SURF, STAR and FAST feature-detectors and found that STAR provides the best performance

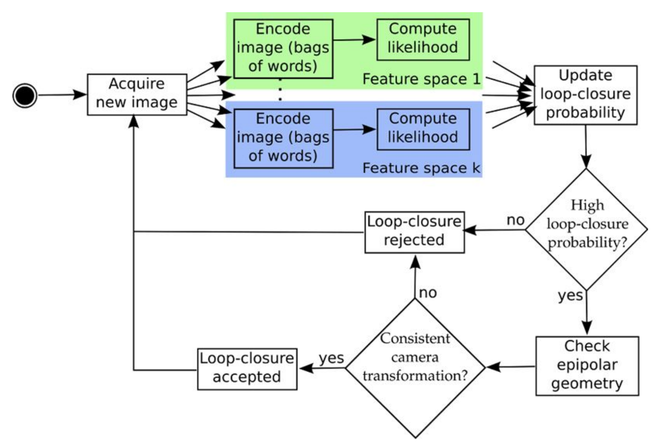


Fig 6. Control - flow diagram for the Place-matching system [4]

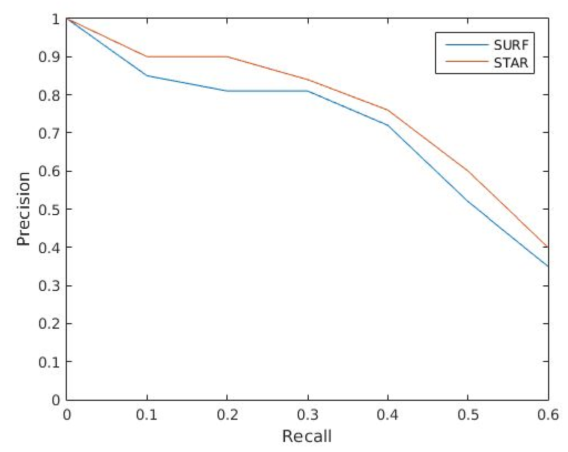


Fig 7. Precision-Recall curve for the place-matcher (SURF and STAR)

Bundle-Adjustment

- A pose-graph formulation for minimizing the error in the estimated poses of the robot and the landmarks.
- Bundle-adjustment is triggered by the place-matching system.
- When a loop-closure is detected, a constraint between the current pose and the candidate pose is added.
- The squared error in the pose-estimates and the reprojection error for the landmarks are minimized
- G2O was used as a graph-solver with Lavenberg-Marquardt as the optimization algorithm

$$\mathbf{e}_{ij}(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{z}_{ij} - \hat{\mathbf{z}}_{ij}(\mathbf{x}_i, \mathbf{x}_j).$$

$$\mathbf{F}(\mathbf{x}) = \sum_{(i,j)\in\mathcal{C}} \underbrace{\mathbf{e}_{ij}^T \Omega_{ij} \mathbf{e}_{ij}}_{\mathbf{F}_{ij}},$$

[3]

