공학박사학위논문

Low-Drift Visual Odometry for Indoor Robotics

실내 로봇을 위한 영상 기반 주행 거리 기록계

2019년 **2**월

서울대학교 대학원

기계항공공학부

김 표 진

Low-Drift Visual Odometry for Indoor Robotics

실내 로봇을 위한 영상 기반 주행 거리 기록계

지도교수 김 현 진

이 논문을 공학박사 학위논문으로 제출함

2018년 **5**월

서울대학교 대학원

기계항공공학부

김 표 진

김표진의 공학박사 학위논문을 인준함

2018년 **6**월

위 원 장 : _____

부위원장 : _____

위 원 : _____

위 원 : _____

위 원 : _____

Low-Drift Visual Odometry for Indoor Robotics

A Dissertation

by

PYOJIN KIM

Presented to the Faculty of the Graduate School of

Seoul National University

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

Department of Mechanical and Aerospace Engineering

Seoul National University

Supervisor : Professor H. Jin Kim

FEBRUARY 2019

Low-Drift Visual Odometry for Indoor Robotics

PYOJIN KIM

Department of Mechanical and Aerospace Engineering

Seoul National University

APPROVED:

Youdan Kim, Chair, Ph.D.

H. Jin Kim, Ph.D.

Chan Gook Park, Ph.D.

Nam Ik Cho, Ph.D.

Chang Yeong Kim, Ph.D.

To my mother, father, and sister with love

To my eternal love, Youngeun Park

Abstract

Low-Drift Visual Odometry for Indoor Robotics

Pyojin Kim
Department of Mechanical and Aerospace Engineering
The Graduate School
Seoul National University

This thesis explores the robust and accurate 6-DoF camera motion estimation from a sequence of images, called visual odometry (VO) or visual simultaneous localization and mapping (vSLAM). We focus on the robustness and high accuracy of the VO and visual localization by explicitly modeling the light changes as an affine illumination model, and utilizing the indoor environmental structures such as lines and planes. This brings the significant advantage to VO that it does not lose estimation accuracy in challenging environments such as light-changing conditions or pure, on the spot rotations.

The first part of the thesis proposes a novel patch-based illumination invariant visual odometry algorithm (PIVO). PIVO employs an affine illumination change model per each patch in the image to compensate unexpected, abrupt, and irregular illumination changes during the direct motion estimation. PIVO infers camera geometry directly from the images, i.e., the raw sensor measurements, without intermediate abstraction, for instance in the form of keypoint matches. We furthermore incorporate a motion prior from feature-based stereo visual odometry in the optimization, resulting in higher accuracy and more stable motion estimates. We evaluate the proposed VO algorithm on a variety of datasets, and demonstrate autonomous flight experiments with an aerial robot, showing that the proposed method successfully estimates 6-DoF pose under significant illumination changes.

In the second part of the thesis, we propose a low-drift VO that separately estimates rotational and translational motion from lines, planes, and points found in RGB-D images. To estimate the rotational motion that is a main source of drift in VO in an accurate and drift-free manner, we

exploit both lines and planes jointly from environmental regularities. We recognize and track the structural regularities with an efficient SO(3)-manifold constrained mean shift algorithm. Once the absolute camera orientation is found, we recover the translational motion from all tracked points with and without depth by minimizing the de-rotated reprojection error. We compare the proposed algorithm to other state-of-the-art VO methods on a variety of RGB-D datasets that include especially challenging pure rotations, and demonstrate improved accuracy and lower drift error.

