Techniques for a Failsafe Visual Inertial SLAM System

by

KUSE, Manohar Prakash

A Thesis Submitted to
The Hong Kong University of Science and Technology
in Partial Fulfillment of the Requirements for
the Degree of Doctor of Philosophy
in the Department of Electronics and Computer Engineering

January 2020, Hong Kong

Authorization

I hereby declare that I am the sole author of the thesis.

I authorize the Hong Kong University of Science and Technology to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize the Hong Kong University of Science and Technology to reproduce the thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

KUSE, Manohar Prakash

January 2020

Techniques for a Failsafe Visual Inertial SLAM System

by

KUSE, Manohar Prakash

This is to certify that I have examined the above PhD thesis
and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by
the thesis examination committee have been made.

Prof. SHEN Shaojie, Thesis Supervisor
Robotics Institute
Department of Electronic and Computer Engineering

Prof. Bertram Emil SHI
Head of Department
Department of Electronics and Computer Engineering

Prof. Shaojie SHEN (Thesis Supervisor)	Dept. of Electronics and Computer Engineering
Prof. Ming LIU	Dept. of Electronics and Computer Engineering
Prof. Bertram Emil SHI	Dept. of Electronics and Computer Engineering
Prof. Xiaojuan Ma	Dept. of Computer Science and Engineering
Prof. Guofeng Zhang (External Examiner)	College of Computer Science Zhejiang University, Hangzhou, P. R. China

Department of Electronics and Computer Engineering

January 2020

Acknowledgments

I would never have completed this work without the help from many people. First of all, I thank my advisor, Prof. Shen Shaojie, for his patience, warm heart-ed mentoring, advice, and encouragement. I have learned from him how to develop, get feedback, express, and defend my ideas. These skills, I feel are most important for my later career.

I thank other members of my thesis committee, Prof. Chan, Prof. Zhang, Prof. Ma, Prof. Shi, Prof. Liu, for their insightful comments on improving this work. I am also grateful for the productive discussions and all the help from the UAV-research group especially, Qin Tong, Li PeiLiang, Joeey, Yi Zhou, Wang Kaixuan, Qiu Kejie, Usman Bhutta, Liang Qing and many other nice people. Also I am thankful to Pan Jie, William Wu and Tianbo Liu for their substantial support and outstanding contributions in terms of hardware. In daily life, we have been good friends. Without them, my graduate study in HKUST would be plain and bland. I thank communication tutor Tania Wilmshurst for her constructive proof reading work.

Last but not least, I thank my parents, fiancÃĺe, my brother, my good friends, for their support and encouragement.

Contents

List of Figures

List of Tables

Techniques for a Failsafe Visual Inertial SLAM System

by KUSE, Manohar Prakash
Department of Electronics and Computer Engineering
The Hong Kong University of Science and Technology

Abstract

Visual-inertial SLAM has been a contemporary research theme with various emerging commercial applications like robot navigation, augmented reality, 3D mapping etc. With the advent of several SLAM systems the theory of multiview geometry has been put to practical use. A typical SLAM system consists of several sub-systems including: visual-odometry, sensor-fusion, place recognition backend, place recognition frontend, posegraph solver. For a successful commercial deployment of SLAM algorithm it is important that the SLAM system be failsafe. In this thesis, we present several techniques for fail-safety of a SLAM system. We start by proposing an edge based visual-odometry method. The advantage of edge based visual odometry over traditional methods based on corner features and optical flow is that such methods also work well in featureless human built environment like corridors. Another advantage is that, the proposed method has a large convergence basin which allows for more reliable odometry computation under large motion or low frame rates. Next we present a learning based whole-image descriptor for loop detection. We demonstrated much higher recall rates compared to existing bag-of-visual-words based loop detection methods. Unlike previous loop detection methods which only evaluate their methods on fronto-parallel scenes, we tested our on datasets involving large viewpoint difference. In addition to higher recall, our method involves an order of magnitude less model storage size compared to bag-of-words dictionary and also an order of magnitude lesser FLOPS (floating point operations) making it suitable for a realtime SLAM system. We also propose a robust feature matching scheme and a local bundle optimization based computation for reliably estimating relative pose at loop detections. Unlike some existing works which merge trajectories from multiple runs offline, we develop a pose graph solver which is able to keep track of multiple co-ordinate systems, identify and recover from kidnaps live and in realtime. Extensive online experimental results are presented throughout the thesis. We conclude by proposing future research opportunities.

Chapter 1

Introduction and Background

We start by introduction of the problem of Simultaneous Localization and Mapping (SLAM) and motivate the need for positional feedback in some of the modern day applications. Additionally we introduce the importance of map making and other related applications for SLAM. Various building blocks that constitute a SLAM system is presented in some detail. We provide a through review of the state-of-the-art literature highlighting the developments and influential works in each of the SLAM building blocks. We touch upon the motivation and directions from the current work for the next generation of SLAM systems. We recognize that modern developments in machine learning and especially the success of the convolutional neural networks (CNN) as an enabler for semantically meaningful scene representations as a way for robust long term data associations.

The proposals are various aspects for robustness of SLAM system. Our first proposal involves a novel odometry method based on optimization of a cost function involving edges under a distance transform field. We then approach the problem of long term data association by proposing to learn a descriptor for scene representation. Next we develop a framework which we call place-recognition backend for computation of the geometry from the recognized long term data association. Finally we propose the use of the disjoint set data structure to handle multiple co-ordinate systems for our system be able to recover from kidnap live and in realtime.

1.1 What is SLAM ?

Simultaneous Localization and Mapping (SLAM) deals with estimation of the structure of the environment (the map) and estimation of the positional state of the robot. The

positional state estimation and mapping can be accomplished using a wide variety of sensors. Commonly used sensors include LiDARs, Laser-range finders, GPS, stereo cameras, RGB-D cameras, monocular cameras, event-cameras, inertial-measurement units (IMUs). A contemporary theme has been sensor fusion, which takes advantage of complementary nature of the sensors. For example, an IMU provides outlier free state estimates (linear acceleration and angular velocities in each of the 3-axis) at a high frame-rate for a short time interval, however over a longer duration the estimate drifts in space. On the other hand GPS can provide driftless estimates at a much lower frame-rate (approximately one measurement every 1 to 5 seconds), however it does not work indoors and gives poor performance in urban canyons. LiDARs provide very accurate 3D point measurements but do not provide color information and are extremely expensive. Depending on the environment, camera based SLAM (called visual-SLAM) is able to provide reliable position estimates but at much higher computational complexity. This thesis is mainly concerned on aspects of visual inertial SLAM (SLAM using cameras and IMU).

The requirement of recovering both position and the map, when neither are known distinguishes SLAM from marker-based tracking (using QR codes for example) because the map aspect (position of the markers) is already known in this case. Positional tracking with a fixed camera rig (for example a motion capture system) is not categorized as SLAM either. Similarly position tracking with RADARs is not SLAM either. The major distinguishing factor of SLAM is the recovery of both camera pose and environment structure (often referred to as *map*) while initially knowing neither. Yet another aspect of SLAM, which distinguish it from image based modelling (also known as 3D reconstruction) is the requirement that it must operate in realtime. What this means is that the pose and map estimation need to happen with streaming sensor measurements and position needed to be reported every 20ms to few seconds depending on the application.

Within the robotics community, research on SLAM was initiated arguably by Smith and Cheeseman [189]. Within the community the first 20 years (1986-2004) is known as the *classical age* [28]. This age saw the introduction of the probabilistic formulation for SLAM including approaches based on Extended Kalman Filters (EKF), Rao-blackwellised Particle Filters, and the maximum likelihood estimation (MLE). A through review on these aspects is presented by Durrant-Whyte and Bailey [12, 44]. Within the computer vision community, various geometric relations between features, points in the scene with

the imaged perspective and relative motion was being developed. All the relavant developments along the lines of multiview geometry are presented in the book by Hartley and Zizzerman [71]. These developments lay the foundation for visual-SLAM, some specific aspects of which are the focus of this thesis. The subsequent period (2004-2015) is referred to as the *algorithmic-analysis* by Cadena *et al.* [28]. This period involved the fundamental theoritical aspects of SLAM, including observability, convergence and consistency. The fundamental contribution from this period is the key role of sparsity towards development of fast and efficient SLAM solvers (for example, ceres-solver[2], g2o [94]). Some of these aspects are covered in the review paper by Dissanayake *et el.* [42]. Current major themes for SLAM research involve sensor-fusion, semantic scene understanding and re-localization, long term autonomy, realtime dense mapping, failsafe implementations etc.

Combining the complementary nature of visual and inertial measurements has been a popular sensor suite for realtime SLAM systems. The visual-inertial fusion approaches can be categorized into a) loosely-coupled [82, 93, 169, 214] and b) tightly coupled [19, 109, 112, 142, 147, 184, 221]. Loosely-couple approaches typically involve an estimate-update framework with IMU accumulation treated as independent module for the estimate step. Vision module is used as an update step in a Kalman Filter framework. Such approaches are easy to implement and runs fast but disregards correlations amongst the internal states of IMU and cameras, which affects the robustness and quality of the solution of the state estimation. Tightly coupled approaches jointly estimate all the sensor states and are based on EKF [19, 112, 142] or graph optimization [109, 147, 184, 221]. A brief review of the state-of-art SLAM systems is presented in Sec. 1.4. Some of the common sensor suites in use for visual SLAM include a) stereo camera, b) stereo camera + IMU, c) RGB-D camera, d) monocular camera + IMU, e) omnidirectional cameras.

1.2 Applications of Online SLAM

One of the major applications of SLAM is control and navigation of mobile robots. Mobile robots typically make use of a feedback-control loops [185], a SLAM system's localization aspect may be able to provide positional feedback. For example, a drone equipped with camera-IMU system can get the position feedback for its stabilization [157, 183]. Such

drones are used in indoor environments for surveillance and inspection application. Similar feedback loops are in use in a large variety of mobile platform like ground vehicles, humanoid robots, quadropods, sea vehicles etc. A SLAM system essentially provides a state estimate for such robotic systems.

The mapping aspects of the SLAM finds application in vision based obstacle avoidance and path planning. The map obtained from a SLAM system is the view of the environment for the mobile platform. This structure is often represented in data structures like Octotrees, polygonal meshes etc. RRT, A* like sampling algorithms are used to come up with a free-space corridors. A obstacle avoiding path is than planned by using polynomial trajectories in a minimum-snap like formulations [135]. Some of the work that use a visual-SLAM systems for planning paths for mobile platforms like UAVs include [49, 50, 117, 122, 179]. Works on use of visual-SLAM for planning paths for autonomous ground vehicles include [13, 103]. Such ground vehicles are in use in various application domains for example warehouses, agriculture, planetary exploration (in Mars rovers), self driving cars etc.

The ability to accurately localize a camera without a prior reference is also crucial to its application in Augmented Reality (AR). With a SLAM localization system in back-end, a rendering engine in an AR application can overlay objects on the viewport. Such systems find application in entertainment, 3D modelling etc. Pioneering work on use of SLAM for AR application was proposed by Klein and Murray [92]. Some other academic works in development of vision based AR systems are [113, 121, 181]. More recently SLAM system have been commercialized for end-user applications. Google Tango (ARCore)and Apple's ARKit are some of the commercial SLAM engines geared towards AR applications.

1.3 Visual-Inertial SLAM System Building Blocks

In this section we will describe the basic visual-SLAM building blocks. We briefly review the relevant influential literature in each of the sub-systems that make up a visual-inertial SLAM system. A typical SLAM system has a front-end and a back-end. The front-end abstracts the scene geometric structure from sensor data. For example, in case of a camera-IMU sensor suite, the front-end extracts relevant features (like corner points, lines etc) from the sensor data and associate this data with scene geometry. Various methods

in the SLAM front-end naturally intersects with methods from Computer Vision, Image Processing, and Signal Processing.

A SLAM back-end uses the abstracted sensor data observations into a non-linear optimization framework to arrive at the scene structure and ego-motion. At the core it involves formulating, analyzing and solving non-linear least square problems. Various aspects on Linear Algebra, Graph Theory, Numerical Methods Optimization, are critical tools to build a SLAM system.

In this thesis, we introduce the terms front-end and back-end for the place-recognition. As has been recognized in the literature [28], long term data associations provide a formidable challenge for a robust SLAM system. This part in often referred to as loop-closure in the literature. In the current SLAM systems, loop-closure is simplistically handled with a bag-of-visual-words framework. In the SLAM community recent advances in Computer Vision and Machine Learning are yet to make a big impact on development of SLAM systems. This thesis is an attempt to bridge this very gap.

The place recognition front-end abstracts the raw sensor data (eg. images) to a place representation using image descriptor and other semantic scene information. This abstract representation is used to provide putative loop candidates to the place-recognition back-end. Revisits occurring under large view point difference, changing weather conditions, insufficient texture etc are the challenges. Recent advances in representational power for the convolutional network presents a great opportunity towards analysis and eventual solution of this problem.

The place recognition back-end computes the relative pose given a loop candidate and the tracked points. This is a challenging aspect as the places can be revisited by the system under vastly varying conditions. Semantically meaningful place representation can be a viable tools towards developing methods which are able to compute camera poses when revisits occur under adversaries. This loop candidate relative pose is used by the SLAM-backend to correct for the drift. An high-level overview for a visual inertial SLAM system is summarized in Fig. 1.1.

1.3.1 Tracking and Visual Odometry

Visual Odometry (VO) is the process of estimating ego-motion from observed scene features. The term VO was coined by Nister in his landmark 2004 paper [155]. A slightly

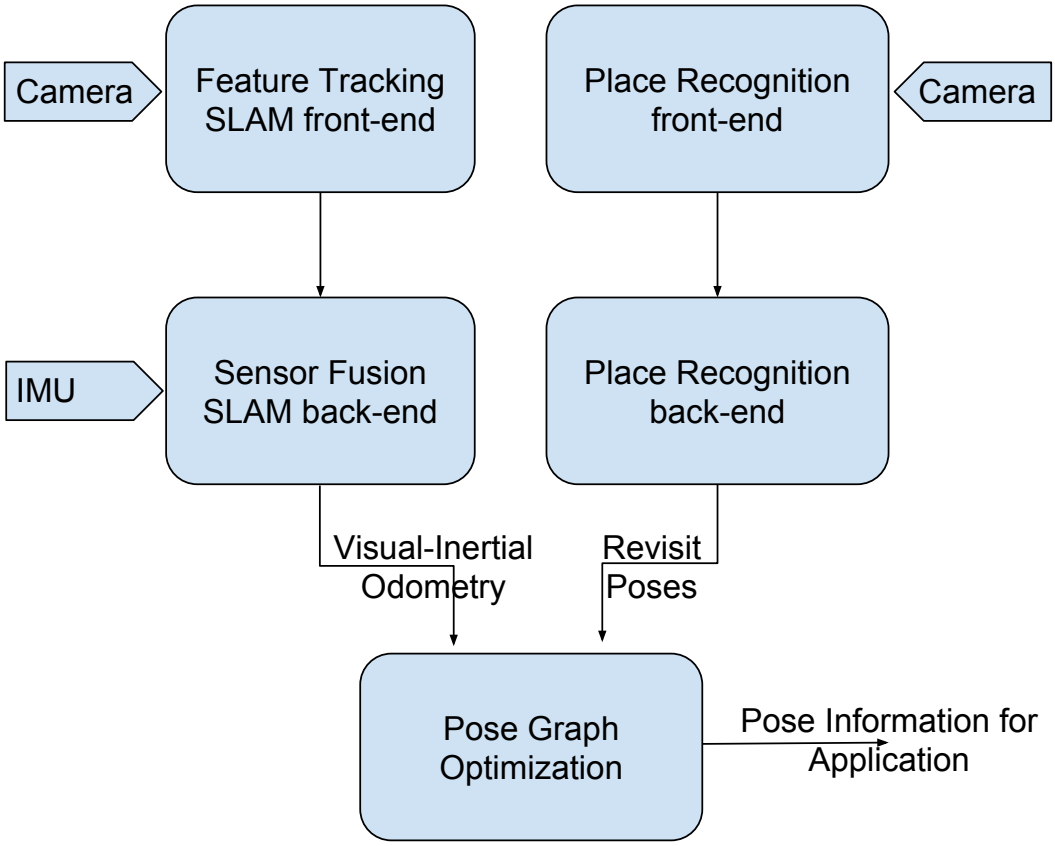

Figure 1.1: Visual Inertial SLAM System Building Blocks

dated review of VO approaches can be found in [55, 178]. Here we briefly describe the formulation for visual odometry. A camera system moves in the environment and taking images at discrete intervals. For monocular camera, the set of images taken at time k are denoted by $\{I_0, I_1, \ldots, I_n\}$. In case of a stereo camera system, there are two set of images at each time instant, viz. left and right denoted by $\{(I_0^l, I_0^r), (I_1^l, I_1^r), \ldots, (I_n^l, I_n^r)\}$. For a RGB-D camera at every time instant, the depth image is available in addition to the intensity image,$\{(I_0, D_0), (I_1, D_1), \ldots, (I_n, D_n)\}$.

The two camera poses at adjacent time instant n and $n-1$ are related by the rigid body transform ${}^kT_{k-1} \in SE(3)$ (special Euclidean group), giving the pose of camera at time instant $k-1$ in the frame of reference of k. Put differently, ${}^kT_{k-1}$ refers to the spatial position of frame $k-1$ when viewed from frame k:

$$ {}^kT_{k-1} = \begin{bmatrix} {}^k\mathbf{R}_{k-1} & {}^k\mathbf{t}_{k-1} \\ \mathbf{0} & 1 \end{bmatrix} \tag{1.1} $$

where ${}^k\mathbf{R}_{k-1} \in SO(3)$, the special orthogonal group or the rotation matrix and ${}^k\mathbf{t}_{k-1} \in \mathbb{R}^3$ representing the translation vector. The full trajectory followed by the camera system

is recovered in the frame-of-reference of the 0^{th} frame by concatenating the transforms ${}^{k}T_{k-1} \forall k = 1, \ldots, n$. In typical implementations a keyframe is fixed for say every m frames or so and the relative poses are computed between this frame and the next set of frames. An iterative refinement over the last m poses can be performed after this step to obtain a more accurate estimate of the local trajectory. This iterative refinement involves minimizing the reprojection errors of the triangulated 3D points over the last m images. This is often also referred as windowed-bundle-adjustment. One common issue with the visual odometry is that since the final pose essentially accumulates the poses from previous estimate, the errors tend to add up resulting in a drift in the trajectory.

At the core, the task of the VO system is relative pose estimation between the reference frame (indexed as r) and the current frame (indexed as c). See Fig. 1.2 for summary of notations. All the approaches in general can be categorized as 3D-2D alignment (minimization of the reprojection error) and 3D-3D alignment (aligning 2 sets of 3D points to arrive at the relative pose). Majority of approaches involving 3D-2D alignment are based on tracked features, making use a variant of a Kanade-Lucas-Tomasi (KLT) algorithm [204], some of them are summarized in [14]. Using the matched features across temporal frames triangulation is performed to infer the 3D positions of the features, in case of monocular sequences. In case of stereo and RGB-D sequences depth can be inferred directly at each time instant to arrive at the 3D positions. In a non-linear least squares formulation the reprojections of these 3D points on the imaged points is minimized. Some of the works based on this general theme are [43, 80, 109, 146, 155]. Some of the approaches based on 3D-3D alignment are [78, 102, 132, 139, 158].

Further the VO approaches can be categorized as being based on point features [92, 146], ones based on lines and edges [63, 219], and direct approaches [46, 54, 88, 97]. Feature based approaches have been the most popular, however direct approaches remain popular in recent times. Although feature based approach work well in textured environment, edge based approaches have shown to be working well in conditions where there are numerous edges as is common in indoor man-made scenes. Recently, Platinsky *et al.* [163] and Yang *et al.* [218] compare feature based approaches and direct approaches.

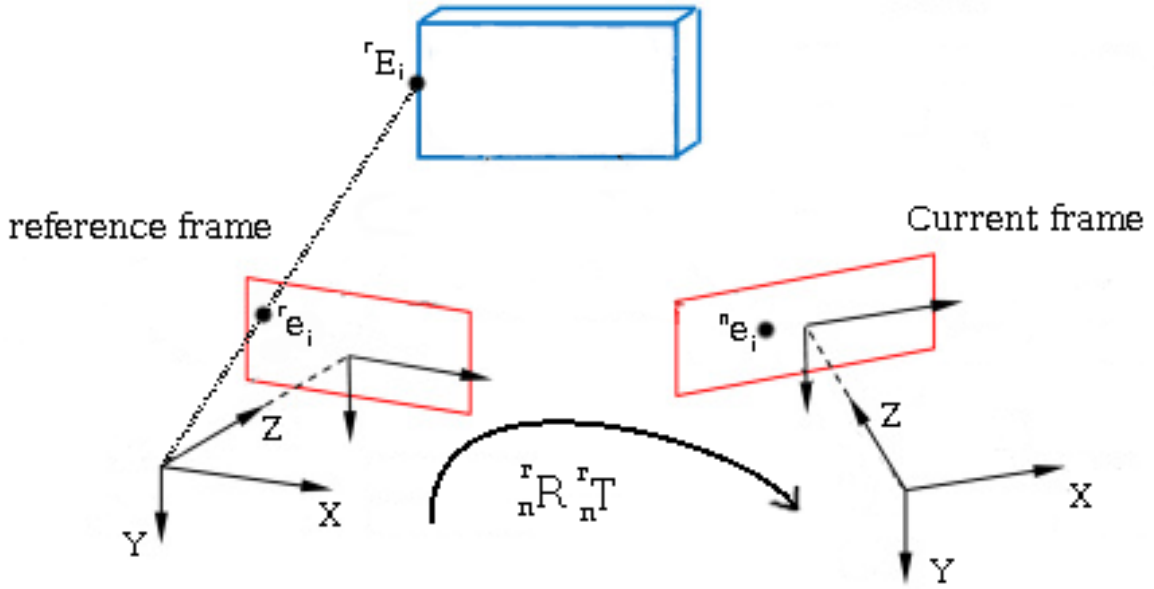

Figure 1.2: Notations and Conventions

1.3.2 Sensor Fusion

Combining the complementary nature of visual and inertial measurements has been a popular sensor suite for realtime SLAM systems. The visual-inertial fusion approaches can be categorized into a) loosely-coupled [82, 93, 169, 214] and b) tightly coupled [19, 109, 112, 142, 147, 184, 221]. Loosely-coupled approaches typically involve an estimate-update framework with IMU accumulation treated as independent module for the estimate step. Vision module is used as an update step in a Kalman Filter framework. Such approaches are easy to implement and runs fast but disregards correlations amongst the internal states of IMU and cameras, which affects the robustness and quality of the solution of the state estimation.

Tightly coupled approaches jointly estimate all the sensor states and are based on EKF [19, 112, 142] or graph optimization [109, 147, 184, 221]. Sliding window based monocular VIO using non-linear least squares with inertial residue and visual residue terms provides an elegant, easy to implement and effective framework for tightly coupled visual inertial odometry. We briefly describe this formulation.

The full state vector in the sliding window is defined as:

$$\chi = [\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n, {}^{b}\mathbf{x}_c, \lambda_0, \lambda_1, \ldots, \lambda_m]$$

$$\mathbf{x}_k = [{}^{w}\mathbf{p}_k, {}^{w}\mathbf{v}_k, {}^{w}\mathbf{q}_k]$$

$\mathbf{x}_k$ $(k = 1, \ldots, n)$ is the IMU state at the time instant k, assumed to be at the instant

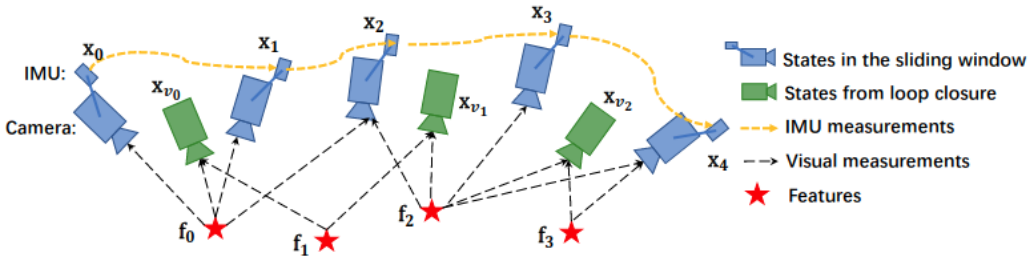

Figure 1.3: Summary of notations. Figure borrowed from [165]

that the image is captured. It contains position (${}^w\mathbf{p}_k$, a 3-vector), velocity (${}^w\mathbf{v}_k$, a 3-vector) and orientation (${}^w\mathbf{q}_k$, represented in Hamiltonian quaternions) of the IMU in the world frame and acceleration bias ($\mathbf{b}_a$, a 3-vector) and gyroscope bias ($\mathbf{b}_g$, a 3-vector) in the IMU body frame. λ_l is the inverse depth of the l^{th} feature. ${}^b\mathbf{x}_c$ is the relative pose of camera (3-vector representing position and 4-vector representing orientation in Hamiltonian notations) in the IMU frame of reference (also referred to as IMU-camera extrinsic calibration).

The following local visual-inertial bundle adjustment formulation which minimizes the sum of prior and the mahalonobis norm of all measurement residuals for IMU ($r_{\mathcal{B}}$) and residuals for camera measurements ($r_{\mathcal{C}}$) to obtain a MAP (maximum posteriori) estimate of the states. $\mathcal{B}$ is the set of IMU measurements and $\mathcal{C}$ is the set of features which have been observed at-least twice in the current sliding window. $\{r_p, H_p\}$ is the prior information from marginalization.

$$\underset{\chi}{\text{minimize}}\{||r_p - H_p\chi||_2^2 + \sum_{k\in\mathcal{B}} ||r_{\mathcal{B}}({}^k\hat{z}_{k+1}), \chi)||_2^2 + \sum_{(l,j)\in\mathcal{C}} ||r_{\mathcal{C}}({}^{c_j}\hat{z}_l), \chi)||_2^2\} \tag{1.2}$$

For detailed derivation for the residue terms refer to our work on edge based visual inertial fusion [118] or by Qin Tong *et al.* VINS-mono [165]. Such formulation can be solved with non-linear least squares solvers like the ceres-solver [2]. Robust norms like the Huber norm [81] are used to reduce the influence of outliers in the estimation process.

In case of monocular VIO, another critical problem to be addressed is that of scale estimation. The scale for camera measurements is unobservable and thus, render these observation impossible to use directly in the above tightly coupled formulation. An estimator for the scale is needed. The scale is initialized in initial few frames by computing vision only structure from motion and aligning it with motion estimation from dead-reckoning the IMU. A detailed discussion on this critical issue is well presented in [165].

1.3.3 Pose Graph Optimization

Pose Graph optimization is a well-known technique to build a consistent and global map. The keyframes with their poses in time are represented as graph nodes. Pose-pose constraints are the edges. Pose-pose constraints could be odometry constraints which come from visual-inertial odometry (VIO) back-end. Loop closure constraints come from the relative poses computed between the putative loop closure candidate by the place-recognition back-end. This part is often also referred as the SLAM back-end. A general tutorial on pose graph based SLAM is presented by Grisetti *et al.* [65].

The graph-based formulation was introduced in the seminal work by Lu and Limios [128], relative motion between two scans was measured and the resulting graph was optimized by iterative linearization. Due to progress in processor speeds and development of sparse direct solvers for the graph based optimization, they can now be solved in real-time. It can be viewed as non-linear least squares optimization which can be solved by re-linearizing at each step and solving the linear equations.

Current state of the art SLAM back-ends are g^2o [94], iSAM2 [86], ceres-solver [2]. These essentially provide a framework to specify the pose graph nodes and the constraints, which are then solved iteratively, taking advantage of the sparse structure of the pose-graph, to arrive at a globally consistent estimate of the poses at every graph node. This essentially help reduce drifts by identifying place revisits.

These being least-squares optimizers are not robust against outliers. Several authors have proposed to use robust norms, like Huber norms however there are some works which deal specifically with spurious constraints which often results from wrong data associations in SLAM front-ends and place recognition front-ends. Sunderhauf and Protzel [198] suggested the use of optimizable switch variable on each of the edge terms in the pose graph. Some other relevant work in this direction include [1, 30, 64, 104, 106, 159].

More recently Calafiore *et al.* [29] has cast the pose graph optimization in complex domain and reformulated it as a semi-definite programming (SDP). However, this approach cannot be used in realtime owing to very high complexity of SDP solvers even for a moderately large sized problem.

In this thesis we proposed a kidnap aware pose graph solver. Our solver is essentially based on the works by Sunderhauf and Protzel [198]. On account of kidnap (blocking the camera view for over a minute and transporting to a totally different location), the VIO

needs to be shut-off and restarted at unkidnap. This results in new co-ordinate references at every unkidnap. Our solver is able to handle multiple such co-ordinate systems and recover from such scenario whenever a revisit is detection. Our relocalization works under complicated kidnap scenarios. Unlike some other relocalization systems which works offline to merge trajectories, our system is able to relocalize live and in realtime.

1.3.4 Place Recognition

A robot performing odometry, ie. accumulating poses from just visual sensors, or fused IMU camera system (both loosely and tightly coupled) views the world as an infinite corridor. As noted earlier, the odometry pose errors get accumulated which results in positional drift. A module which can recognize revisit and compute relative pose between the revisit can help reduce the drift. Lowry *et al.* [127] presents a comprehensive review of the loop detection modules in use and motivation for further developments. Although relocalization can be accomplished from the map developed by the SLAM, appearance based methods remain popular.

Some popular works include FAB-Map[41], DBOW[57], RTAB-Map[100]. These are fundamentally derived from the seminal approach[187]. It describes an image with visual-words derived from clustering of descriptors at sparse point features. Although BOVW-based methods have high specificity, these methods suffer from high miss rates (ie. low recall), especially under larger viewpoint difference and other adversaries.

Current day state-of-the-art SLAM systems make use of the BOVW based relocalization module. Although these methods have achieved some success, the use of sparse point features and static vocabulary leads to discarding of a significant amount of information in the scene, which is detrimental to robustly modelling place representation under camera effects like motion blur, texture deficient environments, noise, etc. Further, the in-variance from viewpoint changes is limited by the representational power of the low-level visual feature descriptor. For example, using a SIFT-based vocabulary can provide better performance compared to a FAST-based vocabulary, at a much higher computational cost.

In our previous work [98], we have presented a weakly supervised method which can learn place representation from data. We have shown, our method to have outperformed

previous approaches on the precision-recall metric. Our method is able to recognize revisits even when they occur in challenging environment including large viewpoint difference, less texture, low lighting, motion blur etc. More details on learning a semantically meaningful place representation and literature review of recent methods we refer to Chp. 4.

After recognition of a revisit by place-recognition front-end, the place recognition back-end computes the relative pose between the putative candidate. Traditionally this relative pose has been computed using approaches similar to 5-point algorithm using matching the descriptors of the tracked features from the SLAM front-end. Although this works in several cases where there are sufficient number of tracked features common in both views of the scene (from the previous visit and the current visit) there often occurs a challenging scenario where the revisits occurs at large viewpoint difference. Computation of relative poses under such challenging cases remain an open challenge. In our work we have attempted this problem. We propose to accumulate keypoint features from several keyframes and solve a local bundle to robustly estimate the relative pose.

In the computer vision community in recent times there has been efforts to use coherence constraints (similar motions for neighbouring pixels) to improve feature matching quality. These approaches have been expensive to compute and not suitable for realtime applications. Wang *et al.* [18] proposed the GMS-matcher. This approach is able to use a simple statistical likelihood measure which is a proxy for coherence constraint. The state-of-the-art semantic alignment methods [22, 69, 91, 156, 170] rely on powerful image representations from a deep convolution network. Although impressive results have been obtained for known objects (using imagenet or similar dataset), matching multiple objects robustly remain an open problem [171]. In this thesis, we propose a novel approach which uses cues from the place representation network to densely match features (even non-textured) and compute pose using an optimization based unified formulation. The details of this are presented in Chp. 5.

1.4 State-of-the-art Visual Inertial SLAM Systems and Future Challenges

Current day state-of-the-art realtime visual inertial SLAM systems are VINS-MONO [165], OKVIS[109]. These approaches are complete systems with visual-odometry, tightly

coupled camera-IMU fusion, loop detection module, pose graph optimization etc, which encompass several of the newest advances in the state-of-the-art in the SLAM literature. There has also been a significant number of vision-only realtime SLAM systems. Some of the notable approaches include PTAM[92], SVO [54], LSD-SLAM[46], DSO [45] and ORB-SLAM [145, 146]. The simplest approaches to integrate these systems with IMU is to use a loosely coupled approach. Leutenegger *et al.* [109] in their OKVIS implementation have clearly demonstrated the advantage of using vision and IMU over vision-only and loosely coupled SLAM approaches. They have been successfully demonstrated to work on various applications ranging from flying UAV autonomously, AR, realtime dense mapping to name a few.

Recently, large cooperates like Google and Apple have developed their own SLAM systems geared especially towards AR applications like games, modelling etc. Google Tango (ARCore) [1] and Apple's ARKit [2] are the current day commercial monocular VINS systems. These systems in are also being demonstrated on smart phones.

Within the robotics community there has been a trend towards long-term autonomy. Most notably the strands-project [72] has demonstrated autonomous operations for 104 days in controlled settings. Long-term autonomy purely based only on a minimalist visual-inertial sensor suite remain a topic of exploration. Robustness issues in addition to hardware failures are connected to incorrect data associations in the front-end and/or in the back-end. Having a sufficiently high sampling rate of the sensor can achieve robust short-term data association. Long-term data association is a more challenging task. This involves intelligent place recognition modules and reliable relative pose computation between long-term data association.

Current SLAM systems use a variant of bag-of-visual-words based approaches to efficiently look up currently observed features in the set of previously observed features. As has been noted earlier, these approaches are not capable of handling severe illumination variations, large viewpoint difference, and other adversaries. While current day SLAM systems have been demonstrated to work on building-scale environment, operation on larger scale environments like city scale etc still remain challenging. Further, the recall of the underlying bag-of-visual-words model is limited by the descriptive power of the underlying feature extractor. There is a need for SLAM methods whose computational

[1] https://developers.google.com/ar/discover/
[2] https://developer.apple.com/arkit/

and memory complexity remain bounded.

More recently there have been attempts to use pretrained CNNs as whole-image-descriptor, pioneered by Sunderhauf *et al.* [199]. The author of this thesis has also developed a place-recognition module for a SLAM system based on popular NetVLAD architecture [5].

Although feature based geometry and optical flow is the defacto standard for odometry computation, more recently direct approaches have been pioneered by Kerl *et al.* [88]. The author of this thesis has attempted to improve upon this method by formulating the odomtery as an edge-alignment problem in their work [97]. Some other authors have also made use of edges for odometry computation [63, 219, 229]. The advantage of using edges is that, even in environments void of corner features visual-odometry can be reliably be estimated. Such environments are common in human built indoor places.

The core theme of this thesis is the development of modules for semantically meaningful place representation of the environment for robust long-term data association. This thesis develops techniques for robustness and failsafety of visual SLAM system in regards to

- Direct edge tracking for visual odometry in environments void of textured features but with lot of edges.
- A novel learning based image descriptor for revisit detection.
- A feature accumulation from several keyframes and local bundle based approach for robust relative pose computation.
- A kidnap aware online relocalization pose graph solver with capability to handle multiple co-ordinate systems.

Chapter 2

Thesis Narrative

In this chapter we provide a general motivation and overview of the various proposed topics. Our major contribution involve a novel edge based odometry method. Our formulation is a direct alignment formulation which does not need to extract features from the scene. The optimization framework was shown to have a much larger convergence basin on account of the novel cost function field. Next we propose a weakly supervised method for place representation. The proposed method provides a whole-image descriptor to represent the scene semantics and recognize a revisit. Our method was shown to be robust towards large view point changes, invariant to rotations. Favourable outcomes where obtained when we compared our method with state-of-the-art loop detection methods. Next we proposed a kidnap aware relocalization system which is able to handle multiple co-ordinate system and provide for general merging of the trajectories online and in realtime. The key contribution in this regard was the use of the datastructure *disjoint-sets* to maintain trajectory-set associations and indirect inference of relative transforms between the co-ordinates in the same set even when there were no loop connections between those very sets. Commuting relative pose for revisits not at fronto-parallel scenes is a challenge due to limited scene overlap and small number of matches. We address this issue by proposing a semi dense feature point matcher. We also deal with the problem of relative pose computation in this case. We proposed to accumulate keypoints from multiple keyframes as opposed to using just a single image pair as common in several contemporary SLAM system. Next we propose an iterative refinement of the local bundle. Our system is open sourced and available as a pluggable module for the popular VINS-fusion system.

2.1 Proposed Edge Based Alignment for Odometry Computation

Reliable feature tracking is a key for short term data associations and to arrive at an initial estimate of the camera poses. A large body of literature in this area has been focused towards selecting good features to track, construction of feature level descriptors, aspects and numerical issues with geometry computation to name a few. In general, feature tracking based methods for odometry work well when the environment has sufficient texture, adequate lighting, distinguishable features and camera sampling rate is sufficient enough. However in environments with little texture, low lighting, these methods often produce insufficient number of point feature matches for robust and accurate computation of camera relative poses. Repeated features in the scene give yet another significant challenge. For example, in a scene with multiple same looking windows, or the one with repeated texture, the corners of a one window can easily be confused with corner of the next window.

Recently, we have seen development of direct methods for computation of short term data association. These methods works directly on all pixels of adjacent frames to produce an estimate of the relative pose, while not having to explicitly detect and match feature points. Kerl *et al.* [88] first introduced this technique. They proposed to register two consecutive RGB-D frames directly by minimizing the photometric error. The formulation can be setup using the non-linear least squares optimization which can be solved by efficient solvers. Although these methods are computationally expensive but still can give realtime performance on hardware with sufficient computing power. The advantage of this method is that it allows for the inclusion of a prior motion model and can deal with the problem of perceptual aliasing in the scene. However, being dependent on image intensity values, this is sensitive to changes in exposure and lighting condition. Some other works in this direction are [46, 54].

We recognize issues with the current day direct methods:

- Sensitivity to image intensity and noise.
- Slow convergence for iterative methods on photometric cost function.
- Small converge basin and tendency of photometric alignment based to get stuck in

local minima.

To deal with these issues we propose a novel, feature-less approach for 3D-2D pose estimation [97]. We propose to align a reference image with the current image by minimization of the sum of squared distances between transformed-projected (on current frame) co-ordinates of the edge-pixels from the reference frame and the nearest edge-pixels in the current frame. We make use of only the edge pixels, relying on our observation that the Distance Transform [52] of the edge-map of the current frame can be used to model the distance between edge pixels from the current frame and the re-projections from the reference frame. In our work [97] we demonstrate a successful application of our method on RGB-D images.

Unlike the previous direct approach by Kerl *et al.* [88], our proposed approach does not rely on the photometric cost function and also does not assume the photo-consistency. The implication of this being, it is robust under changing lightening conditions and has a rather large convergence basin. Also, being a direct method, does not involve sparse features (like SURF, SIFT etc) and feature matches.

Our contributions in edge alignment:

- Observation that the Distance Transform of the edge-map of the current frame can be used to model the distance between edge pixels from the current frame and the re-projections from the reference frame.
- A novel cost function which works directly on edges for estimation of relative transform between the two views.

We extended our method [118] to work with stereo cameras using the edge tracking formulation for visual odometry. Additionally we introduce tightly coupled framework for fusion of IMU measurements along with camera odometry. We explicitly address the problems of lighting variations and estimator convergence using the edge alignment technique. Our method compares favourably with state-of-the-art methods which uses point features especially under aggressive motions. Detailed comparison can be found in our work [118].

Our method of edge tracking can also be used in a monocular settings. We need some estimates on the depth (up-to scale is also sufficient). This can be computed with purely geometric methods for example along the lines of literature on depth from motion [46].

Alternately, with recent advances in neural networks capable on predicting depth [62] our method can be realized in a monocular setting. Since edges are abundant in man made environments like buildings etc, this can boost the performance of the tracking module of a monocular SLAM system, like VINS-mono [165]. More details in regards to the core formulation of our edge alignment along with comparison with competing methods are presented in Chp. 3.

