

Investigating and Improving Common Loop Closure Failures in Visual SLAM

Saran Khaliq

National University of Sciences and Technology

Muhammad Latif Anjum (✉ latif.anjum@seecs.edu.pk)

National University of Sciences and Technology

Wajahat Hussain

National University of Sciences and Technology

Muhammad Uzair Khattak

Mohamed bin Zayed University of Artificial Intelligence

Momen Rasool

National University of Sciences and Technology

Research Article

Keywords:

Posted Date: July 15th, 2022

DOI: <https://doi.org/10.21203/rs.3.rs-1822521/v1>

License:  This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Investigating and Improving Common Loop Closure Failures in Visual SLAM

Saran Khaliq · Muhammad Latif Anjum · Wajahat Hussain · Muhammad Uzair Khattak · Momen Rasool

Received: date / Accepted: date

Abstract We analyse, for the first time, the popular loop closing module of a well known and widely used open-source visual SLAM (ORB-SLAM) pipeline. Investigating failures in the loop closure module of visual SLAM is challenging since it consists of multiple building blocks. Our meticulous investigations have revealed few interesting findings. Contrary to reported results, ORB-SLAM frequently misses large fraction of loop closures on public (KITTI, TUM RGB-D) datasets. One common assumption is, in such scenarios, the visual place recognition (vPR) block of the loop closure module is unable to find a suitable match due to extreme conditions (dynamic scene, viewpoint/scale changes). We report that native vPR of ORB-SLAM is not the sole reason for these failures. Although recent deep vPR alternatives achieve impressive matching performance, replacing native vPR with these deep alternatives will only partially improve loop closure performance of visual SLAM.

Our findings suggest that the problem lies with the subsequent relative pose estimation module between the matching pair. Surprisingly, using *off-the-shelf* SIFT based relative pose estimation (non real-time) manages to close most of the loops missed by ORB-SLAM. This significant performance gap between the two available methods suggests that ORB-SLAM's pipeline can be further matured by focussing on the relative pose estimators, to improve loop closure performance, rather than investing more resources on improving vPR. We also evaluate deep alternatives for relative pose

estimation in the context of loop closures. Interestingly, the performance of deep relocalization methods (e.g. MapNet) is worse than classic methods even in loop closures scenarios. This finding further supports the fundamental limitation of deep relocalization methods recently diagnosed.

Finally, we expose the bias in the well known public dataset (KITTI) due to which these commonly occurring failures have alluded the community. We augment the KITTI dataset with detailed loop closing labels. In order to compensate for the bias in the public datasets, we provide a challenging loop closure dataset which contains challenging yet commonly occurring indoor navigation scenarios with loop closures. We hope our findings and the accompanying dataset will help the community in further improving the popular ORB-SLAM's pipeline.

1 Introduction

Figure 1a shows a long (red) segment in the KITTI dataset [8] where ORB-SLAM [20] misses 55 loop closing chances. Even when the loop is closed, there are many subsequent opportunities missed in the revisited area (KITTI-06, Fig 1b). The situation is not different with TUM RGB-D dataset [27] (Fig. 1c-d). The KITTI dataset consists of relatively structured motion, i.e., the car moving in a particular lane, with little to no change in scene or viewpoint and scale. Our results (Fig. 1e-f) indicate that these loop closure failures are even more frequent in unstructured environments, i.e., SLAM performed in large open indoor spaces with small changes in environment. In this work we investigate, for the first time, the reason behind these failures.

ORB-SLAM [20] is an open-source and widely used visual SLAM pipeline. This open-source pipeline has been the start-of-the-art visual SLAM for good part of the last decade. Interest in this pipeline has led to a family of ORB-

Saran K., Anjum M.L., Wajahat H., Rasool M.
Robotics & Machine Intelligence (ROMI) Lab, School of Electrical Engineering and Computer Science (SEecs), National University of Sciences and Technology, Islamabad. E-mail: {skhaliq,msee16seecs,latif.anjum,wajahat.hussain, mrasool.msee20seecs}@seecs.edu.pk

Khattak M.U.
Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE. E-mail: uzair.khattak@mbzuai.ac.ae

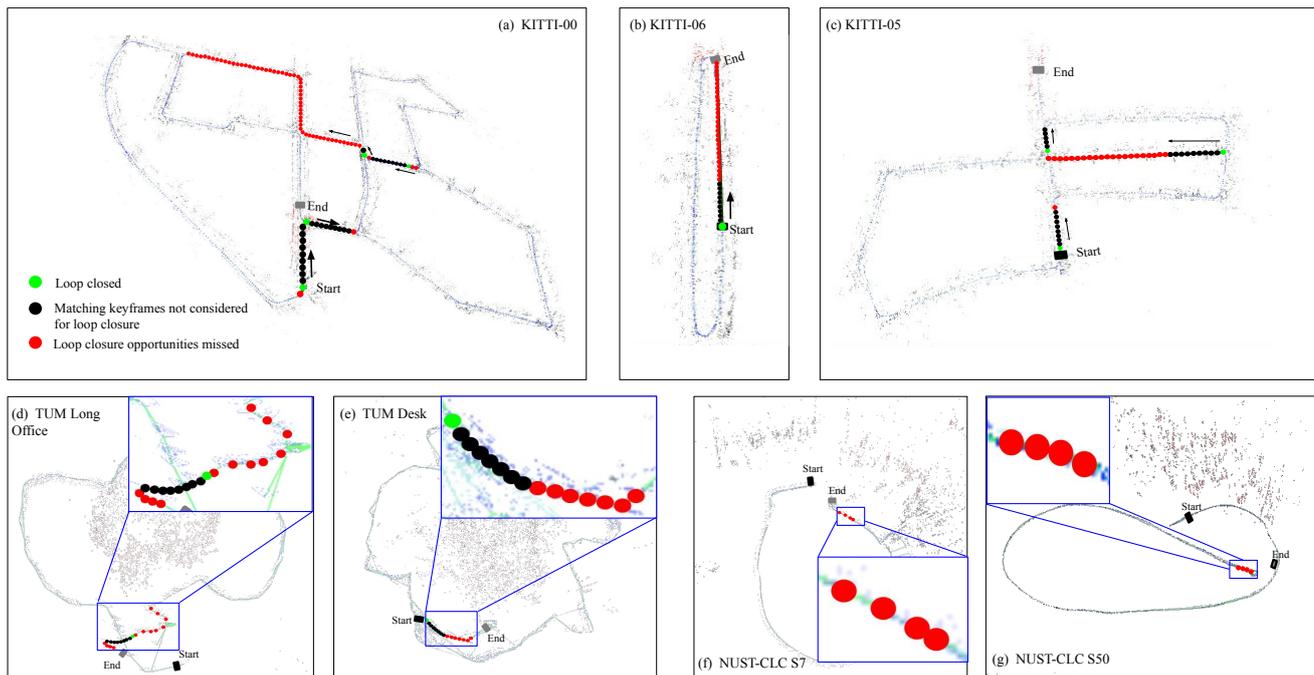


Fig. 1: Is ORB-SLAM closing all loops in KITTI? In KITTI-00, a large revisited sequence (red dots) containing 55 matching keyframes is completely missed. After closing the loop, ORB-SLAM does not consider the next nine keyframes (black dots) for loop closure. There are many missed opportunities (KITTI-05, and KITTI-06) even after these keyframes have passed. The situation remains same in TUM RGB-D. In NUST-CLC, loop closing opportunities are relatively easy but limited (fewer red dots) in each sequence resulting in large fraction of missed loop closures.

SLAM systems (ORB-SLAM [20], ORB-SLAM2 [19], ORB-SLAM3 [4]). This family of SLAM systems consists of many building blocks/threads and has been mainly evaluated at holistic level. The contribution of its individual building blocks towards its success has seen little attention. Recently, Kanwal et al. [22] have demonstrated the limitations of its tracking module in unsupervised settings. They have proposed a deep reinforcement learning approach to reduce its frequent tracking failures. Similarly, Haris et al. [12] have shown the vulnerability of its back-end optimization in case of perceptual aliasing.