Keywords: Aerial Robotics, Visual Navigation, Visual Odometry, Illumination Changes, Structural Regularities, Manhattan World

Student Number: 2013-20663

Table of Contents

Page

Abstract vi

Table of Contents viii

List of Tables xiii

List of Figures xiv

Abbreviations and Acronyms xx

Chapter

1 Introduction 1

1.1 Motivation 1

1.2 Classification of Visual Odometry & SLAM Methods 3

1.2.1 Indirect vs. Direct 3

1.2.2 Filtering vs. Optimization 4

1.2.3 State of the Art 5

1.3 Contribution and Outline 7

1.3.1 Robust Visual Odometry under Dynamic Lighting Changes 8

1.3.2 Low-Drift Visual Odometry in Structured Environments 10

2 Background 12

2.1 Camera Model 12

2.2 Rigid Body Motion 14

2.3 Nonliear Least Squares Optimization 17

2.3.1 Newton Optimization 18

2.3.2 Gauss-Newton Optimization 19

2.3.3 Levenberg-Marquardt Optimization 20

3 Robust Visual Odometry to Irregular Illumination Changes with RGB-D Camera . . . 22

3.1 Introduction . . . 23
3.2 Related Work . . . 26
3.3 Notation and Problem Statement . . . 27
3.4 Proposed Visual Odometry Algorithm . . . 29
3.4.1 Illumination Change Model . . . 31
3.4.2 Planar Patch Selection . . . 31
3.4.3 Direct Motion Estimation . . . 34
3.5 Evaluation . . . 36
3.5.1 Synthetic RGB-D Dataset . . . 37
3.5.2 Author-collected Stationary RGB-D Dataset . . . 43
3.6 Conclusion . . . 46
4 Robust Visual Localization in Changing Lighting Conditions . . . 47
4.1 Introduction . . . 48
4.2 Related Work . . . 50
4.3 Astrobee's Current Localization System . . . 51
4.3.1 Offline Map Construction . . . 52
4.3.2 Online Image Localization . . . 52
4.4 Effect of Changing Lighting Conditions . . . 52
4.5 Illumination-Robust Visual Localization . . . 55
4.5.1 Brightness Recognition . . . 56
4.5.2 Map Recommendation System . . . 58
4.6 Evaluation . . . 58
4.6.1 Constant Exposure Time . . . 58
4.6.2 Dynamic Exposure Time . . . 60
4.7 Conclusion . . . 63
5 Autonomous Flight with Robust Visual Odometry under Dynamic Lighting Conditions 64
5.1 Introduction . . . 65
5.2 Related Work . . . 67

5.3 Notation and Problem Statement . . . 70
5.4 System Overview . . . 72
5.5 Visual Odometry Pipeline . . . 74
5.5.1 Feature-based Estimation . . . 74
5.5.2 Direct Estimation . . . 75
5.5.3 Discussion . . . 77
5.6 Energy Function Analysis . . . 79
5.6.1 Energy Function Convergence . . . 79
5.6.2 Measurement Equation Linearity . . . 83
5.7 Experimental Results . . . 85
5.7.1 Experiments on the Datasets . . . 86
5.7.2 Experiments on an Autonomous Aerial Robot . . . 93
5.8 Conclusion . . . 99
6 Visual Odometry with Drift-Free Rotation Estimation Using Indoor Scene Regularities 105
6.1 Introduction . . . 106
6.2 Related Work . . . 107
6.3 Proposed Method . . . 109
6.3.1 Rotational Motion Estimation . . . 110
6.3.2 Translational Motion Estimation . . . 113
6.4 Evaluation . . . 114
6.4.1 Tests with ICL-NUIM Datasets . . . 115
6.4.2 Tests with Author-collected RGB-D Datasets . . . 117
6.5 Conclusion . . . 119
7 Low-Drift Visual Odometry in Structured Environments by Decoupling Rotational and Translational Motion . . . 120
7.1 Introduction . . . 121
7.2 Related Work . . . 123
7.3 Background 3D Geometry . . . 124