2.2 Proposed Whole-Image-Descriptor for Revisit Detection

Relocalization is a relevant problem studied in context of SLAM. As has been recognized in the literature [28], long-term data association is a more challenging issue and involves loop detection, validation and pose computation potentially from a widely different perspective. A SLAM system without a loop closure module views the world as an infinite corridor. Identifying a place revisit can help reduce the drift which occurs due to error accumulation from the poses as estimated by sliding-window based bundle adjustment methods.

Over the past few years various place recognition systems have been developed in the context of SLAM. Prominent amongst these are FAB-MAP[41], DBOW [57] based on the bag-of-visual-words theme [187]. It describes an image with visual-words derived from clustering of descriptors at sparse point features. Although they provide for an efficient approach for feature look-up, they are not capable in handling severe illumination variations and large view point changes. Major issues with methods based on bag-of-visual-words:

- These methods suffer from high miss rates (ie. low recall), especially under larger viewpoint difference and other adversaries.
- Fails completely in environments with less and/or repeated features.
- Has limited capacity and is much more easily confused with similar looking scenes.

Some of these short comings have already been addressed in the literature by extending bag-of-words methods to use spatial position information of the features. However, to address the fundamental flaw of discarding a large amount of scene information, some

authors have attempted to develop whole image descriptors of the scene. Recently, CNN based approach have been popular. Some of the works using a pretrained CNN network include [32, 33, 200]. Some works also train a network for the specific task [123, 136].

We have proposed a novel scene representation which can be learned in a weakly supervised manner based on NetVLAD [6]. We summarize our contribution in regard to scene descriptor learning:

- We identify the learning instability issue when training a CNN with NetVLAD using the commonly used tripletloss function. We proposed the allpairloss function to mitigate this issue and provide for a quicker and stable learning.

- We proposed to use the decoupled convolutions instead of the usually used convolution. This resulted in over 3X computational speeds of the descriptor computation compared to the descriptor computation based on the standard VGG16 network.

- We proposed a channel squashing approach for dimensionality reduction. This approach was shown to perform about the same as the standard NetVLAD but at a descriptor size of an order of magnitude smaller than it.

A more detailed explanation and related results and comparisons are presented in Chp. 4.

These CNN based image descriptors are high-dimensional vectors (≈1000-D). For a truly scalable SLAM system, the issues of fast and low memory footprint place representation querying is crucial. Methods based on locality-sensitive-hashing have recently emerged [85, 211] in the place recognition community, however, to the best of the authors knowledge they have not yet been studied in the context of relocalization capability in the context of a realtime SLAM system. Robust place representation and efficient querying give rise to opposing objectives. For example, a larger feature descriptor can have larger capacity for place representation, however searching in this high dimensional space can be slow. However, if the dimensionality is reduced, fast searching can happen while the representation power decreases. Development of fast hashing for high dimensional CNN based feature descriptors and other semantic information is one of the emerging themes for research in robot relocalization. Additionally, augmenting scene-text and semantic cues like objects for relocalization could be a future research topic.

2.3 Proposed Kidnap Aware Pose Graph Solver

The next step is integrating the long-term data associations along with the relative pose constraints into the pose graph formulation. An overwhelming majority of SLAM systems use a variant of the 5-point algorithm for relative pose computation for putative matches generated by the underlying bag-of-visual-words method. This works well in cases of revisit under fronto-parallel revisits. However, when the revisits occur at substantial viewpoint changes and under other adversaries like less texture, low lighting etc, on account of small number of matches and the matches laddened with wrong data associations, 5-point algorithm and its robust variants often fails to compute reliable relative poses.

To reliably compute relative pose between the loop candidates at large viewpoint changes we make use of a recently proposed method, GMS-Matches [18]. It relies on a simple and intuitive voting scheme for coherence constraints (similar motions for neighbouring pixels) for feature matches. It takes about 200ms for point feature matching between an image pair of sizes 640x480 to produce a few thousand matching points depending on the nature of the scene. We use several nearby keyframes and accumulate point feature matches. We make use of the switching constraint based outlier rejection mechanism. Alternating minimizations was used to give a coarse estimate of the relative pose at revisit. Next we propose a refinement scene which makes use of the local bundle to minimize the reprojection errors for estimation of relative pose between image sequence. Our robust pose computation method has been integrated in the realtime SLAM system.

The main distinguishing point of our pose graph solver is that it can handle multiple co-ordinate system and relocalize/merge trajectories online and in realtime. We proposed to handle the set associations of various co-ordinate systems using the *disjoint-set* datastructure. We observe that certain kidnap cases are tricky to handle. Details of this can be found in Chp. 5.

2.4 Summary of Contributions

We give a final unified summary of the thesis:

- A fully functional kidnap aware SLAM system based on VINS-Fusion available as an open source package.

- Formulation of the direct edge alignment approach for visual odometry.
- Development, analysis and testing of a CNN based learned weakly supervised approach for realtime place representation and relocalization.
- A robust method for keypoint matching which is able to work even in non-fronto parallel scenes. Pose computation by feature aggregating across keyframes and formulating the problem of pose computation as local bundle alignment.
- A kidnap detector and kidnap aware pose graph solver which is able to handle multiple co-ordinate systems and provide for trajectory merging online and in realtime.

2.5 Open Source Packages Resulting from this Thesis

- Edge-alignment based Visual Odometry `http://github.com/mpkuse/edge_alignment`.
- A tightly coupled VINS using edge alignment `https://github.com/ygling2008/direct_edge_imu`.
- Learning whole image descriptor `https://github.com/mpkuse/cartwheel_train`.
- Cerebro - Whole Image Descriptor based Loop Detection and Relative Pose Computation `https://github.com/mpkuse/cerebro`.
- Kidnap aware Pose graph solver `https://github.com/mpkuse/solve_keyframe_pose_graph`.

Chapter 3

Robust Edge Based Visual Odometry

There has been a paradigm shifting trend towards feature-less methods due to their elegant formulation, accuracy and ever increasing computational power. In this work, we present a direct edge alignment approach for 6-DOF tracking. We argue that photo-consistency based methods are plagued by a much smaller convergence basin and are extremely sensitive to noise, changing illumination and fast motion. We propose to use the Distance Transform in the energy formulation which can significantly extend the influence of the edges for tracking. We address the problem of non-differentiability of our cost function and of the previous methods by use of a sub-gradient method. Through extensive experiments we show that the proposed method gives comparable performance to the previous method under nominal conditions and able to run at 30 Hz in single threaded mode. However, under large motion we demonstrate our method outperforms previous methods using the same run-time configuration for our method. Our codebase for edge alignment based visual odometry is opensourced [1]

3.1 Introduction

Reliable pose tracking of a robotic vehicle is a key element in its precise and stable control for an autonomous operation. To achieve this, the Global Positioning System (GPS) has been a key enabler for autonomous navigation. However, for GPS denied environments localization and navigation using other on-board sensor modalities like lasers, IMUs, monocular and stereo cameras, an RGB-D cameras have been popular research topics [47, 88]. In this work, we propose a direct (feature-less) approach for visual 3D-2D relative 6-DOF

[1] `https://github.com/mpkuse/edge_alignment`

pose estimation (visual odometry) addressing the issue of non-differentiability of the cost function.

RGB-D cameras provide a simple and cost effective way to obtain scene information in 3D. A host of 3D-2D, 3D-3D are available. These approaches can broadly be classified into 1) *direct methods*, 2) *feature-based methods* and 3) *Iterative Closest Point (ICP) based methods* .

Next we review the literature only for visual odometry using an RGB-D cameras. The feature based methods involve extraction of salient image features (eg., SIFT, SURF [209] etc). These features are tracked from a reference frame to the current frame to compute a relative camera pose. These methods involve solving the PNP (Perspective N-Point Projection) problem. Finally, bundle adjustment is applied to reduce the drift [207].

Huang *et al.* [80] employed a non-linear least square solver for the minimization of the distance between inlier matched feature points. Dryanovski *et al.* [43] proposed to use a consistent scene model which is dynamically updated upon new observations using a Kalman filter. Due to the pre-selection of feature points (usually 50-500) most of the information from the image is lost. Further, these approaches are ineffective in featureless environments.

Direct methods (sometimes also referred as featureless methods) do not involve the extraction of keypoints. Instead they use the entire image for motion estimation. This enables them to use more of the information from the images to estimate the relative poses. The methods presented by [190], [210], [88] assume photo-consistency of the scene. These approaches involve estimation of a rigid transformation between the previous frame which is densely aligned with the current image. The estimation is performed by minimization of the photo-metric error between the points in the reference image and reprojected points from the current frame. As a result, these approaches are sensitive to illumination variation and noise in intensities. These approaches have a rather small basin of attraction ie., it does not result in good estimates of transform between the two images (or point clouds) when the motion between the two capture locations is large, as has been noted in [88].

Iterative Closest Point (ICP) based methods directly align 3D point clouds. [191] have employed a method based on ICP for the alignment of point clouds obtained from an RGB-D camera. [174] is a survey on some other attempts which use efficient ICP-like

methods for pose estimation. Fitzgibbon [53] has proposed an algorithm to align two 2D point sets. Their algorithm is also extensible to 3D point sets. [53] uses the distance transform to model the point correspondence function to align 2D curves.

3.1.1 Contribution

In this work, we propose a novel, feature-less approach for 3D-2D relative pose estimation[2] . We propose to align a reference image with the current image by minimization of the sum of squared distances between transformed-projected (on current frame) co-ordinates of the edge-pixels from the reference frame and the nearest edge-pixels in the current frame. We make use of only the edge pixels, relying on our observation that the Distance Transform [52] of the edge-map of the current frame can be used to model the distance between edge pixels from the current frame and the re-projections from the reference frame.

Unlike the previous direct approach by Kerl *et al.* [88], our proposed approach does not rely on the photometric cost function and thus on the photo-consistency assumption. The implication of this being, it is robust under changing lightening conditions. Also, being a direct method, does not involve sparse features (like SURF, SIFT etc) and feature matches. Next we make the case for the use of a sub-gradient method (first order method for non-differentiable cost functions), instead of a Gauss-Newton method, for minimization by arguing the non-differentiability of the cost function used in previous dense methods.

We demonstrate that our method has much larger convergence basin (see Fig. 3.1) when compared to previous dense method [88]. We attribute the larger convergence basin of our method to the use of the distance transform to model the reprojected distances. In particular, the distance transform, by its definition extends the influence of edges much farther.

Our methods runs at 30 Hz on a standard PC core and gives comparable relative pose accuracies when evaluated against the method presented by Kerl *et al.* [88]. We evaluate the proposed method with the TUM-RGBD dataset [194].

[2]The code and more results are available on our website : `http://bit.ly/1KNtprg`. Use password '*icra2015*'. Will be public-access after the review period.

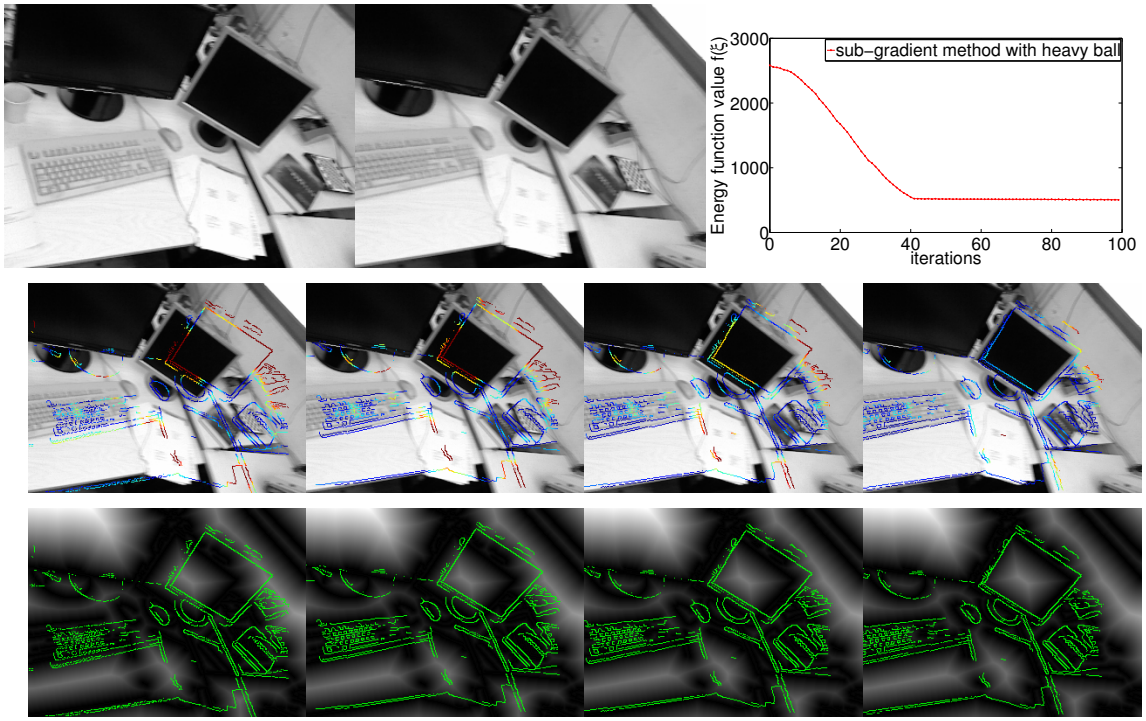

Figure 3.1: Showing reprojections of edge-pixels in the reference frame, onto the current frame as the *sub-gradient* method progresses. The middle row shows the reprojections on the current gray image. They are false colored to represent $v_{e_i}(\xi)$. The last row shows reprojections on the distance transform image of the edge-map of the current frame. Note that the current frame and the reference frame are about 160 ms apart (5 frames) and sub-gradient method progress is shown without pyramids with initial guess as identity. Viewing in color is recommended.

3.2 Direct Edge Alignment (D-EA) Formulation

In this section, we introduce our formulation for relative camera motion estimation using RGB-D camera, which we refer to *D-EA* (*D*irect *E*dge *A*lignment) formulation. It is based on the minimization of geometric error term at each edge pixel to obtain an estimate of the rigid body transform between two frames, ie., to find a pose (rotation and translation matrix) such that the edges of the two images align. This is in contrast to previous direct methods, notably the one proposed by Kerl *et al.* [88] which minimizes the photometric error at every pixel.

The major disadvantage with photometric error minimization approach is that it is very sensitive to noise in the pixel intensities, varying illumination and fast motion. Further, it has a rather small basin of attraction ie. it does not result in good estimates of transform between the two images (or point clouds) when the motion between the two capture locations is large, as has been noted in [88]. Also it assumes photo-consistency which in opinion of authors of this paper is rather strict for deployment in robotics application.

Thus, we propose an energy formulation, which is the sum of squared distances between transformed-projected (on current frame) co-ordinates of the edge-pixels from the reference frame and the nearest edge-pixels in the current frame.

3.2.1 Notations and Conventions

The RGB image collected from RGB-D sensor at timestamp k is denoted as $I_k : \Omega \subset \mathbb{R}^2 \mapsto \mathbb{R}$, where Ω represents the image domain. The depth image at timestamp k is denoted as $Z_k : \Omega \subset \mathbb{R}^2 \mapsto \mathbb{R}$. Let ${}^k\mathbf{P} \in \mathbb{R}^3$ denote a 3D scene point in the co-ordinate system of the camera optical center at time instance k. Since our approach computes the relative pose which are chained to estimate a trajectory, we denote the reference frame (periodically updated) with the script r and the current frame by n. Let ${}^r_n\mathbf{R}$, ${}^r_n\mathbf{T}$ together denote a rigid transformation between the reference and current frames. For convenience of notation we derive our energy formulation using $\mathbf{R}$, $\mathbf{T}$ as the alias to ${}^r_n\mathbf{R}$, ${}^r_n\mathbf{T}$ for a particular instance of reference and current frames.

The camera projection function $\Pi : \mathbb{R}^3 \mapsto \mathbb{R}^2$ projects the visible 3D scene point onto the image domain. The inverse projection function $\tilde{\Pi} : (\mathbb{R}^2, \mathbb{R}) \mapsto \mathbb{R}^3$ back-projects a pixel co-ordinate given the depth at this pixel co-ordinate:

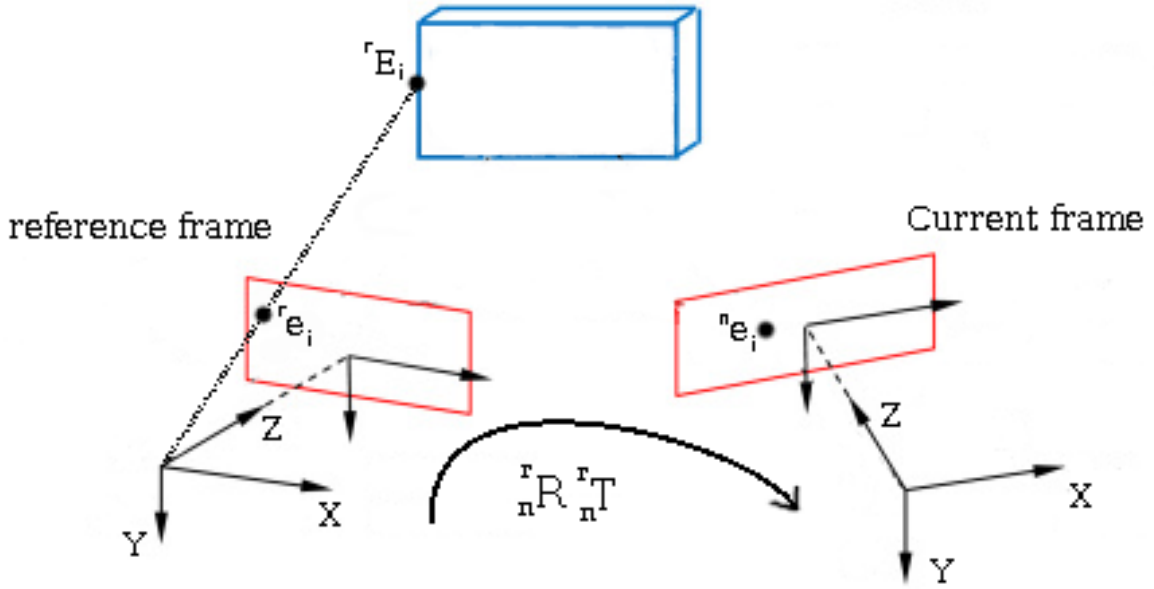

Figure 3.2: Notations and Conventions

$$^{k}\mathbf{u} = \Pi(^{k}\mathbf{P}) \tag{3.1}$$

$$^{k}\mathbf{P} = \tilde{\Pi}(^{k}\mathbf{u}, Z_k(^{k}\mathbf{u})). \tag{3.2}$$

where $^{k}\mathbf{u} \in \mathbb{R}^2$ denotes the image co-ordinates of the 3D point $^{k}\mathbf{P}$.

3.2.2 Relative Motion Estimation

We propose a formulation based on geometric error terms for relative pose estimation of the current frame with respect to a previously set reference frame (the reference frame is periodically updated).

Following our notations, we denote the RGB image and depth map of the reference frame as I_r and Z_r respectively. The i^{th} image point and its corresponding 3D point of the reference image are denoted as $^{r}\mathbf{u}_i$ and $^{r}\mathbf{P}_i$, respectively. Similarly, the RGB image of current frame is denoted by I_n and the i^{th} point of the current frame is denoted as $^{n}\mathbf{u}_i$. Thus, the proposed geometric energy function is the sum of the distances of the re-projections (of edge points from reference image) and nearest edge points in current image:

$$f(\mathbf{R}, \mathbf{T}) = \sum_i \min_j D^2\Big(\Pi[\mathbf{R}^T(^{r}\mathbf{P}_i - \mathbf{T})],\ ^{n}\mathbf{u}_j \Big).$$

where $D : (\mathbb{R}^2, \mathbb{R}^2) \mapsto \mathbb{R}$ denotes the Euclidean distance between those points. The best estimates for the rigid transform can be obtained by solving the following optimization problem.

$$\underset{\mathbf{R},\mathbf{T}}{\text{minimize}}\ f(\mathbf{R}, \mathbf{T})$$
$$\text{subject to } \mathbf{R} \in SO(3)$$

Following the theory of optimization under unitary constraints [133] which proposes to use mapping onto an appropriate manifold at each iteration step. We use the manifold defined by the Lie algebra $se(3)$ corresponding to the Lie group $SE(3)$ which maps the twist coordinates $\xi \in \mathbb{R}^6$ onto the rigid body transform denoted by a rotation matrix $\mathbf{R}$ and translation vector $\mathbf{T}$ using the exponential map (Chp 2. of [149]):

$$\xi = (\mathbf{t}; \mathbf{w})^T \in \mathbb{R}^6.$$

where, $\mathbf{t} \in \mathbb{R}^3$ is the translation component and $\mathbf{w} \in \mathbb{R}^3$ is the rotation component. We denote the rigid body transform on any 3D point in the reference frame, ${}^r\mathbf{P}_i$, corresponding to ξ as $\tau({}^r\mathbf{P}_i, \xi)$:

$$\tau({}^r\mathbf{P}_i, \xi) = \Big[exp(\xi)\Big]^{-1} {}^r\mathbf{P}_i = \mathbf{R}^T({}^r\mathbf{P}_i - \mathbf{T}).$$

Consequently, the constraint optimization formulation can be converted to an unconstrained optimization problem:

$$\underset{\xi}{\text{minimize}} \sum_i \min_j D^2\Big(\ \Pi[\tau({}^r\mathbf{P}_i, \xi)],\ {}^n\mathbf{u}_j\Big).$$

In this approach, we observe that, if the image points corresponding to edge points in the reference image (denoted by ${}^r\mathbf{e}_i \in \mathbb{R}^2$ with corresponding 3D point, ${}^r\mathbf{E}_i$) are pre-selected then the function $\min_j D(\mathbf{u}_i, \mathbf{u}_j)$ is exactly the definition of the Distance Transform [52]. We denote the distance transform of the edge-map of the current image as $V^{(n)} : \mathbb{R}^2 \mapsto \mathbb{R}$. Thus, the energy terms for an edge-pixel of the reference frame is given by:

$$\upsilon_{e_i}(\xi) = V^{(n)}\Big(\ \Pi[\tau(\tilde{\Pi}({}^r\mathbf{e}_i, Z_r({}^r\mathbf{e}_i)), \xi)]\Big)$$

$$f(\xi) = \textstyle\sum_{\forall \mathbf{e}_i} (\upsilon_{e_i}(\xi))^2. \tag{3.3}$$

To summarize, the proposed direct edge alignment (D-EA) formulation is given by

$$\xi^* = \underset{\xi}{\text{argmin}} \sum_{\forall \mathbf{e}_i} (\upsilon_{e_i}(\xi))^2 \tag{3.4}$$

Here, ξ^* denotes the optimal value of $f(\xi)$ for the above stated optimization problem. It has to be noted that ξ^* gives an estimate of the relative transform between the reference frame and the current frame, which can be chained together to obtain a trajectory.

3.3 Solving D-EA with a Sub-gradient Method

In this section we highlight an iterative method to solve the proposed optimization problem (Eq. 3.4). Essentially, we employ a modified *sub-gradient* method for numerical optimization of our optimization problem and provide motivation for the same.

3.3.1 Non-Differentiability & Issues of Gauss-Newton Method

The approach by Kerl *et al.* [88] uses a Gauss-Newton method on the linearized energy function to numerically optimize an energy function with non-linear sum of squared terms :

$$r(\xi, {}^{r}\mathbf{P}_i) = I_n(\Pi[exp(\xi)\ {}^{r}\mathbf{P}_i]) - I_r({}^{r}\mathbf{u}_i)$$

$$\xi^* = \underset{\xi}{\operatorname{argmin}} \sum_i r_{lin}(\xi, {}^{r}\mathbf{P}_i)^2.$$

The major pitfalls of the Gauss-Newton method are that it does not work with non-differentiable functions and that it has no convergence guarantee. Although their method works well in practice, we argue that their energy function is a non-differentiable function of ξ and thus does not satisfy the differentiability requirement of Gauss-Newton methods. From a mathematical stand point the concept of gradient (and Jacobian) do not make sense. Further, the linearization may not track the original non-linear function well as the differential ξ gets a higher value away from iteration's initial estimate. The sub-gradient method is a first order method and is very much like the gradient method but with a few notable differences.

The residue function $r(\xi, {}^{r}\mathbf{P}_i)$ is essentially a composition $I_n \circ \Pi \circ \tau(\xi, {}^{r}\mathbf{P}_i)$. From the theorem on continuity and composition in calculus, which states, if g is continuous at a and f is continuous at $g(a)$, only then $f \circ g$ is continuous at a. That is, $\lim_{x \to a} f(g(x)) = f(g(a))$. We argue that the function I_n being a look-up intensity function on Ω is in general non-continuous everywhere. We further illustrate the nature of I_n using fig. 3.3. We note

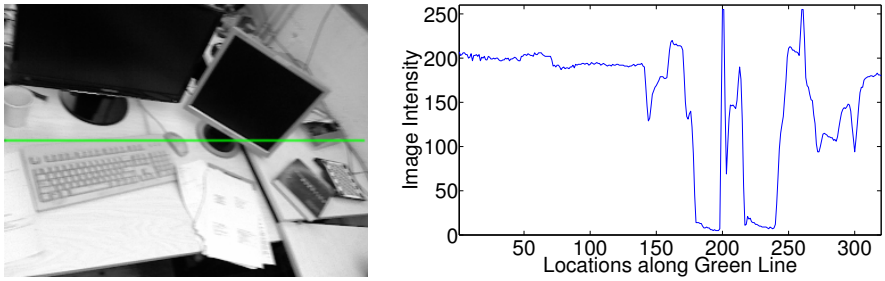

Figure 3.3: Highlighting the non-differentiability of the function I_n at object transition points

that even if one thinks of smooth regions in an image as noise-free at object transition points the function I_n will still be non-differentiable. This validates our argument on non-differentiability of the residue function proposed by Kerl *et al.* [88]. On the contrary, one can argue smoothing I_n can help alleviate the problem of non-differentiability to an extent which in turn can improve the performance of Gauss-Newton iterations, but there is no theoretical guarantee on the convergence.

3.3.2 The Sub-gradient Method

We observe that our energy formulation (Eq. 3.4) is also a non-differentiable function of ξ. We propose to use the *sub-gradient method* which is an algorithm for minimizing a non-differential function. The sub-gradient generalizes the notion of derivatives of a non-differential function. Unlike a gradient of a multivariate function which is a vector (at a particular point), sub-gradient is a set of vectors satisfying an inequality. We refer the reader to Boyd *et al.* [21] for further details on sub-gradient calculus and sub-gradient methods.

Algorithmically, the sub-gradient method is a first order method and is very much like the gradient method but with a few notable differences. For example, the step lengths are not chosen with a line search, instead they are fixed ahead of time. In contrast with the ordinary gradient method, the sub-gradient method is not a descent method and as the iterations progress, the function value can increase.

Similar to a gradient method, the sub-gradient method starts with an initial estimate ($\xi^{(0)}$) and iteratively proceeds in the negative direction of an element in the sub-gradient set (with respect to the twist, ξ).

$$\xi^{(k+1)} = \xi^{(k)} + (-\alpha_k)\ \hbar^{(k)}. \tag{3.5}$$

Here, $\xi^{(k)}$ is the estimate of ξ^* in the k^{th} iteration. $\hbar^{(k)} \in \mathbb{R}^6$ denotes any sub-gradient of f at $\xi^{(k)}$ and $\alpha_k > 0$ is the step size. We use the slashed symbol ($\hbar$) to emphasis yet again the use of sub-gradients. It follows from the convergence proof of the sub-gradient method that, unlike the gradient methods, the step sizes α_k are fixed ahead of time. See section 3.3.4 on choosing step sizes. The operator + denotes combination in the sense of Lie Groups given by

$$\xi_a + \xi_b := log(exp(\xi_a)\ exp(\xi_b)).$$

Also as the sub-gradient method is not a descent method we keep track of the best estimates found so far

$$f_{best}^{(k)} = min(f(\xi^{(0)}), f(\xi^{(1)}), \ldots, f(\xi^{(k)})). \tag{3.6}$$

3.3.3 Computation of a Sub-Gradient

In this section, we derive an expression for a sub-gradient, $\hbar$ with respect to ξ at $\xi = \xi^{(k)}$ of the proposed energy function. By definition, for $\upsilon_{e_i}(\xi)$, defined on the Euclidean set $\mathbb{R}^6$, a vector $\mathbf{c} \in \mathbb{R}^6$ is its sub-gradient if [172]

$$\lim_{\xi \to \xi^{(k)}} \upsilon_{e_i}(\xi) - \upsilon_{e_i}(\xi^{(k)}) \geq \lim_{\xi \to \xi^{(k)}} \mathbf{c} \cdot (\xi - \xi^{(k)}). \tag{3.7}$$

Now, $\upsilon_{e_i}(\xi) = V^{(n)} \circ \Pi \circ \tau(\xi, {}^r\mathbf{P}_i)$. As proved in [172] the chain-rule for differentiable functions is also valid for non-differentiable functions by replacing the sub-gradient in place of gradient. Thus, we write the chain rule for computing the sub-derivate of $\upsilon_{e_i}(\xi)$ with respect to ξ as

$$\mathbf{J}_{e_i} := \frac{\partial V^{(n)}}{\partial e_i} \cdot \frac{\partial e_i}{\partial E_i} \cdot \left.\frac{\partial E_i}{\partial \xi}\right|_{\xi=\xi^{(k)}}. \tag{3.8}$$

where, by our conventions,

$$E_i(\xi^{(k)} + \delta\xi) \approxeq [\hat{\mathbf{R}}(\mathbf{I}_3 + \lfloor \delta\mathbf{w} \rfloor_x)]^T \Big[{}^r\mathbf{E}_i - \hat{\mathbf{T}} - \delta\mathbf{t} \Big].$$

$$\left.\frac{\partial E_i}{\partial \xi}\right|_{\xi=\xi^{(k)}} = \hat{\mathbf{R}}^T \left[\begin{array}{c|c} -\mathbf{I}_3 & \lfloor {}^r\mathbf{E}_i - \hat{\mathbf{T}} \rfloor_x \end{array} \right] \tag{3.9}$$

$\hat{R}$ and $\hat{T}$ represent the rotation and translation matrix of $exp(\xi^{(k)})$ and $\delta\xi = (\delta\mathbf{t}; \delta\mathbf{w})$, ie., the initial estimates at every iteration. The first of the two terms is computed by forward differencing the distance transform image $V^{(n)}$ and the second term is computed from the definition of projection function.

Next we justify why $\mathbf{J}_{e_i}$ using Eq. 3.8 and Eq. 3.9 substituted for $\mathbf{c}$ satisfies Eq. 3.7. Once this is proved, the expression $\mathbf{J}_{e_i}$ will represent a sub-gradient of $\upsilon_{e_i}(\xi)$ evaluated at $\xi = \xi^{(k)}$. We argue that within a sufficiently small hyper-sphere of radius ϵ it is safe to assume that a linear approximation of $\upsilon_{e_i}(\xi)$ closely tracks the original non-linear function $\upsilon_{e_i}(\xi)$. Using the Taylor series expansion, we have

$$\upsilon_{e_i}(\xi) = \upsilon_{e_i}(\xi^{(k)}) + \mathbf{J}_{e_i} \cdot (\xi - \xi^{(k)})$$
$$\text{subject to } ||\xi - \xi^{(k)}|| \leq \epsilon.$$

This is sufficient to conclude that the proposed $\mathbf{J}_{e_i}$ is a sub-gradient of υ_{e_i}, as the above expression trivially satisfies Eq. 3.7. The constraint $||\xi - \xi^{(k)}|| \leq \epsilon$ can be explicitly enforced by use of the *projected sub-gradient* method which is an extension of the sub-gradient method to solve a constrained optimization problem [21]. The projected sub-gradient method works by projecting the updated estimate of ξ ie., $\xi^{(k+1)}$ onto the hyper-sphere [3] [21].

It is relatively straightforward to derive the expression for $\hbar^{(k)}$ (sub-gradient of $f(\xi)$ at $\xi = \xi^{(k)}$):

$$\hbar^{(k)} = \sum\nolimits_{\forall \mathbf{e}_i} 2\ \upsilon_{e_i}(\xi^{(k)})\ \mathbf{J}_{e_i}. \tag{3.10}$$

3.3.4 Analysis on Step Size

Unlike the convergence of the gradient method, which is based on the decrease of the function value at each iteration, the convergence of the sub-gradient method is based on the Euclidean distance to the optimal set. As has been shown by Boyd *et al.* [21]. We present the convergence of a standard sub-gradient method as derived by [21] in this subsection, and in the next subsection we built upon it to derive convergence with the fast convergence strategy.

$$\begin{aligned} ||\xi^{(k+1)} - \xi^*||_2^2 \leq ||\xi^{(k)} - \xi^*||_2^2 - 2\alpha_k(f(\xi^{(k)}) - f^*) \\ + \alpha_k^2 ||\hbar^{(k)}||_2^2. \end{aligned} \tag{3.11}$$

[3] The required projection can be achieved by : $\Gamma(\xi^{(k+1)}) := \frac{\xi^{(k+1)}}{||\xi^{(k+1)}||_2}\epsilon$

Assume that $f(\xi)$ satisfies the Lipschitz condition ie., $|f(\xi_a) - f(\xi_b)| \leq G||\xi_a - \xi_b||_2$. Equivalently bounded sub-gradient, $||\hbar^{(k)}|| \leq G$ (by definition of sub-gradient). Also assume R is an upper bound on the distance of the initial guess to the optimal set ie., $||\xi^{(0)} - \xi^*||_2 \leq R$. One can derive the inequality

$$f_{best}^{(k)} - f^* \leq \frac{R^2 + G^2 \sum_{p=0}^{k} \alpha_p^2}{2 \sum_{p=0}^{k} \alpha_p}. \tag{3.12}$$

Now suppose we set $\alpha_p := \frac{\eta}{(p+1)}$; $p = 0, \cdots, k$ and η is a constant. As $k \to \infty$, $\sum_{p=0}^{k} \alpha_p^2 < \infty$ and $\sum_{p=0}^{k} \alpha_p = \infty$. This leads to convergence of the sub-gradient method under the square summable but not summable step sizes. Thus we use those step sizes:

$$\lim_{k \to \infty} f_{best}^{(k)} = f^*.$$

3.3.5 Fast Convergence Strategies

Although the sub-gradient methods are guaranteed to converge, their convergence speed is rather slow. However, each iteration is of very low complexity as it avoids forming and solving the normal equations.

A general approach to speed up the convergence of a gradient method is to use a class of methods called *heavy ball methods*. For review of literature we direct the reader to the works of Nesterov [151, 152]. Although there are a few variants of the heavy ball methods, we propose to use the update direction as a conic combination ($\beta \in (0, 1)$) of the sub-gradient in current iteration ($\hbar^{(k)}$) and previous iterations. Thus, we use the update direction $\mathbf{s}^{(k)}$ instead of $\hbar^{(k)}$.

$$\mathbf{s}^{(k)} = (1 - \beta)\hbar^{(k)} + \beta \mathbf{s}^{(k-1)}. \tag{3.13}$$

$$\mathbf{s}^{(k)} = (1 - \beta)\left(\sum_{i=1}^{k} \beta^{k-i} \hbar^{(i)} \right) \tag{3.14}$$

Eq. 3.14 can be derived by recursively applying Eq. 3.13. Following the logic from the previous sub-section, the convergence of the modified sub-gradient method is analyzed by the distance of the iterate to the optimal set ($||\xi^{(k+1)} - \xi^*||_2^2$). We obtain the basic inequality ($\mu^{(k)} := \frac{||\mathbf{s}^{(k)}||_2^2}{||\hbar^{(k)}||_2^2}$),

$$f_{best} - f(\xi^*) \leq \frac{R^2 + G^2 \sum_{p=0}^{k} \alpha_p^2 (1 - \beta^p)^2}{2 \sum_{p=0}^{k} \alpha_p \mu^{(p)}} \tag{3.15}$$

We can conclude from this inequality that the iterate will converge towards the optimal value when step sizes (α_p) are square summable but not summable. However, the theoretical analysis on the convergence speed compared to the simple sub-gradient method is largely inconclusive. However, in section 3.5 we experimentally show the effectiveness of the use of the heavy ball scheme with the sub-gradient method for our problem.

3.4 Implementation

We summarize the proposed methodology for numerical minimization of Eq. 3.4 in the listing Algorithm 1. The sub-routine *EdgeMap* returns a list of co-ordinates which are edge pixels in the provided image. We use Canny Edge detection for computing the edge map of the image. The sub-routine *DistanceTransform* computes the distance transform of the input edge-map as proposed in [52]. Whereas, the sub-routine *InverseProjectAllEdgePixels* implements Eq. 3.2.

We update the reference frame periodically (typically every 5-10 frames) and compute the relative poses of subsequent frames as outlined in Algorithm 1. These relative poses can be cumulated to obtain the odometry.

As illustrated in the results section, although the basin of convergence of the proposed algorithm is already much larger than the algorithms based on photometric error minimization, we implement our algorithm with image pyramids. We reason that the use of pyramids gives a better initial guess ($\xi^{(0)}$) for the top most level which helps the proposed method converge faster to produce precise results.

As has been observed by Kerl [88] weighting down large residues can help alleviate the effect of outliers arising due to reflections, occlusions, edge-map misses on the computation of update direction. We use a Laplacian weighting term given by, $W(\upsilon_{e_i}(\xi)) = e^{-\upsilon_{e_i}(\xi)}$.

This solves a weighted least square problem whose sub-gradient can be computed as

$$\hbar^{(k)} = \sum\nolimits_{\forall \mathbf{e}_i} 2W(\upsilon_{e_i}(\xi))\upsilon_{e_i}(\xi^{(k)})\ \mathbf{J}_{e_i}. \tag{3.16}$$

3.5 Results

In this section we perform a series of experiments demonstrating the effectiveness of the proposed formulation for camera pose estimation. We compare our method with another

Algorithm 1 RelativePoseEstimation($I_n, I_r, Z_r, \xi^{(0)}$)

E_n = EdgeMap(I_n)

V_n = DistanceTransform(E_n)

E_r = EdgeMap(I_r)

${}^r\mathbf{P_i}$ = InverseProjectAllEdgePixels(E_r, Z_r)

$k = 1$

$\hbar^{(0)} = 0$

for $k : 1 \rightarrow M$ **do**

  $\hbar^{(k)}$ = GetSubGradient($V_n, {}^r\mathbf{P_i} \forall\ \mathbf{i}, \xi^{(\mathbf{k-1})}$) (Eq. 3.8, 3.10, 3.16)

  $\mathbf{s^{(k)}} = (1 - \beta)\hbar^{(k)} + \beta\mathbf{s^{(k-1)}}$

  $\Delta\xi = \Gamma(-\alpha_k\mathbf{s^{(k)}}\ )$

  if $||\Delta\xi||_2 < \Delta$ **then**

    break

  else

    $\xi^{(k)} = \xi^{(k)} + \Delta\xi$

    $f(\xi^{(k)}) = GetFunctionValue(V_n, {}^r\mathbf{P_i} \forall \mathbf{i}, \xi^{\mathbf{k}})$ (Eq. 3.3)

  end if

end for

return $f_{best}^{(M)} = min(f(\xi^{(0)}), f(\xi^{(1)}), \ldots, f(\xi^{(M)}))$

direct method on RGB-D data by Kerl *et al.* [88] showing the robustness of our method in the presence of large camera motions and changing illumination. We obtained the implementation of [88] from their website and got it working on our PC for evaluation. We used their weighted configuration in realtime parameter setting, which could run in around 20 ms per frame on average (enough for 30 Hz frame rate) at a resolution 320×240 pixels with 6 pyramidal levels. Our method also used the same base resolution with 4 pyramidal levels.

We evaluate our method using the TUM-RGBD dataset [194] which provides RGB-D data for various test cases along with the ground truth obtained from the motion capture system. we conduct our experiments on a PC with Intel Core i7-2600 CPU (3.4 Ghz) with 16 GB of RAM. We also note that our core system, which evaluates the relative position between a pair of images, is single threaded.