In this work, we evaluate, for the first time, its loop closing module which has gained popularity as a separate entity and has been adopted by multiple SLAM solutions. The popularity of this loop closing module, from ORB-SLAM, can be judged from the fact that it has been adopted, out of the box, in many recent SLAM systems including monocular [23], stereo [10], direct [7], multi-sensor [24, 9] and reinforcement learning based [1] SLAM systems. Our analysis reveals the performance gap between this module and the other alternatives already available to the community, which suggests that further attention to this loop closing module can further improve the already popular open-source SLAM pipeline.

Why have these frequent loop closing failures managed to avoid attention? Interestingly, authors of ORB-SLAM report closing all the loops in KITTI dataset. In our opinion, KITTI dataset, although challenging in other aspects, provides long revisited areas (whole street revisited, Fig 1a-b). This provides ample opportunities for loop closure routines to work. Closing one loop in the whole street (containing many opportunities) accounts for the major drift correction. Therefore, the attention is not diverted towards the large number of missed loop closure opportunities which offer diminishing returns.

In order to quantify number of missed loop closure opportunities, we augment the KITTI dataset [8] with detailed loop closing labels. These labels were generated by matching ≈ 27 million image pairs. These loop closing labels are provided as scene graphs. These scene graphs clearly indicate that each sequence contains large number of loop closing opportunities which are easily identified by image registration methods. What will happen when the revisited scene contains limited loop closing opportunities? To fill this gap we provide NUST-CLC dataset which contains 100 navigation sequences, each having loop closures where the revisited scene is limited to 5-10 keyframes (Fig. 1e-f). Furthermore, our dataset contains frequently occurring scene changes (dynamic / viewpoint / scale). Results indicate loop

closure is frequently missed (in 80 sequences out of 100) in such situations.

Investigating the reason behind these surprising failures is not trivial. Loop closing module of ORB-SLAM consists of multiple blocks (visual place recognition (vPR), relative pose estimation). After vPR provides a suitable matching candidate, the subsequent relative pose estimation module provides a constraint for the pose graph for map correction. This relative pose estimation module is based on time tested, handcrafted, robust optimization techniques.

Experiments with state-of-the-art loop closure embedded SLAM system [20] show that the loop closure fails if object/s are displaced/removed from the originally visited scene. The displaced object/s do not need to occupy the whole scene; these could be small objects (a mug, bag, book, etc) occupying a small portion of the scene. What fails loop closure in such situations? A general perception is that visual place recognition fails since the scene has been modified. Our experiments suggest even when vPR suggests a valid loop closing candidate, many times the loop is not closed due to subsequent relative pose estimation module.

Dynamic scenes are not the only challenge in loop closure. State-of-the-art loop closure enabled SLAM system [20] provides limited viewpoint and scale invariance as indicated by our experiments. In situations where a robot passes a few inches/degrees (16.3 inches, 5°) away from a previously visited place, despite finding the correct loop closing candidate, the loop is not closed resulting in missing a good drift reduction opportunity. Our experiments suggest visual place recognition does not fail even in these situations. Subsequent relative pose estimation module is the major reason behind missed loop closure opportunities.

There has been a plethora of work on improving loop closure for visual SLAM in the past two decades. The focus of the majority of these works has been on improving the accuracy and robustness of visual place recognition (vPR) for improving loop closure. Consequently, amazing vPR systems have been developed which can work in extreme conditions (day-night [17, 2], virtual-real world [29], across seasons [21, 14] and dynamic environments [16]). The other part of loop closure routine, the relative pose estimation module, has received less attention. Our investigation suggests, this module is responsible for failing most of the loop closures.

We evaluate state-of-the-art real-time deep vPR [16] against ORB-SLAM's vPR. Interestingly, ORB-SLAM's vPR has little precision. It provides candidates for almost every keyframe, most of them being false positives. Deep vPR, on the other hand, has excellent precision (little to no false positives) and high recall (very few missed matches). Surprisingly, a vPR module with a high recall only partially improves the loop closure because the subsequent relative pose estimation module fails the loop closing pipeline. Our analysis might

help the community dedicate attention to the relative pose estimation block of the loop closing module similar to the attention received by vPR block.

Interestingly, an *off-the-shelf* SIFT based image registration method (VSFM) manages to accurately estimate relative pose for the majority of the scenarios missed by ORB-SLAM. This significant performance gap between the two available methods suggests that ORB-SLAM's pipeline can be further matured by focussing on the relative pose estimators. Perhaps the community may wish to explore using multiple features, tailored for individual modules of SLAM (tracking, mapping, loop closure), instead of the current *one-feature-for-all* strategy. Managing multiple features will affect the computation efficiency and memory requirements. However, recent learning based feature-to-feature transformers [5] might lead to more compact solutions. Additionally, even though tracking in SLAM imposes a hard real-time constraint, loop closure and pose graph optimization do not impose hard real-time constraints [15], paving the way for the use of non real-time SIFT for these modules.

How good are deep pose regressors such as SCoRe [26], PoseNet [13], and MapNet [3] as an alternative to relative pose estimation? We evaluate the use of MapNet [3] for relative pose estimation in visual SLAM. However, apart from scene-specific training requirement of most of them being a bottleneck, we come across the fundamental limitation of these deep pose regressors reported in [25]: these deep methods fare no better than vPR methods. In other words, the pose of the current keyframe is similar to the pose of the best matching image in the training database. Our novel finding suggests that this fundamental weakness of deep relocalizers is much more severe than previously reported.

Contributions of our work:

- We provide in-depth analysis of ORB-SLAM's loop closure routine, outlining its failure cases, finding out what fails the loop closure and testing alternatives for vPR and relative pose estimation. To the best of our knowledge, this performance gap analysis is not available to the community.
- We release a challenging loop closure dataset, NUST-CLC. The dataset can serve as a benchmark for evaluating loop closures in challenging yet common navigation scenarios. We additionally augment KITTI sequences with loop closure ground truth.

2 Dissecting Loop Closure in ORB-SLAM

ORB-SLAM is current state-of-the-art in loop closure enabled visual SLAM and is widely known and used in research community¹. The loop closure routine of ORB-SLAM

¹ ORB-SLAM module is included in MATLAB R2020a release making it the first visual SLAM system to appear in MATLAB.

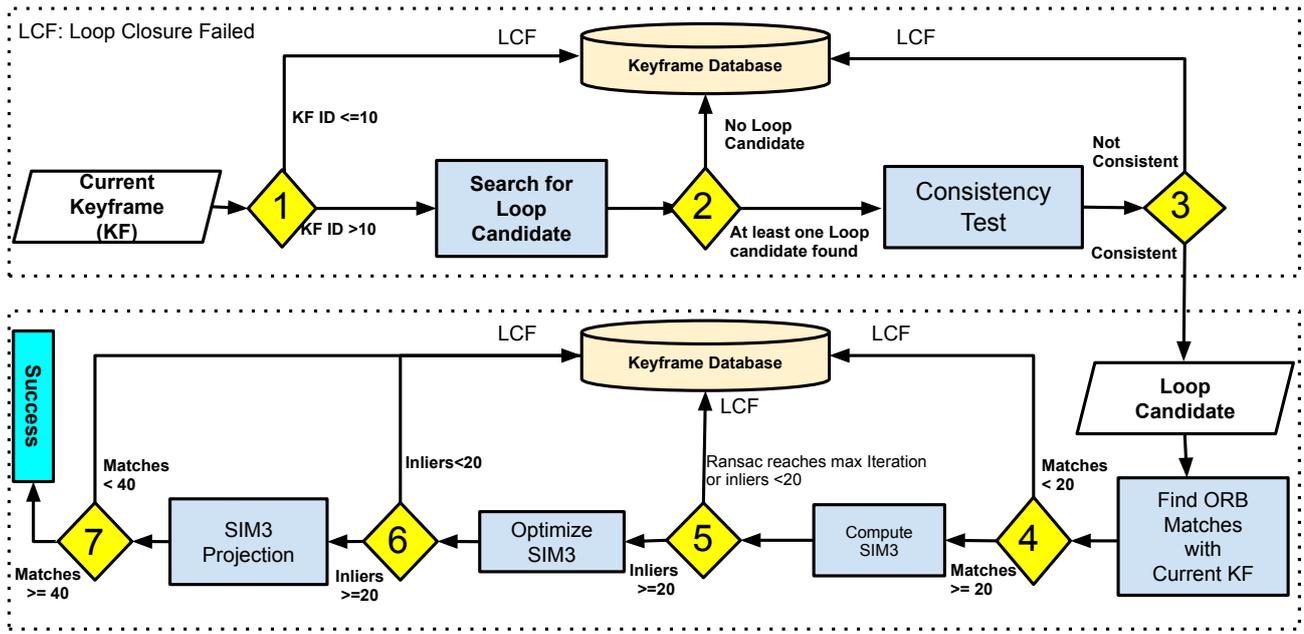


Fig. 2: Detailed flow-chart of loop closure routine in ORB-SLAM

is described in [18]. It uses DBoW2 [6] for place recognition followed by many temporal and geometrical consistency checks. These individual checks are hidden behind programming intricacies making it difficult to analyze failure cases. This section provides a detailed description of the checks on loop closure used in ORB-SLAM. Later in Section 4, we provide loop closure failure cases of ORB-SLAM and identify the checks that fail in those cases. This understanding is essential for developing opportunistic loop closure systems for visual SLAM.