7.3.1 Gaussian Sphere . . . 124
7.3.2 Rotation Motion with Vanishing Directions . . . 125
7.4 Proposed Method . . . 126
7.4.1 Orthogonal Plane-based Visual Odometry . . . 126
7.4.2 Drift-Free Rotation Estimation with Lines and Planes . . . 127
7.4.3 Translation Estimation with All Tracked Points . . . 128
7.5 Evaluation . . . 130
7.5.1 ICL-NUIM Dataset . . . 131
7.5.2 TUM RGB-D Dataset . . . 134
7.5.3 TAMU RGB-D Dataset . . . 137
7.5.4 Author-collected RGB-D Dataset . . . 138
7.6 Conclusion . . . 140
8 Indoor RGB-D Compass from a Single Line and Plane . . . 141
8.1 Introduction . . . 142
8.2 Related Work . . . 144
8.3 Proposed Method . . . 145
8.3.1 Dominant Plane Detection and Tracking . . . 146
8.3.2 One Line and One Plane RANSAC . . . 148
8.3.3 Multiple Lines Refinement . . . 151
8.4 Evaluation . . . 152
8.4.1 ICL-NUIM Dataset . . . 153
8.4.2 TUM RGB-D Dataset . . . 155
8.5 Conclusion . . . 157
9 Linear RGB-D SLAM for Planar Environments . . . 158
9.1 Introduction . . . 159
9.2 Related Work . . . 161
9.3 Proposed Method . . . 162
9.3.1 Line and Plane based Visual Odometry . . . 163

9.3.2 Orthogonal Plane Detection 164
9.3.3 Linear RGB-D SLAM 165
9.4 Evaluation 168
9.4.1 ICL-NUIM Dataset 169
9.4.2 TUM RGB-D Dataset 170
9.4.3 Author-collected RGB-D Dataset 171
9.4.4 Augmented Reality with Linear RGB-D SLAM 172
9.5 Conclusion 173
10 Conclusion 174
Abstract (*in Korean*) 192

List of Tables

3.1 Estimation results with synthetic RGB-D dataset. 38
3.2 Estimation results with author collected RGB-D dataset. 44

4.1 Evaluation Results of Illumination-robust Visual Localization on Constant Exposure Time 60
4.2 Illumination-robust Localization with Dynamic Exposure 61

5.1 Accuracy Improvement of the Proposed Algorithm 88
5.2 Experimental Results on Synthetic EuRoC Benchmark 88

6.1 Evaluation Results on ICL-NUIM Benchmark 115

7.1 Evaluation Results on ICL-NUIM Benchmark 132
7.2 Evaluation Results on TUM RGB-D Benchmark 135

8.1 Comparison of the absolute rotation error (degrees) on ICL-NUIM benchmark [1]. 153
8.2 Comparison of the absolute rotation error (degrees) on TUM RGB-D dataset [2]. 156

9.1 Evaluation Results of ATE RMSE (unit: m) on ICL-NUIM Benchmark 169
9.2 Evaluation Results of ATE RMSE (unit: m) on TUM RGB-D Benchmark 170

List of Figures

1.1 Autonomous drone with cameras checking inventory in the warehouse (left) [3]. Augmented reality running on the Microsoft HoloLens, looking at the design of the building together (right) [4]. 2

1.2 Classification of the current state-of-the-art visual odometry & SLAM methods. . 3

1.3 Indirect (top) vs. direct (bottom) approaches in VO & vSLAM. 4

1.4 Hexacopter aerial robot used in our autonomous flight experiments with varying light conditions by turning on and off the lights repeatedly (top). The estimated (magenta) and true (black) trajectories with the proposed VO method overlap significantly, and the point cloud is consistently recovered despite sudden and severe illumination changes. 9

1.5 Estimated trajectories with the proposed and other VO methods in building-scale environments (top). Augmented reality (AR) rendering results from the proposed linear SLAM approach are shown with the ISS 3D model. 11

2.1 The frontal perspective pinhole camera model [5]. 13

2.2 The relative transformation between two consecutive frames. 16

3.1 Illustrative examples of irregular illumination changes. (a) An automatic exposure control of camera makes the intensity of images change when the camera moves between different global illumination conditions. (b+c) Illumination variation in the TUM dataset ‘fr1/room’ and ‘fr3/struc&notex’. (d) Comparison of the estimation results of DVO [6], EDVO [7], and the algorithm proposed in this paper (namely PIVO) with ‘fr3/struc&notex’. A large drift error takes place under DVO when the illumination changes occur as illustrated in (c). 25