3.5.1 Relative Pose Error (RPE)

Strum *et al.* [194] proposed RPE to measure the local accuracy for visual odometry approaches, which they defined as,

$$\mathbf{E}_i = \left(\mathbf{Q_i^{-1}Q_{i+\delta}}\right)^{-1}\left(\mathbf{B_i^{-1}B_{i+\delta}}\right)$$

Here, $\mathbf{Q_1}, \cdots, \mathbf{Q_n} \in \mathbf{SE(3)}$ is the sequence of GT poses and $\mathbf{B_1}, \cdots, \mathbf{B_n} \in \mathbf{SE(3)}$ is the sequence of estimated poses indexed by time instances. δ is the relative time step. They then proposed to evaluate such statistical measures as RMSE, mean, etc. of the translation component of the sequence $\mathbf{E_1}, \cdots, \mathbf{E_{n-\delta}}$. We report the RMSE values for various values of δ in Table 3.1. We also show the relative pose error (translation component) at each frame for the sequence 'fr1/desk' in Fig. 3.4.

The first three sequences are the sequences for which Kerl *et al.* [88] provide results. These sequences are characterized by rather slower motion without sudden jerks. As a result, the consecutive frames (at 30 Hz) are very near to each other and the photometric error based method. Our method is comparable to [88] for these three sequences.

The next two sequences contain gaps of about 1 second each while in progress. We suspect these might be due to buffering issues in the driver. In real scenarios this might happen.

The next two sequences contain moving objects. Our method and [88] are comparable

Sequence	D-EA		Kerl *et al.* [88]	
	$\delta = 1$	$\delta = 20$	$\delta = 1$	$\delta = 20$
fr2/desk	0.0324	0.1529	0.0333	0.2217
fr1/desk	0.0289	0.0948	0.0346	0.4286
fr1/desk2	0.0335	0.1818	0.0343	0.3658
fr1/floor	0.0355	0.1988	0.0330	0.3380
fr1/room	0.0353	0.2514	0.0307	0.3399
fr2/desk_with_person	0.0125	0.0594	0.0137	0.1516
fr3/sitting_halfsphere	0.0208	0.1462	0.0181	0.2599
fr2/pioneer_slam2	0.0593	0.4447	0.0847	0.4707

Table 3.1: RMSE values of the Relative Pose Errors for various sequences.

in performance for these sequences. We attribute it to the exponential weighting terms in the energy formulation.

In the last sequence the RGB-D camera is placed on a pioneer ground robot and piloted manually. This sequence contains jerks in the motion. Thus the difference between two consecutive frames is quite large at times. Our method clearly outperforms [88] on this sequence.

3.5.2 Effect of Frame Skipping

Next we show the robustness of our method for large motion. In this experiment, we supplied our method as well as the method by Kerl *et al.*[88] with alternate frames of the sequence ie., frame $0, 2, 4, \cdots$ and so on. We also did similar experiments by skipping 3 and 4 frames as well. We plot the translation component of the relative pose error at each frame in the sequence 'fr1/desk' (20.24 sec, or 593 frames). See Fig. 3.8. We observe that the relative pose error for our method does not vary much even when supplied 1 frame of every 4 frames, whereas the relative pose error goes on increasing for [88]. One can think of this property of our method as being robust to fast motion. Note that the run-time configuration was kept the same for all experiments in this subsection.

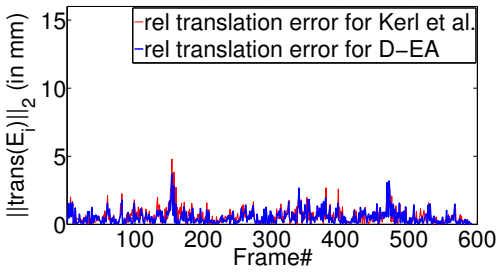

Figure 3.4: Translation component of relative pose error at each frame for the sequence 'fr1/desk'. Best viewed in color.

3.5.3 Demonstration of Large Convergence Basin

We select two frames from the *fr1/rpy* sequence which are about 160 ms apart (5 frames) and show the progress of the sub-gradient method iteration wise in Fig. 3.1. In the illustration there are no pyramids in use and the initial estimate ($\xi^{(0)}$) for sub-gradient method was set as identity. We can clearly see the influence of edges extend which results in a much larger convergence basin for the proposed method.

3.5.4 Effect of Heavy Ball

We demonstrate the effectiveness of the altered gradient direction (heavy ball method). We compare the energy progress for the example reference-current frames shown in Fig. 3.1. The comparison of the simple sub-gradient method and the modified sub-gradient method for number of iterations to convergence is shown in Fig. 3.9

3.6 Conclusion

In this paper, we proposed a direct (feature-less) approach for visual 6-DOF pose estimation with the energy function based on the distance transform applied to an RGB-D camera. We demonstrate with experiments a much larger convergence basin for our method when compared to other direct approaches. We address the issue of non-differentiability of the energy function and thus make the case against the use of Gauss-Newton methods for non-differentiable functions in a strict mathematical sense. Thus, we utilize the sub-gradient method for numerical optimization, which is a class of methods to handle non-differentiable functions. We also analyze the convergence of the modified sub-gradient method (that we use) for our energy function. Our method is comparable to previous

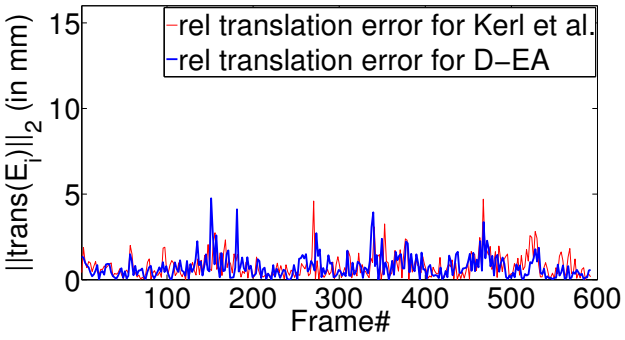

Figure 3.5: Processing only frames $0, 2, 4, 6 \cdots$

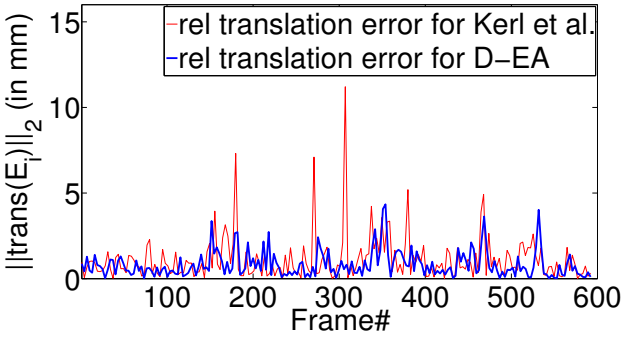

Figure 3.6: Processing only frames $0, 3, 6, 9 \cdots$

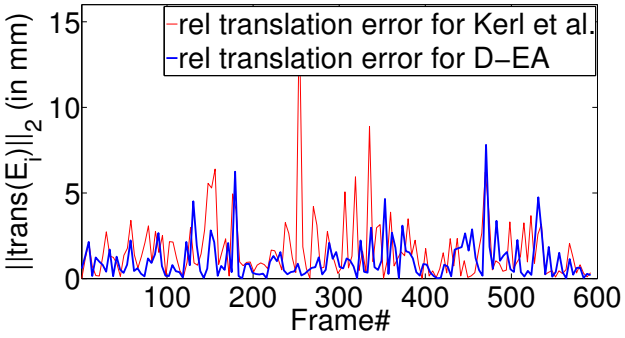

Figure 3.7: Processing only frames $0, 4, 8, 12 \cdots$

Figure 3.8: Robustness for large motions. Relative pose estimation of 'fr1/desk' by skipping frames. Best viewed in color.

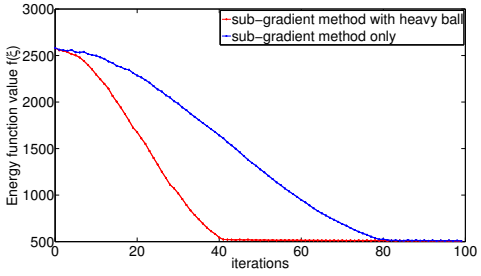

Figure 3.9: Comparison of the sub-gradient method and sub-gradient method with heavy ball acceleration on the frame pair of Fig. 3.1.

method ([88]) for sequences with slower motion and we show with experiments the robustness of our method for sequences with fast motion. This robustness is attributed to our energy function which is based on the minimization of reprojected distances rather than on photo-consistency.

Chapter 4

Place Recognition Front-end

We propose to learn the whole-image-descriptor in a weakly supervised manner based on NetVLAD and decoupled convolutions. We analyze the training difficulties in using standard loss formulations and propose an allpairloss and show its effect through extensive experiments. Compared to standard NetVLAD, our network takes an order of magnitude fewer computations and model parameters, as a result runs about three times faster. We evaluate the representation power of our descriptor on standard datasets with precision-recall. Unlike previous loop detection methods which have been evaluated only on fronto-parallel revisits, we evaluate the performance of our method with competing methods on scenarios involving large viewpoint difference. Our implementation for learning the whole-image descriptor is opensources [1] .

4.1 Introduction

Over the past decade, the SLAM (Simultaneous Localization and Mapping) community has made amazing progress towards increasing the specificity of the odometer and building usable maps of the environment to assist robots in various planning tasks. Systems using visual and inertial information fusion have been a contemporary theme towards reducing drift to less than 0.5% of the trajectory length [28]. Identifying a revisit to a place presents an opportunity to reduce the drift further and also to recover from kidnap scenarios. General place recognition, however, remains an extremely challenging problem [127] due to myriad ways in which visual appearance of a place varies. In our own daily experience, humans describe places to fellow humans as a collection of objects, their color cues, their spatial locations and so on, thereby allowing to disambiguate places even if they

[1] `https://github.com/mpkuse/cartwheel_train`

approach the place from a very different viewpoint. Ideally, the loop detection module should describe a scene in this context. Humans probably do not rely on corner features (a common technique in use for loop detection in existing SLAM systems) to identify a place, instead we humans, most likely represent the scene as a whole in a semantic sense. The proposed system builds on this motivation.

In this work, we propose the use of a framework which learns whole-image descriptors without explicit human labeling to represent a scene in a high dimensional subspace for detecting place revisits. We lay special emphasis on the real-time performance and evaluation of the system in context of visual-SLAM.

Popular past works have considered loopclosure under fronto-parallel scenarios. However, place revisits can happen at substantial viewpoint difference. The underlying place recognition module in SLAM systems to identify place revisits occurring at widely different viewpoints. Past systems based on bag-of-visual-words (BOVW) are limited by the underlying low-level feature descriptors. The learned vocabulary (for BOVW) also have difficulty generalizing under adversaries like large viewpoint difference, noise, low light, changing exposure, less texture [127]. The proposed method can learn a representation that generalizes well and can identify place revisits under non fronto-parallel viewpoints.

We compare our method's run-time performance with a popular bag-of-words approach, DBOW [57] and ibow-lcd by [60] along with recently proposed CNN based approaches for place recognition [199], [136], [123]. On real sequences, our method delivers a similar recognition performance to NetVLAD but at a 3X lower computational time and an order of magnitude fewer training variables. The major advantage of our system is that it has a high place recall rate which makes it effective in detecting loopclosures occurring under larger viewpoint differences.

This chapter is organized as follows. In Section 4.2, we start by reviewing approaches in the Visual Place Recognition (VPR) community and some recent loopclosure methods used in Visual-SLAM community. Next, in Section 4.3, we identify the issue of unstable learning in the original NetVLAD implementation which uses the tripletloss and we propose an allpairloss function to alleviate this issue. In Section 4.4.1, we present experimental evidence to the claim of unstable learning of NetVLAD using the tripletloss and how it was alleviated using the proposed allpairloss function. Unlike other papers in VPR we evaluate our method in context of SLAM using a manually marked loopclosure dataset

in addition to the evaluation on standard datasets from place recognition community.

4.2 Literature Review

We recognize that visual place recognition (VPR) and loopclosure detection in SLAM are related problems. Here we first review recent advances from VPR community and then review state-of-the-art loop-closure methods.

In the context of VPR, Sunderhauf *et al.* [199] pioneered the use of ConvNet features. Compared to SeqSLAM [140] and FAB-MAP [40, 41] use of features from pre-trained network results in better precision-recall performances on standard VPR datasets(Norland, Gardens Point, St. Lucia and Campus). In their subsequent work, Sunderhauf *et al.* [200] proposed to use region proposals and extract ConvNet features on each of the regions. Arandjelovic *et al.* [5] proposed a trainable feature aggregation layer which mimics the popular VLAD (Vector of Locally Aggregated Descriptor). While impressive performance was obtained, these methods rely on nearest neighbour search for retrieval. The image descriptor being very high dimensional (eg. 32K dimensional for [5], 64K for [199]), these methods perform various dimensionality reduction techniques to make nearest neighbour search feasible in reasonable time with some hit to the retrieval performance. WPCA was used by [5] which involve storage of a 32Kx4K matrix costing about 400 MB.

More recently Khaliq *et al.* [89] proposed an approach which make use of region-based features from a light-weight CNN architecture and combines them with VLAD aggregation. The approach from Chen *et al.* [34, 35] identifies key landmark regions directly from responses of VGG16 network which was pre-trained on image classification task. For regional features encoding, bag-of-words was employed on a separate training dataset to learn the code-book. The approach by Hou *et al.* [77] is very similar to [34]. The Disadvantage of using pre-trained models learned on ImageNet object classification, for example, puts more emphasis on objects rather than the nature of the scene itself. Other works in this context include [7, 10, 11, 31, 58, 124, 177, 205]. For a more detailed summary of the works in place recognition we direct the readers to survey on place recognition / instance retrieval [127, 227]. We summarize the literature in Table 4.1.

Although CNN based techniques are considered as state-of-the-art in retrieval and

place recognition tasks, they are still disconnected from overall SLAM and loop-closure detection problems. Commonly employed loop detection methods in state-of-the art SLAM systems rely on sparse point feature descriptors like SIFT, SURF, ORB, BRIEF etc. for representation and an adaptation of BoVW for retrieval. While BoVW provides for an scalable indexed retrieval, the performance of the system is limited by the underlying image representation. Such factors as the quantization in clustering when building vocabulary, occlusions, image noise, repeated structures also affect the retrieval performance.

FAB-MAP [40, 41], DBOW2 [57] and others [15, 148] rely on a visual vocabulary which is trained offline, while recent methods like OVV [153] , IBuILD[90], iBOW-LCD [60], RTAB-MAP [101] and others [4, 59, 193, 225] rely on online constructed visual vocabulary. Authors have also made use of whole-image-descriptors in loopclosure context [140, 161, 226]. Works in the context of loopclosures in SLAM which make use of learned feature descriptors are: [58, 136]. Merrill and Huang [136] learned an auto-encoder from the common HOG descriptors for the whole image. Other miscellaneous work related to our localization system are [36, 37, 51, 87, 188].

We summarize our contributions:

- A novel cost function which deals with the gradient issue observed in standard NetVLAD training.
- Decoupled convolutions instead of standard convolutions result in similar performance on precision-recall basis but at a 3X lower computation cost and about 5-7X fewer learnable parameters, making it ideally suited for real-time loopclosure problems.
- Squashing channels of CNN descriptors instead of explicit dimensionality reduction of image descriptor for scalability. Even a 512-D image descriptor gives reasonable performance.

4.3 Whole Image Descriptor Learning

In this section, we describe the training procedure. We start by reviewing VLAD and NetVLAD [5] (Sec. 4.3.1). We see these methods as a way for pixel-wise feature-map aggregation. Next we describe the learning issues associated with the triplet ranking loss

Representation	Retrieval	Method Description
SPF-real	BOW	[40, 41] soft-real-time (can run @5-10hz)
SPF-binary	BOW	[15, 57, 148] real-time (10hz or more)
SPF-real	Inc-BOW	[4, 101, 153] soft-real-time (1-5 Hz). [16, 208] make use of temporal information to form visual words scene representation.
SPF-binary	Inc-BOW	[59, 60, 90, 225] soft-real-time to 1-5Hz processing
SPF	graph	[192]
Pretrained-CNN	NN	[8, 11, 199] provide for real-time descriptor computation (10-15hz). dimensionality reduction accomplished at 5hz, NN with 64K dim is really slow, NN after dimensionality reduction (4000d) is about 5-15 hz.
Pretrained-CNN	BOW	[34, 35, 77]
Custom-CNN	NN	[5, 31, 124, 205] provides for real-time descriptor computation. dim-reduction and NN search are bottle necks.
Custom-CNN with region-proposals	regionwise-NN	[200] very slow representation vector computation. [89] region descriptor encoding computation 2-3 Hz. Reported matching times is several seconds.
Unsupervised Learning	NN	[58, 123, 136] descriptors are not descriptive enough after dim-reduction. real-time desc computation.
Intensity agnostic	NN optimization	[140, 161] [68, 105, 226] generally slow.

Table 4.1: SPF=Sparse Point Features. BOW=Bag-of-words.

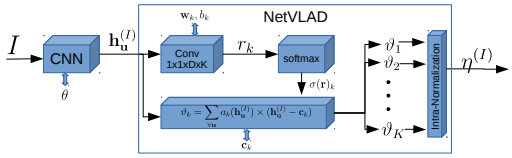

Figure 4.1: Notations and computations for the whole-image descriptor. An image is fed into the CNN followed by the NetVLAD layer. We experiment with VGG16 and propose to use decoupled convolution for its speed. Additionally for dimensionality reduction we propose channel-squashing. Our fully convolutional network, with K=16 produces a 4096-dimensional image descriptor (without channel squashing) and a 512-dimensional image descriptor (with channel squashing). In terms of number of floating point operations (FLOPs) for a 640x480 input image our proposed network is about 25X faster, real computational time is about 3X faster. Details in Sec. 4.3.

function. To mitigate this issue we propose to use a novel all-pair loss function (Sec. 4.3.2). We provide an intuitive explanation along with experimental evidence on why the proposed all-pair loss function leads to faster and stable training.

4.3.1 Review of VLAD and NetVLAD

Let $\mathbf{h}_{\mathbf{u}}^{(I)}(d)$ be the d^{th} dimension ($d = 1 \cdots D$) of the image feature descriptors for image I (width W and height H) at pixel $\mathbf{u} := (i, j), i = 1, \ldots W'; j = 1, \ldots H'$. These per pixel CNN-feature descriptors are assigned to one of the K clusters (K is a fixed parameter, we used 16, 48, 64 in our experiments), with $\mathbf{c}_k \in \Re^D, k = 1 \ldots K$ as cluster centers. The VLAD [5] representation is $D \times K$ matrix, $\mathbf{V} = [\vartheta_1, \vartheta_2, \ldots, \vartheta_K]$, defined as the sum of difference between local descriptor and assigned cluster center,

$$\vartheta_k = \sum_{\forall \mathbf{u}} a_k(\mathbf{h}_{\mathbf{u}}^{(I)}) \times (\mathbf{h}_{\mathbf{u}}^{(I)} - \mathbf{c}_k) \tag{4.1}$$

where $a_k(.)$ denotes a scalar membership indicator function of the descriptor $\mathbf{h}_{\mathbf{u}}^{(I)}$ in one of the K classes. Arandjelovic *et al.* [5] proposed to mimic VLAD in a CNN-based framework. In order that the cluster assignment function, $a_k(.)$, be differentiable and hence learnable with back propagation they defined an approximation of the assignment function $a_k(.)$ using the softmax function. For brevity, we write, $\mathbf{h}_{\mathbf{u}}^{(I)}$ as $\mathbf{h}$:

$$\begin{aligned} \hat{a}_k(\mathbf{h}) &= \frac{e^{-\alpha \ ||\mathbf{h}-\mathbf{c}_k||}}{\sum_{k'=1}^{K} e^{-\alpha \ ||\mathbf{h}-\mathbf{c}_{k'}||}} \\ &= \sigma(\mathbf{r})_k \end{aligned} \tag{4.2}$$

where $r_k = \mathbf{w}_k^T \mathbf{h} + b_k$, $\mathbf{w}_k = 2\alpha \mathbf{c}_k$, $b_k = -\alpha ||\mathbf{c}_k||^2$. $\sigma(\mathbf{r})_k$ is the softmax function. r_k can be computed with convolutions. $\mathbf{w}_k$, b_k and $\mathbf{c}_k$ are learnable parameters in addition to the CNN parameters θ. Fig. 4.1 summarizes the computations and notations. Each of the vectors corresponding to K clusters is individually unit normalized and then the whole vector is unit normalized. This is referred in the literature as Intra-normalization which reduce the effect of burstiness of visual features[83]. Thus, scene descriptor $\eta^{(I)}$, of size $D * K$ is produced using a CNN and the NetVLAD layer,

$$\eta^{(I)} = \aleph(\{\mathbf{h}_\mathbf{u}^{(I)}\}) \tag{4.3}$$

In general any base CNN can be used and the NetVLAD mechanism can be thought of aggregating the CNN pixel-wise descriptors. For experiments, we use the VGG16 network [186]. Additionally, following [79] we propose to use the decoupled convolution, ie. a convolution layer is split into two layers, the first of which does only spatial convolution independently across all the input channels. Second of the two layers, does, 1x1 convolution on the channels. This has been found to boost running time at marginal loss of accuracy for object categorization tasks. We also propose to reduce the dimensions by quashing channels with learned 1x1 convolutions rather than reduce the dimensions of the image descriptor as has been done by the original NetVLAD paper. This eliminates the need to store the whitening matrix (as done by [5]). At the run time it eliminates the need for a large matrix-vector multiplication for dimensionality reduction.

4.3.2 Proposed All-Pair Loss Function

To learn the parameters of the CNN (θ) and of the NetVLAD layer ($\mathbf{w}_k$, b_k and $\mathbf{c}_k$), the cost function needs to be designed such that, in the dot product space, η corresponding to projections of the same scene (under different viewpoints) appear as nearby points (higher dot product value, nearer to 1.0). Let $\eta^{(I_q)}$ be the descriptor of the query image I_q. Similarly, let $\eta^{(P_i)}$ and $\eta^{(N_j)}$ be the descriptors of i^{th} positive and j^{th} negative sample respectively. By positive sample, we refer to a scene which is same as query image scene but imaged from a different perspective. By negative sample, we refer to a scene which is not the same place as the query image. Let the notation, $\langle \eta^{(a)}, \eta^{(b)} \rangle$, denote the dot product of two vectors.

Following [5] we use multiple positive and negative samples $\{I_q, \{P_i\}_{i=1,\ldots,m}, \{N_j\}_{j=1,\ldots,n}\}$

per training sample, however with a novel all-pair loss function. We provide an intuitive explanation for the superiority of the proposed loss function over the standard triplet loss used by [5] for training. We also provide corroborative experimental evidence towards our claims. The commonly used triplet loss function can be rewritten in our notations as, $L_{triplet-loss}$:

$$\sum_{j} \max\Big(0, \langle \eta^{(I_q)}, \eta^{(N_j)} \rangle - \min_i(\langle \eta^{(I_q)}, \eta^{(P_i)} \rangle) + \epsilon\Big) \tag{4.4}$$

where ϵ is a constant margin. Note that [5] preferred to define the loss function in Euclidean space rather than the dot product space. Using any of the spaces is equivalent since for unit vectors, $\mathbf{a}$ and $\mathbf{b}$, the dot product, $\langle \mathbf{a}, \mathbf{b} \rangle$, and the squared Euclidean distance $d(\mathbf{a}, \mathbf{b})$, are related as, $d(a, b) = 2(1 - \langle a, b \rangle)$ with a negative correlation. This has been taken care of by flipped sign in our optimization problem compared to the one used by Arandjelovic *et al.* [5]. This loss function is the difference between the worst positive sample, ie. $\min_i \langle \eta^{(I_q)}, \eta^{(P_i)} \rangle$ and the query with every negative sample.

In an independent study by Bengio *et al.* [17], it was observed that for faster convergence, it is crucial to select triplets from the training dataset, that violate the triplet constraint, ie. result in as few zero-loss as possible. They demonstrated that these zero-loss scenarios lead to zero gradients which in turn slows the training. They suggested to provide easier samples in early iterations and harder samples in later iteration to speed up the learning process. To this effect, Schroff *et al.* [182] proposed a strategy to select triplets using recent network checkpoints, every n (say 1000) training iterations. Instead of using a complicated strategy as done by [182], we rely on a well-designed loss function which gives this effect. Thus, we define a novel loss function based on all-pair comparisons of positive and negative samples with the query image. The proposed loss function is relatively harder to satisfy (resulting in fewer zero loss samples), hence its higher discriminatory power compared to the triplet loss (see Fig. 4.3).

In order to learn highly discriminative descriptors, we want the similarity of query sample ie. $\eta^{(I_q)}$ with positive samples be more than the similarity between query sample and the negative samples. Let us consider two cases a) $\langle \eta^{(I_q)}, \eta^{(N_j)} \rangle > \langle \eta^{(I_q)}, \eta^{(P_i)} \rangle$; b) $\langle \eta^{(I_q)}, \eta^{(N_j)} \rangle < \langle \eta^{(I_q)}, \eta^{(P_i)} \rangle$. Case-b is what we prefer so we do not want to have a penalty (want to have zero loss) for its occurrence. Case-a is opposite of what we prefer thus we add a penalty proportional to the magnitude of the dot product to discourage

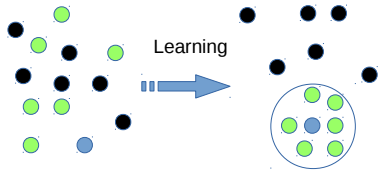

Figure 4.2: Illustration of the effect of learning with proposed loss function. Descriptor of query image ($\eta^{(I_q)}$, in blue). Descriptors of positive set ($\eta^{(P_i)}$, in green) and negative set ($\eta^{(N_j)}$, in black). See also Eq. 4.5.

this event. We propose to add a penalty term as above for all the pairs where the conditions do not hold. For effective learning we propose to compute the loss over every pair of positive and negative sample. The final loss function for one learning sample ($\{I_q, \{P_i\}_{i=1,\ldots,m}, \{N_j\}_{j=1,\ldots,n}\}$) is given as, $L_{proposed}$:

$$L = \sum_{i=1}^{m}\sum_{j=1}^{n} \max(0, \langle \eta^{(I_q)}, \eta^{(N_j)} \rangle - \langle \eta^{(I_q)}, \eta^{(P_i)} \rangle + \epsilon) \tag{4.5}$$

The motivation and the effect of the proposed loss function on termination of learning is summarized in Fig. 4.2.

We further note that the proposed loss function is harder to satisfy (giving a positive penalty) compared to the loss function used by Arandjelovic *et al.*[5] (ie. Eq. 4.4). This has been experimentally observed (see Fig. 4.3). The major drawback of using an easy to satisfy penalty function is the vanishing gradients problem [17], which slows the speed of learning. This is because a zero-loss sample results in zero-gradient during back-propagation.

Loss function in Matrix Notation

For fast and efficient training we express the loss function (Eq. 4.5) in a matrix notation. We firstly define $\Delta^q_{\mathbf{P}}$ to represent dot product between the sample query η_q and descriptors of each of the positive samples

$$\Delta^q_{\mathbf{P}} = \begin{bmatrix} \langle \eta^{(I_q)}, \eta^{(P_1)} \rangle \\ \vdots \\ \langle \eta^{(I_q)}, \eta^{(P_m)} \rangle \end{bmatrix} \tag{4.6}$$

Next, we define $\Delta^q_{\mathbf{N}}$ to represent dot product between the sample query and descriptors

of each of the negative samples.

$$\Delta^q_{\mathbf{N}} = \begin{bmatrix} \langle \eta^{(I_q)}, \eta^{(N_1)} \rangle \\ \vdots \\ \langle \eta^{(I_q)}, \eta^{(N_n)} \rangle \end{bmatrix} \tag{4.7}$$

Let $\mathbf{1}_n$ denote a column-vector of size n and $\mathbf{1}_m$ denote a column-vector of size m with all entries as 1s. Also define $\mathbf{0}_{m \times n}$ as null-matrix of dimensions $m \times n$. The max(.) operator is a point-wise operator. Now we note that Eq. 4.5 can be expressed in matrix notation as:

$$\mathbf{L} = \max\left(\mathbf{0}_{m \times n}, \mathbf{1}_m (\Delta^q_{\mathbf{N}})^T - \Delta^q_{\mathbf{P}} \mathbf{1}_n^T + \epsilon \mathbf{1}_m \mathbf{1}_n^T\right) \tag{4.8}$$

4.3.3 Training Data

In order to train the scene descriptor, only requirement on the data is that we be able to draw positive sample images (views of the same physical scenes) and negative sample images (images of different scenes). One possible way is to bootstrap a video sequence with existing methods for loopclosure detection. Such a preprocessed sequence might not be useful for localization but can provide enough information to draw positive and negative samples for learning a whole-image-descriptor with the proposed method. Several walking, driving, drone videos etc. available on video sharing websites can be used for learning. Another way could be to use 3D mesh-models and render views with nearby virtual-camera locations to obtain positive samples. With the advent of crowd sourcing street scenes and availability of services like mappilary[2] , it is easily possible to assemble a much larger dataset for training. Faster and descriminative learning is even more crucial when making use of such larger training datasets. We differ this until our future work.

For our experiments be comparable with existing methods, we use the Pittsburgh (Pitts250k) [206] dataset which contains 250k images from Google's street-view engine. It provides multiple street-level panoramic images for about 5000 unique locations in Pittsburgh, Pennsylvania over several years. Multiple panoramas are available at a particular place ($\approx$ 10m apart) sampled approximately 30 degrees apart along the azimuth. Another similar dataset is the TokyoTM dataset [206].

[2]https://www.mapillary.com/

4.3.4 Training Hyperparameters

The CNN-learnable parameters are initialized with the Xavier initialization [61]. We initialize NetVLAD parameter $\mathbf{c}_k$ as unit vectors drawn randomly from a surface of a hyper-sphere. b_k and $\mathbf{w}_k$ are coupled with $\mathbf{c}_k$ at initialization. However, as learning progresses these variables are decoupled. We use the AdaDelta solver [224] with batch size of $b = 4$ (each batch with $m = 6$ positive samples and $n = 6$ negative samples)). This configuration takes about 9GB of GPU memory during training. We stop the training at 1200 epochs. Our 1 epoch is 500 randomly drawn tuples from the entire dataset. The learning rate is reduced by a factor of 0.7 if loss function does not decrease in 50 epochs and a regularization constant is set to 0.001 (to make regularization loss about 1% of fitting loss). Data augmentation (rotation, affine scale, random cropping, random intensity variation) is used for robust learning, which we begin after 400 epochs. This explains the rise in the loss function values in our experiments in Fig. 4.5, 4.6, 4.7.

The output descriptor size is $K \times D$. K is the number of clusters in NetVLAD, we use K=16,32,64. D is the number of channels of the output CNN. For both VGG16 and the decoupled network it is 512. Some other authors notably Arandjejovic [5] and Sunderhauf [199] have used whitening-PCA and gaussian random projections respectively to reduce the dimensions of the image descriptor from about 64K to 4K. We suggest to use learnable squashing channels (to say 32) with channel-wise convolutions before feeding the pixel-wise descriptors to the NetVLAD layer rather than reduce the dimensionality of the image descriptor. We have experimentally compared the effect of this channel squashing in Fig. 4.11.

4.4 Experiments

In this section we describe our experiments. We evaluate the effect of using the allpair loss function compared to the commonly used triplet loss function (Sec. 4.4.1), while keeping the same backend CNNs. Next the effect on running time and memory consumption by the use of decoupled convolutions are tabulated (Sec. 4.4.2). In Sec. 4.4.3 we evaluate the precision-recall performance of the proposed algorithm with other competing methods in the SLAM community and in the visual place recognition community. Finally in Sec. 4.4.4 we evaluate the performance of the proposed method on a real world SLAM sequences

captured under challenging conditions especially revisits occurring with non-fronto parallel configurations and in-plane rotations.

4.4.1 Evaluation Metrics for Loss Function

We evaluate the effect of the proposed loss function on NetVLAD compared to the original triplet loss which was used in [5]. We evaluate our proposed loss function using a) relative loss declines b) number of correctly identified pairs from the validation tuple. We also plot the variance dot products of the positive samples amongst themselves. For validation we use the Pitts30K dataset and for training we use the TokyoTM dataset.

Similar to the training tuple, a validation tuple contains a query image, nN (= 6) negative samples and nP (= 6) positive samples. For evaluation, we propose to count the number of correctly identified image pairs which were actually similar to query images (ground truth) and identified as similar based on the image descriptor dot product scores. Ideally, the evaluation metric should be the SLAM sequences however it becomes infeasible to evaluate with SLAM sequence every say 20 epochs so we stick to this workaround. We plot these metrics for the training data and the validation data as the iterations progress. See Fig. 4.5, 4.6, 4.7 for the plots. The summary of the observation:

- Using the same backend convolutional network, the same parameters of the NetVLAD layer, and same learning hyper-parameters, the network trained with the proposed all-pair loss function performs better as evaluated against the validation metric, count of pairs correctly identified.
- It can also be inferred that the gradients obtained from the use of proposed all-pair loss function are more stable, hence the faster convergence.
- Our all pair loss function was found to perform better even when using the decoupled convolutions, decoupled convolution with channel quashing vs the original VGG16 network.

Number of Zero Loss Tuples

In this experiment, we train with batch size 24. But since this won't fit in the GPU memory we use gradient accumulation. We plot the number of zero loss sample as iterations progress in Fig. 4.3. When using the triplet loss, we get more number of zero

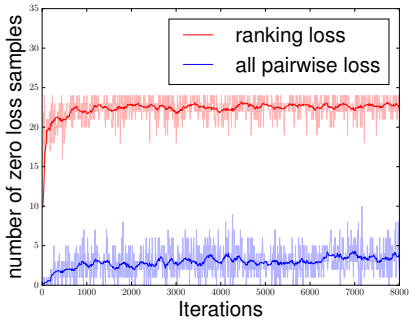

Figure 4.3: The number of batches with zero loss as iterations progress for learning with proposed cost function (in blue, Eq. 4.5) compared to using the triplet ranking loss [5] (in red, Eq. 4.4).This experiments used a batch size of 24 with gradient accumulation. Having a higher count for zero-loss samples is detrimental to learning as it leads to zero-valued gradients. Best viewed in color.

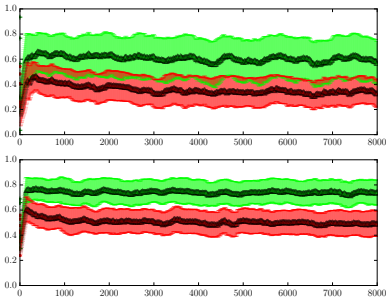

Figure 4.4: Showing spreads ($\mu \pm \sigma$) of $\langle \eta_q, \eta_{P_i} \rangle$ (in green) and spreads of $\langle \eta_q, \eta_{N_i} \rangle$ (in red) as the learning progresses. Fig. 4.4 (top) corresponds to [5], with the triplet loss function. Fig. 4.4 (bottom) corresponds to the proposed allpair loss function. We observe a lower spread amongst positive samples and larger separation between positive and negative samples.

loss samples. This results in zero-gradient updates and hence slow learning compared to proposed allpairloss. This can be attributed to allpairloss function being harder to satisfy resulting in better gradients during training.

Spreads of Positive and Negative Samples

As experimentally observed in Fig. 4.4, the use of proposed allpair loss function results in a more discriminative image descriptor as compared to the network trained with the triplet loss. We observe a lower spread amongst the positive samples and a larger separation in positive and negative samples. This has the effect that the deployment as loopclosure module being less sensitive to slight changes in dot product thresholds.

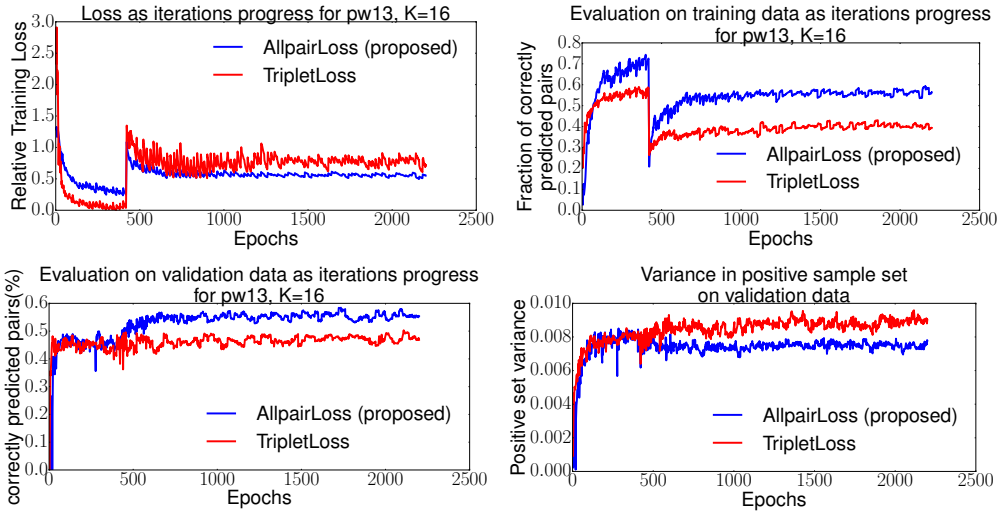

Figure 4.5: Comparing the effect of using allpairloss and tripletloss for training with decoupled net (deepest layer) with K=16. (a) Shows the relative training loss as iterations progress (lower is better). We show the evaluation metric, ie. the count of correctly identified pairs in (b) and (c) for training data and a separate validation data (higher is better). (d) show the variance in the positive set in dot product space as iterations progress (lower is better).

TripletLoss vs AllpairLoss on Decoupled Net

We compare the effect of different loss function when using the decoupled network. The network trained with allpair loss is able to correctly identify almost 60% of the pairs from the tuples drawn from the validation data, compared to when network was trained with tripletloss which is able to identify about 35% correctly under identical conditions. See Fig. 4.5.

TripletLoss vs AllpairLoss on VGG16 Net

Even when trained with VGG16 as the backend CNN, allpairloss performed better than the tripletloss under identical training conditions. See Fig. 4.6.

TripletLoss vs AllpairLoss on Decoupled Net with Channel Squashing

When using decoupled network with channels squashing (for dimensionality reduction) we observe a better performance when trained with allpairloss. In this configuration the descriptor size is just 512 per image. The training was arguably more unstable in this case (we observe oscillations). Possibly with a lower learning rate this effect can be reduced. See Fig. 4.7.

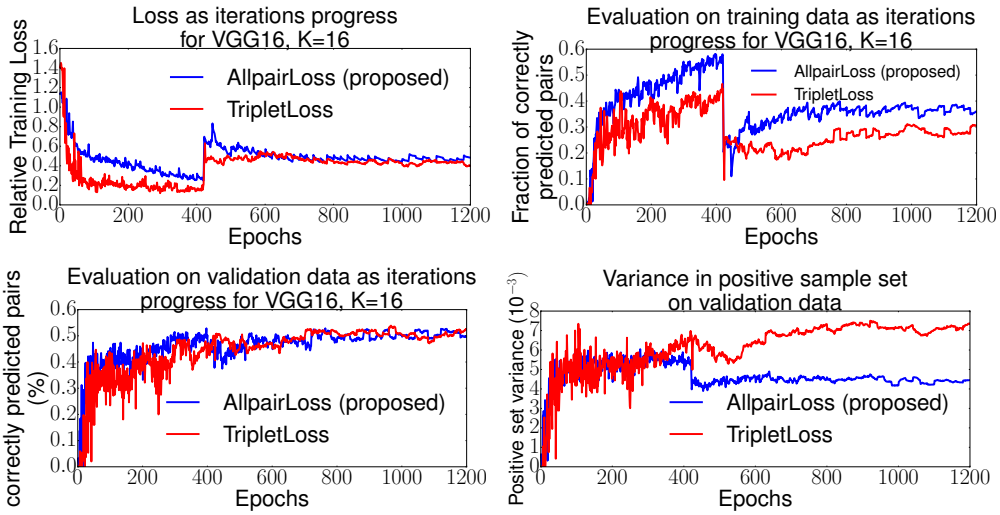

Figure 4.6: Effect of tripletloss and allpairloss with backend CNN as the VGG16 net with K=16. (a) shows the relative training loss as iterations progress (lower is better). (b) and (c) shows the training and validation evaluation metric (higher is better). Evaluation metric is the percentage of pairs correctly identified. (d) shows the variance of positive set descriptors in dot product space.