Loop closure in ORB-SLAM contains seven checks which have to be passed before the loop can be finally closed. The complete flow diagram is given in Figure 2. Individual checks are explained below.

1: Initialization: The first check is to avoid loop closure at the same place without significant motion. The search for the loop candidate starts after 9 keyframes have passed. Every keyframe before that is being added to the central keyframe database.

2: Search for Loop Candidate: To search for a viable loop candidate, bag-of-words based matching score is calculated between the current keyframe and its three connected frames in the co-visibility graph. The minimum of those scores is selected as a reference score. With this reference score, the algorithm queries the central keyframe database, where bag-of-words based matching score is calculated between the current keyframe and every frame in the keyframe database. Keyframe having a matching score greater than the reference score is considered a loop candidate. If a loop

candidate is not found, the current keyframe is added to the central keyframe database aborting the loop closure routine.

3: Temporal Consistency Test: If check 2 is passed, we have the current keyframe and its loop candidate frame. The loop candidate is validated through temporal consistency check. Temporal consistency is established if three previous keyframes (of the current keyframe) have also passed check 2 and have valid loop candidates. Furthermore, co-visibility graph (keyframes having common map features) of loop candidates for these four keyframes (current and three previous) should have at least one common keyframe. If consistency test fails, the current keyframe is added to the central keyframe database aborting the loop closure routine.

4: Finding ORB Matches: After validating the loop candidate through a temporal consistency test, the next step is to find ORB matches between the current keyframe and its loop candidate. If at least 20 ORB matches are not found, the current keyframe is added to the central keyframe database aborting the loop closure routine.

5: Geometrical Consistency (Compute $SIM(3)$): At this step, RANSAC iterations are performed with matched features to find similarity transformation using method described in [11] followed by a guided search for more correspondences. This similarity transformation is essential to estimate the error accumulated in the trajectory. The similarity transformation should be found with enough inliers (at least 20), failing which the current keyframe is added to the central keyframe database aborting the loop closure routine.

6: Optimize $SIM(3)$: With the matches (3D correspondences) between current keyframe and loop candidate avail-

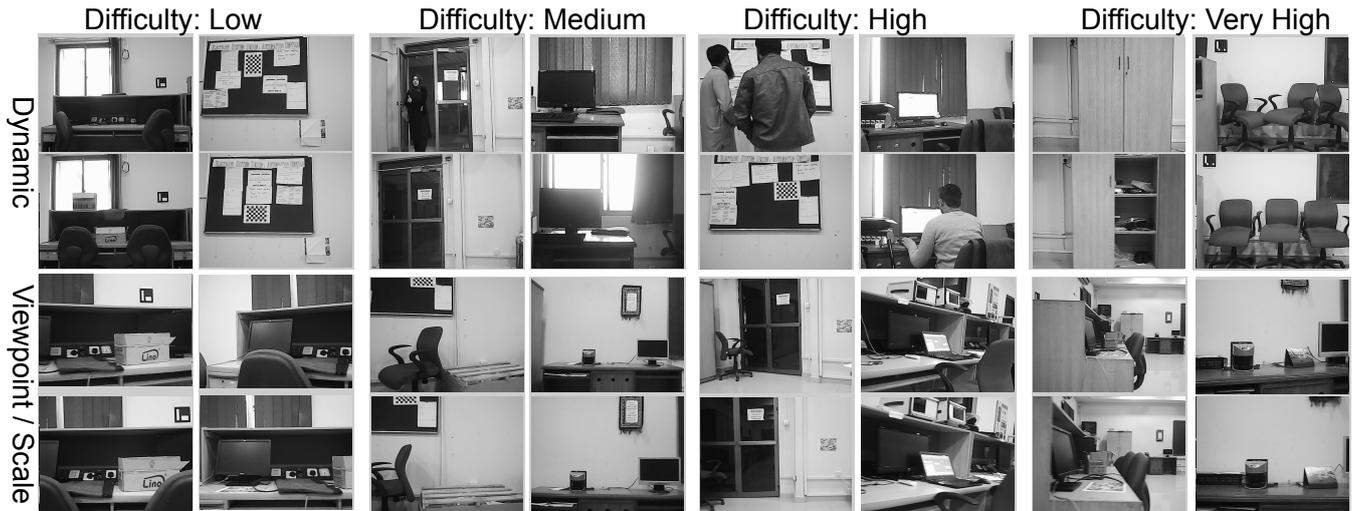


Fig. 3: Sample original (top) and loop closing candidate (bottom) images in NUST-CLC dataset. Most of the dynamic scenes contain frequently occurring daily life situations e.g. an additional notice placed on the board, door open/close, human presence etc.

able, $SIM(3)$ is optimized to minimize reprojection error in both frames. ORB-SLAM uses $g2o$ optimization tool for the purpose. This optimization should result in at least 20 inliers, failing which the current keyframe is added to the central keyframe database aborting the loop closure routine.

7: Map-point Projection using $SIM(3)$: Finally map-points of the loop candidate and its connected keyframes in co-visibility graph are projected onto the current keyframe using $SIM(3)$. The number of matches after the projection should be greater than 40, failing which the current keyframe is added to the central keyframe database aborting the loop closure routine.

3 Challenging Loop Closure Dataset

Exploiting the scarcity of challenging loop closure datasets, we release a new loop closure dataset, called NUST Challenging Loop Closures (NUST-CLC), containing loop closures that are not only challenging but also occur frequently in indoor navigation². The dataset contains 100 indoor navigation episodes (of length ≈ 319 mins, ≈ 0.27 million images), released as videos as well as ROS bags for ready usage. Each navigation episode contains one challenging loop closure opportunity. The dataset has been divided into two categories:

- **Dynamic Scenes:** Dynamic scenes include situations where a robot revisits a place with exact same viewpoint but the scene has changed slightly (e.g. few objects moved,

removed or added or replaced within the scene). Our experiments show loop closure fails in such situations if the disturbed part of the scene is feature rich compared to the remainder of the scene.

- **Viewpoint/Scale Changes:** Revisiting a previously visited place with the exact same pose is a very rigid and practically difficult requirement. In practice, a navigating robot can only see a previously visited place from a different viewpoint and location. Current state-of-the-art ORB-SLAM’s loop closing pipeline provide limited viewpoint and scale invariance. Consequently, a navigating robot with a large drift, passes around the previously visited place, capturing the same scene from a slightly different viewpoint or position, without closing the loop, and without even knowing there is a loop closure opportunity in the vicinity.