3.2 The notations and setting of the proposed visual odometry algorithm. 28

3.3 Overview of the proposed visual odometry algorithm. 30
3.4 Input and output images of the proposed visual odometry algorithm. (c) shows residual image between the patch based keyframe and the current image frame. . 32
3.5 Planar patch selection results based on the plane RANSAC algorithm. 33
3.6 Example images from the synthetic RGB-D dataset. 37
3.7 Comparison of VO results with the ground truth. 39
3.8 Absolute trajectory errors of each tested dataset. 40
3.9 Weighting functions with respect to the residuals of each method. 41
3.10 Residual distribution of each method. 42
3.11 The true and estimated illumination change model parameters. 43
3.12 The RGB image sequences in the author-collected RGB-D dataset. 44
3.13 Comparison of ATE and weighted residual errors from 'LAB2'. 45

4.1 Astrobee, a free-flying robot designed to autonomously navigate on the International Space Station (ISS), can localize robustly under changing lighting conditions within multiple maps reconstructed offline using structure from motion (SfM). Two pictures show light conditions on the ISS during day (left) and night (right). 49
4.2 Astrobee's mapping and localization algorithms. 51
4.3 Astrobee (left) and experimental environment in the granite lab (right) simulating the interior of ISS. Red circles on the right denote the adjustable lights. The green circle is the place where the intensity of light is measured with a digital light meter. The blue circles indicate the AR tag and overhead camera for providing the ground-truth position. The yellow circle indicates Astrobee's navigation camera. 53
4.4 Images taken under different lighting conditions in the granite lab. The wall panels imitate the interior of the ISS. 54
4.5 Brightness distributions of the images in Figure 4.4. Each pixel is represented as 8-bit grayscale from 0 (black) to 255 (white). 54

4.6 The success rate for various lighting / map combinations. 55
4.7 The illumination-robust visual localization pipeline. The proposed algorithm (blue box) is inserted into the original pipeline (Figure 4.2). 56
4.8 Illustration of the proposed brightness recognition algorithm. 57
4.9 Evaluation results comparing the proposed and current localization methods with constant exposure time, a bright map, and no motion. The dotted vertical lines represent the time instants at which each snapshot is taken. The current estimated lighting condition (green line) shows similar behavior to the brightness level of the actual images in the third row. Although the lights dimmed from frames 200 to 250, the proposed algorithm (red line) shows no failure and maintains the proper number of inliers whereas the current method (blue line) cannot localize. 59
4.10 'Circle' (left) and 'Sideways' (right) trajectories estimated by our method (in red) are very similar to the ground truth trajectories (in black). 60
4.11 Translational error of each method in the experiments. 61
4.12 Results on Astrobee in the stationary case with dynamic exposure time. The exposure time setting (cyan line) changes if the estimated lighting condition (green line) is too dark or too bright. Failures occasionally occur when the lighting condition is too dark to detect features (almost black). 62

5.1 Hexacopter aerial robot in our autonomous flight experiments with varying light conditions by turning on and off the lights repeatedly. 66
5.2 Stereo camera model and image coordinate systems. 71
5.3 Overview of the proposed stereo visual odometry pipeline. 73
5.4 Topological representation of the proposed algorithm. 73
5.5 Kernels for feature detection and circular matching strategy. (figures courtesy of Andreas Geiger) 74
5.6 A motion prior from feature-based VO stabilizes the direct VO significantly. . . . 78

5.7 Our sequential VO shows the best accuracy among other direct only [8] or feature only VO methods [9]. 79
5.8 Tendency of (a) the reprojection and photometric error for transformation with respect to each translational and rotational direction, and (b) the first derivatives of the (a). 80
5.9 Convergence history of the feature-based and direct estimation with the error. . . 82
5.10 Dimensionless linearity index of the feature-based and direct estimation with respect to the (a) translational and (b) rotational motion of the camera. 84
5.11 Extracts from the synthetic EuRoC dataset and author-collected dataset. 86
5.12 Red squares in (c) denote the image patches for compensating irregular illumination changes, and (d) shows the true and estimated trajectories. 89
5.13 (a) The photometric error of the three direct VO methods is drawn in a logarithmic scale where photometric disturbances occur between the gray dotted lines. (b) The true and estimated photometric parameters with the proposed method overlap significantly. 90
5.14 Comparison of the proposed and other VO methods on the multistoried stairway from the 1st to 6th floor. 92
5.15 Schematic diagram of the data flow in our experimental setup. 94
5.16 The image areas in the red patches show successful photometric compensation. . 96
5.17 Comparison of the proposed and other VO methods in illumination changes. . . . 96
5.18 Runtime evaluation of the proposed algorithm on the aerial robot. 97
5.19 Flight experiment results in a light-changing environment. 98

