Robust Adaptive Systems Lab,
The Robotics Institute,
Carnegie Mellon University

Results and Analysis

"Robust Loop Closures in visual Odometry using RGB-D cameras"

By Siddharth S Jha
IIT Kharagpur

Submitted To:
Prof. Nathan Michael
Carnegie Mellon University

As a part of a 9 week long summer internship

Overview of the Task

Using only a visual camera input, check for similar looking images that may be used as potential loop closures in a visual odometry problem. Calculate the transforms between the candidates for loop closure for addition into the overall pose graph.

Approach Used

The backbone of this task was the open source library DLoopDetector by Dorian Galvez. It is available at https://github.com/dorian3d/DLoopDetector. The library itself is based upon DBoW2, a bag of words library by the same person. It calculates bag of word vectors for every image and matches them with a pre-trained BRIEF descriptor vocabulary in order to detect loop closures. It also verifies the structural sanity and performs outlier rejection in order to eliminate false positives. A wrapper for the DLoopDetector library was writen, and an essential matrix estimation approach with RANSAC outlier rejection was added to it, in order to detect loop closures. For the transform estimation, GTSAM based Perspective n points based method was used, where n = 30 (MAX), and the reprojection error was minimized using a Levenberg Marquadt optimizer. The 3D point information was obtained from the depth camera, and consistency check for the depths was added in order to eliminate instability during optimization. For tracking, another approach used was the Gauss Newton tracker based on SSD (sum of squared differences) from the dense visual tracker package. The entire loop detection and pose estimation was performed on keyframes in order to minimize dependance on CPU. Keyframes were detected using forward-backward optical flow and appended using the heuristic comparison of the area covered by the tracked keypoints.

The overall codebase consists of 5 .cc files and their corresponding headers. A separate header for dense visual tracking is present.

Codebase

KeyframeAppender.cc:
Implements the forward backward optical flow algorithm to detect keyframes from the default video sequence, so that the loop closer and the trajectory optimization does not go very hard on CPU resources. It calls the LoopDetector and PnPSolver/SSDTracker.

LoopDetector.cc
Wrapper for the DLoopDetector Library that detects loops based on a bag of words approach, using FAST corners and BRIEF descriptors on a pre-trained vocabulary.

PnPSolver.cc
This implements a sparse optimization based PnP solver based on RGBD images data. Points considered as matches between consecutive keyframes (calculated using Forward-Backward Optical Flow) or loop closure keyframes (using FAST features & BRIEF descriptors), are sent to this optimizer to roughly estimate the motion of the camera.

DeadReckoning.cc
This file implements the trajectory optimizer for the overall keyframe based trajectory estimation algorithm. Uses GTSAM to estimate the position of the camera based on the priors provided via SSD Tracker, OpenCV based PnP solver or GTSAM based PnP solver. It currently uses the Levenberg Marquadt optimizer, but ISAM2 can also be used if required.

Visualization.cc
Function to visualize optimized trajectories using RViz. TODO: Change the blob markers to actual tf2 objects that can lead to better visualization.

Running the Codebase

A bag file/live sensor with topics:
- /camera/rgb/image_rect_color
- /camera/depth/image_rect
- /vicon/areneae07_odom (if ground truth comparisons need to be seen).

roslaunch vins_estimator dead_reckoning.launch

Saves images in home directory.

Results of Loop Closure after outlier rejection:
The white circles show some of the feature points matched between the two images after outlier rejection. The original image would show more points than tracked images.

<u>Dataset 1 (inside Master's lounge) (31 seconds long)</u>

4-7 loop closures detected

Output 1:

Ground Truth (VICON)

Translation from current to previous keyframe:
-0.291756, -0.00395886, 0.103857

Rotation matrix from current to previous keyframe:
[0.9980852301159218, 0.06150162034576986, 0.006589697966707622;
 -0.06062765140511196, 0.9938409219084025, -0.09276049711680577;
 -0.0122540323790933, 0.09218336419930387, 0.9956666440407405]

Method 1 (SSD Dense Tracker)
Estimated Translation:
-0.297048
-0.032777
0.106225

Estimated Rotation matrix:

0.998285	0.0578879	0.00871338
-0.0566048	0.992485	-0.108482
-0.0149277	0.107803	0.99406

Method 2 (GTSAM PnP solver)

Translation Mat:
[-0.2848433127385341;
 -0.02398772945580716;
 0.0914167947220129]

Rotation Mat:
[0.998181908093789, 0.06011099322312575, 0.004629592783412123;
 -0.05929996664022179, 0.9927594366575041, -0.104465021323426;
 -0.01087563527252184, 0.1040004883906344, 0.9945178295835077]

Loop Closure Candidate (keyframe 13)

Current Keyframe (keyframe 41)

Output 2:

Ground Truth (VICON)
Translation from current to previous keyframe:
-0.409177, -0.158026, 0.0464627

Rotation from current to previous keyframe:
[0.993980461926425, 0.06259573734852701, -0.08991448701029607;
 -0.06345162172497007, 0.9979624961020155, -0.006689400145445668;
 0.0893125579580038, 0.01235435306395282, 0.9959270239085639]