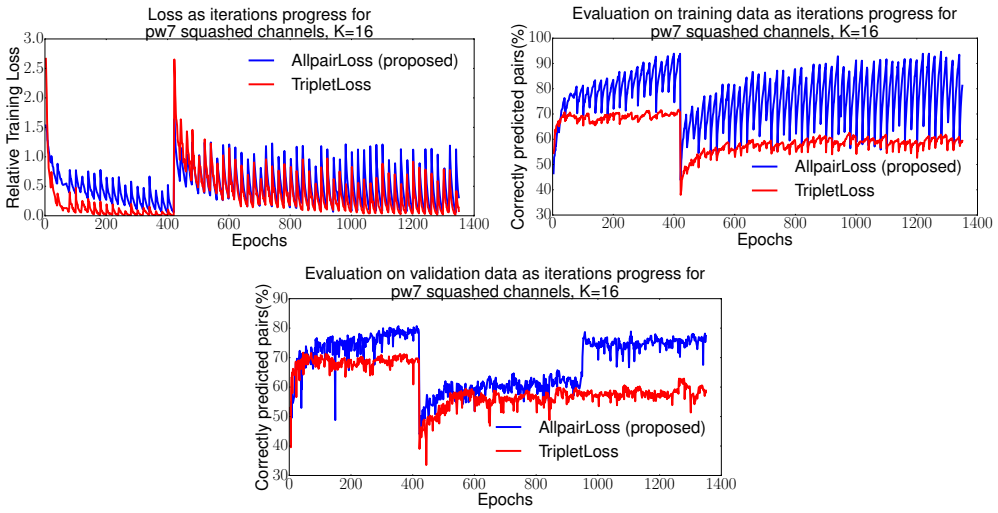

Figure 4.7: Effect of tripletloss vs allpairloss for decoupled net, K=16 with channel squashing. The descriptor size in this case was just 512. Arguably the learning in this case can be improved with lower learning data due to the osscillating losses we observe. (a) shows the relative training loss. (b) and (c) shows the percentage of pairs correctly identified for training and a separate validation data.

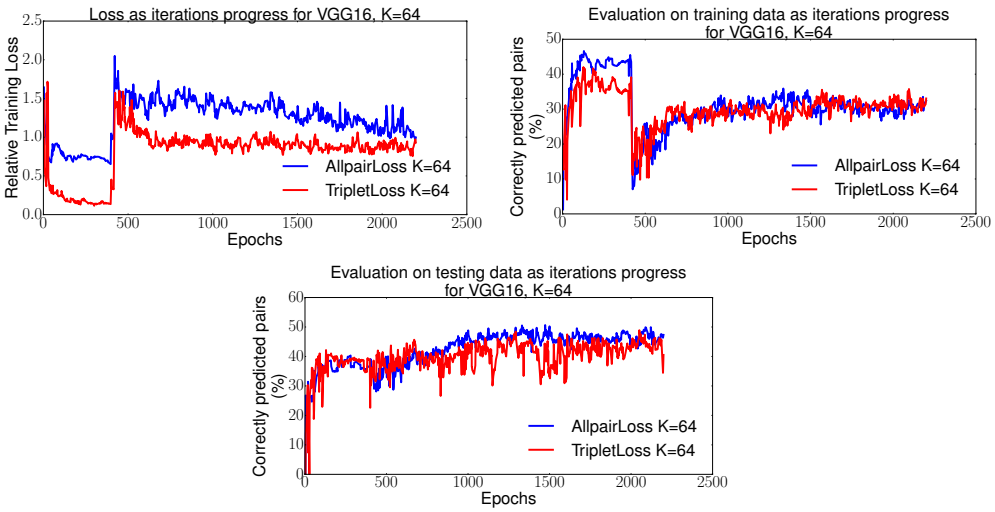

Figure 4.8: Effect of tripletloss vs allpairloss for VGG16 net, K=64. (a) shows the relative training loss (ratio of loss at i^{th} and 0^{th} iteration). (b) and (c) shows the percentage of pairs correctly identified for training and a separate validation data.

4.4.2 Running Times

Currently there is rapid progress in compute-capabilities of GPUs. Under such circumstances it is much more appropriate to report the number of floating point operations (FLOPs) for the networks rather than absolute running times in seconds (or milli-seconds). We tabulate in Table 4.2 the Giga-FLOPs of the networks under various parameter settings. For reference, the forward pass with VGG base network can be computed in about 40-50ms and with decoupled nets in about 10-15ms for 640x480 3 channel images on Titan X (Pascal). VGG16 with $K = 64$ is the recommended configuration from Arandjelovic *et al.* [5], which results in a 32K-dimensional descriptor which is reduced to 4096-dimensional by a linear transformation. This linear transformation takes about 400 MB of memory.

On the other hand, our proposed network which uses a network with decoupled convolutions as the base CNN with channel squashing and $K = 16$ results in 512-dimensional descriptor (4096-dimensional if not using channel quashing). It is able to run 3-4x faster than NetVLAD [5] while having about 20x fewer floating point operations for a 640x480 image and 5x fewer number of learnable parameters. See table 4.2. It is worth noting that most of the computational load is in the computation of per pixel descriptors and the NetVLAD layer itself takes negligible computations compared to base CNN.

For training the networks, we use Intel i7-6800 CPU with Titan X (Pascal), 12GB GPU RAM. It takes about 1 seconds per iteration for forward and backward pass. Note that one iteration with batchsize 4, involve 52 images $((6 + 6 + 1) \times 4)$.

CNN Layer	# L	D-Size	Model (MB)	Fwd pass memory (MB)		GFLOPS	
VGG16_K16				**320x240**	**640x480**	**320x240**	**640x480**
block5_pool	14.7M	8192	56.19	234.94	767.56	47.04	188.08
block4_pool	7.6M	8192	29.19	174.06	725.32	47.05	188.117
block3_pool	1.7M	4096	6.65	165.47	641.86	47.05	188.167
VGG16_K64				**320x240**	**640x480**	**320x240**	**640x480**
block5_pool	14.78M	32768	56.38	234.32	711.46	47.05	188.11
block4_pool	7.70M	32768	29.38	203.53	696.23	47.08	188.26
block3_pool	1.76M	16384	6.75	158.86	635.26	47.13	188.46
decoup_K16				**320x240**	**640x480**	**320x240**	**640x480**
pw13	3.2M	8192	12	197.97	792.21	1.742	7.01
pw10	1.36M	8192	5	189.1	734.48	1.749	7.03
pw7	554K	4096	2	164.9	652.25	1.749	7.04
decoup_K16_r				**320x240**	**640x480**	**320x240**	**640x480**
pw13	3.5M	512	12	211.58	793.46	1.742	7.01
pw10	1.49M	512	5	193.86	739.73	1.749	7.03
pw7	686K	512	2	167.67	657.97	1.749	7.04
decoup_K64				**320x240**	**640x480**	**320x240**	**640x480**
pw13	3.33M	32768	12	210.98	805.21	1.76	7.08
pw10	1.40M	32768	5	189.38	734.58	1.78	7.18
pw7	600K	16384	2	162.86	652.35	1.78	7.186

Table 4.2: Tabulation of run time memory requirements, learnable parameters (# L), descriptor size (D-Size), model size in Mega-bytes, giga floating point operation (GFLOPs) for various configurations. We note that *block*5*_pool* for VGG16 network is equal in depth to *pw*13 for decouped network. *block*4*_pool* and pw10 have equal depth; *block*3*_pool* and pw7 have equal depth. K (eg. K16, K64) refers to the number of clusters in NetVLAD layer. We report data for input image size 320x240 and 640x480. We conclude that our proposed decoupled network is 20X faster computationally with an order of magnitude less number of parameters, while delivering about the same performance as the original NetVLAD. Our squashed channel network 'decoup_K16_r' gives a descriptor size of 512 with about 5% additional forward pass memory and 2% increase in parameter size with hardly noticeable computation time increase. NetVLAD [5] uses a whitening PCA for reducing descriptor dimensionality which needs to store a matrix of size 32Kx4K that takes about 400 MB.

4.4.3 Precision-recall Comparison

We evaluate the performance on the following datasets: a) **GardensPoint** dataset, b) **CampusLoop** dataset [136], c) our **CampusConcourse** dataset. Each of the datasets contains two sequences, 'live' and 'memory'. Note that every image in live sequence has a corresponding image in the memory sequence. For evaluation, we load the memory sequence in the database and compare this database with each of the images in live sequence using a basic nearest neighbour search. Further we also evaluate our performance for the mappilary Berlin streetview dataset [200] i)**berlin-kundamm** ii) **berlin-halenseeestrasse** and iii) **berlin-A100** as has been common amongst visual place recognition community.

We compare the proposed method with the recently proposed learning based loop detection methods **CALC** by Merril and Huang [136]. Additionally we also compare with **DBOW** [57] (Bag-of-visual words). We evaluate with prominent approaches amongst visual place recognition community, **AlexNet** by Sunderhauf *et al.* [199], **LA-Net**, by Lopez-Antequera *et al.* [123]; **original NetVLAD** [5]; Chen *et al.* [35].

The main idea of this evaluation is to gauge the recall rates and discriminative performance of various methods. We take the matches as correctly identified if match's index is within six indices of itself. For our evaluation, we use the total number of positive matches as the length of the sequence, since every image in the live sequence has a correspondence in the memory-sequence. Total number of accepted matches are those which satisfy the loop hypothesis. The precision-recall curves are formed by sweeping through all the thresholds within the full range of thresholds. The results presented here differ from those presented in [136] as it is not exactly clear how recall=1 was achieved by them, how the DBoW was used to generate these results and any heuristics, if any, was used to identify false positives.

As noted by Merrill and Huang [136], and by Sunderhauf *et al.* [199] superior precision-recall does not fully prove the superiority of a method in real loop-closure of a SLAM system. Such factors as repeated objects in scenes, similar looking scenes, in-variance to rotation & scale, computation time are important when considering a place recognition system for SLAM's loop closure. Another issue about such an evaluation is that it cannot gauge a method's ability to return 'no matches' in case the query scene is not found in the database. Also all these datasets are rather small (about 80-200 frames) and we

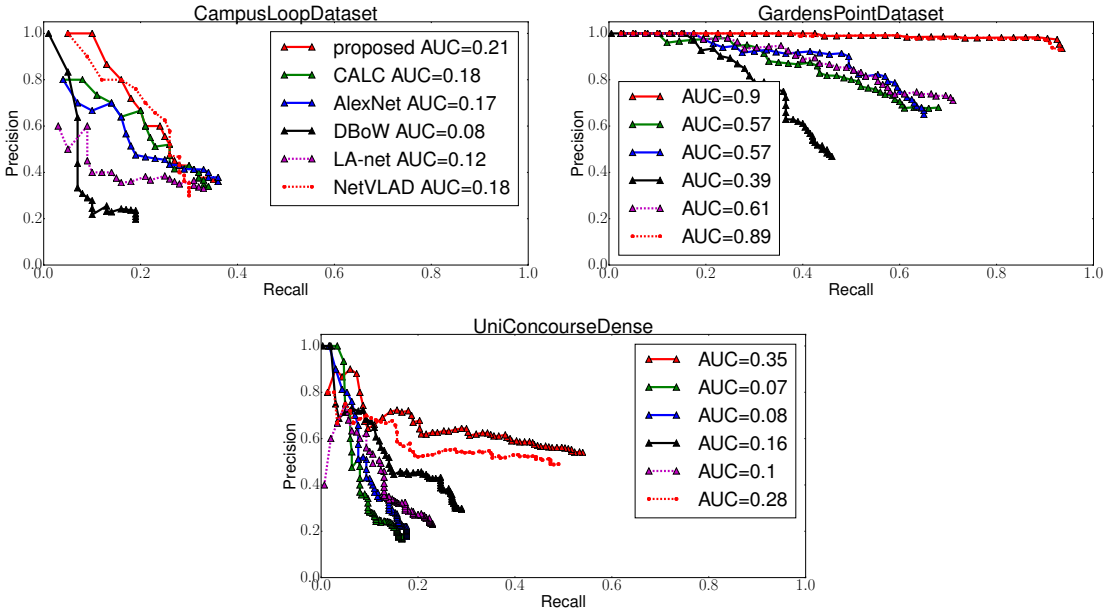

Figure 4.9: Precision-recall curves for various methods for loop detection. Our method using the decoupled net as the backend CNN gives comparable performance in CampusLoop dataset which contains appearance changes due to snowy weather. Our method gives a comparable performance to the NetVLAD in other two datasets which has only large viewpoint and in-plane rotational changes. Which is far better than other relevant methods.

cannot evaluate the generalizability of the scene description by each of the methods. For example a dataset with multiple similar looking scene is needed to thoroughly evaluate a method's performance. Rotation and scale variance of the method cannot be evaluated with these datasets. So for a better perspective of the usability of the methods we also evaluate them on live SLAM sequences with manually marked loopclosure detections for evaluation (details in section 4.4.4).

Walking-apart Sequence

For precision-recall curves see Fig. 4.9. The *CampusLoop sequence* contains appearance variation due to changing weather condition. Our method does not explicitly deal with this kind of variation as it is primarily based on color cues. The method CALC, for example, is based on scene structure. Our method delivers comparable performance in this sequence. For the other two testing sequences, viz. *GardenPoint* and *UniConcourse*, our method performs better than previous methods. This is attributed to the fact that the descriptors learned by our method are able to generalize well into identifying place revisits at large viewpoint difference and rotational variance, which is the case in these

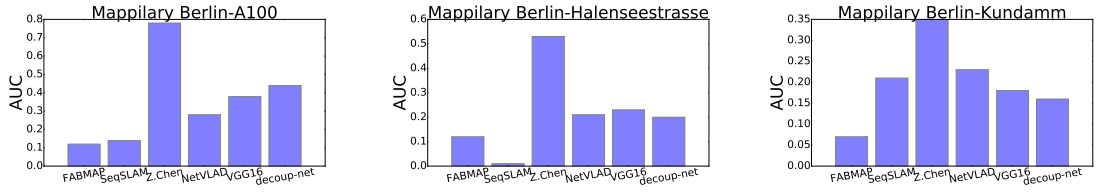

Figure 4.10: Comparing the methods with area under the curve (AUC) of the precision-recall plots for the mappilary dataset. The following methods were compared: FABMAP [41], SeqSLAM [140], Z.Chen [35], NetVLAD [5], proposed with VGG16 backend net, proposed with decoupled net as backend net.

two sequences. When compared to NetVLAD [5] which uses the triplet loss and VGG network, we observe a slight boost of recall rates. Since these sequences are very small, the higher capacity of the proposed method is not observable in this case.

AUC Performance on Mappilary Dataset

We evaluate our method with other state of the art methods with area-under-the-curve (AUC) of the precision-recall plot on the mappilary Berlin-streets dataset in Fig. 4.10. Each of the three test sequences contain two sets of images. Note that each image in second of the two sets has a pre-image in the first set. These datasets are 80-200 frames each. Although considerable viewpoint and light variation exists amongst the two sets, there is no rotation variation. We test our method with VGG16 backend CNN and with decoupled-net backend CNN. We compare with FAB-MAP [41], SEQSLAM [140], Chen *et al.* [35], NetVLAD [5]. In this case, Chen *et al.*'s method outperform. It is worth noting that Chen's method [35] makes use of region proposal and is not a real-time (or near real-time) method.

4.4.4 Online Loop Detections

We compare the performance of the descriptors produced from the proposed method to some of the relevant methods with real world sequences. We introduce three sequences and refer them as 'Live Walks Dataset', each is about 10min of walking. The main differentiating point compared to the standard KITTI dataset is that ours contains adversaries like revisits under large viewpoint difference, moving objects (people), noise, lighting changes, in-plane rotation to name a few. Two of which were captured with a gray scale camera and one of it was captured with a color camera. We also provide manually marked ground truth labels for loop detections along with odometry of the poses

for visualization. The odometry was not used for identifying loops. Note that for a real sequence with N keyframes there are a little less than N^2 pair of loop-frames. The human was shown every pair and asked to mark the pairs which were the same place. Using these manual annotations, there are 3 kinds of pairs. a) pairs not detected by the algorithm, ie. missed pairs b) wrongly detected pairs, ie. pairs which were in reality different places but algorithm identified it as the same place. c) pairs correctly identified, ie. pairs which were marked by the algorithm as same places and were in reality same places.

We compare our method with some of the relevant methods on our real world datasets. We plot the precision-recall under various threshold settings. We define precision as the fraction of candidate loops which were actually loops. By recall we mean the fraction of actual loops identified. We do not use any geometric verification step to boost our precision, the results shown in this section are from raw image descriptor comparison. With geometric verification, precision of almost 100% can be easily accomplished.

We use our method in various configurations a) decoupled net as base CNN, K=16 (descriptor size of 4096), b) VGG16 as base CNN, K=16, c) decoupled net with squashed channels, K=16 (descriptor size of 512). We compare with i) NetVLAD [5], ii) Merrill and Huang [136], iii) Sunderhauf *et al.* [199], iv) DBOW [57] and v) ibow-lcd [60]. We acknowledge the superior performance of Z. Chen *et al.* [35] method and possibly also of Sunderhauf *et al.* [200] on the mappilary dataset. However, it was not practical to test it on our datasets which are an order of magnitude larger than those dataset. It takes about 1-1.5 sec/frame for descriptor computation and about 800ms-1.2 sec/pair for descriptor comparison. For our dataset of 5000 keyframes the provided MATLAB implementation would take almost 100 days (number of comparisons would be $5000 + 4999 + 4998 + ...$). Arguably a faster implementation could accomplish the task in about a day or two for a 5000 frame or 15 min walking video. Thus this method is no where close to being real-time. Hence it was not compared. We also note that the running time for Sunderhauf *et al.* [200] is in similar range to Z. Chen *et al.*'s method.

4.5 Conclusion

We proposed a data-driven, weakly supervised approach to learn a scene representation for use in loopclosure module of a SLAM system.

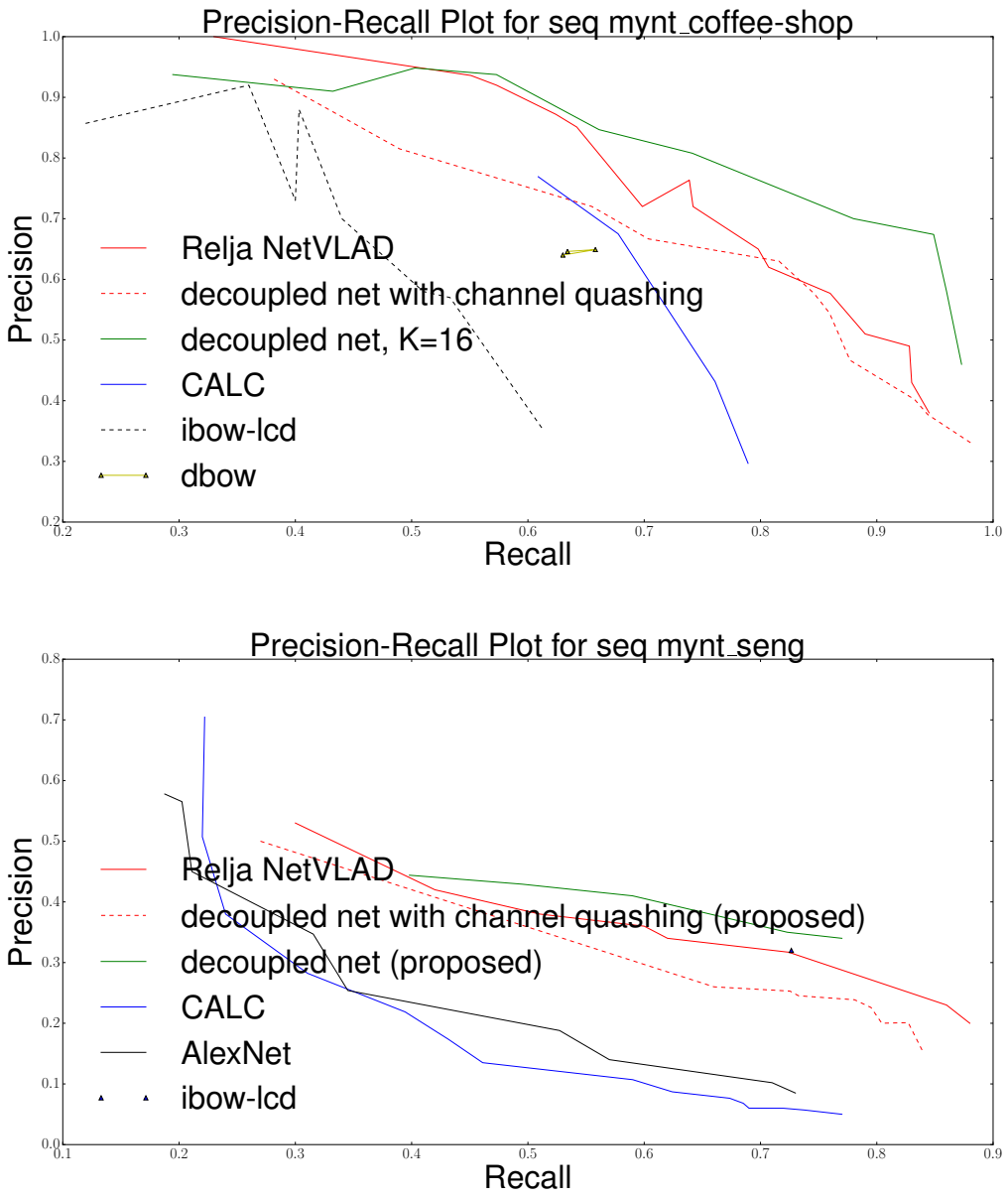

Figure 4.11: Precision-recall plot for the sequences ‘mynt_coffee-shop‘ and ‘mynt_seng‘ when compared to manual annotations of loop candidates and threshold varied. We compare the following methods: Relja NetVLAD [5], decoupled net with channel squashing (proposed), decoupled net without channel squashing, CALC [136], ibow-lcd [60] and DBOW [57].

Unstable learning was observed for the original NetVLAD [5] which made use of triplet-loss for training. This was observed to be especially prominent when trained with smaller number of clusters. The issue was mitigated with use of the proposed allpairloss function. This resulted in higher performance even with a smaller number of cluster in the NetVLAD layer. For realtime performance we made use the decoupled convolutional layer instead of the standard convolutions for speed. The network with decoupled convolutions are almost 3X faster in computation time with 5-7X fewer learnable parameters.

To evaluate precision-recall performance for loopclosure detection in a real SLAM system, we compare our method with popular BOVW-based methods along with state-of-the-art CNN-based methods on real world sequences. Qualitative and quantitative experiments on standard datasets as well as self-captured challenging sequences with adversaries including revisits at large viewpoint difference, in-plane rotation, dim lighting etc. suggest that proposed method can identify loopcandidates under substantial viewpoint difference. We also observe a boost in recall rates when compared to training with original NetVLAD. Also our descriptors are found to be fairly invariant to rotation and lighting changes.

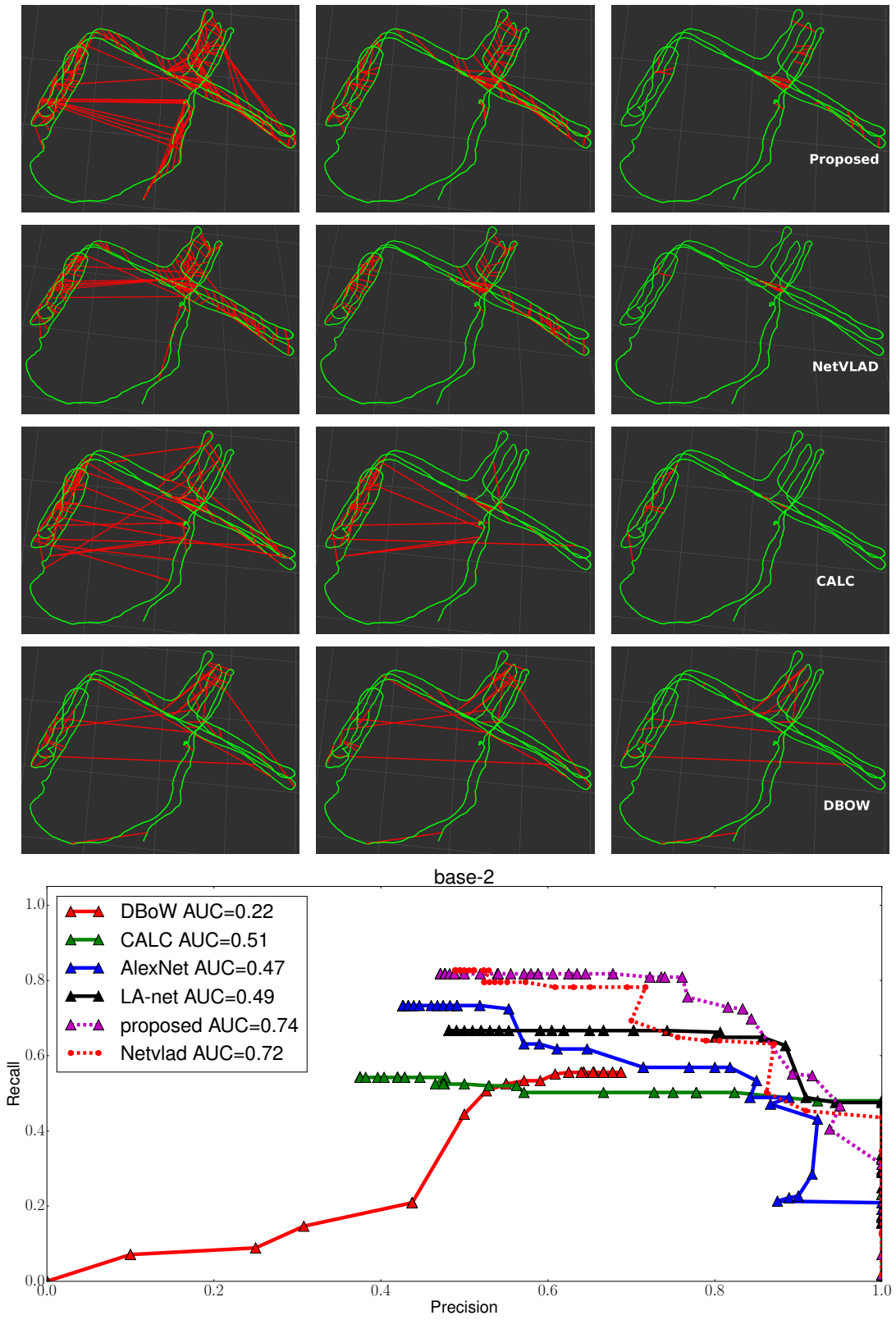

Figure 4.12: Loop closure candidates (in red) as we vary the thresholds on VIO (green) for sequence 'base-2' for the proposed method (in row-1); NetVLAD [5] (in row-2); CALC[136] (in row-3) and DBOW [57] (in row-4). Along the columns are various thresholds. Leftmost is for loosest, rightmost is for tightest. Row-5 shows the PR-curve for each method where compared to human marked loop-candidates.

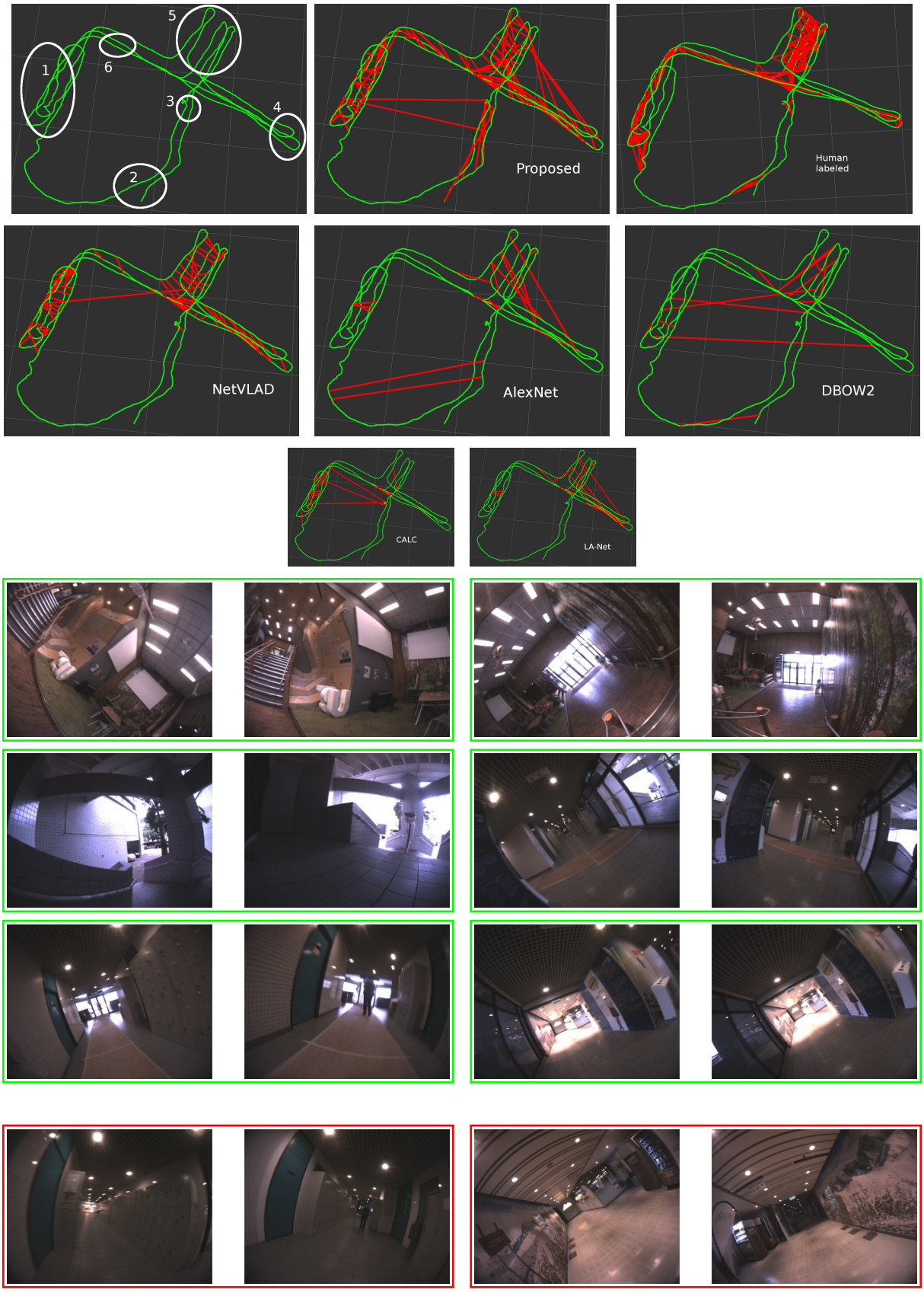

Figure 4.13: Top row: Plot of visual-inertial odometry of the sequence 'base-2'; Loop candidates by our proposed method; human marked loop candidates; **2nd row**: NetVLAD [5]; CALC [136] ; LA-Net[123] ; Alexnet [200] ; DBOW2 [57]. **Row 3 and 4**: Examples of correct detections by the proposed method in each of the regions. **Row 5**: Examples of wrong detections.

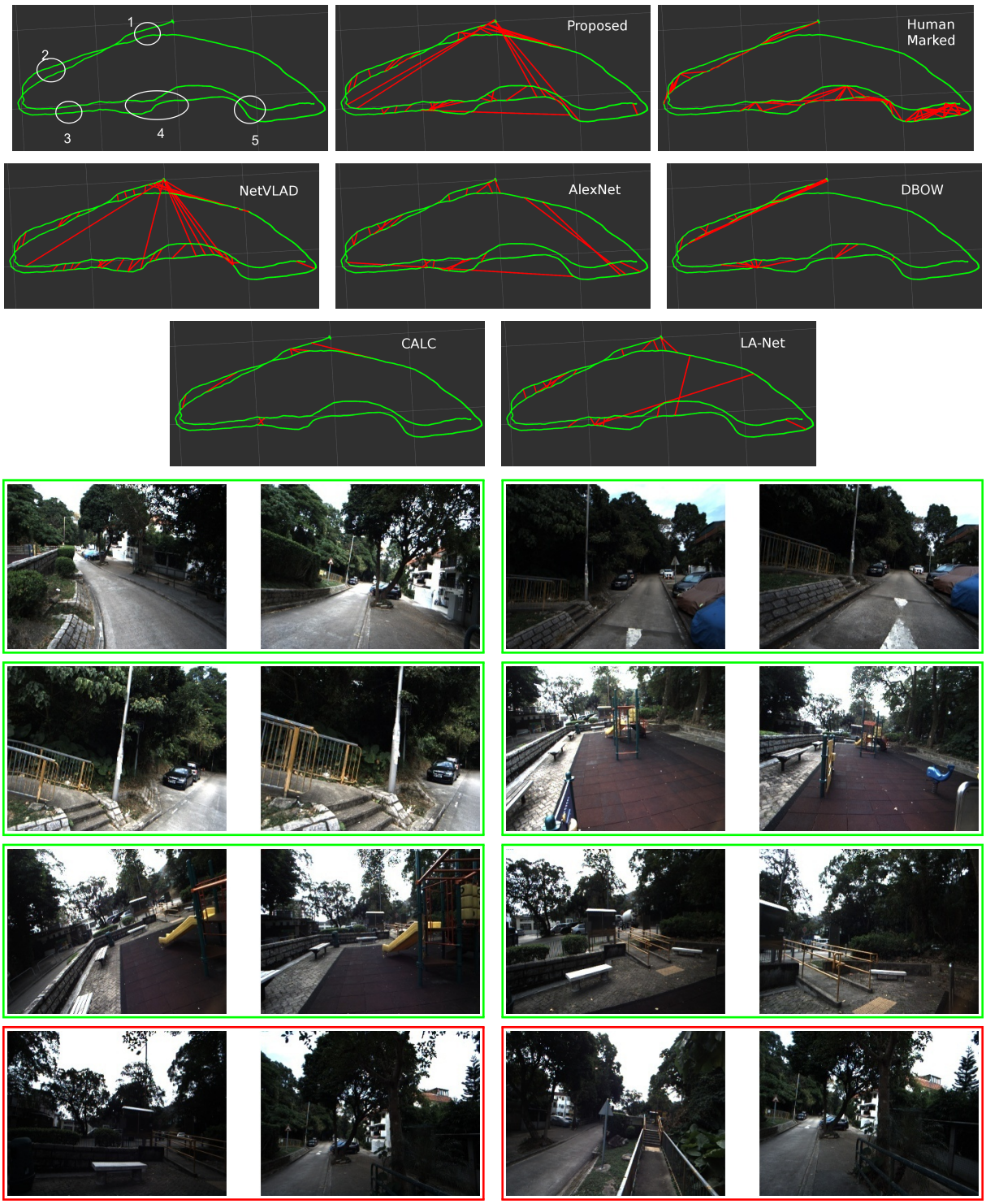

Figure 4.14: Top row: Plot of visual-inertial odometry of the sequence 'tpt-park'; Loop candidates by our proposed method; human marked loop candidates; **2nd row**: NetVLAD [6]; CALC [136] ; LA-Net[123] ; Alexnet [200] ; DBOW2 [57]. **Row 3 and 4**: Examples of correct detections by the proposed method in each of the regions. **Row 5**: Examples of wrong detections.

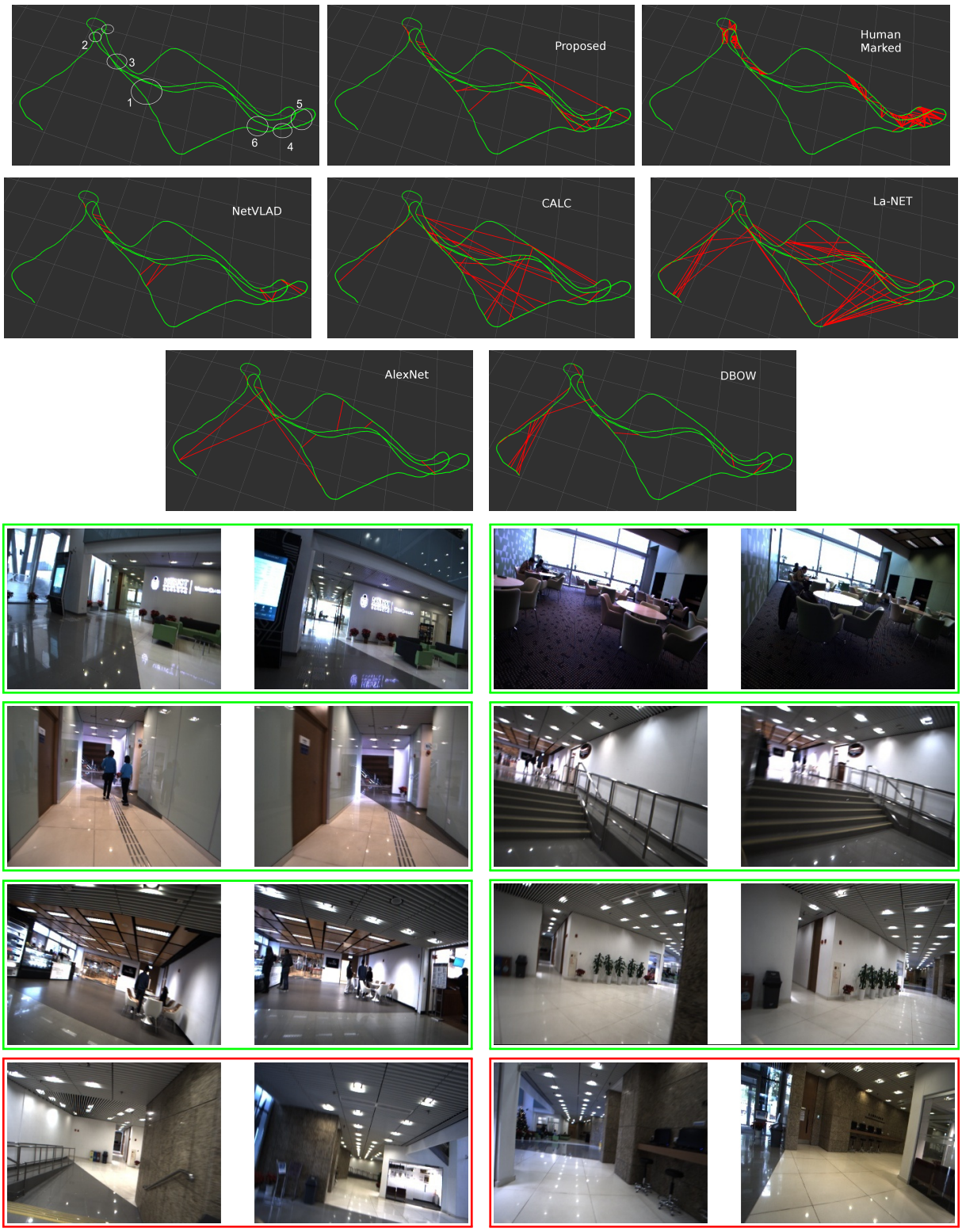

Figure 4.15: Top row: Plot of visual-inertial odometry of the sequence 'lsk-1' with human marked place revisits 1 to 6; detections by the proposed method for this sequence; NetVLAD [6]. **2nd row**: CALC [136] ; LA-Net[123] ; Alexnet [200] ; DBOW2 [57]. **Row 3 and 4**: Examples of correct detections by the proposed method in each of the regions. **Row 5**: Examples of wrong detections.