6.1 The drift of the rotation estimate is the main source of position inaccuracy in VO. 107
6.2 Overview of the algorithm that separately estimates rotation and translation. . . . 112
6.3 Some motion estimation results of the proposed algorithm in the ICL-NUIM dataset. 115

6.4 The inferred MF orientation is drawn in the top right corner of (b), and (d) shows the reconstructed trajectory and consistent point cloud. 116
6.5 The rotation matrix errors for the proposed and other VO algorithms. 117
6.6 Example images from the author-collected RGB-D dataset. 118
6.7 Motion estimation results with the proposed algorithm compared to other VO methods on the author-collected RGB-D dataset in a single-loop (a) and multiple-loop (b) sequences. 118

7.1 Example of a structured environment exhibiting strong orthogonal spatial regularities. 122
7.2 Geometric relationship between the lines, planes, and the Gaussian sphere. 125
7.3 Clustered lines and segmented planes are overlaid on the RGB image. 128
7.4 Overview of the proposed LPVO algorithm. 129
7.5 Estimated trajectories with LPVO (magenta), OPVO (dark green), and ground-truth (black) in the ICL-NUIM dataset Living Room 0, 2 and Office Room 1, 3. 133
7.6 Absolute rotational error (top) and translational error (bottom) for the proposed and other VO algorithms are plotted. 135
7.7 Estimated trajectories with LPVO (magenta), OPVO (dark green), and ground-truth (black) in the TUM fr3_longoffice, fr3_struc_notex_far, fr3_struc_tex_near, and fr3_large_cabinet. 136
7.8 Example image from 'Corridor-A-const', the clustered lines/planes, and the inferred MF orientation are shown on the left. 137
7.9 Example images from the author-collected RGB-D dataset. 138
7.10 Estimated trajectories with the proposed and other VO methods on the author-collected dataset in a single-loop (left) and multiple-loop (right) sequences. . . . 139

8.1 A single line and a single plane from RGB-D images. 143

8.2 Geometric relationships between the line, plane, and the Gaussian sphere in the MW. 146
8.3 Overview of the proposed algorithm. 147
8.4 The normal vector of a dominant plane from the distribution of the surface normals. 147
8.5 Orthogonal distance metric from the endpoints of the line l to an auxiliary line $\hat{l}$. . 149
8.6 Improved performance to recognize the regularities of structural space. 150
8.7 Examples of the MW from the ICL-NUIM [1] and TUM RGB-D [2] datasets. . . 152
8.8 Experimental results in the ICL-NUIM dataset (a) 'Living Room 0' and (b) 'Office Room 1'. 154
8.9 Comparison with and without refinement step (NO) versus the other algorithms. . 155
8.10 Inferred MW (VPs) orientation, the clustered lines and plane with the proposed method are overlaid on top of the RGB images in the TUM RGB-D dataset (a) 'fr3_longoffice' and (b) 'fr3_nostruc_notex'. 156
8.11 The statistical distribution of the absolute rotation error from the 'fr3_struc_tex'. . 157

9.1 Linear RGB-D SLAM generates a consistent global planar map using a linear KF. 160
9.2 Overview over the complete L-SLAM algorithm. 163
9.3 Results of orthogonal plane detection are overlaid on top of the RGB images. . . 165
9.4 Selected motion estimation results in the ICL-NUIM dataset. 169
9.5 Trajectories with L-SLAM (magenta) and true (black) for the TUM RGB-D dataset. 170
9.6 Estimated trajectories with the proposed and other SLAM methods on the author-collected RGB-D dataset in a square corridor sequence. 171
9.7 Augmented reality (AR) implementation results. 172