We have kept the revisited area limited to 5-10 keyframes in each sequence. This feature along with carefully incorporated dynamic/viewpoint/scale changes make our dataset useful in evaluating robust loop closing pipelines. There are 50 navigation episodes of each category. Each episode has been flagged as low, medium, high, or very high depending upon the difficulty of loop closure. The difficulty is assessed by running ORB-SLAM and finding if the loop is closed. Low difficulty flag is associated with those episodes where ORB-SLAM closes the loop at standard thresholds. We successively lower thresholds (see Section 4.4), and assess the difficulty of loop closure. Episodes where loop closure fails even at lowest thresholds are flagged with very high difficulty. There are in total 20 low, 23 medium, 23 high, and 34 very high difficulty loop closures in our dataset. Some sam-

² <https://github.com/sarankhaliq2326/NUST-CLC>

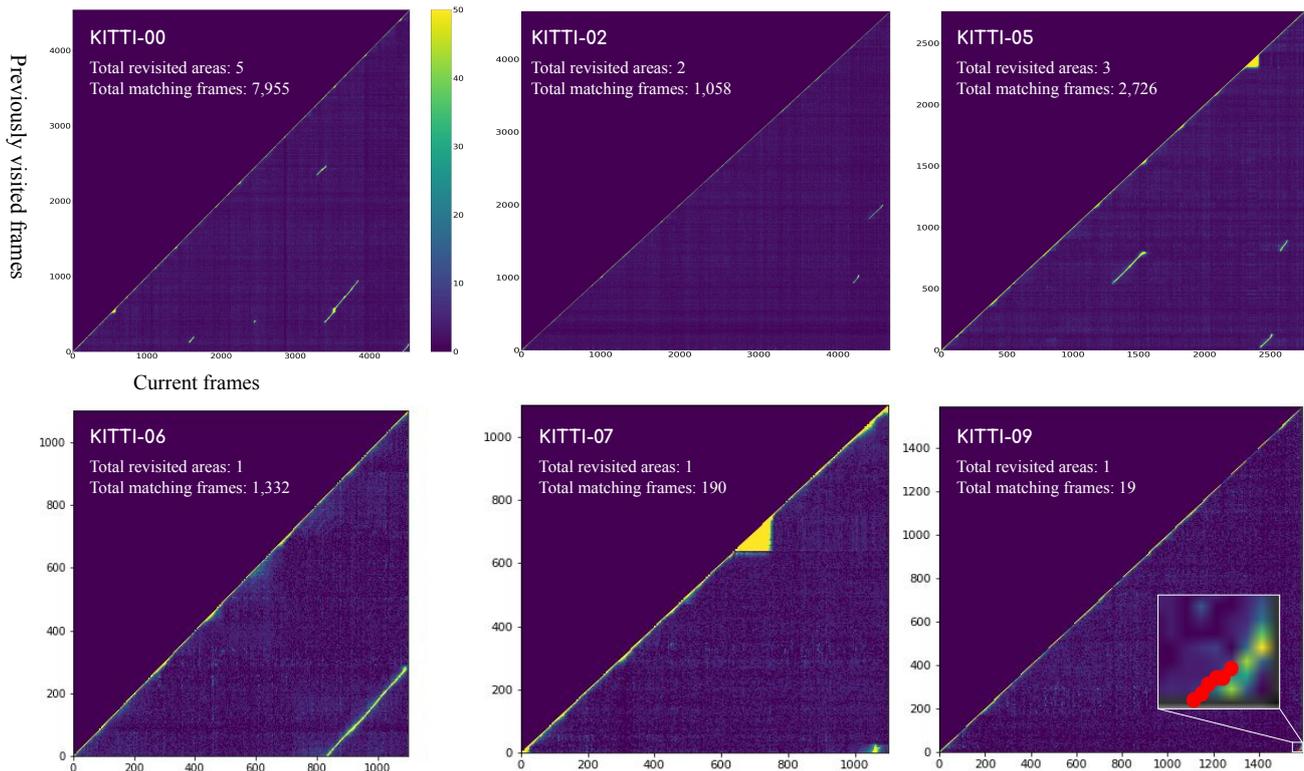


Fig. 4: Scene graphs of six KITTI sequences that include loop closure. Part of the figure below secondary main diagonal shows number of SIFT matches surviving RANSAC between current frame and previously visited frames sampled @ 10 fps. We have counted total matching frames @ threshold = 50. False negatives are manually shown in red color (KITTI-09), and enlarged for better viewing. Figure is better viewed with a digital zoom.

ple loop closure scenes in our dataset are shown in Figure 3 for each flagged category. We release loop closure ground truth along with the dataset.

4 Experimental Results

Experiments have been conducted on our dataset, NUST-CLC, as well as on public datasets and are presented accordingly.

4.1 Augmenting KITTI for Loop Closures

Frame level inspection of KITTI sequences (Figure 4) for loop closures reveals very interesting observations. There are revisited areas in six KITTI sequences, some of them containing multiple ones (yellow(ish) diagonals in the lower triangle of scene graphs in Figure 4). Most of the revisited areas are long enough (indicated by lengths of non-main diagonals in Figure 4) to provide many loop closure opportunities. The performance is usually reported by closing one loop (with any one of the multiple available opportunities)

in a revisited area. We augment KITTI with loop closure ground truth by releasing scene graphs (Figure 4) for six KITTI sequences which contain loop closures.

Each sequence is sampled at 10 fps as most of the SLAM systems are evaluated at this or lower frame rates. Each frame is compared with all previously visited frames, and SIFT feature matches surviving RANSAC are counted and visually shown with a color scheme (Figure 4). Each yellow(ish) color in the lower half indicates a revisited area. The yellow(ish) color in the secondary main diagonal (upper right to lower left) indicates each frame matching with its nearby frames and thus not considered as valid loop closures.

We have counted total matching frames in each sequence with a threshold of 50 SIFT matches surviving RANSAC. Besides these graphical representations, we release csv files for all sequences containing exact SIFT matches surviving RANSAC for all frames. Users can select threshold of their choice and count loop closures from the csv files. Challenging cases of loop closures would be the cases with matches close to or lower than 50. SIFT features have been used to produce the ground truth because of their superior performance and higher robustness to viewpoint and scale changes.

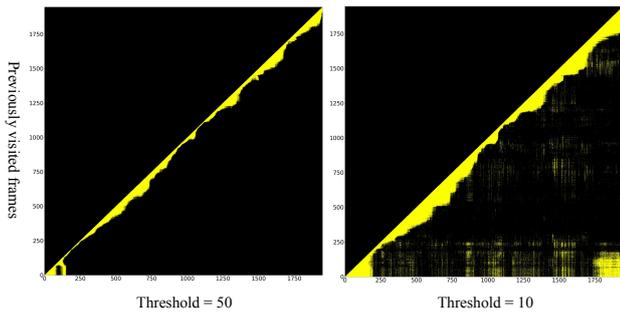


Fig. 5: Scene graphs for NUST-CLC S4 sequence. Since the loop closure is challenging, scene graph does not show significant matches at threshold = 50. We lowered the threshold down to 10, and plotted binary color graph (yellow for above 10, black for all others), showing revisited area (lower right corner, threshold = 10). Please note that, lowering threshold to 10 shows many irrelevant matching pairs (yellow dots spread all over).

The generation of these scene graphs required significant computing effort as ≈ 27 million image pairs required computation of SIFT features and RANSAC survival. On our intel(R) core(TM) i7-4720HQ processor with 2.60 GHz clock frequency, it took ≈ 374 hours.

We manually inspected each frame ($\approx 17,000$ in total) to remove any false negatives (loop closures missed by SIFT matching). There is an initial part of the revisited scene in KITTI-09 which should have been labeled as loop closure but its SIFT matches are lower than threshold (of 50), as indicated with red dots (Figure 4, KITTI-09, enlarged inset). Similarly, there are negligible false positives (12 in all sequences) after manual inspection. These missed opportunities a.k.a false negatives (FNs) and incorrect loop closures a.k.a false positives (FPs) have been flagged in associated csv files.

Interestingly, these scene graphs reveal that simple image registration method manages to find large number of correct matches (loop closures) in KITTI dataset. However, as shown later ORB-SLAM missed large number of these opportunities indicating a large performance gap.