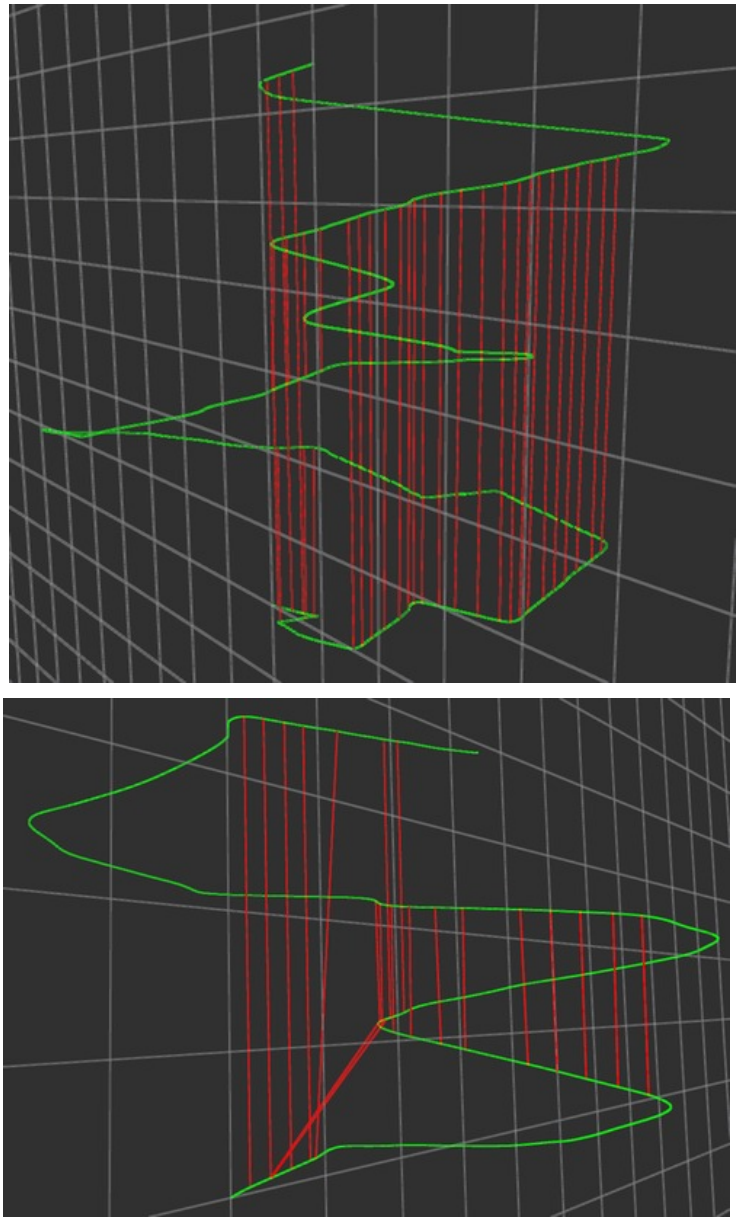

Figure 4.16: The results of the proposed method on KITTI00 and KITTI05. The XY plane is the 2d location of the trajectory. z-axis represents the frame number. In this dataset the revisits occur at similar viewpoints, the performance of all the compared methods is almost the same.

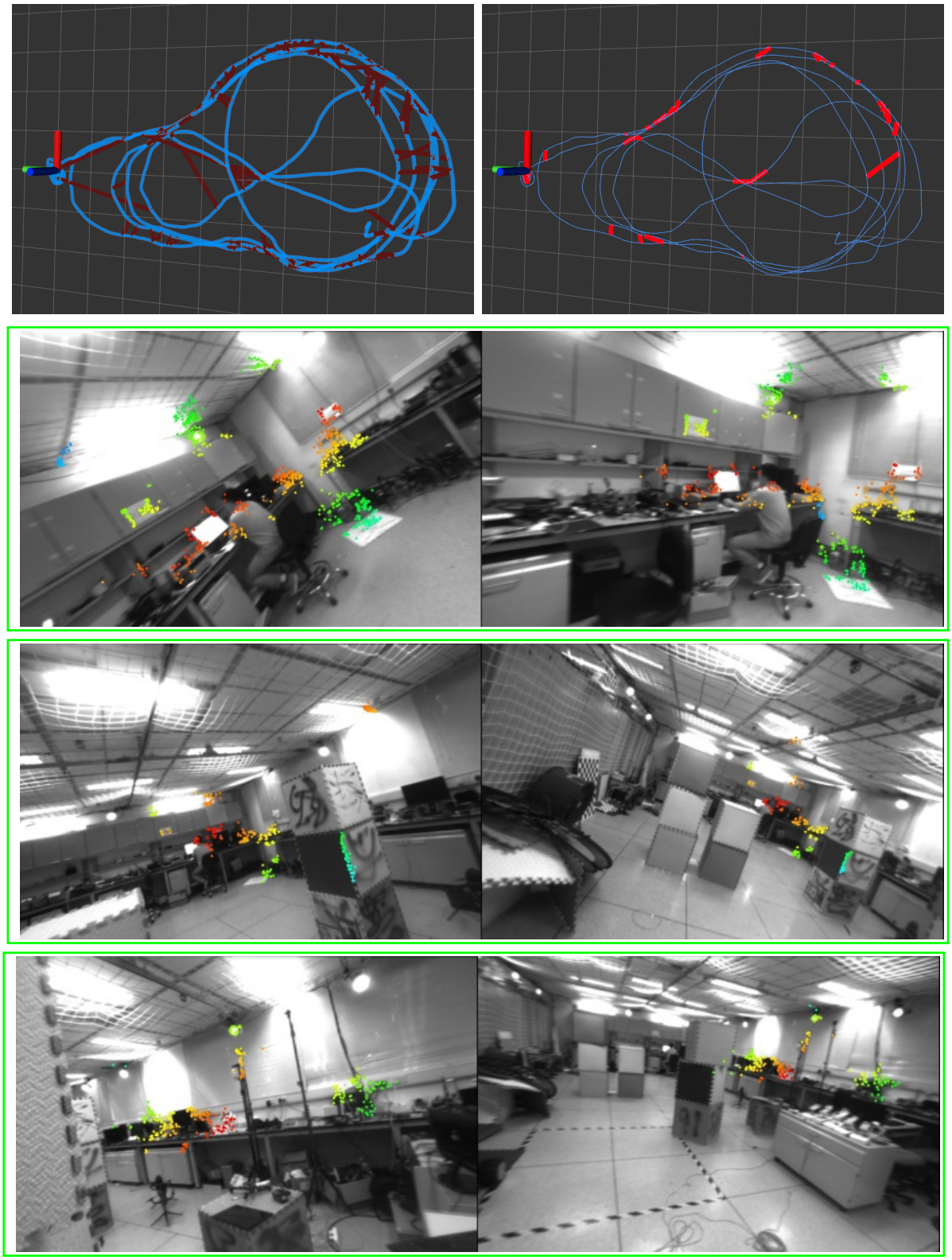

Figure 4.17: Comparing revisit detections of the proposed method (top-left) and VINS-Fusion, which uses DBOW2 (top-right). This sequence contains repeated traversal in a hall of 15mx5m at various rotations and viewpoints. Although bag-of-words based method perform well under fronto-parallel view it has very low recall compared to our method on larger viewpoint difference. A side-by-side live run of this sequence is available at `https://youtu.be/dbzN4mKeNTQ`. Row-2 to row-4 shows some representative loop-pairs which we identified by our methods as loops but were missed out by DBOW2 in VINS-Fusion. The pointfeature matches were produced live, details of which are described in Chapter 5.

Chapter 5

Place Recognition Back-end

Performance of visual slam systems have been steadily improving. Camera and Inertial sensor fusion has been a popular contemporary theme towards making SLAM systems robust and consistent. Continuing from our previous work on learning powerful place representation, in this work we deal with the problem of image feature matching and relative pose computation under large view point difference. We develop a fast and low complexity voting scheme for incorporating spatial smoothness constraint into point matching. We demonstrate the use of this simple technique to robustly match tracked features of the place revisits. Finally we propose a novel unified non-linear optimization based framework to obtain relative poses by accumulating multiple revisit image pairs which typically contain low number of tracked features to obtain a high quality relative poses for the pose graph optimization.

Some past works have also built full SLAM system with multi-session map merging capability. The *maplab* system [180] provides an online tool, *ROVIOLI* which is essentially a visual-inertial odometry and localization front-end. Although it provides for a console based interface offline for multi-session map merging, it cannot identify kidnaps and recover from them online. This is the major distinguishing point of our system. Also the relocalization system by Tong *et al.* [166] can merge multiple sessions live it only merges with the first co-ordinate frame, any loop connections between co-ordinate systems not involving the first co-ordinate systems are ignored. Our system on the other hand is able to maintain multiple co-ordinate systems and their relative poses, set associations and merge the trajectories online in real-time. We open source our fully functional system [1] as an add-on for the popular VINS-Fusion.

[1] `https://github.com/mpkuse/cerebro`

5.1 Introduction

In a camera-IMU setup, tracking drift gets quite significant due to accumulating errors from visual inertial odometry. An extremely low (and non accumulating) drift is essential for rich user experience which uses SLAM, for instance AR (Augmented Reality). The major tasks in reducing drifts in SLAM system can be identified as a) revisit detection b) computation of relative pose between revisits c) pose graph optimization.

Previous successful visual-inertial SLAM systems, VINS-Mono [165], ORB-SLAM[145, 146], OKVIS[109] make use of a bag-of-visual-words based algorithm to detect place revisits. Works by Cummins *et el.* [41], Galvez *et al.* [57] have been popular.

These techniques work well when the revisits occur at almost the same viewpoint as the past visit. However, the performance of these algorithms is limited by the underlying feature descriptor in addition to the vocabulary used. Another negative point about these methods is that they essentially uses a very small fraction of the image information to identify revisits. As demonstrated in our previous work [98] method based on bag-of-visual-words suffer from a high miss rate especially when revisits occur as a large viewpoint difference. Other adversaries like motion blur, few corner features, changing lighting condition present a difficult challenge to these techniques.

Recently, CNN based approach have been popular. Chen *et al.* [33], Sunderhauf *et al.* [200] made use of descriptors from off-the-shelf object classification networks to identify place revisits. Lopez-Antequera *et al.* [123], Merrill and Huang [136], Kuse and Shen [98] attempted to learn image descriptors especially for the task of place recognition. Our method to detect place revisit [98] was found to have outperformed all the previous techniques when tested using challenging datasets.

Finding correspondence given a true positive revisit image pair potentially at large viewpoint difference, is another fundamental issue. This is crucial for relative pose computation for reducing the drift. Traditionally, feature correspondences are computed by matching tracked features using local descriptors like SIFT [126], ORB [173], A-SIFT[141], Haris Corners [70], affine covariant region detectors [138] etc. followed by pruning incorrect matches by the ratio test and fundamental matrix test. The problem with sole reliance on descriptors is the difficulty in differentiating true positives and false positives. This problem is increasingly evident with revisits occurring at large viewpoint difference and other adversaries like motion blur, in-plane rotation etc. Further, feature matching

if often times not feasible due to the limited number of tracked features in each view.

Recently several techniques [114, 116, 131, 162] have focused on separating true and false matches using match distribution constraints. These techniques result in complex smoothing terms which are difficult to understand and expensive to minimize. However, it has been observed by Lin *et al.* [114] that this results in elimination of large fraction of true matches which in turn is counter productive to accurate pose estimation. This issue is even more important under large view point difference as missing out on true point matching can be disastrous, given that so few features are common in such views.

Coherence constraints (similar motions for neighbouring pixels) can dramatically improve feature matching quality. Recently, [120, 129, 212] are some of the works that demonstrated superior performance in feature matching at a much higher computational complexity. For these approaches, the minimization is often complex thus limiting their ability for real-time usage. Stereo matching, a closely related problem, has also been attempted with similar approaches [73, 195].

There are some works which deal with point based coherence [115, 150] and patch-match based matchers [67]. These techniques can have powerful effect on matching quality however owing to their computational burden cannot be used on real-time SLAM systems albeit suitable for offline 3D reconstruction.

Wang *et al.* [18] proposed the GMS-matcher. Starting with a large number of feature points, the coherence constraint is incorporated as statistical likelihood measure based on the number of neighbouring matches.

The state-of-the-art semantic alignment methods [69, 91, 156] rely on powerful image representations from a deep convolution network. Although impressive results have been obtained for known objects (using image-net or similar dataset), matching multiple objects robustly remain an open problem [171].

More recently, Rocco *et al.* [170] proposed a network architecture which mimics the standard feature extraction, description and matching process. It was trained using synthetically generated imagery. It was demonstrated to work with image pair containing a single image category (for example bike, cat, dig etc.) It currently does not work well for a general scene. Related to [170] is the work by Brachmann *et al.* [22]. They proposed a layer which mimics RANSAC. Higher computation cost and difficulty to obtain ground truth matches are some of the major issues for this approach.

A large body of literature exists for the relative pose computation. Fundamentally three types of relative pose computations are possible, viz. 2D-2D (based on 5 point algorithm [110, 154]), 3D-3D (ICP based [164, 175]) and 3D-2D (based on perspective-n-points [107]). We review, some recent work in realtime pose computation and the current pose computation techniques in use in state-of-the-art SLAM systems.

The classical approach for pointset registration by Arun *et al.* [9] often fails due to noise in correspondences, wrong correspondences etc. More recently, the typical work flow is global method (which needs no initial guess) followed by iterative refinement. Global methods usually start with a 3D-3D point correspondences [66]. Some of the recent methods in this category include [23, 25, 74]. Branch-and-bound based methods [48, 111, 217] often produce very high quality solutions by exploring the pose space. However these are computationally very expensive and likely not suitable for realtime SLAM systems. Approaches by Bustos et al [25], Yang et al [216] and variants of 4-points congruent sets (4PCS) [201] are some recent approaches which explicitly model the noise in their formulation.

A large body of iterative methods are ICP based, some of the recent relavant methods are [20, 26, 84, 164, 223]. The main bottleneck in ICP based approaches is the fact that at every involves expensive nearest neighbour computations. The computational effort is expended on testing candidate alignments that are subsequently discarded. To this effect, Zhou *et al.* [228] proposed a fast global registration method. Their approach is computationally less complex and does not involve iterative refinement, initial guess and works quite well in presence of noise, partially overlapping scenes and wrong correspondences. The proposed approach in our relocalization pose computation is inspired from [228].

We review the relative pose computations in the relocalization / loop-closure systems for popular recent SLAM systems. VINS-Mono [167] relies on a tightly coupled relocalization, initial guess of loop pose is estimated by perspective-n-points (PNP). The image-image matching is performed using the ORB descriptors and the 3D points from monocular tracking. The relocalization system of ORB-SLAM [145, 146] consist of the bag-of-visual words approach and PNP for pose computation. LSD-SLAM[46] performs the scale aware sim(3) direct image alignment for relocalization pose computation. Some other SLAM systems which follow a similar pipeline for loop detection include RKSLAM[121], OKVIS[108], Ling *et al.* [118].

5.1.1 Descriptor Extraction and Comparison

We propose a pluggable system to the popular VINS-Fusion[2] by Qin *et al.*[165]. Our system receives keyframes from the visual-inertial odometry sub-system to produce loop-closure candidates. We use a naive store-and-compare strategy to find loop-candidates. The descriptors at all previous keyframes, ie. $\eta^{(I_t)}$ $t = 1, \ldots, t$ are stored indexed with time. When a new keyframe arrives, say $\eta^{(I_{t+1})}$, we perform $\langle \eta^{(I_{t_i})}, \eta^{(I_{t+1})} \rangle$ $i = 1, \ldots, t-T$. T is typically 150, ie. ignore latest 150 frames (or 15 seconds) for loopclosure candidates. These are a measure of likelihoods for loopclosure at each of the keyframe timestamps. We accept the loopclosure hypothesis if the query score is above a set threshold (fixed for all the sequences) and if three consecutive queries retrieve descriptors within six keyframes of the first of the three queries. In a real implementation, the threshold can be set a little lower and wrong hypothesis can be eliminated with geometric verifications heuristics. We note that loopcandidates with large viewpoint difference provide a formidable challenge for tracked feature matching between the two views. In this work for comparison of descriptor's precision-recall, we do not perform any geometric verification.

Our naive comparison (matrix-vector multiplication) takes about 50 ms for comparison with 4000 keyframes (about 10 min sequence) on a desktop CPU with image descriptor dimension of 8192 and about 10ms for 512 dimensional image descriptor. While the comparison times grow unbounded as number of keyframes increase, the objective of this paper is to demonstrate the representation power of learned whole image descriptors over the traditional BOVW on sparse feature descriptor loopclosure detection framework and the recently proposed CNN-based image descriptors in terms of detection under large viewpoint difference. Dealing with scalability could be a future research direction. In our opinion scalability can be achieved by sophisticated product quantization approaches similar to Johnson *et al.*[85] or by maintaining a marginalized set of scene descriptors, along with scene object labels and dot product comparison on this smaller subset.

5.2 Coherence Constrainted Robust Point Matching

In our previous work [98], we proposed to learn a whole image descriptor for scene representation. The next step after we have a putative loop candidates, is to compute relative

[2]https://github.com/HKUST-Aerial-Robotics/VINS-Fusion

pose between the matches, which could be used in a graph based pose graph optimization framework to correct trajectory drift. In order to compute the relative pose we need feature matches between the putative candidates. In scenarios where revisits occur at a similar viewpoints, tracked feature descriptors can be matched and the relative pose can be computed using a 5-point algorithm with RANSAC for robustness.

The revisits that occur at a large view point difference pose a significant challenge to feature matching and pose computation. Due to this viewpoint difference, the features tracked by the SLAM front-end (eg. ORB features) produce too few matches. This is since they are sparse features and the scene overlap area is rather narrow.

In order to effectively leverage the gains from the proposed loop closure system which produces matches even at wide view point difference we develop a novel frame work for feature matching. Our method is inspired from recently proposed GMS matcher [18] and is an effective technique to provide good matching performance at much lower computational cost.

This section starts with a brief review of the DAISY dense descriptors which the proposed framework is based upon. Next we propose an alternate notion of neighbourhood regularization by voting. We propose to leverage the assignment map of the neural network (see Fig. 5.1 for example association map) along with the dense *DAISY* descriptor [203]. We note that, it is also possible to make use of the pixel-wise descriptors from the neural network. However it doesn't produce effective matches at a reduced descriptor length. We defer this topic to a future paper and in this paper concentrate on developing robust matching heuristics which work in real-time and are effective at computing relative pose at larger viewpoint difference.

5.2.1 The *DAISY* Descriptors

Tola *et al.* [202, 203] proposed the *DAISY* descriptor. It is a histogram based local region descriptor which is computationally efficient to compute on all the pixels of an image. The advantage of *DAISY* descriptor is that pixels with no texture (but having some texture in the neighbourhood) can also be matched to the corresponding pixel in another view. This is unlike other commonly used descriptors like ORB, SURF etc, which are effective at matching only corner points. Additionally, Tola *et al.* [203] also show its robustness towards scale, contrast and brightness changes.

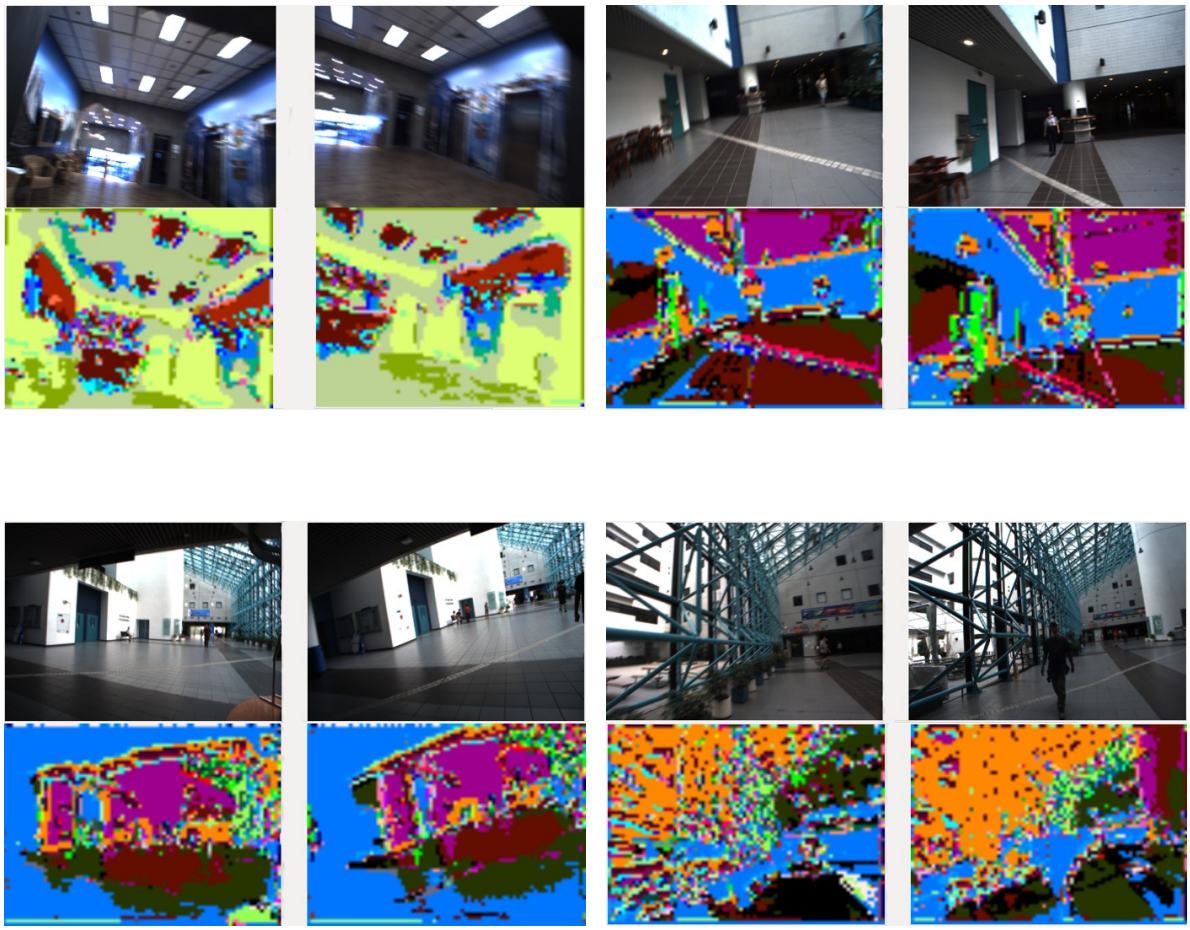

Figure 5.1: Show some example image pairs and their association maps

Winder *et al.* [215] demonstrated a minimalist *DAISY* configuration, in which even a 20-dimensional descriptor provided excellent performance. Since it can be computed quickly (and hence no need to store these) at all pixels in about 28-ms for a 320x240 image on a desktop computer and about 120-ms on CPU of Nvidia TX2 embedded computer we use it in our experiments. Dense DAISY descriptors are efficient to compute because histograms computed for one region can be reused for all neighboring pixels, and also the DAISY computation pipeline enables a very efficient memory access pattern.

For the sake of further discussion we assume, $\mathbf{d}_{I_t}(\mathbf{u}) \in \Re^{\mathbf{20}}$ denote the *DAISY* descriptor at pixel $\mathbf{u}$ of the image with timestamp t. It is worth noting that *DAISY* descriptors can be compared with simple Euclidean distances.

Wang *et al.* [18] proposed the GMS-matcher, which our work is inspired from. It starts with a large number of feature points ($\approx 10,000$). The coherence constraint is incorporated as statistical likelihood measure based on the number of neighbouring matches. Our approach for enforcing coherence constraints is inspired from the GMS-matcher [18].

Unlike GMS-matcher, our approach does not start with large number of point features. Instead we rely on a much more descriptive descriptors of the point features, the *DAISY* descriptor and an efficient voting scheme to enforce the coherence constraint.

We propose the matching method in two distinct use cases. a) Given two point sets (tracked features in each image) to match them robustly. See section 5.2.2. b) Given two images with no point sets to generate dense feature matches and feature tracks in adjacent views. See section 5.2.3.

5.2.2 Guided Matching

Given two images I_t (current) and I_τ (previous), each with n and m number of tracked features by the SLAM-front-end. We denote the tracked features set as $\mathbf{v} = \{\mathbf{u}_i | i = 1 \cdots n\}$ and $\mathbf{v}' = \{\mathbf{u}'_i | i = 1 \cdots m\}$ respectively for I_t and I_τ. Let $N_j(\mathbf{u_i}) \in \Re^2$ denote the j^{th} pixel in the vicinity of $\mathbf{u}_i$. Here we describe our proposed voting scheme to achieve speedy coherence constrainted matches.

We start by computing the *DAISY* descriptor for each of the two images. Next we assign set A to be *DAISY* descriptors at each tracked pixel in v along with its neighbour pixels chosen randomly in the neighbourhood. Similarly, set A' is created using v'.

$$A = \{\mathbf{d}_{I_t}(\mathbf{u}_i)\} \cup \{\mathbf{d}_{I_t}(N_j(\mathbf{u}_i)\} \tag{5.1}$$

$$A' = \{\mathbf{d}_{I_\tau}(\mathbf{u'}_i)\} \cup \{\mathbf{d}_{I_\tau}(N_j(\mathbf{u'}_i)\} \tag{5.2}$$

We find the nearest neighbours (in descriptor space) of each item in set A from items in set A'. Note that this can be accomplished extremely fast with approximate nearest neighbour [144]. Further, since consecutive elements in A, likely match with the same elements from B, leads to more cache hits and increases overall performance.

Let i^{th} item from set A have j^{th} item from A' as the nearest neighbour in descriptor space. Let l be the index of pixel location associated with i^{th} element of set A. Let p be the index of pixel location associated with j^{th} element of set B. We create a sparse voting matrix $\mathbf{V} \in \Re^{\mathbf{n} \times \mathbf{m}}$. The i^{th} element (from set A) casts a vote for the j^{th} (in set A') at voting matrix location (l, p). This is accomplished by incrementing $V(l, p)$ by 1. For robustness we also experiment with voting with multiple nearest neighbour with tapering voting weight. For k^{th} nearest neighbor we use the weight as $\frac{1}{1+k^2}$.

It can be noted that a row of the matrix $\mathbf{V}$ (say l^{th} row) denotes a likelihood of u_l (a feature location from set v) being matched to a pixel location in v'. We accept the match if it gets a majority of votes (50% or more). Thus, the proposed process retains only those matches whose nearest neighbours in set B are consistent with those that of its spatial neighbours. This effective being a proxy for coherence constraint. Put differently, if the l^{th} feature from v is a true match of p^{th} feature from v' iff their spatial neighbours are consistent. With the retained matches we compute the Fundamental matrix with RANSAC and further eliminate matches which are not consistent with this fundamental matrix [71].

In some cases where the viewpoint difference is very high, there are not enough tracked feature matches to be matched by the method proposed. We accumulate such matches, abeit being small in number. Matches from multiple putative loop candidates are merged in a unified framework to compute a relative pose between revisits. Details of these is presented in Sec. 5.4. The inspiration is drawn from boosted learners like AdaBoost [56] which integrate multiple weak-learners into a powerful learner.

5.2.3 Dense Matching

We propose a fast and efficient method for computing new set of point matches using the *DAISY* descriptors and the cluster association images $q^{(t)}(\mathbf{u})\ \forall\mathbf{u}$ and $q^{(\tau)}(\mathbf{u}')\ \forall\mathbf{u}'$. Note that we do not need to extract key-points, rather we can directly match descriptors at every pixel and retain top matches.

We rely on the assumption that given a matching image pair, pixels in same clusters represent similar parts of the scene. Accordingly, we use cluster association maps, to vastly reduce the search space of point features. We match a pixel $\mathbf{u}_l$ (in cluster $q^{(t)}(\mathbf{u}_l)$) with a pixel $\mathbf{u}_p$ (in cluster $q^{(\tau)}(\mathbf{u}_p)$), such that $q^{(t)}(\mathbf{u}_l) = q^{(\tau)}(\mathbf{u}_p)$. See figure 5.2 and figure 5.3 for illustration of this process. Note that the cluster association maps ($q^{(t)}$ and $q^{(\tau)}$) are images of size 60x80 (ie. 1/4 dimensions of original input image) thus, the pixel co-ordinates need to be scaled appropriately.

We iterate over clusters indices which are present in both images. As noted earlier, descriptor matching(in *DAISY* space) can be done very efficiently with approximate nearest neighbour algorithm [144]. For every pixel we compute two nearest neighbors and eliminate those matches which do not satisfy the Lowe's ratio test[126]. We also use a voting

scheme as described in section 5.2.2 to enforce the coherence constraints on these matches.

We take a union of matches for each of the clusters. On this larger set of matches, we retain only those matches which satisfy the epipolar constraint [71]. Thus we arrive at point feature matching without extracting key-points, which also works for image pair with large view point difference. This produces an order of magnitude larger number of matches since it can not only match corner points but also points with no texture (but having some texture in the surrounding).

Match Expansion

Given dense point matches between image I_t and I_τ, we expand these matches on the neighbouring key-frames. We keep on expanding in the next key-frame until at-least 50% of the features can be tracked.

Thus, the features visible in I_t (current key-frame) are tracked onto the set $S = \{I_{t-1}, I_{t-2}, \cdots, I_{t-s}\}$. Similarly we also expand the dense matches in I_τ on set $F^{(+)} = \{I_{\tau+1}, I_{\tau+2}, \cdots, I_{\tau+p}\}$ and set $F^{(-)} = \{I_{\tau-1}, I_{\tau-2}, \cdots, I_{\tau+p'}\}$.

Since I_t and I_{t-1} are adjacent key-frames, a pixel $\mathbf{u_l}$ in I_t should occur in a W-neighbourhood in I_{t-1}. We search each of the matches in a 40x40 neighbourhood in I_{t-1}. This searching can be done efficiently with a fast approximate nearest neighbour search. The comparison is in *DAISY* space. For robust estimation we accept a match using a voting scheme described earlier.

5.2.4 Match Quality Assessment

There have been attempts to evaluate descriptor performances. Notable amongst these are by Mikolajczyk and Schmid [137] and more recently, by Madeo and Bober [130]. These methods suggest to use a precision-recall to evaluate descriptors. These are helpful to make a decision on the feature descriptor to use. However, after having fixed the descriptor, these methods are not suitable to evaluate performance on a per image basis. We develop a simple yet effective heuristic for quality evaluation of a matching on a per image basis. Our method is based on intuitive notions of goodness of a match. We use this heuristic score to accept the match or reject the match.

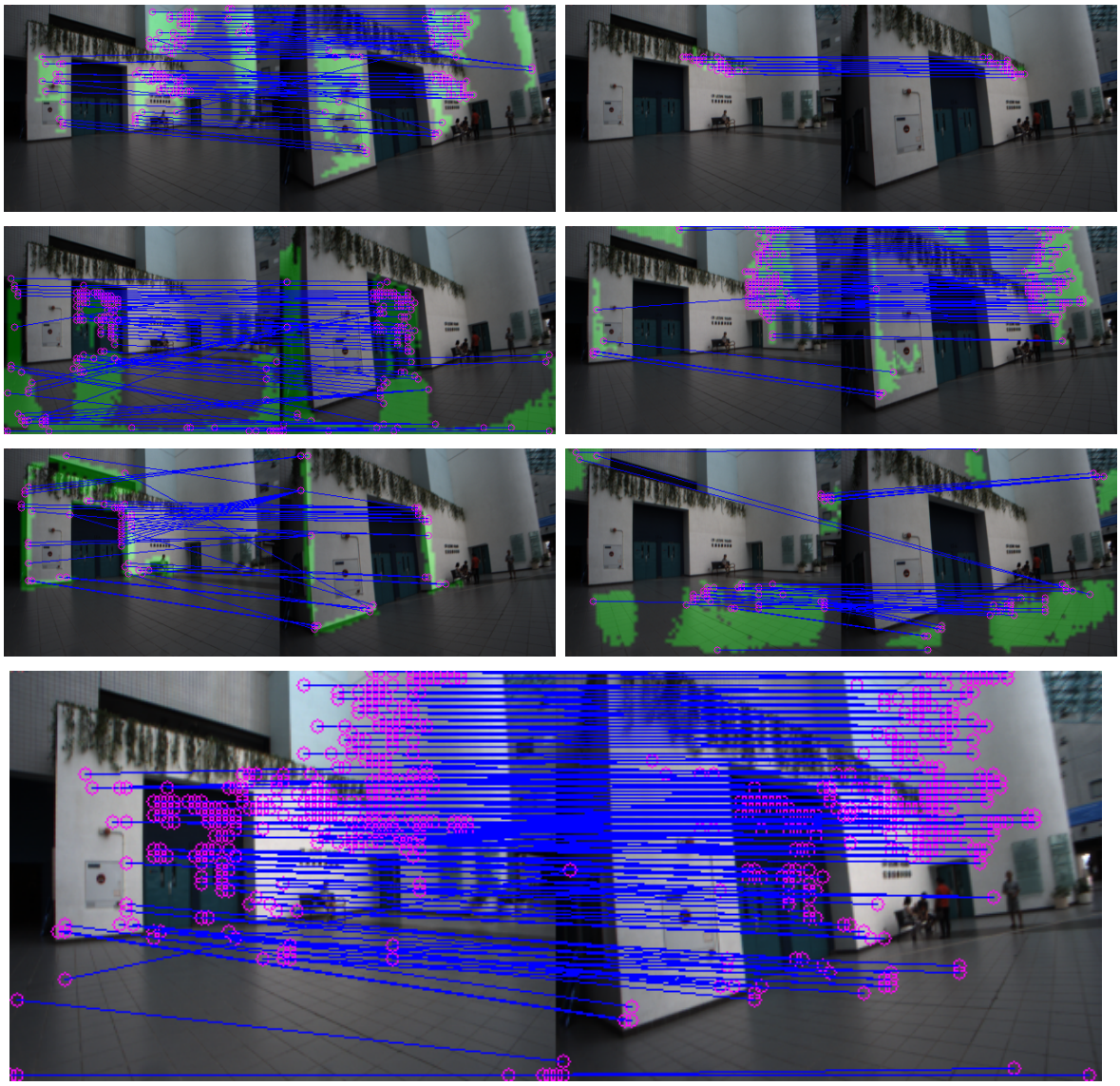

Figure 5.2: 22 consistent matches with sparse features. 180 matches with proposed method. The green overlays are the clusters with same index from association map.

Matching Quality Heuristics

The ratio of distance between the nearest neighbour and 2nd nearest neighbour gives a metric for quality of individual feature point matching[126]. Lower the ratio indicates a better match. We infer that, the distribution of the ratios gives us an estimate of the quality of the overall match. We divide the distribution into quartiles. We arrive at a matching quality score based on the number of matches in top two quartiles.

A higher fraction of retained matches, after the voting scheme proposed in section 5.2.2, indicate a more confident match. The converse however is not true. Particularly, there could be less overlapping area between the two views and the tracked features being uniformly distributed on image space. Additionally, less number of matches eliminated

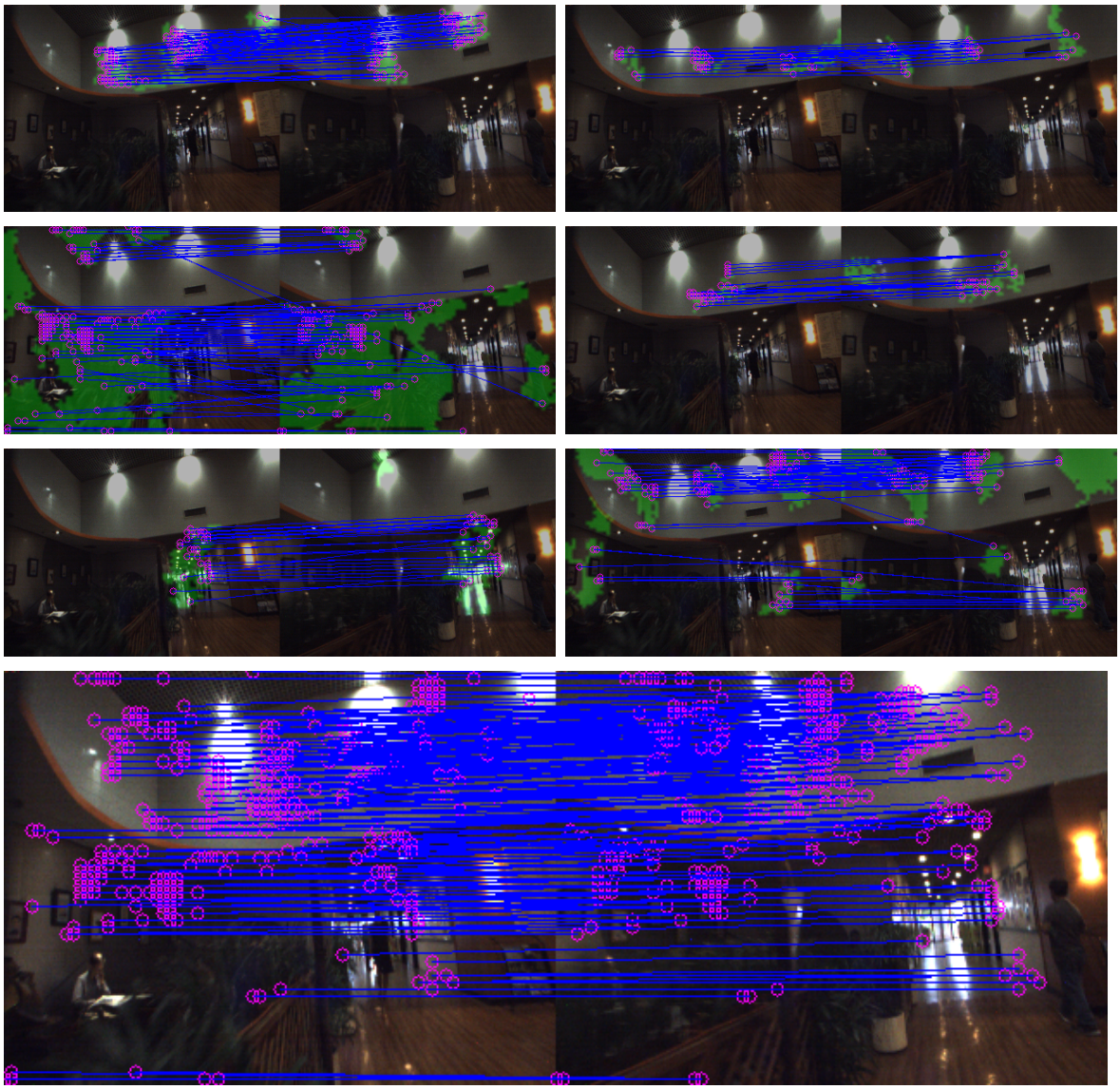

Figure 5.3: 23 consistent matches with sparse features. 312 matches with proposed method. The green overlays are the clusters with same index from association map.

by F-test also indicates a confident match. For this statement, the converse is true.

Asynchronous Implementation

In order to make our implementation quick to respond to putative matches even on embedded CPUs and to effectively use processor parallelism we use the producer-consumer model. This paradigm has also the advantage of keeping GPU always busy with data while an independent CPU thread can compute the relative pose.

The main thread queues the incoming key frames. These are consumed by the GPU thread to produce the learned image descriptors. This representation is queued into another queue which is consumed by the score computation thread. The score computation

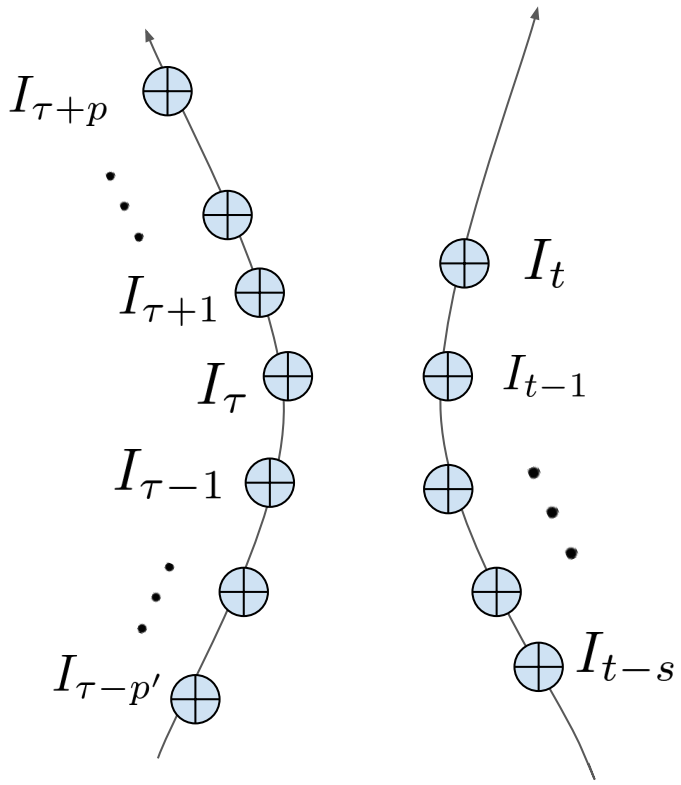

Figure 5.4: Given feature associations in vicinity of the current frame (I_t) and the previous frame (I_τ). Feature associations are a) accumulated from tracked feature matches as described in Sec. 5.2.2 and b) obtained from dense matching and match expansion as described in Sec. 5.2.3.

thread produces putative matches which are in turn queued into an independent queue. This queue of putative matches is consumed by multiple threads (in our implementation 4) which does the Daisy matching as described in earlier subsection.