Method 1 (SSD Dense Tracker)

Estimated Translation:
-0.394487
-0.145345
 -0.0167239

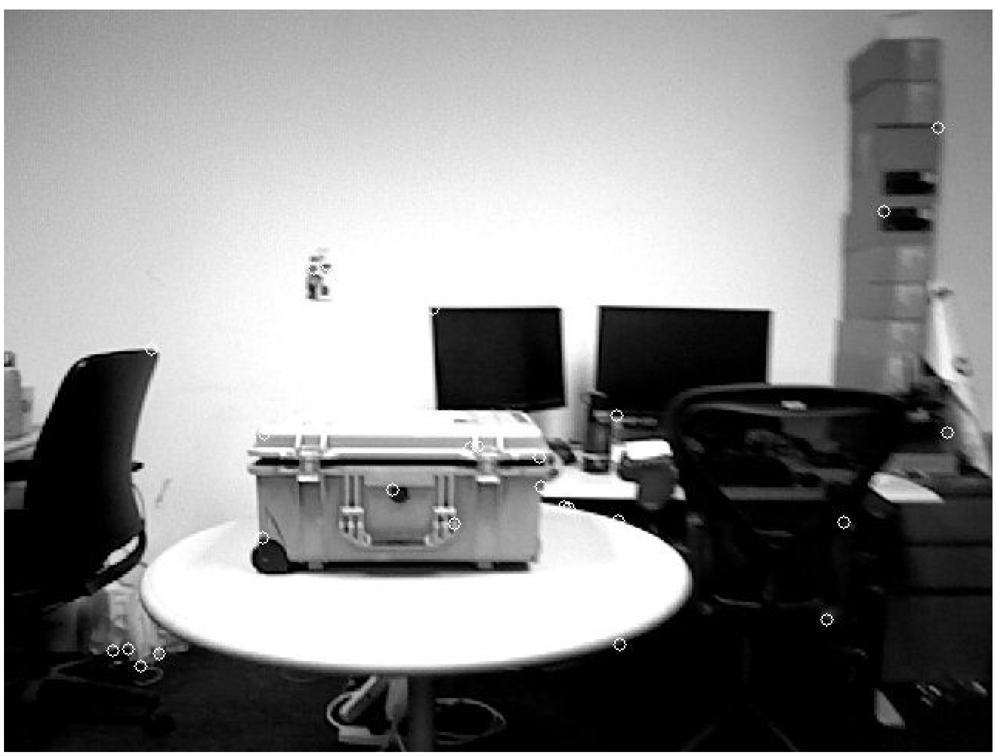
Estimated Rotation:
0.99475 0.0615372 -0.0817606
-0.0616275   0.998099 0.00142076
 0.0816925 0.00362542   0.996651

Method 2 (GTSAM PnP solver)

Translation Mat:
[-0.3954343389447448;
 -0.1852108998028772;
 0.02870883117642043]

Rotation Mat:
[0.9963189620226237, 0.04551872370508989, -0.07265300716712043;
 -0.0456152616795877, 0.9989593938078668, 0.0003307792296361073;
 0.07259236305648803, 0.002984549204258008, 0.9973572720058896]

Loop Closure Candidate (keyframe 16)



Current Keyframe (keyframe 44)

Set of all detected good matches for dataset 1 for a two runs are [uploaded here.]()

Dataset 2 (outside) (193 seconds long)

40-50 loop closures

Note 1: This dataset was taken on Areneae01 robot for which the camera is inverted wrt the body, but the results are in body frame.

Note 2: No ground truth data is available for this dataset. So I am just showing the results of loop closure detection, SSD Tracking and GTSAM PnP.

Output 1:

Method 1 (SSD Dense Tracker)

Estimated Translation:
0.108813
-0.188657
-0.00142989

Estimated Rotation:

0.971471	-0.236579	-0.0166064
0.235927	0.971176	-0.0339897
0.024169	0.029102	0.999284

Method 2 (GTSAM PnP solver)

Translation Mat:
[0.1021736884754924;
-0.1913546389664584;
-0.003728865136764189]

Rotation Mat:
[0.9709254846777025, -0.2387810176711783, -0.01700539683150215;
0.2379869706454538, 0.9704811887592102, -0.03910530681651739;
0.02584108626378701, 0.03392126446585493, 0.9990904247719539]

Loop Closure Candidate (Keyframe 221)

Current Keyframe (Keyframe 403)

Output 2:

Method 1 (SSD Dense Tracker)

Estimated Translation:
-0.149242
0.287291
0.0302946

Estimated Rotation:

0.994056	0.107592	0.0166088
-0.1088	0.987159	0.116957
-0.00381191	-0.118069	0.992998

Method 2 (GTSAM PnP solver)

Translation Mat:
[-0.1606927872448934;
0.2739995900521049;
-0.03292123167995982]

Rotation Mat:
[0.99420627212484, 0.1061590582651309, 0.01691229365515753;
-0.1073989054673451, 0.9876673803465997, 0.1139269805292242;
-0.004609269407289424, -0.1150831389852414, 0.9933452135951174]