Abbreviations and Acronyms

2D : 2-Dimensional
3D : 3-Dimensional
DoF : Degrees of Freedom
GPS : Global Positioning System
IMU : Inertial Measurement Unit
VO : Visual Odometry
VIO : Visual-Inertial Odometry
EKF : Extended Kalman Filter
UKF : Unscented Kalman Filter
SLAM : Simultaneous Localization And Mapping
MW : Manhattan World
MF : Manhattan Frame
VP : Vanishing Point
SfM : Structure-from-Motion
ROS : Robot Operating System
RMSE : Root Mean Squared Error
RPE : Relative Pose Error
ATE : Absolute Trajectory Error
ARE : Absolute Rotation Error
MAV : Micro Aerial Vehicle
UAV : Unmanned Aerial Vehicle
VR : Virtual Reality
AR : Augmented Reality
ISS : International Space Station
RANSAC : RANdom SAmple Consensus

1 Introduction

1.1 Motivation

The human being understands the world with the eyes. Although humans have and use other various sensory organs such as touch and hearing, we accept about 85% of the information they acquire through the visual perception system, the eyes. The significant portion about 30% of the human brain devoted to visual processing compared to 8% for tactile sensation and 3% for hearing [10]. They indicate both the importance and the complexity of the ability to perceive the surrounding 3D world around us from 2D image sequences captured by our eyes. While our vision system, the eyes and brain, enables us to perform various vision tasks such as object detection, tracking, and classification, one of the most fundamental abilities is to precisely recognize our own six degrees of freedom movements in a three-dimensional (3D) space. This space perception ability allows us to explore the unknown environments, recognize where we are now in 3D space, and move without colliding with the unidentified objects, and so on.

Recent emerging technologies from autonomous cars and drones to augmented and virtual reality (AR/VR) need such space perception ability to perceive, reconstruct and ultimately un-

Figure 1.1: Autonomous drone with cameras checking inventory in the warehouse (left) [3]. Augmented reality running on the Microsoft HoloLens, looking at the design of the building together (right) [4].

derstand the 3D world like the human being does: a drone that flies autonomously needs to know where it is now, and the 6-DoF camera pose and the geometry of the scene are required to render virtual objects into the image correctly, and allow them to interact real-world objects as shown in Fig. 1.1. To convincingly simulate a virtual object standing on a real-world table, both the 6-DoF pose of the observer's head and the location of the real-world table should be known exactly and precisely.

Many researchers in robotics and computer vision have been actively investigating for decades the 3D spatial perception capabilities, commonly called various terminologies such as visual odometry (VO), visual simultaneous localization and mapping (vSLAM), or structure and motion (SaM).

Therefore, this dissertation mainly focuses on the robustness and high accuracy of the VO and visual localization by explicitly modeling the light changes as a simple affine illumination model, and utilizing the indoor environmental structures such as lines and planes. This has the major advantage in VO that it does not lose estimation accuracy in challenging environments such as light-changing conditions or pure, on the spot rotations.