5.3 Naive Relative Pose Computation

Let the set $f_Q = \{f_1, f_2, \ldots, f_q\}$ be global ids of all the features visible in set Q. Let $f'_Q = \{f'_1, f'_2, \ldots, f'_q\}$ be all the features visible in set Q'. Let $\mathbf{F_i}$ denote the 3D co-ordinates of the feature f_i in the frame-of-reference of I_t. Similarly, let $\mathbf{F'_i}$ denote the 3D co-ordinate of the feature f'_i in the frame-of-reference of I_τ.

We have a list of tuples, each containing the global ids of the feature association: $L = [L_k = \{f_i, f'_j\}|$ f_i and f_j estimates to be corresponding feature]. As noted earlier, we also maintain a inverted index of the tracked features, with which we can query the 3D co-ordinates and the frames in which the feature was visible and its image co-ordinates.

However, for the dense matching case this is not needed.

3d3d Alignments

Since we know the feature associations and their corresponding 3D points, we can essentially align the 2 3D point sets to arrive at a pose. Ee can solve the following optimization problem to arrive at the relative poses between I_t and I_τ, ie. ${}^tT_\tau$.

$$\underset{{}^t\mathbf{T}_\tau}{\text{minimize}} \sum_{\{f_i, f'_j\} \in L} ||\mathbf{F}_i - {}^t\mathbf{T}_\tau . \mathbf{F'}_j||_2 \tag{5.3}$$

3d2d alignment

A pose which minimize the reprojection of the 3D point from set Q onto the imaged points as observed in set Q' with the feature association set L is a solution to finding relative poses. Lets assume a feature F_i (in set Q) is imaged on multiple frames in set Q' at views $v_1, v_2, \ldots$ at image co-ordinates $u_{f_j}^{(v_1)}, u_{f_j}^{(v_2)}, \ldots$ respectively. Note that $\{i, j\} \in L$.

$$\underset{{}^{v_1}T_t, {}^{v_2}T_t, \ldots}{\text{minimize}} \sum_k \sum_{\forall \{k,j\} \in L} \sum_{\forall v} ||\mathbf{u}_{f'_j}^{(v)} - \pi({}^v\mathbf{T}_t . \mathbf{F}_k)||_2 \tag{5.4}$$

A similar equation can be constructed which uses 3D points from the set Q' and the imaged points from set Q. Assuming the points are imaged at views $v'_1, v'_2, \ldots$ the corresponding optimization problem is:

$$\underset{{}^{v'_1}T_t, {}^{v'_2}T_\tau, \ldots}{\text{minimize}} \sum_k \sum_{\forall \{i,k\} \in L} \sum_{\forall v'} ||\mathbf{u}_{f'_i}^{(v')} - \pi({}^{v'}\mathbf{T}_\tau . \mathbf{F'}_k)||_2 \tag{5.5}$$

5.4 Non-linear Optimization based Pose Estimation

In this section we start by describing the notations. Next we formulate the problem of spatial alignment of image sequence as a distance minimization for their corresponding 3D points. After that we show an alternating minimization based algorithm can be used for fast and robust computation of the 6-DOF relative pose between the image sequence. Finally we propose an iterative local bundle refinement step.

5.4.1 Notations

Assume, we have two local image sequences (with subscripts a and b). Let $I_{a_0}, I_{a_1}, ...I_{a_n}$ and $D_{a_0}, D_{a_1}, ...D_{a_N}$ be the intensities and depth images for sequence-a. Similarly, let $I_{b_0}, I_{b_1}, ...I_{b_m}$ and $D_{b_0}, D_{b_1}, ...D_{b_M}$ be the intensities and depth images for sequence-b. Additionally, we also are given as input the camera poses for both the sequences which we denote by ${}^{w}\mathbf{T}_{a_0}, {}^{w}\mathbf{T}_{a_1}, ...{}^{w}\mathbf{T}_{a_N}$ and ${}^{w'}\mathbf{T}_{b_0}, {}^{w'}\mathbf{T}_{b_1}, ...{}^{w'}\mathbf{T}_{b_M}$. Note that the relative poses for nearby frames (odometry) is accurately known from the underlying odometry system. The relative poses between the two sequences ${}^{w}\mathbf{T}_{w'}$ as obtained from the odometry, at best has significant drift and at worse in case of kidnap is unknown, and just cannot be directly used for refinement with an iterative method. The camera poses can be obtained by a visual-inertial system (like VINS-MONO [167], ORB-SLAM [146], etc.). The camera poses are expressed in a reference frame denoted by w (for sequence-a) and w' (for sequence-b). The world frames of both the sequences can in general be different.

Let ${}^{a_0}X_j^{(a)}\ \forall j = 1, 2, ..., K$ and ${}^{b_0}X_j^{(b)}\ \forall j = 1, 2, ..., K$ be the corresponding 3D-3D pointset. ${}^{a_0}X_j^{(a)}$ is a 3D point (indexed by j) in the sequence-a expressed in the frame-of-reference of the first camera in that sequence (ie. a_0). Similarly, ${}^{b_0}X_j^{(b)}$ be a 3D point from sequence-b expressed in the frame-of-reference of the first camera in that sequence (ie. b_0). Note that the 3D points may or may not be visible in the 1st frame of the sequence, they are merely expressed in that co-ordinate system.

5.4.2 Problem Formulation

Our objective is to find a rigid transform (rotation and translation) that align the point sets. Since the correspondences are obtained by image level features, they are prone to false matches and noise. We formulate the problem of alignment of image sequences as a distance minimization problem between the correspondence. Precisely, we define,

$$f(X_j, \mathbf{R}, \mathbf{t}) = || \ {}^{a_0}X_j^{(a)} - \mathbf{R}\ {}^{b_0}X_j^{(b)} - \mathbf{t}||_2^2 \tag{5.6}$$

where, $\mathbf{R}, \mathbf{t}$ together is the pose of camera b_0 from camera a_0, ie. the relative pose between the two sequences.

For identification of outliers and spurious matches we make use of switch constraints optimization variables. The core idea is to have additional optimization variables (s_j) for

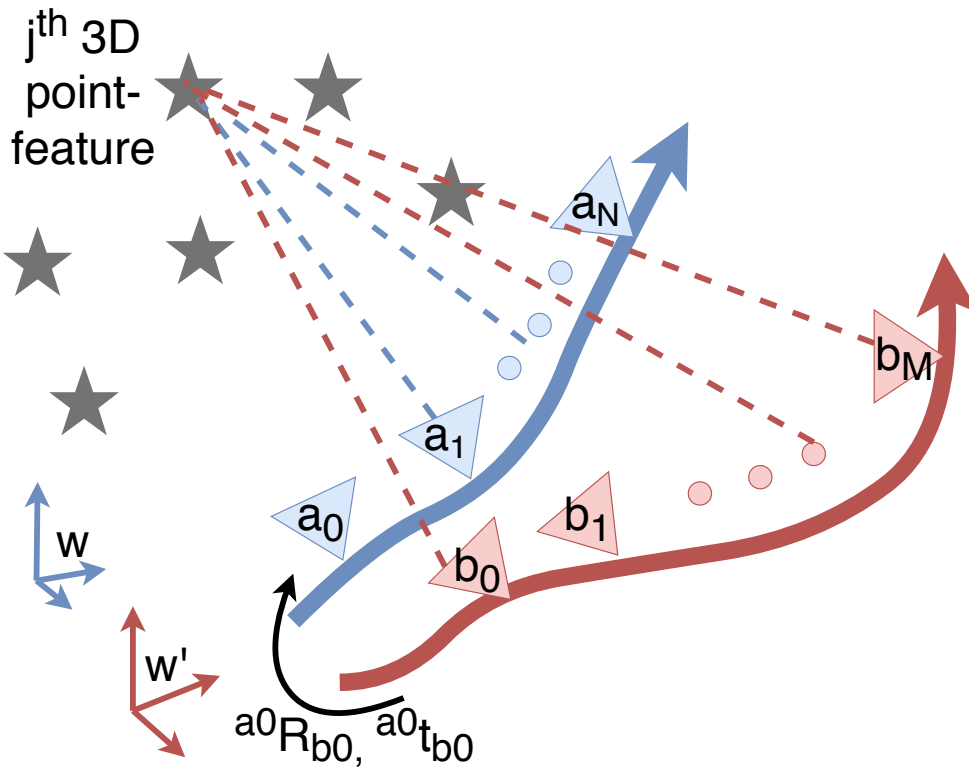

Figure 5.5: Notations for the proposed method.

every point-feature correspondence. In order to avoid the $s_j = 0$ as the trivial solution, we need to augment the cost function to penalize the reduction of s_j. We note that such approach is inspired by Sunderhauf *et al.* [198] who used it for the solution of pose graph optimization problem in SLAM under wrong loop pose graph edges. Zhou *et al.* [228] have applied a similar formulation for point cloud registration problem.

$$\begin{aligned} \underset{\mathbf{R},\mathbf{t},s_j j=1...K}{\text{minimize}} \quad & F(\mathbf{R},\mathbf{t},\mathbf{s}) = \sum_{j=1}^{K} s_j^2 f(X_j,\mathbf{R},\mathbf{t}) + \lambda(1-s_j)^2 \\ \text{s.t.} \quad & \mathbf{R} \in SO(3) \end{aligned} \tag{5.7}$$

where, λ is a fixed constant. $SO(3)$ denotes the special orthogonal group. We shall collectively refer to the $s_j, j = 1...K$ as boldfaced $\mathbf{s}$.

5.4.3 Image Level Feature Correspondence Aggregation

Often times one pair of images do not have sufficient number of point matches. Also in case of larger view point difference just one image pair cannot give sufficient number of point matches for reliable pose computation. In this approach we propose to use multiple image pairs and aggregate the feature correspondences. Since we know the depth images at each of the image sequence, we could get the 3D point by inverse projection. In case of

monocular camera, we could get such depth estimate by tracking-triangulation of these keypoints in nearby frames. Further since we also know the camera pose at each of the frame, we could represent all the 3D point in one co-ordinate frame reference for that sequence (without loss of generality, the first frame of the sequence).

We proceed by drawing several random image pairs, one image from sequence-a (a_p) and another image from sequence-b (b_q). Next we compute feature correspondence in image I_{a_p} and I_{b_q}. We experiment by using widely used ORB features and descriptor [143] with ratio test [126]. Also we experiment with GMS-matcher [18]. Let $\mathbf{u}_i := (u_i, v_i)$ and $\mathbf{u'}_i := (u'_i, v'_i)$ be one of the S number of image correspondences (indexed by i). Knowing the depth value at that pixel location using D_{a_p} and D_{b_q}, we can obtain the 3D point in the camera frame as:

$$\begin{aligned} {}^{a_p}X_i^{(a_p)} &= \pi^{-1}((u_i, v_i), D_{a_p}) \\ {}^{b_q}X_i^{(b_q)} &= \pi^{-1}((u'_i, v'_i), D_{b_p}) \end{aligned} \tag{5.8}$$

where, $\pi(.) : \mathbb{R}^3 \to \mathbb{R}^2$ is the perspective projection function and $\pi^{-1}(.) : (\mathbb{R}^2, R) \to \mathbb{R}^3$, is the inverse reprojection function for the camera model. For our experiments, we use the standard pinhole camera model. Finally, we transform the co-ordinate reference to respective first frame in the sequence and concatenate the matches by drawing multiple random image pairs.

$$\begin{aligned} {}^{a_0}X_i^{(a)} &= ({}^{w}\mathbf{T}_{a_0})^{-1}\ {}^{w}\mathbf{T}_{a_p}\ {}^{a_p}X_i^{(a_p)} \\ {}^{b_0}X_i^{(b)} &= ({}^{w'}\mathbf{T}_{b_0})^{-1}\ {}^{w'}\mathbf{T}_{b_q}\ {}^{b_q}X_i^{(b_q)} \end{aligned} \tag{5.9}$$

Thus, we shall have several 3D-3D correspondences derived from image correspondences. We are well aware that it is also possible to obtain 3D-3D correspondence from point-cloud feature descriptors like FPFH [176]. However as noted by Yang and Carlone [216] such point-cloud feature result in large number of false matches. This problem is even more severe in case of a sparse point-cloud. Also, the computational time to build local point cloud is much higher than computation of image-image point correspondences.

5.4.4 Solving with Alternating Minimizations

For solving the proposed optimization problem (equation 5.7), we adopt the strategy of Alternating Minimizations (AM) [27, 39]. It provides for a fast and efficient solution without involving Gauss-Newton iterations. The AM algorithm involves an objective

function with multiple optimization variable blocks. The iterations proceed by starting with an initial guess of only one of the optimization variable blocks (say all s_j). This is in contrast to the Gauss-Newton like algorithms which start with an initial guess for all the optimization variables.

We shall superscript the optimization variables to denote the iteration number. The iterations start with an initial guess only for $\mathbf{s}$, $\mathbf{s}^{(0)} := \mathbf{1}$. The iterations cycle between the following two steps,

$$\begin{aligned} \mathbf{R}^{(n-1)}, \mathbf{t}^{(n-1)} &= \underset{R,t}{\operatorname{argmin}}\ F(\mathbf{R}, \mathbf{t}, \mathbf{s}^{(n-1)}) \\ s^{(n)} &= \underset{s}{\operatorname{argmin}}\ F(\mathbf{R}^{(n-1)}, \mathbf{t}^{(n-1)}, \mathbf{s}) \end{aligned} \tag{5.10}$$

Turns out each of these steps can be solved in closed form. We give details in this regard in the following subsections. In the next two subsections for notational brievety, we denote ${}^{a_0}X_j^{(a)}$ by a_j and ${}^{b_0}X_j^{(b)}$ by b_j.

Optimize $\mathbf{R}, \mathbf{t}$ when $\mathbf{s}$ are constant

$$\underset{\mathbf{R},\mathbf{t}}{\operatorname{argmin}}\ h = \sum_{j=1}^{K} s_j^2 ||a_j - \mathbf{R}b_j - \mathbf{t}||_2^2 \tag{5.11}$$

When $\mathbf{s}$ is fixed, this can be viewed as the weighted sum of squared distance between 3d-3d correspondence. This objective can be solved in the closed form [9, 75, 125]. The closed form method, involve mean centering, singular value decomposition of a 3×3 matrix and matrix multiplications. This usually scales very well. Thus, when given the weights as input, ie. $\mathbf{s}^{(n-1)}$ we can compute $\mathbf{R}^{(n-1)}, \mathbf{t}^{(n-1)}$.

Optimize $\mathbf{s}$ when $\mathbf{R}, \mathbf{t}$ are constant

$$\underset{s_j j=1\ldots K}{\operatorname{argmin}} g = \sum_{j=1}^{K} s_j^2 f(X_j, \mathbf{R}^{(n-1)}, \mathbf{t}^{(n-1)}) + \lambda(1 - s_j)^2 \tag{5.12}$$

At the minimum point (ie. $\mathbf{s}^*$), the gradient of the function, g, has to be zero. Thus $\frac{\partial g}{\partial s_k}\Big|_{s_k^*} \to 0,\ \forall\ k = 1\ldots K$.

$$\begin{aligned} \frac{\partial g}{\partial s_k} &= 2s_k f(X_j, \mathbf{R}^{(n-1)}, \mathbf{t}^{(n-1)}) - 2\lambda(1 - s_k) \\ &= 2s_k[f(X_j, \mathbf{R}^{(n-1)}, \mathbf{t}^{(n-1)}) + \lambda] - 2\lambda \\ \text{so, } s_k^* &= \frac{\lambda}{f(X_j, \mathbf{R}^{(n-1)}, \mathbf{t}^{(n-1)}) + \lambda} \end{aligned} \tag{5.13}$$

thus, we can compute the next iterate of $\mathbf{s}$ in closed form given the inputs $\mathbf{R^{(n-1)}}, \mathbf{t^{n-1}}$, $s_k^n \leftarrow s_k^* \ \forall \ k = 1...K$. We additionally note that all s_k are thus restricted to have values between zero and one, since $f(X_j, R, t)$ is always positive for all inputs. This ${}^wT_{w'} =^{a0} \hat{T}_{b0} := [\mathbf{R}^*|\mathbf{t}^*]$.

5.4.5 Local Bundle Refinement

Next we do the relative pose refinement by formulating the problem as a local bundle refinement for the sequences. The optimization variables are the poses at frames. ${}^w\mathbf{T}_{a_0}, {}^w\mathbf{T}_{a_1}, ... {}^w\mathbf{T}_{a_N}$ and ${}^{w'}\mathbf{T}_{b_0}, {}^{w'}\mathbf{T}_{b_1}, ... {}^{w'}\mathbf{T}_{b_M}$. The optimization problem involves two types of residue terms ie. 1) *Odometry Residues*, 2) *Reprojection Residues*.

Odometry Residues

The observed relative poses between the frames of one sequence are denoted by, ${}^{a_i}\hat{\mathbf{T}}_{a_{i+f}}, f = 1, ..., F$, we use $F = 4$. These are obtained from the outputs of the SLAM system. The main motivation for these terms is to constraint the relative poses between the frames of a sequences.

$$\delta_{ai}^{odom} = ({}^w\mathbf{T}_{a_i})^{-1} \ {}^w\mathbf{T}_{a_{i+f}} \ominus {}^{a_i}\hat{\mathbf{T}}_{a_{i+f}} \tag{5.14}$$

where the operator $A \ominus B; A \in SE(3), B \in SE(3)$, is the pose difference and defined as $A^{-1} \times B$. We also add similar terms for sequence-B, and denote it as δ_b^{odom}. The error terms (scalars) for δ_a^{odom} is obtained as:

$$\begin{aligned} r_a^{odom} := \sum_{\forall a_i} & ||euler\_angles(\delta_{ai}^{odom})||_2^2 \\ & + \lambda ||translation(\delta_{ai}^{odom})||_2^2 \end{aligned} \tag{5.15}$$

Reprojection Residues

We have several randomly picked image pairs $\gamma = (a_i, b_j), |\gamma| = V$. One of the image from sequence-a, a_i and another from sequence-b, b_j. For each of the image-pairs (viz. I_{a_i} and I_{b_j}) we also have K number of point correspondences. The point feature correspondences $(\mathbf{u}_k^{(a_i)} \leftrightarrow \mathbf{u}_k^{(b_j)})$ are indexed by k, where, $\mathbf{u}_k^{(a_i)} := (u_k^{(a_i)}, v_k^{(a_i)})$ and $\mathbf{u}_k^{(b_j)} := (u_k^{(b_j)}, v_k^{(b_j)})$.

The residue using the 3d points from I_{a_i} and 2d points from I_{b_j}:

$$C_{b_j}^{a_i} = ||\pi\Big({}^{bj}\mathbf{T}_{b0}\ {}^{b0}\hat{\mathbf{T}}_{a0}\ {}^{a0}\mathbf{T}_{ai}\ \pi^{-1}[\mathbf{u}_k^{(a_i)}, Z_k^{(a_i)}]\Big) - \mathbf{u}_k^{b_j}|| \tag{5.16}$$

The residue using the 2d points from I_{a_i} and 3d points from I_{b_j}:

$$C_{a_i}^{b_j} = ||\pi\Big({}^{ai}\mathbf{T}_{a0}\ {}^{a0}\hat{\mathbf{T}}_{b0}\ {}^{b0}\mathbf{T}_{bj}\ \pi^{-1}[\mathbf{u}_k^{(b_j)}, Z_k^{(b_j)}]\Big) - \mathbf{u}_k^{a_i}|| \tag{5.17}$$

Solution to the Minimization

Combining the above terms, we solve the resulting non-linear least squares optimization (equation 5.18).

$$\min_{\substack{{}^{w}\mathbf{T}_{ai}\ ,\forall a_i \\ {}^{w'}\mathbf{T}_{bi}\ ,\forall b_i}} r_a^{odom} + r_b^{odom} + \sum_{\forall (i,j)\in\gamma} (C_{b_j}^{a_i})^2 + (C_{a_i}^{b_j})^2 \tag{5.18}$$

We make use of the ceres-solver [3] to this effect. We parameterize the poses with 7 parameters (4 quaternions, 3 translation). For a practical implementation we hold the variables ${}^{w}\mathbf{T}_{ai}$ as constant and optimize only the poses for b_j. Also instead of using the frame pose in the world co-ordinate system, it makes more sense to express the pose in the first frame of the sequence. We initialize the poses of the sequence-b using the coarse estimate of relative transform between the two sequence as described in the previous section.

After solving the optimization, the relative poses, $({}^{w}\mathbf{T}_{ai})^{-1} \times^{w'} \mathbf{T}_{bj}$ represent the observed relative poses at loop candidates and can be used with a pose graph optimization engine to correct the drift or recover from kidnap.

5.4.6 Loop Hypothesis Sequence Construction

The above process for initialization of a pose from 3D points and non-linear refinement assume that we already know a loop sequence. In our work, we move from the term loop-closure candidate to loop-sequence hypothesis to indicate that a revisit is not just defined by a pair of images but by a sequence of images. As described in previous section, we draw random image pairs from this sequences for robust pose estimation. For image description, we use from our previous work [99], weakly supervised whole-image descriptor. Every image is represented by the 1024-D vector. The learned whole-image descriptor has the property that images from the same physical scene but different view point has dot

product nearer to 1.0 and the dot products between image descriptors of different physical scenes has a lower dot product value. In other words, similar image-pair have a high dot product value than dis-similar image-pair.

We propose a grid based voting-cummulation based scheme for detecting coherent loop sequences. The main motivation comes from the observation that the nearest neighbours of the current frame and the nearest neighbours in the W-window of the current frame are also in a W'-window of the nearest neighbour frame of the current frame. In literature, this property is often referred as coherence. See figure 5.6 for an illustration of coherent and non-coherent sequences. Such voting scheme provide a quick way to implement temporal coherence. Such voting schemes have been used in other domain, we are inspired from the GMS-matcher [18] to adapt such a scheme for our work.

In our implementation, we make a temporal grid of, $W = 30$ keyframes in each bin. We take 5 nearest neighbours for each keyframe which have a dot product higher than the threshold (we used 0.75). Each nearest neighbour casts a vote which are cummulated for every 50 keyframes (2 sec) to obtain a loop hypothsis sequence.

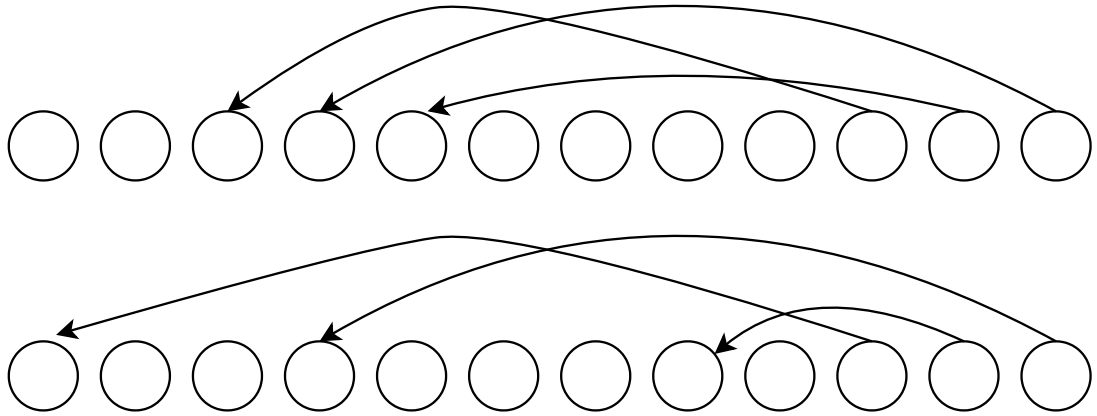

Figure 5.6: The circles represent the image frames arranged in temporal order. The arrow represents the nearest neighbours of the keyframe in the descriptor space. Illustration of temporally coherent loop sequence(top). Botton image shows non coherent loop sequence and more likely to be a false match.

5.5 Kidnap Detection and Recovery

In our system we also deal with the kidnap recovery mechanism. By kidnap we refer to the camera's view being blocked and the camera teleported to another location several 10s to 100s of meters away in 10s to 100s of seconds. The teleported location may or may not be a previously seen location. We make use of a simple criterion like the current

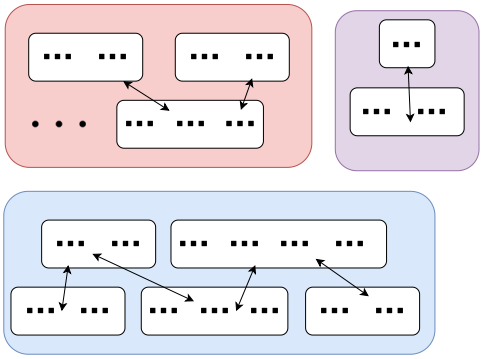

Figure 5.7: The solid black squares represent the nodes in the pose graph. Arrows show the loopcandidates. The white rectangles show each of the individual worlds. The colored rectangular enclosures are the worlds belonging to the same set.

number of tracked features falling to a very low value (like less than 10) to determine if the camera system is kidnapped. Once we determine that the camera system has been kidnapped we stop the visual inertial odometry subsystem. When sufficient number of features are again being tracked we reinitialize the odometry/sensor fusion system. It is to be noted that in such a case it starts with a new co-ordinate reference. Hence forth we refer to the new co-ordinate systems as world-0 (w_0), world-1 (w_1), world-2 (w_2) and world-k (w_k) in general. We denote the nodes n by a superscript to identify the world that it is in. For instance $n_i^{(k)}$, means the i^{th} node (i is the global index of the node) is in the k^{th} world. The odometry pose of the node is denoted as ${}^{(k)}\mathbf{T}_i$.

The incoming loopcandidate pair can be categorized into two kinds (refer to figure 5.7 for a visual explanation). a) Intra worlds (eg. $n_i^{(k)} \leftrightarrow n_j^{(k)}$) and b) Inter worlds ($n_i^{(k)} \leftrightarrow n_j^{(k')}$).

5.6 Loop Edge Pose Computation

In this section we detail some of the issues involved in pose computation under large viewpoint changes.

Direct Pose Computation Between w_k and $w_{k'}$

The inter-world loopcandidates $n_i^{(k)} \leftrightarrow n_j^{(k')}$ can be used to infer the relative poses between the co-ordinate system w_k and $w_{k'}$. In this section, we describe the computation of the relative pose between the worlds k and k', ie. ${}^{(k)}\mathbf{T}_{(k')}$ from the relative pose between the two nodes (${}^{i}\mathbf{T}_j$) and the odometry poses of the nodes in their respective worlds ie. ${}^{(k)}\mathbf{T}_i$

and ${}^{(k')}\mathbf{T}_j$ respectively.

$$ {}^{(k)}\mathbf{T}_{(k')} = {}^{(k)}\ \mathbf{T}_i \times^i \mathbf{T}_j \times ({}^{(k')}\mathbf{T}_j)^{-1} \tag{5.19} $$

Indirect Pose Computation Between w_k and w_{k_1}

It is easy to see that if we have two inter-world loopcandidates like: $n_i^{(k)} \leftrightarrow n_j^{(k')}$ and $n_{i_1}^{(k_1)} \leftrightarrow n_{j_1}^{(k')}$, the three worlds w_k, $w_{k'}$ and w_{k_1} are said to be in same set. It is also possible to indirectly infer the relative pose between worlds w_k and w_{k1} even though no loopcandidate exists between these two sets. This estimate is needed for correctly initializing the poses to solve the pose graph optimization problem.

We make use of the data structure disjoint sets [38] to maintain the world information that are in the same set. The advantage of the disjoint set datastructure is it provides for a constant time set-union and sub-linear time set-association query. Each world starts in its own set, everytime we encounter the inter-world looppair we merge these two sets of the worlds into a single set.

When we assert that two different worlds are in the same set, we imply that a relative transform between these two worlds can be determined. However that a loopcandidate between these two pairs of worlds may or may not exist. In case no loopcandidate exists between the two worlds but these two worlds are in the same set, the relative poses between the worlds can be determinted by finding a graph-path between the two worlds and chaining the relative pose estimates between the adjacent world pairs in the path.

In a general scenario, this can be accomplished by constructing a directed graph of the worlds with nodes being the worlds in the same set and edges being the relative poses between these two worlds, ie. ${}^{(k)}\mathbf{T}_{(k')}$. A breadth-first search on this graph is sufficient to determine an estimate of relative poses between arbitrary pairs of worlds by chaining the relative poses of the path generated by the graph search.

5.7 Implementation Details

Our full system (see Fig. 5.8) employs multiple threads. It uses the producer-consumer programming paradigm for managing and processing the data. In our system, thread-1 produces image descriptors of all incoming keyframe images. Thread-2 consumes the image descriptors to produce candidate matches. Thread-3 consumes the candidate matches

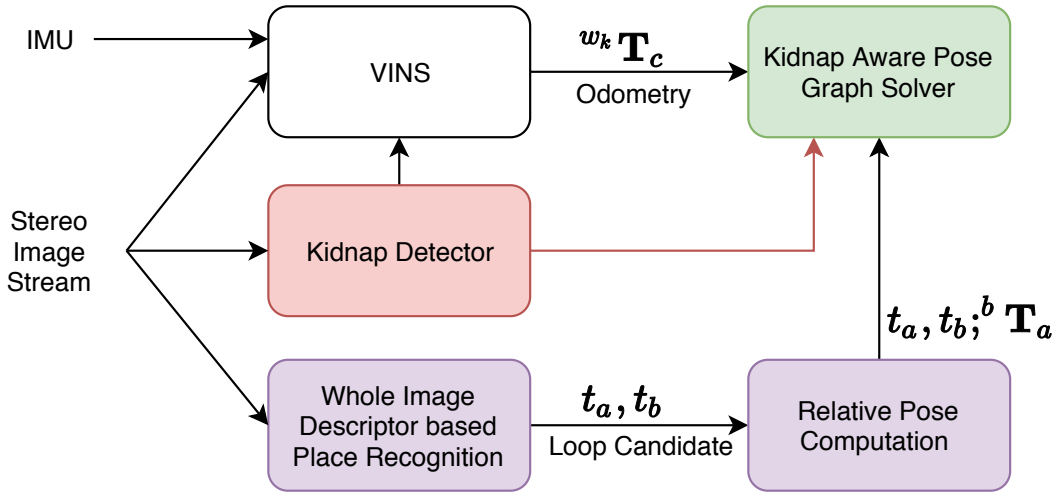

Figure 5.8: System Overview

to produce feature correspondences. Thread-4 uses the feature correspondence to produce the relative pose ${}^i\mathbf{T}_j$. Thread-5 monitors the number of tracked features to know if the system has been kidnap. Thread-6 uses the loopcandidates and their relative poses to construct the disjoint set data structure and maintain the relative poses between multiple co-ordinate systems as detailed in Sec. 5.6 and 5.6. Thread-7 incrementally constructs and solves the pose graph optimization problem while carefully initializing the initial poses and making use of poses between the worlds. Our pose graph solver is based upon the work of Sunderhauf *et al.* [198]. A separate thread is used for visualizing the poses. Even though we use 7 threads, the effective load factor on the system is about 2.0. This means about two cores are occupied by our system (this does not include the processing for VINS-Fusion Odometry System).

We demonstrate the working of our full system (see Fig. 5.9). It is worth noting that in addition to reducing the drift on account of loopclosures, our implementation can reliably identify and recover from kidnap scenarios lasting longer than a minute online in realtime. We attribute such robustness to the high recall rates of the NetVLAD based image descriptor engine. The results in regard to the operation of the full system can be found in the attached video. We record our own data for demonstrating the kidnap cases. Some of the kidnap cases are labelled hard which include the need for indirect inference which is not available in the previous slam systems including the relocalization system from Qin *et al.*[166].

5.8 Experiments

We also experiment with our entire system involving relative pose computations at the loopcandidates and pose graph solver with kidnap recovery mechanism. Our experimental setup involves just the 'MYNT EYE D' [3] camera. It includes a stereo camera pair and an 200 Hz IMU with frame and IMU sync of about 1 ms. We kidnap the camera by blocking the view of the camera and transporting it to another location. Additionally, we also experiment with the EuRoC MAV dataset [24] which also have stereo camera data and IMU.

A representative live real-time run of the system is shown in Fig. 5.9. Our system can identify and recover and relocalize from kidnaps online and in realtime. The main contribution here was the use of disjoint sets to hold the set associations of the co-ordinate systems and the use of breadth-first-search to infer relative poses between co-ordinate systems.

5.8.1 Accuracy on varying number of edge-points

Next we present the effect of using varying number of edge point on accuracy of pose computation. For comparison we use adjacent keyframes and the baseline pose from VIO system. We compute the pose for the same pair of images using varying number of edge points. The sampling was done randomly. The general trend is that more the number of edge points we use, more accurate is the relative pose estimates. Fig. 5.11 and Fig. 5.12 shows an example image with about equal proportion of strong edges and texture. In such case using too few points results in good estimates, this can be attributed to high proportion of inlier strong edge points. As we increase the number of points we get worse estimates, this effect is due to fact that more texture points being added which negatively affects the accuracy. As we increase the number of points further we see a higher accuracy for pose computation. Fig. 5.13, shows an example with mostly strong edges. In this case, even with a small number of points we are able to produce rotation error in the range of about 0.5 degrees. Using more points results in marginal improvement of accuracy.

[3] https://www.mynteye.com

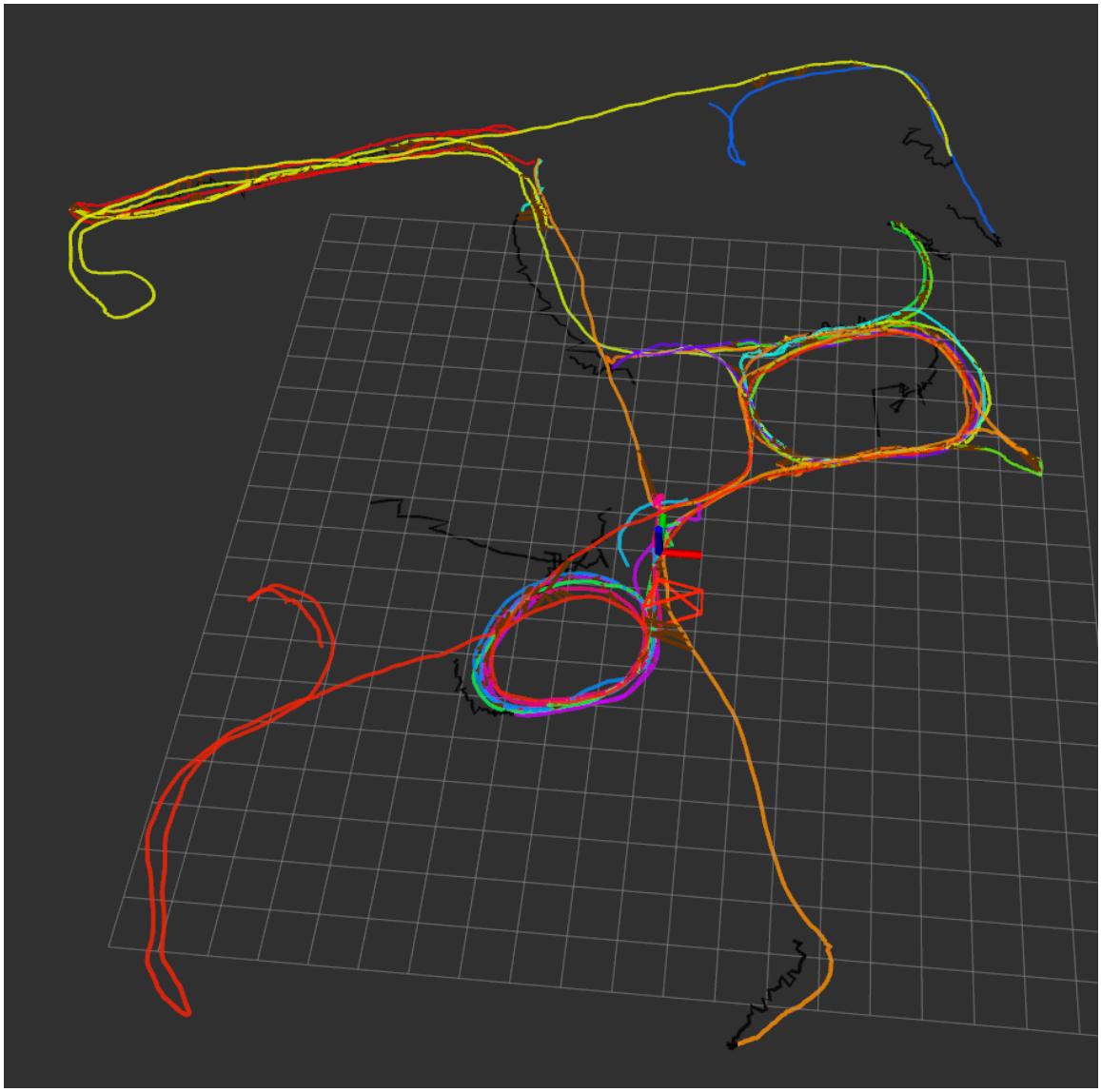

Figure 5.9: Shows the corrected trajectories (different colors for different worlds) merged according to the inter-world loop candidates. Note that the merging occurs live (not offline) in real-time as the loop candidates are found. We also note that such cases cannot be handled by Qin *et al.*[166] which just merges with the world-0 (first world) and ignore any inter-world loop candidates not involving world-0. In this sequence involve multiple kidnaps lasting from 10s to 30s. The video for the live run is available at the link: `https://youtu.be/3YQF4_v7AEg`. Live runs videos are available for more sequences through this link: `https://www.youtube.com/playlist?list=PLWyydx20vdPzs5VVhZu0TGsReT7U17Fxp`

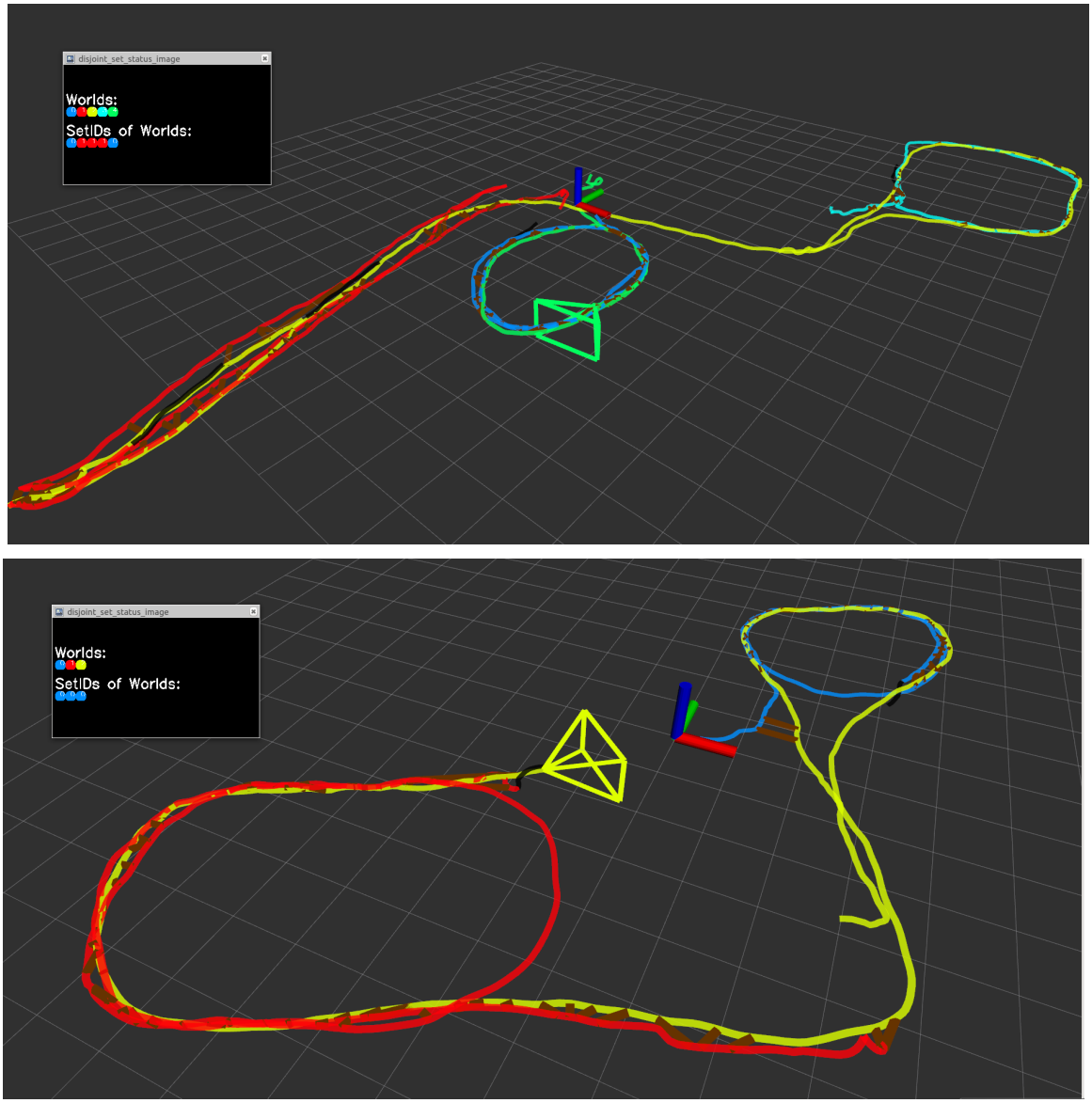

Figure 5.10: Live kidnap detection and relocalization. These sequences involve large kidnaps. The set associations are managed with a disjoint-set datastructure. The live run videos for these sequences can be accessed through `https://youtu.be/h8uuR17b0xM` (top) and `https://youtu.be/KDRo9LpL6Hs` (bottom).