Similar scene graph for one sequence of NUST-CLC dataset is shown in Figure 5. Since NUST-CLC is purposely generated loop closure dataset, simple image registration methods are unable to detect loop closures (Figure 5), as there are no significant geometrically correct SIFT matches in revisited area. The situation remains similar in other sequences (not shown) of NUST-CLC. Therefore, in our opinion, generating scene graphs for our dataset does not provide adequate loop closure ground truth. Manual labeling is, therefore, the most suitable method for the purpose. Figure 6 shows a visual representation of loop closure ground truth

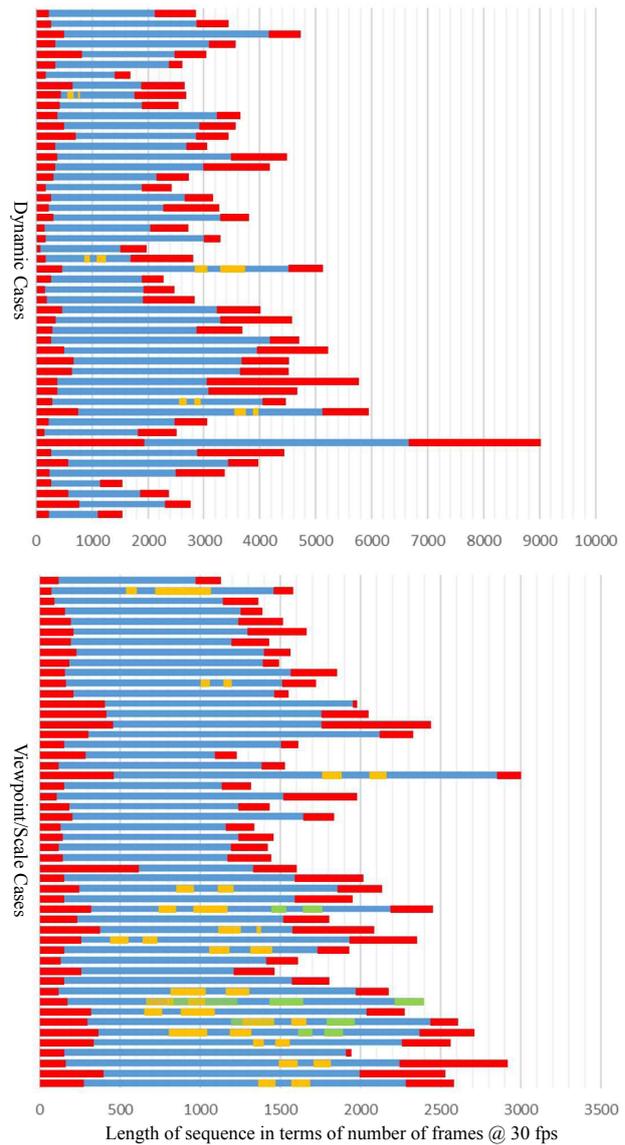


Fig. 6: Loop closure ground truth for NUST-CLC dataset. The ground truth is generated manually. The first and second visit are indicated by same colors. Each sequence contains at least one revisited region. There are large number of revisited frames because of high frame rate (30 fps) we are using to generate this ground truth.

for our NUST-CLC dataset generated manually. Each horizontal bar represents the length of each sequence in our dataset in terms of number of frames sampled at 30 fps. The first bar indicates the first visit followed by the second bar indicating second visit. Besides this graphical representation, we release the data in csv files for users.

Table 1: Performance of ORB-SLAM’s loop closure on public datasets and NUST-CLC. Good performance on public datasets is due to many available opportunities in each revisited area.

Dataset	No. of revisited areas	No. of loop closed	Total matching KFs in all revisited areas
KITTI-00	5	4	91
KITTI-02	2	2	42
KITTI-05	3	3	49
KITTI-06	1	1	44
KITTI-07	1	1	4
KITTI-09	1	1	6
TUM RGB-D (3 sequences)	3	3	11, 14, 3
NUST-CLC (100 sequences)	100	20	5-15 KFs in each sequence

4.2 Evaluation on Public and Private Datasets

Table 1 shows performance of ORB-SLAM’s loop closure routine on KITTI, TUM RGB-D and our NUST-CLC dataset. Interestingly, the popular loop closing block misses a large number of loop closing chances (last column). Note that KITTI and TUM RGB-D sequences contain long revisited trajectories resulting in abundant matching keyframes in each revisited street (indicated by the length of non-main diagonals in Figure 4), and therefore the loop is eventually closed.

We encourage the reader to view our supplementary video. Our NUST-CLC sequences contain short revisited areas, which results in 80% (37/50 in dynamic, and 43/50 in viewpoint/scale sequences) of the loop closures being missed. This is intriguing since our revisits consist of little variance (viewpoint/scale/dynamics).

4.3 What Fails the Loop Closure?

Intrigued by this large-scale failure on both public and private datasets, we dissected loop closure routine at modular level to understand which particular component of the routine is failing the loop closure. Results are tabulated in Table 2. Since this is failure analysis, only those sequences of NUST-CLC dataset are analyzed where loop closure failed. Interestingly, search for loop candidate and temporal consistency test was successful for all sequences, indicating vPR is providing a potential match. Apparently the loop closure failure is triggered by subsequent geometrical registration (relative pose estimation). This trend is the same in both dynamic as well as viewpoint/scale changes sequences. This raises interesting questions. Is the relative pose estimation module rigid? Did the vPR module provide a valid candidate?

Table 2: Why loop closure fails in dynamic scenes and viewpoint/scale changes? A large percentage of failure is triggered by failure to geometrically register the two images in both dynamic scenes as well as under viewpoint/scale changes.

Loop Closure Checks	No. (and %) of times the check fails	
	Dynamic Scenes	Viewpoint/Scale
Initialization	0/37 (0%)	0/43 (0%)
Search for Loop Candidate	0/37 (0%)	0/43 (0%)
Temporal Consistency Test	0/37 (0%)	0/43 (0%)
Finding ORB Matches	18/37 (48.64%)	2/43 (4.66%)
Geomet. Consistency Test	16/37 (43.24%)	27/43 (62.79%)
Optimize $SIM(3)$	3/37 (8.12%)	14/43 (32.55%)
Map-point Projection	0/37 (0%)	0/43 (0%)

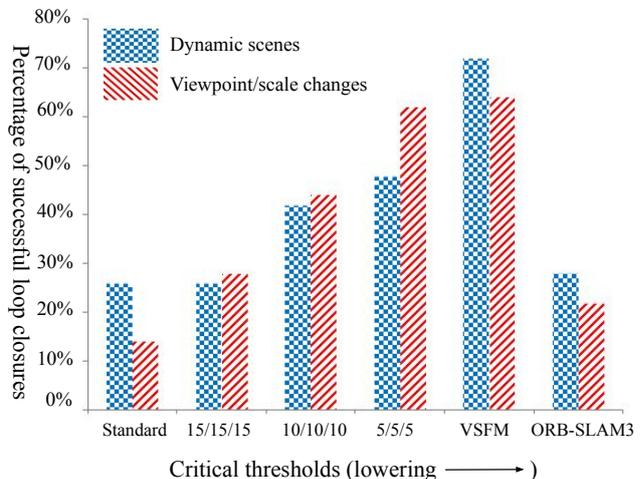


Fig. 7: Performance of ORB-SLAM’s loop closure on our dataset at standard thresholds and after lowering thresholds. We indicate our thresholds as a/b/c, where no. of ORB matches required are (a), no. of inliers during geometrical consistency test are (b), and no. of inliers during $SIM(3)$ optimization are (c). Improvement in performance with lowering thresholds is evident. Using VSFM for relative pose estimation achieves performance comparable to the lowest thresholds. ORB-SLAM3 [4], at standard thresholds, does not outperform original ORB-SLAM despite improved place recognition.

4.4 Relaxing Thresholds in ORB-SLAM

In order to investigate the rigidity of the relative pose estimation block, we reduced the thresholds involved in geometric registration. Table 2 shows that the most common checks that cause loop closure failure in challenging situations are: a) inability to find ORB matches (at least 20 are required), b) geometrical consistency test failure ($SIM(3)$)

Table 3: Detailed analysis of what fails ORB-SLAM’s loop closure in challenging situations at various thresholds. It is interesting to note that the three checks (bold faced) identified in Table 2 continue to be major causes of failure despite lowering their required thresholds. Data at standard thresholds is repeated here for ready comparison.