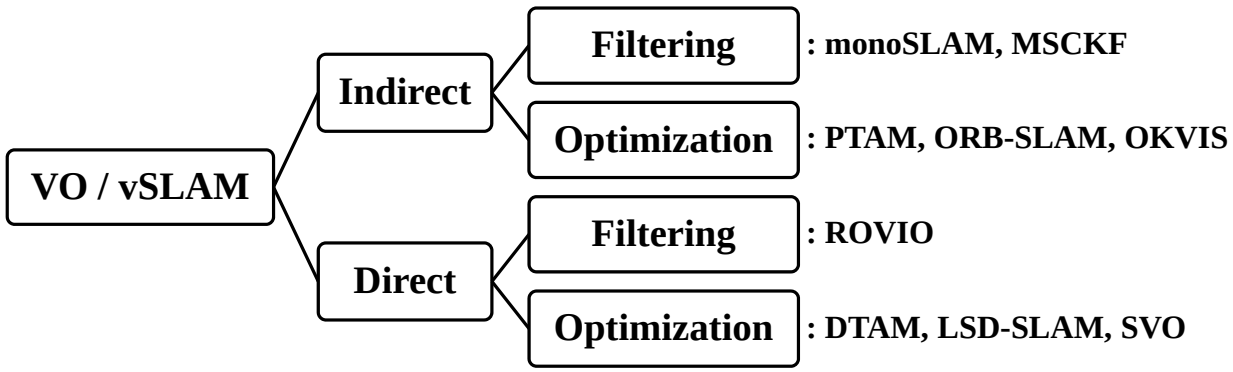


Figure 1.2: Classification of the current state-of-the-art visual odometry & SLAM methods.

1.2 Classification of Visual Odometry & SLAM Methods

This section defines two ways to classify the current state-of-the-art visual odometry and visual SLAM approaches: indirect vs. direct and filtering-based vs. optimization-based methods as shown in Fig. 1.2. These classification criteria can be applied independently regardless of the used camera sensor modalities. They apply in the context of different sensor modalities including monocular, stereo, visual-inertial, and RGB-D cameras.

1.2.1 Indirect vs. Direct

The underlying most of the mathematical formulations in VO and vSLAM is a probabilistic model that takes noisy measurements from the camera $\mathbf{Y}$ as input, and estimates $\mathbf{X}$ for the hidden, unknown model parameters representing 6-DoF camera motion and surrounding 3D maps. Typically, a maximum likelihood method is employed, which finds the model parameters $\mathbf{X}$ that maximize the probability, i.e.,

$$\mathbf{X}^* := \underset{\mathbf{X}}{\operatorname{argmax}} P\left(\mathbf{X} \mid \mathbf{Y}\right) \tag{1.1}$$

Indirect methods divide the estimating 6-DoF camera motion and 3D geometry from images into two sequential steps: 1) keypoint detection and matching, 2) geometric optimization on the computed point correspondences. First, we pre-process the raw sensor measurements (image intensity values) to generate a condensed, intermediate image representation. In this step, we typ-

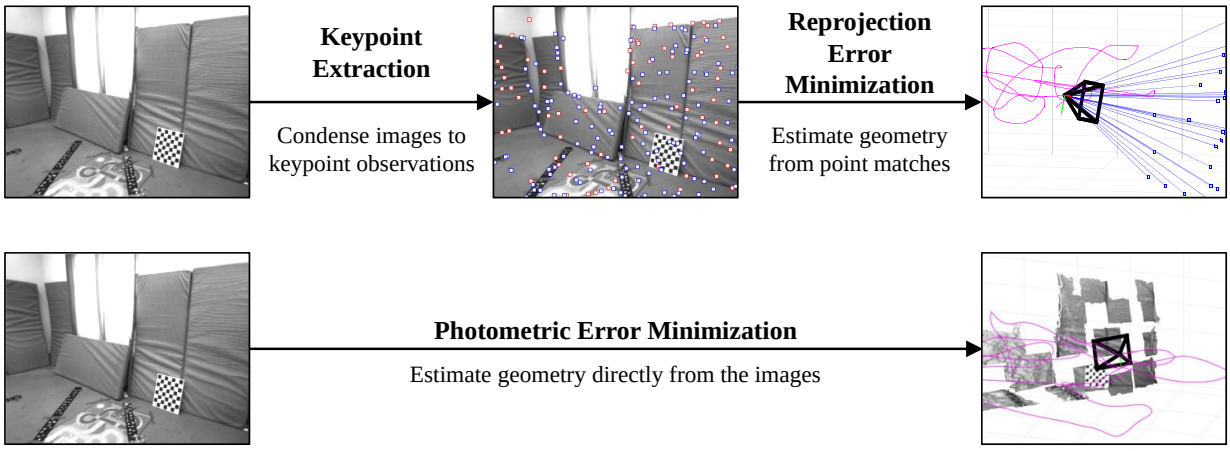


Figure 1.3: Indirect (top) vs. direct (bottom) approaches in VO & vSLAM.

ically extract and match a sparse set of keypoints; one could also use alternative methods such as establishing correspondences with the dense optical flows. Second, we interpret the intermediate image representations as for the noisy measurements $\mathbf{Y}$ in Eq. (1.1) in a probabilistic model to estimate the 6-DoF camera motion and surrounding 3D geometry. Since the pre-computed values such as keypoint positions are geometric quantities, indirect methods optimize the geometric error. Note that we can extract and match parametric representations of other geometric primitives than only points, such as lines, planes, and curve segments.