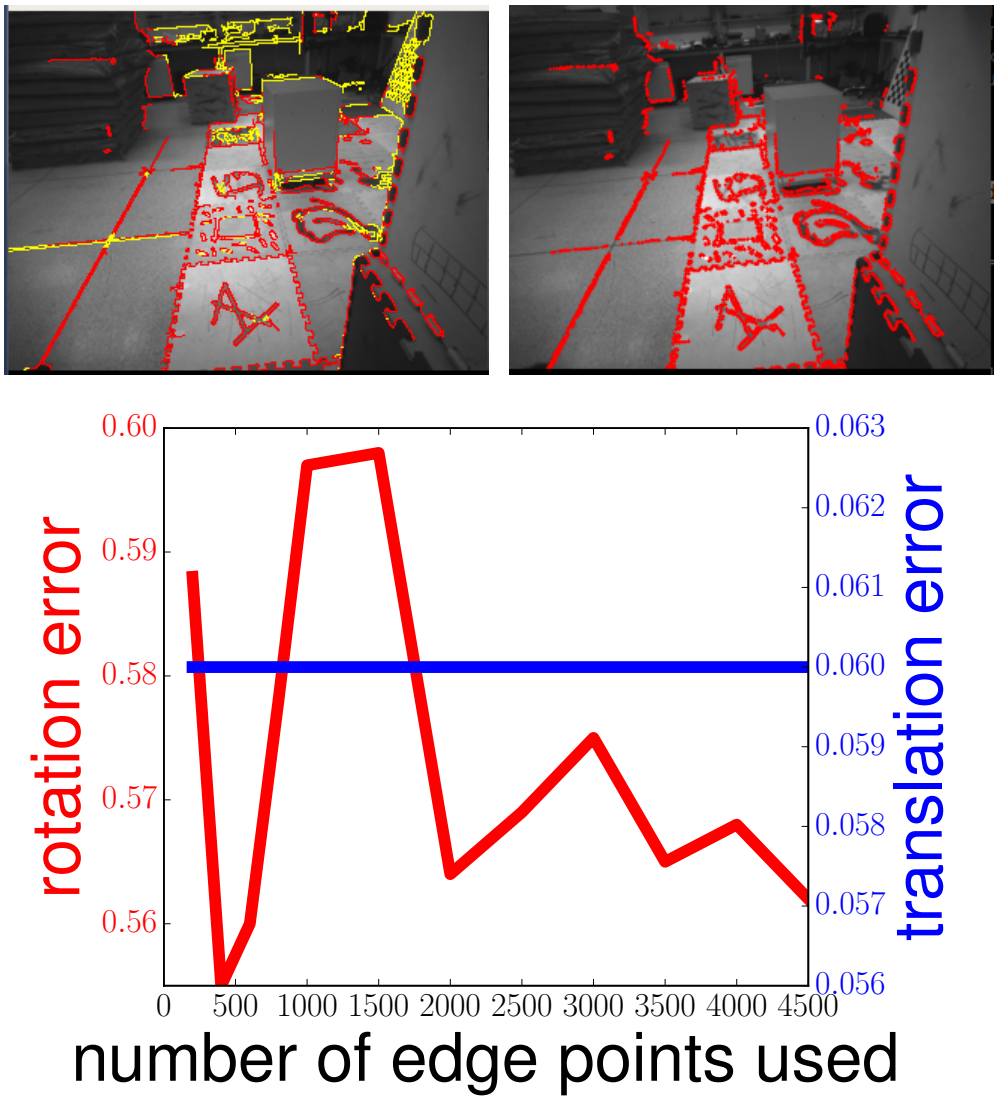

Figure 5.11: The effect of using varying number of edge-points (x-axis) on the accuracy of pose computation. Top images shows detected edge-points (left) and reprojection using the computed pose with 4500 edge-points. Rotation errors in degrees (left-axis, in red), translation errors in meters (right-axis in blue). Best viewed in color.

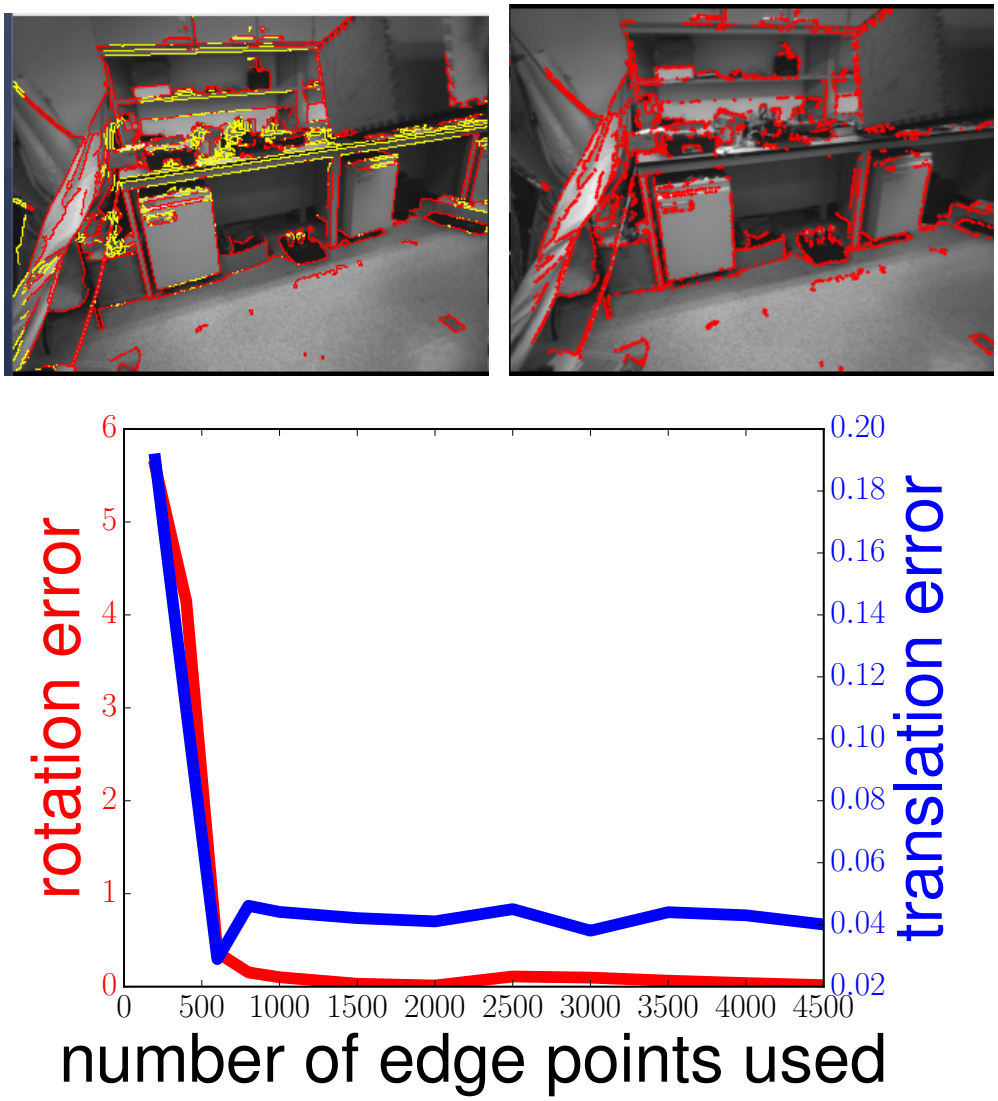

Figure 5.12: The effect of using varying number of edge-points (x-axis) on the accuracy of pose computation. Top images shows detected edge-points (left) and reprojection using the computed pose with 4500 edge-points. Rotation errors in degrees (left-axis, in red), translation errors in meters (right-axis in blue). Best viewed in color.

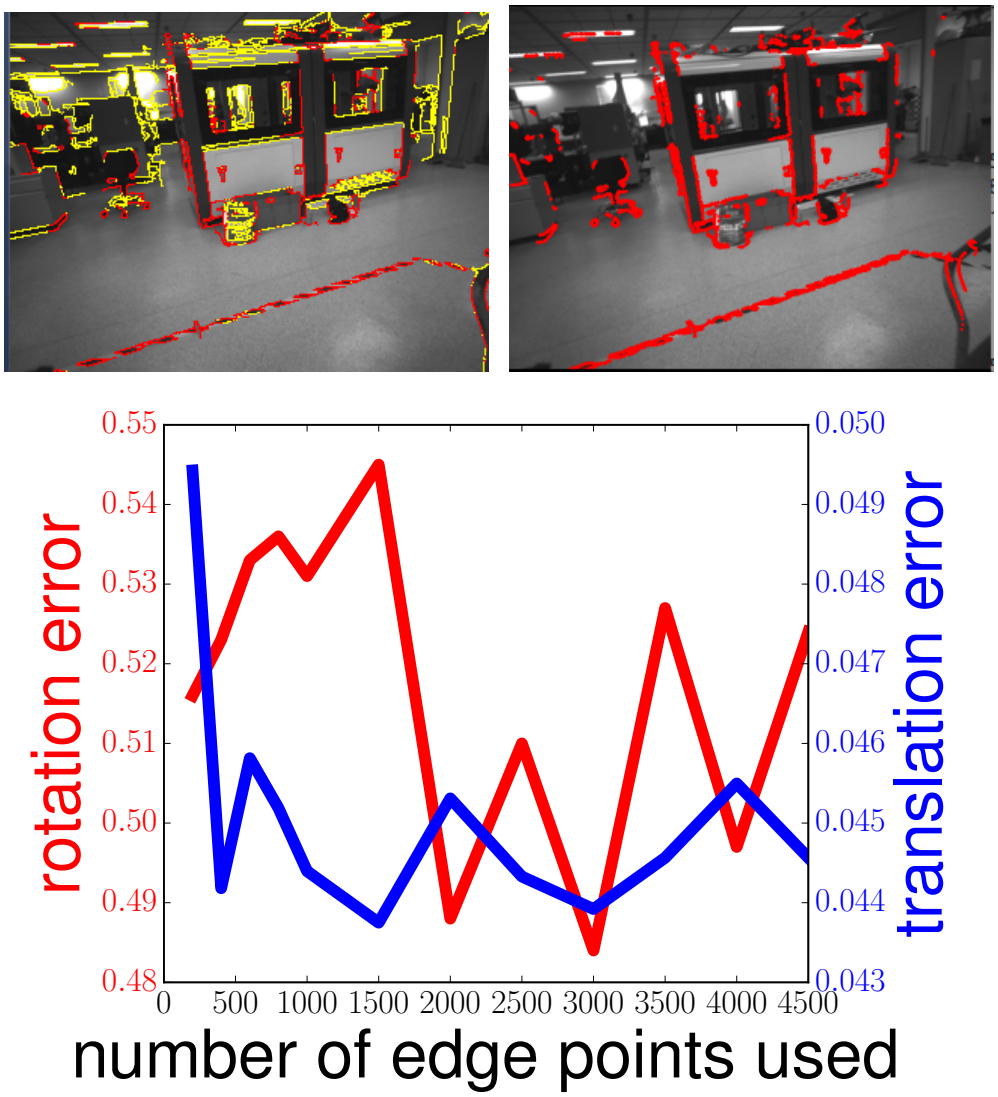

Figure 5.13: The effect of using varying number of edge-points (x-axis) on the accuracy of pose computation. Top images shows detected edge-points (left) and reprojection using the computed pose with 4500 edge-points. Rotation errors in degrees (left-axis, in red), translation errors in meters (right-axis in blue). Best viewed in color.

	EA-1		EA-2		PNP	
sequence	mean	std. dev.	mean	std. dev.	mean	std. dev.
db-a	0.415	0.44	0.32	0.31	0.52	0.38
db-b	0.383	0.76	0.25	0.54	0.55	0.72
db-c	0.556	0.83	0.29	0.27	0.47	0.37
db-d	0.607	0.69	0.43	0.42	0.67	0.63
db-e	0.543	0.83	0.35	0.43	0.63	0.65

Table 5.1: Quantitative comparison of reprojection error for edge-alignment (EA) and Perspective-n-points on sparse point feature matching. EA-1: Pose refinement with EA using identity as the initial guess for the pose. EA-2: Pose refinement with initial guess obtained with closed form 3d-3d alignment. PNP: Uses ORB sparse point features, closed form 3d-3d alignment as initial guess for PNP refinement (minimization of reprojection errors at sparse point-features).

5.8.2 Quantitative Comparison between Edge-alignment and PNP

In this experiment we compare the performance of pose computation using a) proposed edge-alignment based method b) traditional sparse feature and refinement with Perspective-n-points (PNP). For the PNP although we minimize the reprojection errors on the sparse point features, we report (see Table 5.1) the reprojection errors on all the edge-points. This in our opinion gives a better gauge on the performance. We provide for several datasets the mean and standard deviations across the images in those datasets. Each dataset consists of about 100 keyframes. We observe that the reprojection errors tend to be smaller in case of the edge-alignment, indicating higher accuracy of pose computation.

Next we compare the pose estimates in adjacent keyframes using the edge-alignment and standard PNP based method. For comparison we use the odometry estimates from the VIO system as the baseline. Although performance is comparable in all the sequences we would like to point out the case with the sequence 'db-b'. It contains motion blurs and fewer texture features. In this case the edge based approach is able to provide a more accurate estimate. In other sequences the difference between the means is in the range of 0.5 degrees for rotation and 0.05 m for translation.

	EA				PNP			
sequence	μ^o	σ^o	μ^{tr}	σ^{tr}	$\hat{\mu^o}$	$\hat{\sigma^o}$	$\hat{\mu^{tr}}$	$\hat{\sigma^{tr}}$
db-a	0.88	2.85	0.04	0.12	0.52	0.30	0.02	0.01
db-b	0.32	0.64	0.02	0.03	0.88	1.60	0.03	0.07
db-c	0.75	1.32	0.03	0.04	0.49	0.23	0.02	0.01
db-d	0.64	1.18	0.03	0.04	0.44	0.30	0.01	0.01
db-e	1.29	2.60	0.07	0.10	0.74	0.97	0.03	0.03

Table 5.2: Showing the mean (μ^o) and std deviations (σ^o) in degrees for errors in Euler angle rotation estimates. μ^{tr} and σ^{tr} in meters are the mean and std deviations of errors in translation estimates respectively.

5.8.3 Qualitative Comparison of Edge-alignment and PNP

We give quantitative comparison between a) the standard sparse-features followed by perspective-n-point (PNP) and b) proposed edge alignment. We provide two real world examples (see Fig. 5.14 and Fig. 5.15). These were detected as true loop-candidates by our proposed place recognition system. Using the proposed edge-alignment approach we are able to get tighter estimates of the relative poses. For comparison we show the reprojections of several points plotted on reference image. We see a more consistent reprojection of the edges when using the proposed edge alignment based approach. For scenes with strong egdes, alignment of edge provide a superior pose computation accuracy compared to locally optimizing sparse point based methods. Next we also compare standard PNP with the proposed edge-alignment for pose computation in some real-world scenario (see Fig. 5.16 and Fig. 5.17). In challenging cases with large number of similar looking edges the edge-alignment based pose computation is able to produce reliable pose estimates.

5.9 Conclusion

In this chapter we present a robust method for feature matching, pose computation and pose graph solver capable to recovering from kidnap. To take advantage of the high recall rates of the learning based method there is a need for feature matching and pose computation to produce reliable estimates. Our feature matching takes advantage of the association maps which are a by-product of our place recognition framework. We take advantage of the coherence constraint in the form of a simple voting scheme.

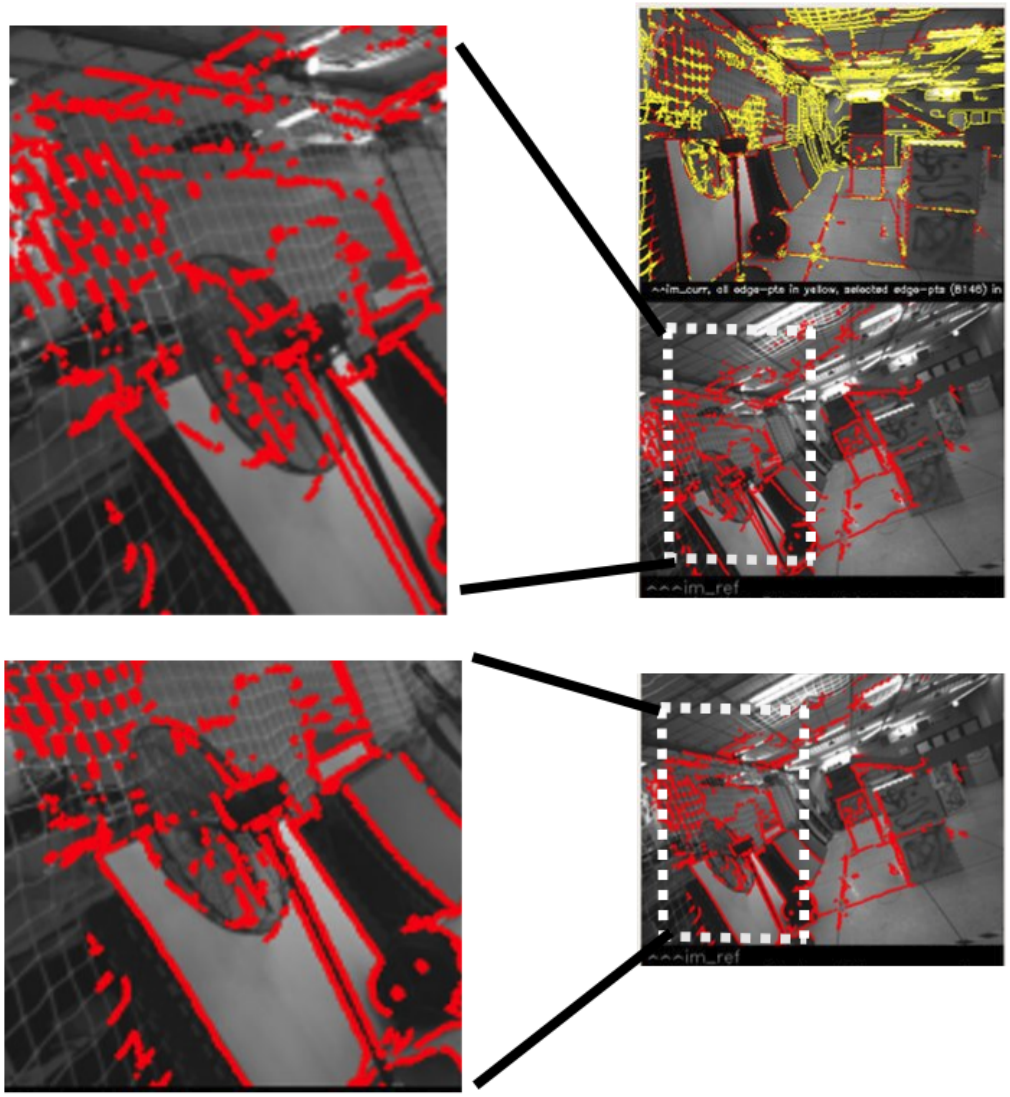

Figure 5.14: Qualitative comparison of alignment using edge-alignment and using sparse point features and perspective-n-points (PNP). Top figure shows current image (top) and all detected edge-points marked in yellow and edge-points used for alignment computation in red (${}^{c}X$). Middle row shows the reprojection of detected points on reference frame using the pose computed with sparse point ORB-features and perspective-n-points (PNP), ie. ${}^{r}T_{c}^{(PNP)} \times {}^{c}X$. Bottom row shows the reprojection of detected edge-points of current frame using pose computed with the proposed edge-alignment algorithm, ie. ${}^{r}T_{c}^{(EA)} \times {}^{c}X$. Best viewed in color.

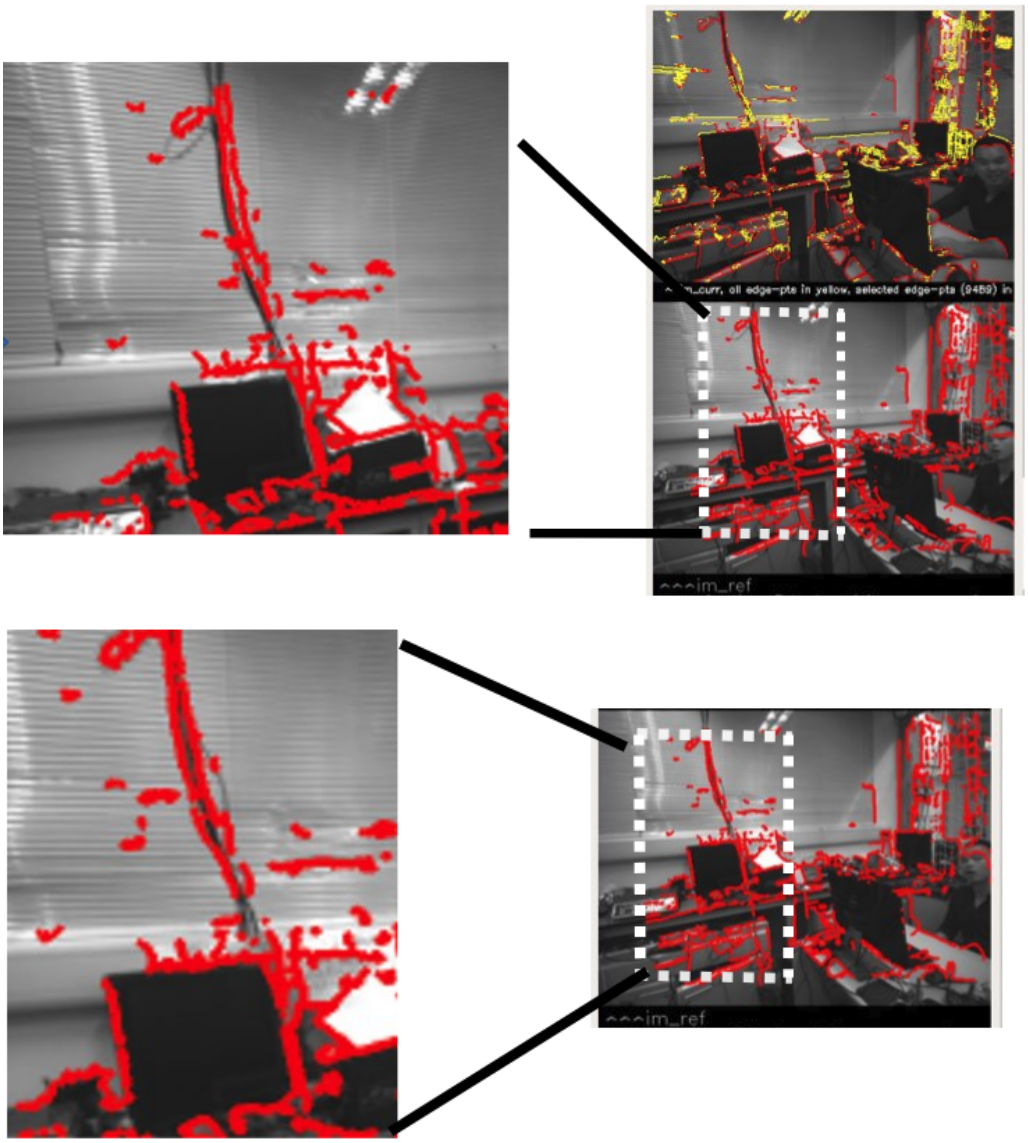

Figure 5.15: Qualitative comparison of alignment using edge-alignment and using sparse point features and perspective-n-points (PNP). Top figure shows current image (top) and all detected edge-points marked in yellow and edge-points used for alignment computation in red (${}^{c}X$). Middle row shows the reprojection of detected points on reference frame using the pose computed with sparse point ORB-features and perspective-n-points (PNP), ie. ${}^{r}T_{c}^{(PNP)} \times {}^{c}X$. Bottom row shows the reprojection of detected edge-points of current frame using pose computed with the proposed edge-alignment algorithm, ie. ${}^{r}T_{c}^{(EA)} \times {}^{c}X$. Best viewed in color.

Figure 5.16: Top-row left: Detected edge-points marked on the current image. In yellow are all detected points. In red are the points with valid depths, ie. ${}^{c}X$. **Top-row right:** Reprojected edge-points of current image plotted on the reference image. In blue is the reprojection using pose computed with PNP(+RanSAC) on ORB sparse-point matches. In red is the reprojection using pose computed with PNP (+RanSAC) on matches with GMS-Matcher [18], ie. ${}^{r}T_{c}^{(PNP)} \times {}^{c}X$. **Bottom:** Reprojection of detected edge points of current image plotted in reference image using pose computed by proposed edge-alignment method, ${}^{r}T_{c}^{(EA)} \times {}^{c}X$.

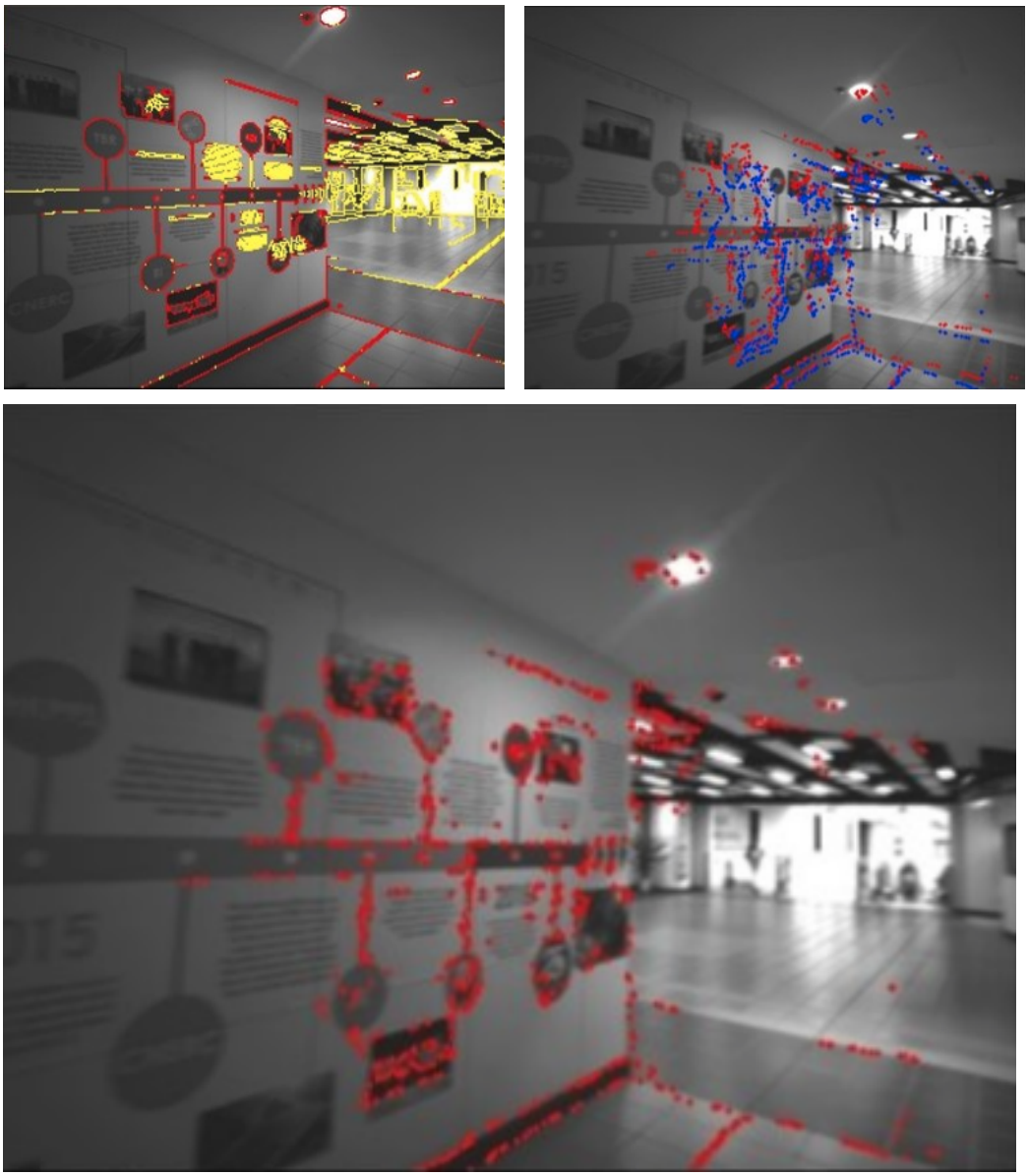

Figure 5.17: Top-row left: Detected edge-points marked on the current image. In yellow are all detected points. In red are the points with valid depths. **Top-row right:** Reprojected edge-points of current image plotted on the reference image. In blue is the reprojection using pose computed with PNP(+RanSAC) on ORB sparse-point matches. In red is the reprojection using pose computed with PNP (+RanSAC) on matches with GMS-Matcher [18]. **Bottom:** Reprojection of detected edge points of current image plotted in reference image using pose computed by proposed edge-alignment method.

For a reliable relative pose computation under large viewpoint changes, we propose a two step approach a) global computation followed by b) pose refinement. Since under large viewpoint difference the overlapping area is often times less resulting in fewer sparse feature matches. For this we rely on GMS-matcher [18]. We aggregate feature matching (and their 3D points) from multiple image pairs. We formulate a 3D-3D alignment method and solve it with alternating minimization to produce a coarse estimate of the pose. This is especially critical in long loops or kidnap scenarios where the relative pose from odometry is not suitable as an initial guess for refinement. For the refinement stage we propose a local bundle with odometry and reprojection constraints.

Everytime at the kidnap, the odometry system resets resulting in a new co-ordinate reference. For handling kidnaps we need to handle multiple co-ordinate systems and their associations with each other. We propose to use the disjoint-set data structure to this effect. Experimentally we show that we can reliable handle upto 20 co-ordinate frames and merge them reliable to recover from complicated kidnap scenarios.

Chapter 6

Conclusion and Future Directions

SLAM related techniques such as visual inertial odometry, re-localization etc. are finding its way to real world products and services starting from self-driving cars, maps, augmented reality etc. Cadena *et al.* [28] proposed that we are now in *'robust-perception-age'* and that open questions lie in four categories: robust performance, high-level understanding, resource awareness, and task-driven inference. We address some of the issues in the category of robust performance in this thesis and some remain a topic of future exploration.

Particularly, we explore the case of odometry estimation in environments which lack corner key features but are plentiful in edge features. Our edge based tracker is able to estimate reliable relative poses in corridor-like scenes which lack good corner features to track. The traditional corner based methods still tend to perform better in feature rich environment. For the visual odometry part, we recommend a simple intermediate system which can gauge the environment as being feature-rich or edge-rich depending on its output, perform traditional method or the proposed edge-based method. This would provide tracking fail-safety in environments which all corner features.

Next when revisits occur at non fronto-parallel views, the existing bag-of-words based visual approach for relocalization has a high miss rates. The performance of the bag-of-words method as a retrieval method is limited to the underlying feature descriptors. The descriptors derived from sparse point features tend to not use a large portion of information from the image. Additionally, since it relies just on corner features, the flatter regions are effectively ignored. Also the limiting factors comes from the quantization error of the clustering of the features to form the visual words. Those visual words might not be true representative of the real scenes. CNN based descriptors provide for a more richer scene descriptors. Our system is able to provide precision-recall performance which is on

par with the standard NetVLAD while at a computational cost and model size an order of magnitude lower. Our descriptor is suitable for realtime SLAM systems.

Relative pose computation under large viewpoint difference and recovering from kidnap are yet another challenge that we address in this thesis. In general pose computation at large view point difference is an extremely challenging problem. Changes in weather, day-night scenes, fixed objects like furniture displaced from their positions, dynamic scenes are some of the major challenges. We address the case of non-frontal scenes where the objects are within the range of the stereo camera baseline (about 5m for a 10cm baseline). Having known the depth image of the scene we leverage the scene edges and nearby frames, we formulate the problem as a local bundle adjustment problem. While we have increased the fail-safety of the SLAM system, in general this problem is still unsolved.

Relocalization from kidnap is another issue we address for the fail-safety. The disjoint set data structure provides a convenient way to manage the pose relationships between arbitrary number of co-ordinate systems. This helps us build a multi-coordinate pose graph whose solution can provide for a kidnap recovery and relocalization mechanism live and in realtime. This mechanism of handling co-ordinate systems can also help with manage a swarm of robots. Also it can provide for advanced AR (Augmented-reality) applications like multiple AR devices seeing the same virtual objects at the same spot. This is current beyond the realm of current commercial AR systems.

6.1 Summary of contributions

We proposed our fully functional, kidnap aware SLAM system which in general can be plugged into any visual-inertial system. Our open-source implementation is crafted towards an add-on system for VINS-Fusion. Our system can detect revisits at large viewpoint difference and recover from the drift.

In this thesis, we present our contribution to the state-of-the-art towards fail-safety aspects of a SLAM system. In this thesis we started with a general introduction of the subsystems of a SLAM system and its applications. In Chp. 3 we developed a visual odometry system based upon alignment of edges. The core differentiating point was the direct formulation and the use of distance transform as metric of alignment. The advantage of such odometry is that, due to its large convergence basin, it is able to work

under fast motions, low frame rates and also under limited corner features (but sufficient edges). We compare our work with direct methods as well as feature based methods.

Our next contribution (described in Chp. 4) was a weakly supervised image description method, aka. the place recognition front-end. When compared to computationally expensive learning based methods, we get a comparable precision-recall. When compared to the bag-of-words approach which is the most common approach in modern SLAM system, our method has a much higher recall rate. Our method was an order of magnitude faster compared to NetVLAD and needed two order of magnitude less storage size when compared to it. When comparing model sizes to vocabulary (model) of the bag-of-words approach our model size was an order of magnitude smaller. The main advantage of a learning based method over bag-of-visual-words is that the former has a much higher recall rate.

To take advantage of the high recall rates of the place recognition front end we need robust methods which can estimate relative poses reliably under large viewpoint changes. Simple 5-point method based relative pose computation method often fail in such scenarios. We propose a two step approach to this. We aggregate feature correspondences from multiple image pairs at the revisit. We formulate a 3D-3D alignment and solve it with alternating formulation to produce a coarse estimate of the relative pose. This is especially critical in long loops or kidnap scenarios where the relative pose from odometry is not suitable as an initial guess for refinement. For the refinement stage we propose a local bundle with odometry and reprojection constraints.

6.2 Future Work and Challenges

In addition to kidnap recovery mechanism as addressed in this thesis, we overview of some advances in the field in relation to robust methods in the SLAM pipelines. Yang and Shen [220] addressed the issue of IMU-camera calibration and proposed an online calibration method. Ling and Shen [119] proposed a markerless and online stereo calibration method. Qin and Shen [168] proposed an online method for camera and IMU alignment. Such methods when integrated in the SLAM systems provide robustness in performance by providing online calibration.

Although there is progress in realtime dense mapping [213] which is suitable for navigation, current state-of-the-art SLAM systems are unable to provide a high-level semantic understanding of the geometry of the surrounding world. Some recent works however provides some solutions to this issue. In their creatively titled paper, Paul *et al.* [160] proposed online visual summarization of scenes. They used a topic vector representation and a graph clustering for online organization of the semantic data. Sunderhauf *et al.* [196, 197] proposed a place categorization and semantic approach for object-level entities and their geometric representation. Some other approaches in regard to semantic SLAM include Fusion++[134], DS-SLAM [222] Hosseinzadeh *et al.* [76] to name a few. Although point-cloud based maps are sufficient for robot navigation tasks, semantic maps can help robots make intelligent decisions in regard to their possibly dynamic environment. Several open challenges remain in this regard.

We hope that this thesis takes one step further towards long term autonomy of robots.

References

[1] Pratik Agarwal, Gian Diego Tipaldi, Luciano Spinello, Cyrill Stachniss, and Wolfram Burgard. Robust map optimization using dynamic covariance scaling. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 62–69. Ieee, 2013.

[2] Sameer Agarwal, Keir Mierle, et al. Ceres solver, 2012.

[3] Sameer Agarwal, Keir Mierle, and Others. Ceres Solver. `http://ceres-solver.org`.

[4] Adrien Angeli, David Filliat, Stephane Doncieux, and Jean-Arcady Meyer. Fast and incremental method for loop-closure detection using bags of visual words. *IEEE Transactions on Robotics*, 24(5):1027–1037, 2008.

[5] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5297–5307, 2016.

[6] Relja Arandjelovic, Petr Gronát, Akihiko Torii, Tomás Pajdla, and Josef Sivic. NetVLAD: CNN Architecture for Weakly Supervised Place Recognition. In *CVPR*, pages 5297–5307. IEEE Computer Society, 2016.

[7] Relja Arandjelovic and Andrew Zisserman. DisLocation: Scalable descriptor distinctiveness for location recognition. In *Asian Conference on Computer Vision*, pages 188–204. Springer, 2014.

[8] Roberto Arroyo, Pablo F Alcantarilla, Luis M Bergasa, and Eduardo Romera. Fusion and binarization of CNN features for robust topological localization across

seasons. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 4656–4663. IEEE, 2016.

[9] K Somani Arun, Thomas S Huang, and Steven D Blostein. Least-squares fitting of two 3-D point sets. *IEEE Transactions on pattern analysis and machine intelligence*, (5):698–700, 1987.

[10] Artem Babenko and Victor Lempitsky. Aggregating local deep features for image retrieval. In *Proceedings of the IEEE international conference on computer vision*, pages 1269–1277, 2015.

[11] Dongdong Bai, Chaoqun Wang, Bo Zhang, Xiaodong Yi, and Xuejun Yang. Sequence searching with cnn features for robust and fast visual place recognition. *Computers & Graphics*, 70:270–280, 2018.

[12] Tim Bailey and Hugh Durrant-Whyte. Simultaneous localization and mapping (SLAM): Part II. *IEEE Robotics & Automation Magazine*, 13(3):108–117, 2006.

[13] Abhinav Bajpai, Guy Burroughes, Affan Shaukat, and Yang Gao. Planetary monocular simultaneous localization and mapping. *Journal of Field Robotics*, 33(2):229–242, 2016.

[14] Simon Baker and Iain Matthews. Lucas-kanade 20 years on: A unifying framework. *International journal of computer vision*, 56(3):221–255, 2004.

[15] Loukas Bampis, Angelos Amanatiadis, and Antonios Gasteratos. High order visual words for structure-aware and viewpoint-invariant loop closure detection. In *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, pages 4268–4275. IEEE, 2017.

[16] Loukas Bampis, Angelos Amanatiadis, and Antonios Gasteratos. Fast loop-closure detection using visual-word-vectors from image sequences. *The International Journal of Robotics Research*, 37(1):62–82, 2018.

[17] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48. ACM, 2009.

[18] JiaWang Bian, Wen-Yan Lin, Yasuyuki Matsushita, Sai-Kit Yeung, Tan-Dat Nguyen, and Ming-Ming Cheng. Gms: Grid-based motion statistics for fast, ultra-robust feature correspondence. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2828–2837. IEEE, 2017.

[19] Michael Bloesch, Sammy Omari, Marco Hutter, and Roland Siegwart. Robust visual inertial odometry using a direct EKF-based approach. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 298–304. IEEE, 2015.

[20] Sofien Bouaziz, Andrea Tagliasacchi, and Mark Pauly. Sparse iterative closest point. In *Proceedings of the Eleventh Eurographics/ACMSIGGRAPH Symposium on Geometry Processing*, pages 113–123. Eurographics Association, 2013.