Loop Closure Checks	Dynamic Scenes. No. (and %) of times the check fails			
	Threshold Standard (20/20/20)	Threshold Scheme: 15/15/15	Threshold Scheme: 10/10/10	Threshold Scheme: 5/5/5
Initialization	0/37 (0%)	0/37 (0%)	0/29 (0%)	0/26 (0%)
Search for Loop Candidate	0/37 (0%)	0/37 (0%)	0/29 (0%)	0/26 (0%)
Temporal Consistency Test	0/37 (0%)	0/37 (0%)	0/29 (0%)	0/26 (0%)
Finding ORB Matches	18/37 (48.64%)	9/37 (24.32%)	10/29 (34.48%)	4/26 (15.38%)
Geometrical Consistency Test	16/37 (43.24%)	17/37 (45.95%)	16/29 (55.17%)	19/26 (73.08%)
Optimize $SIM(3)$	3/37 (8.12%)	4/37 (10.81%)	3/29 (14.34%)	3/26 (11.54%)
Map-point Projection	0/37 (0%)	7/37 (18.92%)	0/29 (0%)	0/26 (0%)

Loop Closure Checks	Viewpoint/Scale Changes. No. (and %) of times the check fails			
	Threshold Standard (20/20/20)	Threshold Scheme: 15/15/15	Threshold Scheme: 10/10/10	Threshold Scheme: 5/5/5
Initialization	0/43 (0%)	0/36 (0%)	0/28 (0%)	0/19 (0%)
Search for Loop Candidate	0/43 (0%)	0/36 (0%)	0/28 (0%)	0/19 (0%)
Temporal Consistency Test	0/43 (0%)	0/36 (0%)	0/28 (0%)	0/19 (0%)
Finding ORB Matches	2/43 (4.66%)	1/36 (2.78%)	1/28 (3.57%)	0/19 (0%)
Geometrical Consistency Test	27/43 (62.79%)	16/36 (44.44%)	13/28 (46.42%)	10/19 (52.63%)
Optimize $SIM(3)$	14/43 (32.55%)	15/36 (41.62%)	14/28 (50%)	9/19 (47.37%)
Map-point Projection	0/43 (0%)	4/36 (11.11%)	0/28 (0%)	0/19 (0%)

should be computed with at least 20 inliers), and c) $SIM(3)$ optimization failure (again 20 inliers are required here). Our experiments suggest lowering of these thresholds improves the loop closure success rate. Figure 7 shows results after lowering these three thresholds (a/b/c above) to 15/15/15, 10/10/10, and 5/5/5 in the same order.

Detailed analysis of the cause of each loop closure failure at various thresholds is provided in Table 3. It is interesting to observe that despite lowering of thresholds, the three checks identified at standard threshold continue to be the major reasons of failure. The improvement in loop closure success rate should be interpreted carefully as lowering the thresholds increases the chance of false loop closure and adversarial attacks [12]. There is, however, no false loop closure observed with these lowered thresholds in our dataset.

4.5 Reason Behind Strict Thresholds

We manually inspected ORB-SLAM’s loop closure (*for more than 10,000 keyframes*) to establish how frequently a valid candidate was provided by the native vPR. Results (Table 4) show that the native vPR has very low precision (Figure 8), providing a false candidate for a large number of keyframes (high false positives). In our opinion, this low precision of vPR places an additional burden of rejecting false candidates on the subsequent geometric registration block. As a result of this extra vigilance (strict thresholds), good candidates are often rejected by subsequent geometric registration.

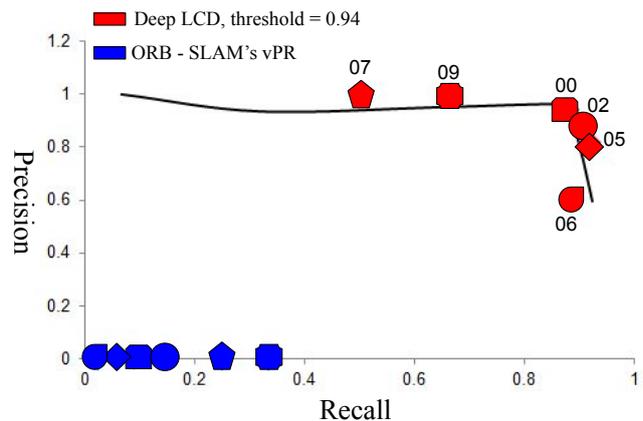


Fig. 8: Precision-recall comparison for Deep LCD and ORB-SLAM’s vPR on KITTI sequences mentioned on data points. Black curve represents Deep LCD PR curve at various thresholds for KITTI-00. Threshold 0.94 provides best precision and recall and is, therefore, used for other sequences while comparing with ORB-SLAM’s vPR. High number of false positives in ORB-SLAM’s vPR results in close to zero precision.

4.6 How Much a Better vPR Improves Loop Closures?

For comparison, we evaluate the performance of a recent real-time deep vPR method (Deep-LCD [16]). We used statistical analysis on KITTI-00 to find a suitable matching

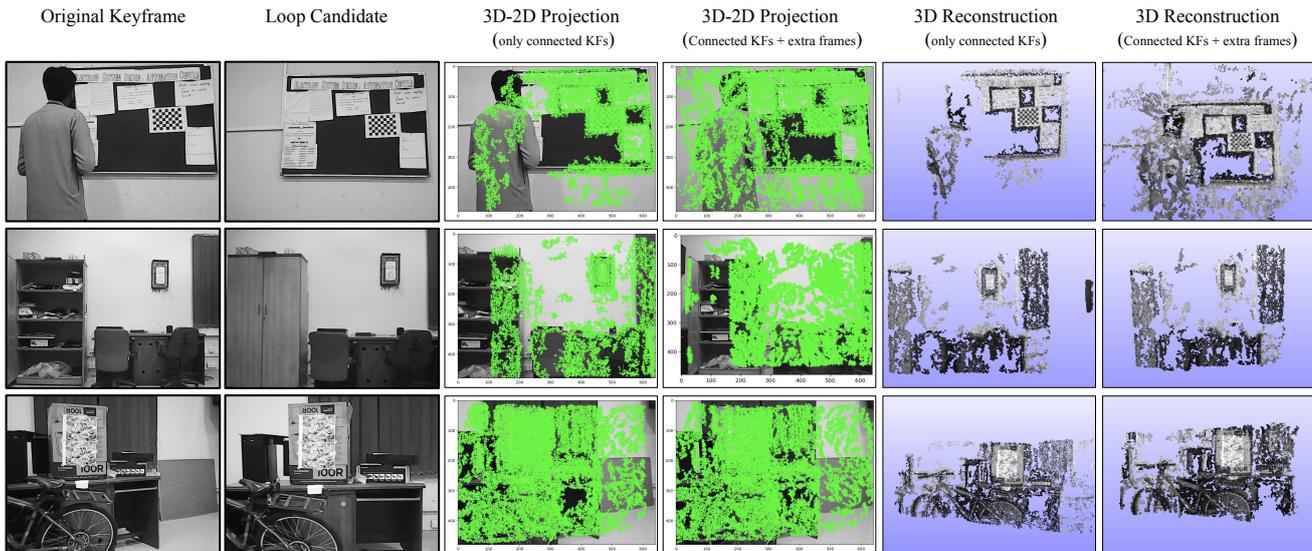


Fig. 9: How good is the pose estimated by VSFM? Three cases from NUST-CLC where ORB-SLAM fails to close the loop despite valid loop candidate due to failure of relative pose estimation. VSFM [28] not only manages to register the keyframes, but reconstructs semi-dense map (second last column) using only connected keyframes and provides accurate relative pose as shown by well aligned 3D-2D projection (third column). Visual comparison of 3D reconstructions (last two columns) shows little improvement by adding intermediate frames indicating accurate pose estimation using only connected keyframes.

Table 4: How often a correct candidate is found by vPR of loop closure in ORB-SLAM in comparison with deep approach [16]?

Sequence	True Positives	False Positives	False Negatives
	ORB-SLAM:Deep	ORB-SLAM:Deep	ORB-SLAM:Deep
KITTI-00	8 : 80	1685 : 1	83 : 11
KITTI-02	6 : 38	999 : 29	36 : 4
KITTI-05	3 : 45	1005 : 0	46 : 4
KITTI-06	1 : 39	372 : 10	43 : 5
KITTI-07	1 : 2	444 : 1	3 : 2
KITTI-09	2 : 4	815 : 1	4 : 2
Average	3.50 : 36.66	886.67 : 7.00	35.83 : 4.66

threshold for Deep LCD (black curve in Fig. 8), and generated precision-recall of all sequences of KITTI (Figure 8) for ready comparison with ORB-SLAM. Deep vPR has impressive precision (fewer false positives) and recall (fewer missed chances) as compared to native vPR of ORB-SLAM.