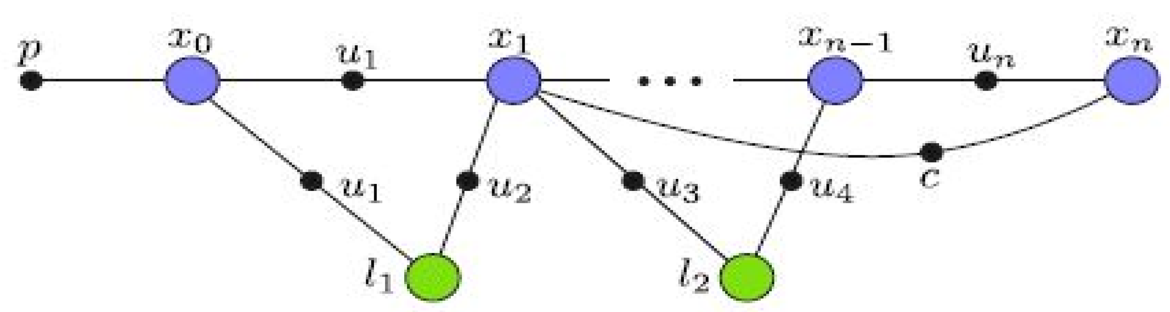


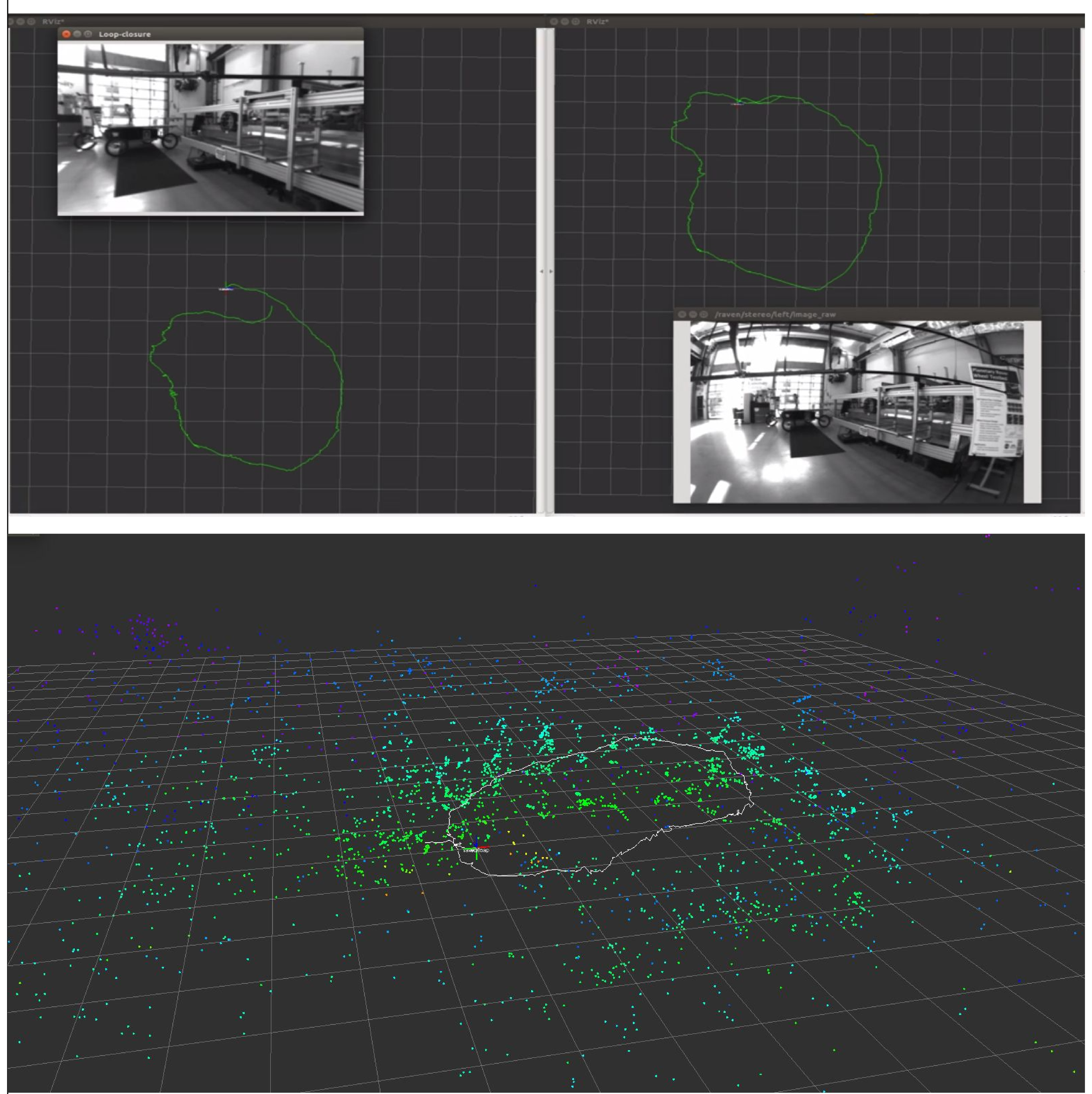
Fig 8. A pose-graph representation. Every node in the graph corresponds to a robot pose. Nearby poses are connected by edges that model spatial constraints between robot poses arising from measurements. [5]

Datasets

- A total of 10 data-sets along 5 different trajectories were collected (each trajectory done twice)
- Each data-set contains:
  - Raw images from the left camera at 20 Hz
  - Raw images from the right camera at 20 Hz synced with the left image
  - IMU measurements at 200 Hz
  - IMU measurements corresponding to the images (20 Hz)
  - Time-stamps and sequence-id for each element

Fig 9. A sample of the data-set. A raw image from the left-camera

Results



Length	16.4 m
End-point drift	0.5 m
# Landmarks	5213
# Pose-nodes	733
Optimization Time (Pose)	4 ms
Optimization Time (Landmarks)	714 ms

Fig 10. Results. (a) The system in action. Loop-closure detection candidate image(top-left). Actual trajectory snapshot (bottom-right). The path estimated by odometry (bottom-left). The corrected path (top right). (b) The corrected map with landmarks. (c) Some numbers related to hte optimization and the trajectory

References

[1] Mark Cummins and Paul Newman, “Highly Scalable Appearance-Only SLAM - FAB-MAP 2.0” RSS 2009, Seattle
[2] Mark Cummins and Paul Newman, "FAB-MAP: Appearance-Based Place Recognition and Mapping using a Learned Visual Vocabulary Model", Invited Applications Paper, ICML 2010
[3] Grisetti, G.; Kümmerle, R.; Stachniss, C.; Burgard, W., "A Tutorial on Graph-Based SLAM," in Intelligent Transportation Systems Magazine, IEEE , vol.2, no.4, pp.31-43, winter 2010
[4] Angeli, A.; Filliat, D.; Doncieux, S.; Meyer, J.-A., "Fast and Incremental Method for Loop-Closure Detection Using Bags of Visual Words," in *Robotics, IEEE Transactions on* , vol.24, no.5, pp.1027-1037, Oct. 2008
[5] http://people.csail.mit.edu/kaess/isam/doc/Tutorial.html
[6] http://cogrob.ensta-paristech.fr/loopclosure.html