[21] Stephen Boyd, Lin Xiao, and Almir Mutapcic. Subgradient methods. *lecture notes of EE392o, Stanford University, Autumn Quarter*, 2004:2004–2005, 2003.

[22] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. DSAC-Differentiable RANSAC for camera localization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 3, 2017.

[23] Jesus Briales and Javier Gonzalez-Jimenez. Convex global 3d registration with lagrangian duality. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4960–4969, 2017.

[24] Michael Burri, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W Achtelik, and Roland Siegwart. The EuRoC micro aerial vehicle datasets. *The International Journal of Robotics Research*, 2016.

[25] Álvaro Parra Bustos and Tat-Jun Chin. Guaranteed outlier removal for point cloud registration with correspondences. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2868–2882, 2017.

[26] Erik Bylow, Jürgen Sturm, Christian Kerl, Fredrik Kahl, and Daniel Cremers. Real-time camera tracking and 3D reconstruction using signed distance functions. In *Robotics: Science and Systems*, volume 2, 2013.

[27] Charles L Byrne. Alternating minimization and alternating projection algorithms: A tutorial. *Sciences New York*, pages 1–41, 2011.

[28] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332, 2016.

[29] Giuseppe Calafiore, Luca Carlone, and Frank Dellaert. Pose graph optimization in the complex domain: Lagrangian duality, conditions for zero duality gap, and optimal solutions. *arXiv preprint arXiv:1505.03437*, 2015.

[30] Luca Carlone, Andrea Censi, and Frank Dellaert. Selecting good measurements via âĎŞ 1 relaxation: A convex approach for robust estimation over graphs. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 2667–2674. IEEE, 2014.

[31] Z. Chen, A. Jacobson, N. SÃijnderhauf, B. Upcroft, L. Liu, C. Shen, I. Reid, and M. Milford. Deep learning features at scale for visual place recognition. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3223–3230, May 2017.

[32] Zetao Chen, Adam Jacobson, Niko Sünderhauf, Ben Upcroft, Lingqiao Liu, Chunhua Shen, Ian Reid, and Michael Milford. Deep learning features at scale for visual place recognition. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 3223–3230. IEEE, 2017.

[33] Zetao Chen, Obadiah Lam, Adam Jacobson, and Michael Milford. Convolutional neural network-based place recognition. *arXiv preprint arXiv:1411.1509*, 2014.

[34] Zetao Chen, Lingqiao Liu, Inkyu Sa, Zongyuan Ge, and Margarita Chli. Learning context flexible attention model for long-term visual place recognition. *IEEE Robotics and Automation Letters*, 3(4):4015–4022, 2018.

[35] Zetao Chen, Fabiola Maffra, Inkyu Sa, and Margarita Chli. Only look once, mining distinctive landmarks from convnet for visual place recognition. In *2017 IEEE/RSJ*

International Conference on Intelligent Robots and Systems (IROS), pages 9–16. IEEE, 2017.

[36] T. Cieslewski and D. Scaramuzza. Efficient decentralized visual place recognition from full-image descriptors. In *2017 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*, pages 78–82, Dec 2017.

[37] T. Cieslewski and D. Scaramuzza. Efficient Decentralized Visual Place Recognition Using a Distributed Inverted Index. *IEEE Robotics and Automation Letters*, 2(2):640–647, April 2017.

[38] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2009.

[39] I Csiszár and G Tusnády. Information Geometry and Alternating Minimization Problems. *Statistics & Decision, Supplement Issue No*, 1, 1984.

[40] Mark Cummins and Paul Newman. FAB-MAP: Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research*, 27(6):647–665, 2008.

[41] Mark Cummins and Paul Newman. Appearance-only SLAM at large scale with FAB-MAP 2.0. *The International Journal of Robotics Research*, 30(9):1100–1123, 2011.

[42] Gamini Dissanayake, Shoudong Huang, Zhan Wang, and Ravindra Ranasinghe. A review of recent developments in simultaneous localization and mapping. In *Industrial and Information Systems (ICIIS), 2011 6th IEEE International Conference on*, pages 477–482. IEEE, 2011.

[43] Ivan Dryanovski, Roberto G Valenti, and Jizhong Xiao. Fast visual odometry and mapping from RGB-D data. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 2305–2310. IEEE, 2013.

[44] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part I. *IEEE robotics & automation magazine*, 13(2):99–110, 2006.

[45] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence*, 4, 2017.

[46] Jakob Engel, Thomas Schöps, and Daniel Cremers. LSD-SLAM: Large-scale direct monocular SLAM. In *European Conference on Computer Vision*, pages 834–849. Springer, 2014.

[47] Jakob Engel, Jürgen Sturm, and Daniel Cremers. Camera-based navigation of a low-cost quadrocopter. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 2815–2821. IEEE, 2012.

[48] Olof Enqvist, Klas Josephson, and Fredrik Kahl. Optimal correspondences from pairwise constraints. In *2009 IEEE 12th international conference on computer vision*, pages 1295–1302. IEEE, 2009.

[49] Matthias Faessler, Flavio Fontana, Christian Forster, Elias Mueggler, Matia Pizzoli, and Davide Scaramuzza. Autonomous, vision-based flight and live dense 3d mapping with a quadrotor micro aerial vehicle. *Journal of Field Robotics*, 33(4):431–450, 2016.

[50] Davide Falanga, Philipp Foehn, Peng Lu, and Davide Scaramuzza. PAMPC: Perception-Aware Model Predictive Control for Quadrotors. *arXiv preprint arXiv:1804.04811*, 2018.

[51] Xiaohan Fei, Konstantine Tsotsos, and Stefano Soatto. A Simple Hierarchical Pooling Data Structure for Loop Closure. *CoRR*, abs/1511.06489, 2015.

[52] Pedro F Felzenszwalb and Daniel P Huttenlocher. Distance Transforms of Sampled Functions. *Theory of computing*, 8(1):415–428, 2012.

[53] Andrew W Fitzgibbon. Robust registration of 2D and 3D point sets. *Image and Vision Computing*, 21(13):1145–1153, 2003.

[54] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. SVO: Fast semi-direct monocular visual odometry. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 15–22. IEEE, 2014.

[55] Friedrich Fraundorfer and Davide Scaramuzza. Visual odometry: Part ii: Matching, robustness, optimization, and applications. *IEEE Robotics & Automation Magazine*, 19(2):78–90, 2012.

[56] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

[57] Dorian Gálvez-López and Juan D Tardos. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197, 2012.

[58] Xiang Gao and Tao Zhang. Unsupervised learning to detect loops using deep neural networks for visual SLAM system. *Autonomous robots*, 41(1):1–18, 2017.

[59] Emilio Garcia-Fidalgo and Alberto Ortiz. On the use of binary feature descriptors for loop closure detection. In *Emerging Technology and Factory Automation (ETFA), 2014 IEEE*, pages 1–8. IEEE, 2014.

[60] Emilio Garcia-Fidalgo and Alberto Ortiz. iBoW-LCD: An Appearance-based Loop Closure Detection Approach using Incremental Bags of Binary Words. *arXiv preprint arXiv:1802.05909*, 2018.

[61] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 249–256, 2010.

[62] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, volume 2, page 7, 2017.

[63] Ruben Gomez-Ojeda, Jesus Briales, and Javier Gonzalez-Jimenez. PL-SVO: Semi-direct Monocular Visual Odometry by combining points and line segments. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4211–4216. IEEE, 2016.

[64] Matthew C Graham, Jonathan P How, and Donald E Gustafson. Robust incremental slam with consistency-checking. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 117–124. IEEE, 2015.

[65] Giorgio Grisetti, Rainer Kummerle, Cyrill Stachniss, and Wolfram Burgard. A tutorial on graph-based SLAM. *IEEE Intelligent Transportation Systems Magazine*, 2(4):31–43, 2010.

[66] Yulan Guo, Mohammed Bennamoun, Ferdous Sohel, Min Lu, Jianwei Wan, and Ngai Ming Kwok. A comprehensive performance evaluation of 3D local feature descriptors. *International Journal of Computer Vision*, 116(1):66–89, 2016.

[67] Yoav HaCohen, Eli Shechtman, Dan B Goldman, and Dani Lischinski. Non-rigid dense correspondence with applications for image enhancement. In *ACM transactions on graphics (TOG)*, volume 30, page 70. ACM, 2011.

[68] Fei Han, Hua Wang, Guoquan Huang, and Hao Zhang. Sequence-based sparse optimization methods for long-term loop closure detection in visual SLAM. *Autonomous Robots*, pages 1–13, 2018.

[69] Kai Han, Rafael S Rezende, Bumsub Ham, Kwan-Yee K Wong, Minsu Cho, Cordelia Schmid, and Jean Ponce. SCNet: Learning semantic correspondence. In *International Conference on Computer Vision*, 2017.

[70] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 10–5244. Citeseer, 1988.

[71] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[72] Nick Hawes, Christopher Burbridge, Ferdian Jovan, Lars Kunze, Bruno Lacerda, Lenka Mudrova, Jay Young, Jeremy Wyatt, Denise Hebesberger, Tobias Kortner, et al. The strands project: Long-term autonomy in everyday environments. *IEEE Robotics & Automation Magazine*, 24(3):146–156, 2017.

[73] Heiko Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on pattern analysis and machine intelligence*, 30(2):328–341, 2008.

[74] Dirk Holz, Alexandru E Ichim, Federico Tombari, Radu B Rusu, and Sven Behnke. Registration with the point cloud library: A modular framework for aligning in 3-D. *IEEE Robotics & Automation Magazine*, 22(4):110–124, 2015.

[75] Berthold KP Horn, Hugh M Hilden, and Shahriar Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *JOSA A*, 5(7):1127–1135, 1988.

[76] Mehdi Hosseinzadeh, Kejie Li, Yasir Latif, and Ian Reid. Real-time monocular object-model aware sparse SLAM. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 7123–7129. IEEE, 2019.

[77] Yi Hou, Hong Zhang, and Shilin Zhou. BoCNF: Efficient image matching with bag of ConvNet features for scalable and robust visual place recognition. *Autonomous Robots*, 42(6):1169–1185, 2018.

[78] Andrew Howard. Real-time stereo visual odometry for autonomous ground vehicles. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 3946–3952. IEEE, 2008.

[79] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[80] Albert S Huang, Abraham Bachrach, Peter Henry, Michael Krainin, Daniel Maturana, Dieter Fox, and Nicholas Roy. Visual odometry and mapping for autonomous flight using an RGB-D camera. In *International Symposium on Robotics Research (ISRR)*, pages 1–16, 2011.

[81] Peter J Huber et al. Robust estimation of a location parameter. *The annals of mathematical statistics*, 35(1):73–101, 1964.

[82] Vadim Indelman, Stephen Williams, Michael Kaess, and Frank Dellaert. Factor graph based incremental smoothing in inertial navigation systems. In *Information Fusion (FUSION), 2012 15th International Conference on*, pages 2154–2161. IEEE, 2012.

[83] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. On the burstiness of visual elements. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1169–1176. IEEE, 2009.

[84] Bing Jian and Baba C Vemuri. Robust point set registration using gaussian mixture models. *IEEE transactions on pattern analysis and machine intelligence*, 33(8):1633–1645, 2010.

[85] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *arXiv preprint arXiv:1702.08734*, 2017.

[86] Michael Kaess, Hordur Johannsson, Richard Roberts, Viorela Ila, John J Leonard, and Frank Dellaert. iSAM2: Incremental smoothing and mapping using the Bayes tree. *The International Journal of Robotics Research*, 31(2):216–235, 2012.

[87] Chingiz Kenshimov, Loukas Bampis, Beibut Amirgaliyev, Marat Arslanov, and Antonios Gasteratos. Deep learning features exception for cross-season visual place recognition. *Pattern Recognition Letters*, 100:124–130, 2017.

[88] Christian Kerl, Jürgen Sturm, and Daniel Cremers. Robust odometry estimation for RGB-D cameras. In *ICRA*, pages 3748–3754. IEEE, 2013.

[89] Ahmad Khaliq, Shoaib Ehsan, Michael Milford, and Klaus McDonald-Maier. A Holistic Visual Place Recognition Approach using Lightweight CNNs for Severe ViewPoint and Appearance Changes. *arXiv preprint arXiv:1811.03032*, 2018.

[90] Sheraz Khan and Dirk Wollherr. Ibuild: Incremental bag of binary words for appearance based loop closure detection. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 5441–5447. IEEE, 2015.

[91] Seungryong Kim, Dongbo Min, Stephen Lin, and Kwanghoon Sohn. DCTM: Discrete-Continuous Transformation Matching for Semantic Flow. *arXiv preprint arXiv:1707.05471*, 2017.

[92] G. Klein and D. Murray. Parallel tracking and mapping for small ar workspaces. pages 225–234, 2007.

[93] Kurt Konolige, Motilal Agrawal, and Joan Sola. Large-scale visual odometry for rough terrain. In *Robotics research*, pages 201–212. Springer, 2010.

[94] Rainer Kümmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. g 2 o: A general framework for graph optimization. In *Robotics and*

Automation (ICRA), 2011 IEEE International Conference on, pages 3607–3613. IEEE, 2011.

[95] Manohar Kuse and Sunil Prasad Jaiswal. Graph modelling of 3D geometric information for color consistency of multiview images. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 1394–1398. IEEE, 2015.

[96] Manohar Kuse, Sunil Prasad Jaiswal, and Shaojie Shen. Deep-mapnets: A residual network for 3D environment representation. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 2652–2656. IEEE, 2017.

[97] Manohar Kuse and Shaojie Shen. Robust camera motion estimation using direct edge alignment and sub-gradient method. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 573–579. IEEE, 2016.

[98] Manohar Kuse and Shaojie Shen. Learning Whole Image Descriptors for Loop Detection under Large Viewpoint Difference. 2018.

[99] Manohar Kuse and Shaojie Shen. Learning Whole-Image Descriptors for Real-time Loop Detection and Kidnap Recovery under Large Viewpoint Difference. *CoRR*, abs/1904.06962, 2019.

[100] Mathieu Labbé and François Michaud. Online global loop closure detection for large-scale multi-session graph-based SLAM. In *IROS*, pages 2661–2666. IEEE, 2014.

[101] Mathieu Labbe and Francois Michaud. Appearance-based loop closure detection for online large-scale and long-term operation. *IEEE Transactions on Robotics*, 29(3):734–745, 2013.

[102] Simon Lacroix, Anthony Mallet, Raja Chatila, and Laurent Gallo. Rover self localization in planetary-like environments. In *Artificial Intelligence, Robotics and Automation in Space*, volume 440, page 433, 1999.

[103] Henning Lategahn, Andreas Geiger, and Bernd Kitt. Visual SLAM for autonomous ground vehicles. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1732–1737. IEEE, 2011.

[104] Yasir Latif, César Cadena, and José Neira. Robust loop closing over time for pose graph SLAM. *The International Journal of Robotics Research*, 32(14):1611–1626, 2013.

[105] Yasir Latif, Guoquan Huang, John J Leonard, and JosÃĺ Neira. An Online Sparsity-Cognizant Loop-Closure Algorithm for Visual Navigation. In *Robotics: Science and Systems*, 2014.

[106] Gim Hee Lee, Friedrich Fraundorfer, and Marc Pollefeys. Robust pose-graph loop-closures with expectation-maximization. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 556–563. IEEE, 2013.

[107] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o (n) solution to the pnp problem. *International journal of computer vision*, 81(2):155–166, 2009.

[108] Stefan Leutenegger, Paul Furgale, Vincent Rabaud, Margarita Chli, Kurt Konolige, and Roland Siegwart. Keyframe-based visual-inertial slam using nonlinear optimization. *Proceedings of Robotis Science and Systems (RSS) 2013*, 2013.

[109] Stefan Leutenegger, Simon Lynen, Michael Bosse, Roland Siegwart, and Paul Furgale. Keyframe-based visual–inertial odometry using nonlinear optimization. *The International Journal of Robotics Research*, 34(3):314–334, 2015.

[110] Hongdong Li and Richard Hartley. Five-point motion estimation made easy. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 1, pages 630–633. IEEE, 2006.

[111] Hongdong Li and Richard Hartley. The 3D-3D registration problem revisited. In *2007 IEEE 11th international conference on computer vision*, pages 1–8. IEEE, 2007.

[112] Mingyang Li and Anastasios I Mourikis. High-precision, consistent EKF-based visual-inertial odometry. *The International Journal of Robotics Research*, 32(6):690–711, 2013.

[113] Peiliang Li, Tong Qin, Botao Hu, Fengyuan Zhu, and Shaojie Shen. Monocular visual-inertial state estimation for mobile augmented reality. In *2017 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 11–21. IEEE, 2017.

[114] Wen-Yan Lin, Siying Liu, Nianjuan Jiang, Minh N Do, Ping Tan, and Jiangbo Lu. Repmatch: Robust feature matching and pose for reconstructing modern cities. In *European Conference on Computer Vision*, pages 562–579. Springer, 2016.

[115] Wen-Yan Lin, Siying Liu, Yasuyuki Matsushita, Tian-Tsong Ng, and Loong-Fah Cheong. Smoothly varying affine stitching. 2011.

[116] Wen-Yan Daniel Lin, Ming-Ming Cheng, Jiangbo Lu, Hongsheng Yang, Minh N Do, and Philip Torr. Bilateral functions for global motion modeling. In *European Conference on Computer Vision*, pages 341–356. Springer, 2014.

[117] Yi Lin, Fei Gao, Tong Qin, Wenliang Gao, Tianbo Liu, William Wu, Zhenfei Yang, and Shaojie Shen. Autonomous aerial navigation using monocular visual-inertial fusion. *Journal of Field Robotics*, 35(1):23–51, 2018.

[118] Yonggen Ling, Manohar Kuse, and Shaojie Shen. Edge alignment-based visual–inertial fusion for tracking of aggressive motions. *Autonomous Robots*, 42(3):513–528, 2018.

[119] Yonggen Ling and Shaojie Shen. High-precision online markerless stereo extrinsic calibration. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1771–1778. IEEE, 2016.

[120] Yaron Lipman, Stav Yagev, Roi Poranne, David W Jacobs, and Ronen Basri. Feature matching with bounded distortion. *ACM Transactions on Graphics (TOG)*, 33(3):26, 2014.

[121] Haomin Liu, Guofeng Zhang, and Hujun Bao. Robust keyframe-based monocular slam for augmented reality. In *Mixed and Augmented Reality (ISMAR), 2016 IEEE International Symposium on*, pages 1–10. IEEE, 2016.

[122] Giuseppe Loianno, Chris Brunner, Gary McGrath, and Vijay Kumar. Estimation, control, and planning for aggressive flight with a small quadrotor with a single camera and IMU. *IEEE Robotics and Automation Letters*, 2(2):404–411, 2017.

[123] Manuel Lopez-Antequera, Ruben Gomez-Ojeda, Nicolai Petkov, and Javier Gonzalez-Jimenez. Appearance-Invariant Place Recognition by Discriminatively Training a Convolutional Neural Network. *Pattern Recognition Letters*, 92:89–95, jun 2017.

[124] A. Loquercio, M. Dymczyk, B. Zeisl, S. Lynen, I. Gilitschenski, and R. Siegwart. Efficient descriptor learning for large scale localization. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3170–3177, May 2017.

[125] Adele Lorusso, David W Eggert, and Robert B Fisher. *A comparison of four algorithms for estimating 3-D rigid transformations*. University of Edinburgh, Department of Artificial Intelligence, 1995.

[126] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[127] Stephanie Lowry, Niko Sünderhauf, Paul Newman, John J Leonard, David Cox, Peter Corke, and Michael J Milford. Visual place recognition: A survey. *IEEE Transactions on Robotics*, 32(1):1–19, 2016.

[128] Feng Lu and Evangelos Milios. Globally consistent range scan alignment for environment mapping. *Autonomous robots*, 4(4):333–349, 1997.

[129] Jiayi Ma, Ji Zhao, Jinwen Tian, Alan L Yuille, and Zhuowen Tu. Robust point matching via vector field consensus. *IEEE Transactions on Image Processing*, 23(4):1706–1721, 2014.

[130] Simone Madeo and Miroslaw Bober. Fast, compact, and discriminative: Evaluation of binary descriptors for mobile applications. *IEEE Transactions on Multimedia*, 19(2):221–235, 2017.

[131] Josef Maier, Martin Humenberger, Markus Murschitz, Oliver Zendel, and Markus

Vincze. Guided matching based on statistical optical flow for fast and robust correspondence analysis. In *European Conference on Computer Vision*, pages 101–117. Springer, 2016.

[132] Mark Maimone, Yang Cheng, and Larry Matthies. Two years of visual odometry on the mars exploration rovers. *Journal of Field Robotics*, 24(3):169–186, 2007.

[133] Jonathan H. Manton. Optimization algorithms exploiting unitary constraints. *IEEE Transactions on Signal Processing*, 50(3):635–650, 2002.

[134] John McCormac, Ronald Clark, Michael Bloesch, Andrew Davison, and Stefan Leutenegger. Fusion++: Volumetric object-level slam. In *2018 International Conference on 3D Vision (3DV)*, pages 32–41. IEEE, 2018.

[135] Daniel Mellinger and Vijay Kumar. Minimum snap trajectory generation and control for quadrotors. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 2520–2525. IEEE, 2011.

[136] Nate Merrill and Guoquan Huang. Lightweight Unsupervised Deep Loop Closure. *arXiv preprint arXiv:1805.07703*, 2018.

[137] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE transactions on pattern analysis and machine intelligence*, 27(10):1615–1630, 2005.

[138] Krystian Mikolajczyk, Tinne Tuytelaars, Cordelia Schmid, Andrew Zisserman, Jiri Matas, Frederik Schaffalitzky, Timor Kadir, and Luc Van Gool. A comparison of affine region detectors. *International journal of computer vision*, 65(1-2):43–72, 2005.

[139] Annalisa Milella and Roland Siegwart. Stereo-based ego-motion estimation using pixel tracking and iterative closest point. In *Computer Vision Systems, 2006 ICVS'06. IEEE International Conference on*, pages 21–21. IEEE, 2006.

[140] Michael J Milford and Gordon F Wyeth. SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1643–1649. IEEE, 2012.

[141] Jean-Michel Morel and Guoshen Yu. ASIFT: A new framework for fully affine invariant image comparison. *SIAM journal on imaging sciences*, 2(2):438–469, 2009.

[142] Anastasios I Mourikis and Stergios I Roumeliotis. A multi-state constraint Kalman filter for vision-aided inertial navigation. In *Robotics and automation, 2007 IEEE international conference on*, pages 3565–3572. IEEE, 2007.

[143] Marius Muja and David G Lowe. Fast matching of binary features. In *2012 Ninth conference on computer and robot vision*, pages 404–410. IEEE, 2012.

[144] Marius Muja and David G Lowe. Scalable nearest neighbor algorithms for high dimensional data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2227–2240, 2014.

[145] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[146] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.

[147] Raúl Mur-Artal and Juan D Tardós. Visual-inertial monocular SLAM with map reuse. *IEEE Robotics and Automation Letters*, 2(2):796–803, 2017.

[148] RaÃžl Mur-Artal and Juan D TardÃşs. Fast relocalisation and loop closing in keyframe-based SLAM. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 846–853. IEEE, 2014.

[149] R. M. Murray, Zexiang Li, and S. S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Ann Arbor, 1994.

[150] Andriy Myronenko, Xubo Song, and Miguel A Carreira-Perpinán. Non-rigid point set registration: Coherent point drift. In *Advances in Neural Information Processing Systems*, pages 1009–1016, 2007.

[151] Yu Nesterov. Smooth minimization of non-smooth functions. *Mathematical programming*, 103(1):127–152, 2005.

[152] Yu Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161, 2013.

[153] Tudor Nicosevici and Rafael Garcia. Automatic visual bag-of-words for online robot navigation and mapping. *IEEE Transactions on Robotics*, 28(4):886–898, 2012.

[154] David Nistér. An efficient solution to the five-point relative pose problem. *IEEE transactions on pattern analysis and machine intelligence*, 26(6):0756–777, 2004.

[155] David Nistér, Oleg Naroditsky, and James Bergen. Visual odometry. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–I. Ieee, 2004.

[156] David Novotny, Diane Larlus, and Andrea Vedaldi. Anchornet: A weakly supervised network to learn geometry-sensitive features for semantic matching. In *Proc. CVPR*, volume 2, 2017.

[157] Helen Oleynikova, Michael Burri, Simon Lynen, and Roland Siegwart. Real-time visual-inertial localization for aerial and ground robots. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 3079–3085. IEEE, 2015.

[158] Clark F Olson, Larry H Matthies, Marcel Schoppers, and Mark W Maimone. Robust stereo ego-motion for long distance navigation. In *cvpr*, page 2453. IEEE, 2000.

[159] Edwin Olson and Pratik Agarwal. Inference on networks of mixtures for robust robot mapping. *The International Journal of Robotics Research*, 32(7):826–840, 2013.

[160] Rohan Paul, Daniela Rus, and Paul Newman. How was your day? online visual workspace summaries using incremental clustering in topic space. In *2012 IEEE International Conference on Robotics and Automation*, pages 4058–4065. IEEE, 2012.

[161] E. Pepperell, P. I. Corke, and M. J. Milford. All-environment visual place recognition with SMART. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1612–1618, May 2014.

[162] Daniel Pizarro and Adrien Bartoli. Feature-based deformable surface detection with self-occlusion reasoning. *International Journal of Computer Vision*, 97(1):54–70, 2012.

[163] Lukas Platinsky, Andrew J Davison, and Stefan Leutenegger. Monocular visual odometry: sparse joint optimisation or dense alternation? In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 5126–5133. IEEE, 2017.

[164] François Pomerleau, Francis Colas, Roland Siegwart, and Stéphane Magnenat. Comparing ICP variants on real-world data sets. *Autonomous Robots*, 34(3):133–148, 2013.

[165] T. Qin, P. Li, Z. Yang, and S. Shen. VINS-Mono. https://github.com/HKUST-Aerial-Robotics/VINS-Mono, 2017.

[166] Tong Qin, Peiliang Li, and Shaojie Shen. Relocalization, global optimization and map merging for monocular visual-inertial SLAM. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1197–1204. IEEE, 2018.

[167] Tong Qin, Peiliang Li, and Shaojie Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4):1004–1020, 2018.

[168] Tong Qin and Shaojie Shen. Online temporal calibration for monocular visual-inertial systems. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3662–3669. IEEE, 2018.

[169] Ananth Ranganathan, Michael Kaess, and Frank Dellaert. Fast 3D pose estimation with out-of-sequence measurements. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 2486–2493. IEEE, 2007.

[170] Ignacio Rocco, Relja Arandjelovic, and Josef Sivic. Convolutional neural network architecture for geometric matching. In *Proc. CVPR*, volume 2, 2017.

[171] Ignacio Rocco, Relja Arandjelović, and Josef Sivic. End-to-end weakly-supervised semantic alignment. *arXiv preprint arXiv:1712.06861*, 2017.

[172] R. Tyrrell Rockafellar. *Convex analysis*. Princeton Mathematical Series. Princeton University Press, Princeton, N. J., 1970.

[173] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to SIFT or SURF. In *Computer Vision (ICCV), 2011 IEEE international conference on*, pages 2564–2571. IEEE, 2011.

[174] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the ICP algorithm. In *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*, pages 145–152. IEEE, 2001.

[175] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the ICP algorithm. In *3dim*, volume 1, pages 145–152, 2001.

[176] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (FPFH) for 3D registration. In *2009 IEEE International Conference on Robotics and Automation*, pages 3212–3217. IEEE, 2009.

[177] Torsten Sattler, Michal Havlena, Konrad Schindler, and Marc Pollefeys. Large-scale location recognition and the geometric burstiness problem. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1582–1590, 2016.

[178] Davide Scaramuzza and Friedrich Fraundorfer. Visual odometry [tutorial]. *IEEE robotics & automation magazine*, 18(4):80–92, 2011.

[179] Korbinian Schmid, Philipp Lutz, Teodor Tomić, Elmar Mair, and Heiko Hirschmüller. Autonomous vision-based micro air vehicle for indoor and outdoor navigation. *Journal of Field Robotics*, 31(4):537–570, 2014.

[180] T. Schneider, M. T. Dymczyk, M. Fehr, K. Egger, S. Lynen, I. Gilitschenski, and R. Siegwart. maplab: An Open Framework for Research in Visual-inertial Mapping and Localization. *IEEE Robotics and Automation Letters*, 2018.

[181] Thomas Schöps, Jakob Engel, and Daniel Cremers. Semi-dense visual odometry for AR on a smartphone. In *Mixed and Augmented Reality (ISMAR), 2014 IEEE International Symposium on*, pages 145–150. IEEE, 2014.

[182] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015.

[183] Shaojie Shen, Nathan Michael, and Vijay Kumar. Autonomous multi-floor indoor navigation with a computationally constrained MAV. In *Robotics and automation (ICRA), 2011 IEEE international conference on*, pages 20–25. IEEE, 2011.

[184] Shaojie Shen, Nathan Michael, and Vijay Kumar. Tightly-coupled monocular visual-inertial fusion for autonomous flight of rotorcraft MAVs. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 5303–5310. IEEE, 2015.

[185] Roland Siegwart, Illah Reza Nourbakhsh, and Davide Scaramuzza. *Introduction to autonomous mobile robots*. MIT press, 2011.

[186] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[187] Josef Sivic and Andrew Zisserman. Video Google: A text retrieval approach to object matching in videos. In *null*, page 1470. IEEE, 2003.

[188] E. Sizikova, V. K. Singh, B. Georgescu, M. Halber, K. Ma, and T. Chen. Enhancing Place Recognition using Joint Intensity - Depth Analysis and Synthetic Data. *European Conference on Computer Vision (ECCV) Workshop on Virtual/Augmented Reality for Visual Artificial Intelligence (VARVAI)*, 2016.

[189] Randall C Smith and Peter Cheeseman. On the representation and estimation of spatial uncertainty. *The international journal of Robotics Research*, 5(4):56–68, 1986.

[190] Frank Steinbrücker, Jürgen Sturm, and Daniel Cremers. Real-time visual odometry from dense RGB-D images. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 719–722. IEEE, 2011.

[191] Jörg Stückler and Sven Behnke. Model Learning and Real-Time Tracking Using Multi-Resolution Surfel Maps. In *AAAI*, 2012.

[192] Elena Stumm, Christopher Mei, Simon Lacroix, Juan Nieto, Marco Hutter, and Roland Siegwart. Robust visual place recognition with graph kernels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4535–4544, 2016.

[193] Elena S. Stumm, Christopher Mei, and Simon Lacroix. Building Location Models for Visual Place Recognition. *Int. J. Rob. Res.*, 35(4):334–356, April 2016.

[194] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A Benchmark for the Evaluation of RGB-D SLAM Systems. In *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.

[195] Jian Sun, Nan-Ning Zheng, and Heung-Yeung Shum. Stereo matching using belief propagation. *IEEE Transactions on pattern analysis and machine intelligence*, 25(7):787–800, 2003.

[196] Niko Sünderhauf, Feras Dayoub, Sean McMahon, Ben Talbot, Ruth Schulz, Peter Corke, Gordon Wyeth, Ben Upcroft, and Michael Milford. Place categorization and semantic mapping on a mobile robot. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 5729–5736. IEEE, 2016.

[197] Niko Sünderhauf, Trung T Pham, Yasir Latif, Michael Milford, and Ian Reid. Meaningful maps with object-oriented semantic mapping. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5079–5085. IEEE, 2017.

[198] Niko Sünderhauf and Peter Protzel. Switchable constraints for robust pose graph SLAM. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 1879–1884. IEEE, 2012.

[199] Niko Sünderhauf, Sareh Shirazi, Feras Dayoub, Ben Upcroft, and Michael Milford. On the performance of convnet features for place recognition. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 4297–4304. IEEE, 2015.

[200] Niko Sünderhauf, Sareh Shirazi, Adam Jacobson, Feras Dayoub, Edward Pepperell, Ben Upcroft, and Michael Milford. Place recognition with convnet landmarks:

Viewpoint-robust, condition-robust, training-free. *Proceedings of Robotics: Science and Systems XII*, 2015.

[201] Pascal Willy Theiler, Jan Dirk Wegner, and Konrad Schindler. Keypoint-based 4-Points Congruent Sets–Automated marker-less registration of laser scans. *ISPRS journal of photogrammetry and remote sensing*, 96:149–163, 2014.

[202] Engin Tola, Vincent Lepetit, and Pascal Fua. A fast local descriptor for dense matching. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, volume 32, pages 1–8. IEEE, IEEE, 2008.

[203] Engin Tola, Vincent Lepetit, and Pascal Fua. Daisy: An efficient dense descriptor applied to wide-baseline stereo. *IEEE transactions on pattern analysis and machine intelligence*, 32(5):815–830, 2010.

[204] Carlo Tomasi and Takeo Kanade. Detection and tracking of point features. 1991.

[205] A. Torii, R. ArandjeloviÄĞ, J. Sivic, M. Okutomi, and T. Pajdla. 24/7 place recognition by view synthesis. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1808–1817, June 2015.

[206] Akihiko Torii, Josef Sivic, Toma Pajdla, and Masatoshi Okutomi. Visual place recognition with repetitive structures. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 883–890. IEEE, 2013.

[207] Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. Bundle Adjustment - A Modern Synthesis. In *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice*, ICCV '99, pages 298–372, London, UK, UK, 2000. Springer-Verlag.

[208] Konstantinos A Tsintotas, Loukas Bampis, and Antonios Gasteratos. Assigning visual words to places for loop closure detection. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–7. IEEE, 2018.

[209] Tinne Tuytelaars and Krystian Mikolajczyk. Local invariant feature detectors: a survey. *Foundations and Trends® in Computer Graphics and Vision*, 3(3):177–280, 2008.

[210] Tommi Tykkälä, Cédric Audras, Andrew Comport, et al. Direct iterative closest point for real-time visual odometry. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 2050–2056. IEEE, 2011.

[211] Olga Vysotska and Cyrill Stachniss. Relocalization under substantial appearance changes using hashing. In *Proc. Int. Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. Workshop Planning, Perception Navig. Intell. Veh.*, 2017.

[212] Chao Wang, Lei Wang, and Lingqiao Liu. Density maximization for improving graph matching with its applications. *IEEE Transactions on Image Processing*, 24(7):2110–2123, 2015.

[213] Kaixuan Wang, Fei Gao, and Shaojie Shen. Real-time scalable dense surfel mapping. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 6919–6925. IEEE, 2019.

[214] Stephan Weiss, Markus W Achtelik, Simon Lynen, Margarita Chli, and Roland Siegwart. Real-time onboard visual-inertial state estimation and self-calibration of mavs in unknown environments. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 957–964. IEEE, 2012.

[215] Simon Winder, Gang Hua, and Matthew Brown. Picking the best daisy. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 178–185. IEEE, 2009.

[216] Heng Yang and Luca Carlone. A polynomial-time solution for robust registration with extreme outlier rates. *arXiv preprint arXiv:1903.08588*, 2019.

[217] Jiaolong Yang, Hongdong Li, Dylan Campbell, and Yunde Jia. Go-ICP: A globally optimal solution to 3D ICP point-set registration. *IEEE transactions on pattern analysis and machine intelligence*, 38(11):2241–2254, 2015.

[218] Nan Yang, Rui Wang, and Daniel Cremers. Feature-based or Direct: An Evaluation of Monocular Visual Odometry. *arXiv preprint arXiv:1705.04300*, 2017.

[219] Shichao Yang and Sebastian Scherer. Direct monocular odometry using points and lines. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3871–3877. IEEE, 2017.

[220] Zhenfei Yang and Shaojie Shen. Monocular visual–inertial state estimation with online initialization and camera–IMU extrinsic calibration. *IEEE Transactions on Automation Science and Engineering*, 14(1):39–51, 2016.

[221] Zhenfei Yang and Shaojie Shen. Monocular visual–inertial state estimation with online initialization and camera–imu extrinsic calibration. *IEEE Transactions on Automation Science and Engineering*, 14(1):39–51, 2017.

[222] Chao Yu, Zuxin Liu, Xin-Jun Liu, Fugui Xie, Yi Yang, Qi Wei, and Qiao Fei. Ds-slam: A semantic visual slam towards dynamic environments. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1168–1174. IEEE, 2018.

[223] Pan Yue, Yang Bisheng, Liang Fuxun, and Dong Zhen. Iterative Global Similarity Points: A robust coarse-to-fine integration solution for pairwise 3D point cloud registration. In *2018 International Conference on 3D Vision (3DV)*, 2018.

[224] Matthew D Zeiler. ADADELTA: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[225] Guangcong Zhang, Mason J Lilly, and Patricio A Vela. Learning binary features online from motion dynamics for incremental loop-closure detection and place recognition. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 765–772. IEEE, 2016.

[226] Hao Zhang, Fei Han, and Hua Wang. Robust Multimodal Sequence-Based Loop Closure Detection via Structured Sparsity. In *Robotics: Science and Systems*, 2016.

[227] Liang Zheng, Yi Yang, and Qi Tian. SIFT meets CNN: A decade survey of instance retrieval. *IEEE transactions on pattern analysis and machine intelligence*, 40(5):1224–1244, 2018.

[228] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Fast global registration. In *European Conference on Computer Vision*, pages 766–782. Springer, 2016.

[229] Yi Zhou, Hongdong Li, and Laurent Kneip. Canny-vo: Visual odometry with rgb-d cameras based on geometric 3-d–2-d edge alignment. *IEEE Transactions on Robotics*, 35(1):184–199, 2019.

Deliverables from this Thesis

Thesis Information

- **Title:** Techniques for a Failsafe Visual Inertial SLAM System
- **Author:** KUSE, Manohar Prakash

Publications Related to this Thesis

- Yonggen Ling, Manohar Kuse, and Shaojie Shen. Edge alignment-based visual–inertial fusion for tracking of aggressive motions. *Autonomous Robots*, 42(3):513–528, 2018
- Manohar Kuse and Shaojie Shen. Learning Whole-Image Descriptors for Real-time Loop Detection and Kidnap Recovery under Large Viewpoint Difference. *CoRR*, abs/1904.06962, 2019
- Manohar Kuse and Shaojie Shen. Robust camera motion estimation using direct edge alignment and sub-gradient method. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 573–579. IEEE, 2016

Open Source Software Packages

- Direct Edge Alignment:
  `https://github.com/mpkuse/edge_alignment`
- Weakly Supervised Whole Image Descriptor:
  `https://github.com/mpkuse/cartwheel_train`
- Place Recognition and Pose Computation Plugin for VINS-Fusion:
  `https://github.com/mpkuse/cerebro`
- Kidnap Aware Pose Graph Solver:
  `https://github.com/mpkuse/solve_keyframe_pose_graph`

Demo Videos

- "VINS-Fusion+Cerebro: Highlight Video" `https://www.youtube.com/watch?v=lDzDHZkInos`
- "Relocalization from stored maps" `https://www.youtube.com/watch?v=OViEEB3rINo`
- "AR Demo under kidnap" `https://www.youtube.com/watch?v=HL7Nk-fBNqM`
- "Visual-Inertial Odometry with Edge Alignment" `https://www.youtube.com/watch?v=Pctn3jrBk4w`

Other Publication during Study

- Manohar Kuse and Sunil Prasad Jaiswal. Graph modelling of 3D geometric information for color consistency of multiview images. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 1394–1398. IEEE, 2015
- Manohar Kuse, Sunil Prasad Jaiswal, and Shaojie Shen. Deep-mapnets: A residual network for 3D environment representation. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 2652–2656. IEEE, 2017

Most Influencial Citations

- Christian Kerl, Jürgen Sturm, and Daniel Cremers. Robust odometry estimation for RGB-D cameras. In *ICRA*, pages 3748–3754. IEEE, 2013
- Relja Arandjelovic, Petr Gronát, Akihiko Torii, Tomás Pajdla, and Josef Sivic. NetVLAD: CNN Architecture for Weakly Supervised Place Recognition. In *CVPR*, pages 5297–5307. IEEE Computer Society, 2016
- Niko Sünderhauf and Peter Protzel. Switchable constraints for robust pose graph SLAM. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 1879–1884. IEEE, 2012
- Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Fast global registration. In *European Conference on Computer Vision*, pages 766–782. Springer, 2016

- Tong Qin, Peiliang Li, and Shaojie Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4):1004–1020, 2018

- Sameer Agarwal, Keir Mierle, and Others. Ceres Solver. `http://ceres-solver.org`