Replacing native vPR of state-of-the-art SLAM with deep vPR might partially improve the loop closing ability, due to fewer false positives, allowing relaxed thresholds for geometric checks. However, even at very low thresholds (5/5/5), **ORB-SLAM missed 50% of the chances**, indicating improving vPR might not substantially improve loop closure. We have tested recently released ORB-SLAM3 [4], which uses an improved place recognition for loop closure, on NUST-CLC. There is no significant improvement over original ORB-

SLAM (Figure 7) despite improved place recognition, which strengthens our view that improving place recognition only partially improves loop closure.

4.7 How to Improve Relative Pose Estimation?

Lowering the registration thresholds results in substantial correct loop closures. In this section, we evaluate SIFT based registration for relative pose estimation. To this end, we selected a SIFT based *SfM* pipeline [28], and tried to register current keyframe, for which vPR provided a valid loop closing candidate but ORB-SLAM was unable to estimate $SIM(3)$.

Instead of registering the corresponding keyframes directly, we extracted their connected keyframes (keyframes having common features in ORB-SLAM map) for both counterparts. Interestingly, *off-the-shelf* VSFM [28], without any parameter tuning, managed to build a sub-map for the majority of the cases missed by ORB-SLAM at standard thresholds (Figure 7). Qualitative results (third and fifth column in Figure 9) indicate that the connected keyframes contain enough feature content to build a semi-dense map (third column in Figure 9), and estimate valid relative pose as shown by well aligned 3D-2D projection (third column in Figure 9), whereas ORB-SLAM loop closure routine could only match negligible features.

In order to judge the quality of relative pose, we added a few intermediate frames in addition to connected keyframes of loop closing counterparts. Visual comparison of 3D reconstructions (last two columns in Figure 9) shows little improvement, authenticating the accuracy of relative pose estimate using only the connected keyframe sub-map.

These findings have exposed a performance gap between the relative pose estimation module of ORB-SLAM and the currently available (and implemented) relative pose estimation to the community. Therefore, in order to further refine the state-of-the-art visual SLAM, more attention may be diverted towards the current open source ORB-SLAM pipeline. Additionally, ORB-SLAM’s *one-feature-for-all* strategy has its own advantages with respect to system complexity, memory management, and computation. However, with computational power increasing steadily, we argue using custom features for different modules of SLAM may improve its performance.

4.8 How Good are Deep Alternatives for Relative Pose Estimation?

In this section, we evaluate the relative pose estimation by deep pose regressor [3]. Deep learning based methods have shown impressive invariance (viewpoints, light conditions). However, recently deep pose regressors have been diagnosed with a fundamental limitation [25] i.e. for a given frame, the pose provided by these deep methods is similar to the pose of the best matching frame in the training database. We evaluate a well known deep pose regressor (MapNet [3]) in an even simpler scenario of loop closures where the training set contains an image similar to the query image since the place has been revisited.

Details of training and inference of MapNet are given in Algorithm 1. For every keyframe \mathbf{K}^i , we train a pose regressor \mathbf{M}^i using the MapNet architecture. Input to this deep pose regressor is a 2D image and its output is 6 DoF pose. The training data for \mathbf{M}^i consists of the keyframe \mathbf{K}^i (2D image) and its accompanying 6 DoF absolute pose \mathbf{P}^i . Training MapNet using this single image will lead to over-fitting. Therefore, in order to increase the robustness, we extract the connected keyframes of \mathbf{K}^i , i.e., $\mathbf{K}^\alpha, \mathbf{K}^\beta, \dots, \mathbf{K}^\zeta$, and their accompanying poses ($\mathbf{P}^\alpha, \mathbf{P}^\beta, \dots, \mathbf{P}^\zeta$) from the ORB-SLAM map.

Assuming ORB-SLAM’s vPR proposes keyframe \mathbf{K}^i as the potential loop closing candidate for the now current keyframe \mathbf{K}^j . Our aim is to estimate the relative pose between \mathbf{K}^j and \mathbf{K}^i . We assume that the ORB-SLAM’s map has drifted therefore, \mathbf{P}^i and \mathbf{P}^j do not provide valid constraint for improving the posegraph of ORB-SLAM. Instead we use \mathbf{M}^i to estimate the pose $\tilde{\mathbf{P}}^j$ of the current keyframe \mathbf{K}^j .

Algorithm 1 Estimating pose using MapNet

```

INPUT:  $\mathbf{K}^j$  current keyframe,  $\mathbf{K}^i$  loop candidate keyframe,  $\mathbf{P}^i$  pose of the  $i$ th frame
 $\{\mathbf{K}^\alpha, \mathbf{K}^\beta, \dots, \mathbf{K}^\zeta, \mathbf{P}^\alpha, \mathbf{P}^\beta, \dots, \mathbf{P}^\zeta\} \leftarrow \text{getConnectedKeyframesData}(\mathbf{K}^i, \text{3D Map})$ 
 $\mathbf{M}^i \leftarrow \text{trainMapNet}(\mathbf{K}^i, \mathbf{K}^\alpha, \mathbf{K}^\beta, \dots, \mathbf{K}^\zeta, \mathbf{P}^i, \mathbf{P}^\alpha, \mathbf{P}^\beta, \dots, \mathbf{P}^\zeta)$ 
 $\tilde{\mathbf{P}}^j \leftarrow \text{getDeepPose}(\mathbf{K}^j, \mathbf{M}^i)$ 
return  $\tilde{\mathbf{P}}^j$ 

```

If we trust our vPR and are unable to estimate the relative pose between \mathbf{K}^j and \mathbf{K}^i , we can assume identity relative pose between these two keyframes. We call this naive vPR solution. Considering the simple scenario of only translational motion between the current keyframe and its loop closing candidate, the amount of error introduced in the ORB-SLAM’s pose graph, due to this noisy identity constraint, is equal to translation error between the current keyframe and its vPR candidate using their ground truth poses (red bars in Figure 10). Does MapNet provide a better solution than this naive approach? To estimate this we calculated the translation error between the actual ground truth pose of the keyframe \mathbf{P}^j and the estimated pose $\tilde{\mathbf{P}}^j$ provided by MapNet inference (blue bars in Figure 10).

Results show that 33 times out of 65, naive vPR solution, i.e., assigning the current keyframe the pose of its vPR matching keyframe, is better than deep solution. Although 32 times (out of 65) deep solution is better than the naive vPR solution, on average its performance is worse. Deep pose regressors are improving but they are not as accurate as hand-crafted image registration solutions yet.

5 Conclusion

Visual SLAM has evolved into a sophisticated application consisting of numerous carefully developed blocks (tracking, mapping, loop closing etc). SLAM family has dedicated commendable effort into optimizing each building block. However, SLAM systems are still mainly evaluated on a holistic level using aggregate metrics (average trajectory error (ATE)). In our opinion, this holistic evaluation, although informative, might drown the limitation of the individual building blocks. Individual block level evaluation is tedious and perhaps less glamorous. On the other hand, building a brand new SLAM system (feature based, direct, deep) is perhaps more attractive, which again is evaluated at a holistic level. In our opinion, the time and effort invested by the SLAM community in building complete SLAM systems is substantial. Instead of reinventing the entire pipeline, in this work, we performed in-depth analysis of the loop closure block of state-of-the-art SLAM which resulted in interesting findings. Firstly, in addition to (suspecting and) maturing vPR solutions, further investigation to develop robust real-time relative pose estimation modules might be more beneficial. Furthermore, pushing for All-Deep solutions is perhaps not the way forward. While deep vPR partially improves the