Loop Closure Candidate (Keyframe 218)

Current Keyframe (Keyframe 437)

Output 3:

Method 1 (SSD Dense Tracker)
Estimated Translation:
-0.0388997
0.0714455
0.0250539

Estimated Rotation:

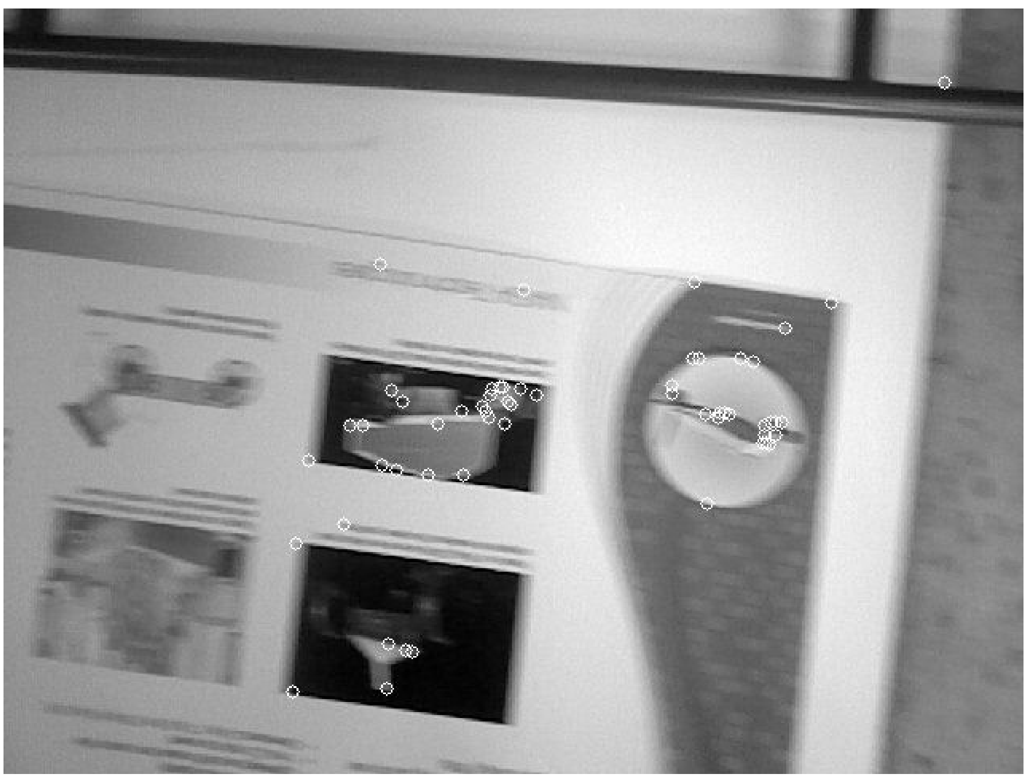
0.999211	-0.02487	-0.0309857
0.0239774	0.999296	-0.0288527
0.0316814	0.0280869	0.999103

Method 2 (GTSAM PnP solver)

Translation Mat:
[-0.03831004703670274;
0.06147120911834879;
0.01849852507641268]

Rotation Mat:
[0.9987672714236954, -0.03165191655609451, -0.03826196292516093;
0.03060981107731478, 0.9991528016270305, -0.02752178596434812;
0.03910072139807555, 0.02631667088335479, 0.9988887117958438]

Loop Closure Candidate (Keyframe 282)



Current Keyframe (Keyframe 496)

Rest of the loop closed images of the dataset for 2 runs are [uploaded here.]

Discussions and Future Work

The ground truth comparisons show that both the dense SSD tracker and the GTSAM PnP algorithms are good at tracking scenes over small variations. The output translations are quite close to the ground truth values. A loop closure is considered successful if and only if both SSD tracker and GTSAM based PnP solver agree with each other. The dense SSD tracker is guaranteed working only for small distances and is very accurate for those distances (<20cm). The GTSAM estimator although can theoretically track distances greater than that, but due to the sparse nature of optimization and unavoidable noise in depth data, especially with distances greater than 3.5m, it is important to check for agreement of both methods in order to trigger a loop closure. A weighting function may be developed in the future that gives more weight to the SSD tracking result if the distance is very low and more weight to the GTSAM estimate if the distance is more.

A very achievable target for near-future is trajectory generation using the estimates between consecutive keyframes. The keyframe based approach is very fast compared to an approach where every image frame is added to the optimization, although the latter allows for efficient use of IMU preintegration to correct scale and make the estimation observable, concerning which a monocular keyframe based approach will definitely fail as IMU preintegration goes out of bounds if performed for a time >= 0.5 seconds.

Timing

On a Core i7 4500U laptop with 8 gigabytes of RAM, running Ubuntu 14.04, the loop closure algorithm took a time of ~0.3 seconds per detection after the addition of 500 keyframes. Since loop closures are not triggered every time a keyframe is added, the implementation still runs in real time on a single core.