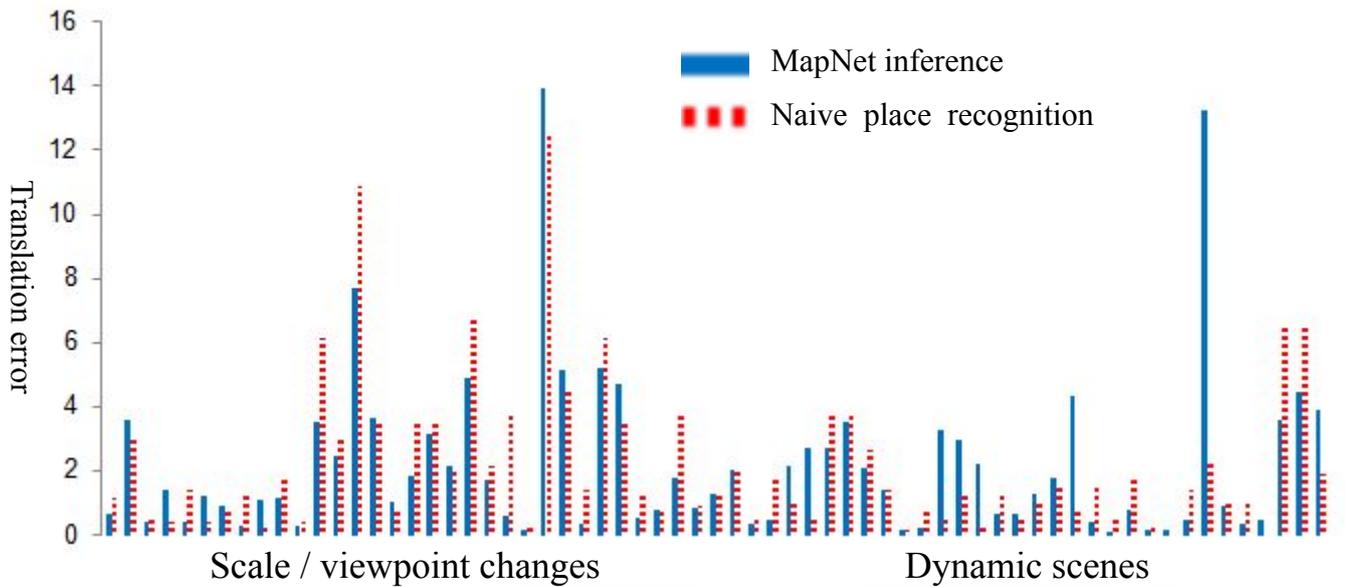


Fig. 10: **Understanding the limitations of deep relocators in loop closing scenarios.** Comparing the translation error shows that inferring pose of a given frame using the deep approach (MapNet) fares no better than naive place recognition solution, where the query frame is assigned pose of its nearest neighbour (best match) frame in the database. Note that the training dataset of MapNet already contains the image similar to the query image since it is a revisited location. vPR solution is better 33 times out of 65.

loop closures, deep relocators still suffer from fundamental limitations. This finding vindicates the decades-long effort of the SLAM community in designing robust registration methods. We hope our findings and the accompanying datasets will assist the community in further improving the popular ORB-SLAM’s pipeline.

Author Contributions

Saran Khaliq performed most of the experiments with ORB-SLAM, generating NUST-CLC dataset, experimenting at various thresholds, and finding out what fails the loop closure. **Muhammad Latif Anjum** and **Wajahat Hussain** supervised the research, steer-heading the project. The manuscript has been mainly written by these supervisors. **Muhammad Uzair Khattak** performed experiments on alternative relative pose estimator (e.g. VSFM, Colmap, and MapNet). **Momen Rasool** was involved in testing deep vPR (deep-LCD) and generating KITTI ground truth (scene graphs).

Acknowledgements

This work has been funded by Higher Education Commission (HEC), Govt. of Pakistan through two research grants 10023/Federal/NRPU/RD/HEC/2017 and 20-13396/NRPU/RD/HEC/2020.

Ethical Statement

The research did not involve testing on any human or animal. The project has been completed within all ethical boundaries of research.

References

1. Prasad et al V (2016) Learning to prevent monocular slam failure using reinforcement learning. arXiv preprint arXiv:160707558
2. Anosheh A, Sattler T, Timofte R, Pollefeys M, Van Gool L (2019) Night-to-day image translation for retrieval-based localization. In: ICRA, IEEE, pp 5958–5964
3. Brahmbhatt S, Gu J, Kim K, Hays J, Kautz J (2018) Geometry-aware learning of maps for camera localization. In: CVPR, pp 2616–2625
4. Campos C, Elvira R, Rodríguez JJG, Montiel JM, Tardós JD (2021) Orb-slam3: An accurate open-source library for visual, visual-inertial, and multimap slam. IEEE Transactions on Robotics 37(6):1874–1890
5. Dusmanu M, Miksik O, Schönberger JL, Pollefeys M (2020) Cross-descriptor visual localization and mapping. arXiv preprint arXiv:201201377

6. Gálvez-López D, Tardos JD (2012) Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics* 28(5):1188–1197
7. Gao X, Wang R, Demmel N, Cremers D (2018) Ldso: Direct sparse odometry with loop closure. In: *IROS, IEEE*, pp 2198–2204
8. Geiger A, Lenz P, Urtasun R (2012) Are we ready for autonomous driving? the kitti vision benchmark suite. In: *CVPR, IEEE*, pp 3354–3361
9. Geneva P, Maley J, Huang G (2019) An efficient schmidt-ekf for 3d visual-inertial slam. In: *CVPR*, pp 12105–12115
10. Gomez-Ojeda R, Moreno FA, Zuñiga-Noël D, Scaramuzza D, Gonzalez-Jimenez J (2019) Pl-slam: a stereo slam system through the combination of points and line segments. *IEEE Transactions on Robotics*
11. Horn BK (1987) Closed-form solution of absolute orientation using unit quaternions. *Josa* 4(4):629–642
12. Ikram MH, Khaliq S, Anjum ML, Hussain W (2022) Perceptual aliasing++: Adversarial attack for visual slam front-end and back-end. *IEEE Robotics and Automation Letters* 7(2):4670–4677
13. Kendall A, Grimes M, Cipolla R (2015) PoseNet: A convolutional network for real-time 6-dof camera relocalization. In: *ICCV*, pp 2938–2946
14. Kenshimov C, Bampis L, Amirgaliyev B, Arslanov M, Gasteratos A (2017) Deep learning features exception for cross-season visual place recognition. *Pattern Recognition Letters* 100:124–130
15. Lajoie PY, Hu S, Beltrame G, Carlone L (2019) Modeling perceptual aliasing in slam via discrete–continuous graphical models. *IEEE Robotics and Automation Letters* 4(2):1232–1239
16. Merrill N, Huang G (2018) Lightweight unsupervised deep loop closure. In: *RSS, Pittsburgh, Pennsylvania*, DOI 10.15607/RSS.2018.XIV.032
17. Milford MJ, Wyeth GF (2012) Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights. In: *ICRA, IEEE*, pp 1643–1649
18. Mur-Artal R, Tardós JD (2014) Fast relocalisation and loop closing in keyframe-based slam. In: *ICRA, IEEE*, pp 846–853
19. Mur-Artal R, Tardós JD (2017) Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics* 33(5):1255–1262
20. Mur-Artal R, Montiel JMM, Tardos JD (2015) Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics* 31(5):1147–1163
21. Naseer T, Ruhnke M, Stachniss C, Spinello L, Burgard W (2015) Robust visual slam across seasons. In: *IROS, IEEE*, pp 2529–2535
22. Naveed K, Anjum ML, Hussain W, Lee D (2022) Deep introspective slam: deep reinforcement learning based approach to avoid tracking failure in visual slam. *Autonomous Robots* pp 1–20
23. Pascoe G, Maddern W, Tanner M, Piniés P, Newman P (2017) Nid-slam: Robust monocular slam using normalised information distance. In: *CVPR*, pp 1435–1444
24. Qin T, Li P, Shen S (2018) Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics* 34(4):1004–1020
25. Sattler T, Zhou Q, Pollefeys M, Leal-Taixe L (2019) Understanding the limitations of cnn-based absolute camera pose regression. In: *CVPR*, pp 3302–3312
26. Shotton J, Glocker B, Zach C, Izadi S, Criminisi A, Fitzgibbon A (2013) Scene coordinate regression forests for camera relocalization in rgb-d images. In: *CVPR*, pp 2930–2937
27. Sturm J, Engelhard N, Endres F, Burgard W, Cremers D (2012) A benchmark for the evaluation of rgb-d slam systems. In: *IROS, IEEE*, pp 573–580
28. Wu C (2013) Towards linear-time incremental structure from motion. In: *2013 International Conference on 3D Vision-3DV 2013, IEEE*, pp 127–134
29. Zhang J, Tai L, Yun P, Xiong Y, Liu M, Boedecker J, Burgard W (2019) Vr-goggles for robots: Real-to-sim domain adaptation for visual control. *IEEE Robotics and Automation Letters* 4(2):1148–1155

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [FinalRevisionvideo.mp4](#)