Direct methods skip the intermediate representation, and directly estimate the 6-DoF camera motion and 3D geometry on the raw camera sensor measurements, taking as the noisy measurements $\mathbf{Y}$ in a probabilistic model. Since the camera sensor provides the photometric measurements (intensity values from 0$\sim$255), the direct approach estimates the best 6-DoF pose of the camera by minimizing the photometric error. Note that other sensor modalities like RGB-D cameras, which directly measure the depth information, direct formulations can also optimize a geometric error. Figure 1.3 shows the difference between the indirect and the direct approaches.

1.2.2 Filtering vs. Optimization

The most significant difference between the filtering-based and optimization-based methods is whether to estimate the 6-DoF camera motion and 3D geometry within the filtering framework

like Kalman filter, or to solve the problems within nonlinear optimization formulations.

Filtering-based methods estimate and continuously update a joint probability distribution (covariance) over all relevant parameters in the state vector. In the prediction (propagation) step, filtering-based approaches add new model parameters with large initial uncertainty or increases the uncertainty of existing states. New measurements are used in the update (correction) step to update the observed state vector for reducing uncertainty. This approach is commonly implemented as Kalman filtering framework. Since this state vector representation allows us to marginalize old state variables easily, filtering-based methods typically keep the current 6-DoF camera pose only in the state vector. Recent filtering-based methods such as the multi-state constrained Kalman filter (MSCKF) employ a sliding window of past camera poses and internal calibration parameters.

Optimization-based methods compute the 6-DoF camera pose and 3D geometry in the form of a non-linear energy function, which is minimized by utilizing the optimization techniques such as Gauss-Newton, Levenberg–Marquardt (LM) algorithms. To facilitate this, they aggressively drop available information and only keep a small subset of frames called the keyframes. Furthermore, it allows linearizations to be evaluated after better estimates are available. [11] concludes that filtering-based approach has a better accuracy per unit of computation for small problems, whereas optimization-based methods achieve a better trade-off for larger systems by using more observations.

1.2.3 State of the Art

We briefly review the relevant state-of-the-art VO and vSLAM algorithms that are the representatives of each category shown in Fig. 1.2. We list the algorithms which are based on using one or more RGB and depth cameras.

MonoSLAM: Real-time Single Camera SLAM [12] is a point-based (indirect) visual odometry and SLAM approach within the extended Kalman filtering. It uses a constant velocity model to propagate the variables (camera poses, 3D points) in the state vector, and models the residuals in the correction step as the nonlinear reprojection error. Similar work using inverse depth param-

eterization is done in EKF monoSLAM [13]. Multi-State Constraint Kalman Filter (MSCKF) [14] is a similar filtering-based, indirect VO method combined with an IMU. The MSCKF tightly couples both camera and IMU modalities, and formulates the estimation problem as the extended Kalman filter, keeping as state a sliding window of recent camera frames.

ORB-SLAM: A Versatile and Accurate SLAM Systems [15] is a classical indirect SLAM algorithm, which is based on the nonlinear optimization. Current 6-DoF camera motion is estimated in real-time using model-based tracking, while the global, consistent point map is being optimized in the back-end using traditional, nonlinear bundle adjustment in parallel like PTAM [16]. It minimizes a non-linear error function called the reprojection error at every frame to estimate the camera motion similar to StereoScan [9] and DEMO [17]. It works really well for many scenarios, and is very well engineered. Furthermore, it includes loop closure detection and relocalization modules, resulting in large and consistent 3D global maps. OKVIS [18] optimizes a nonlinear cost function consisting of visual terms and inertial terms over a sliding window, which is the optimization-based version of the MSCKF.

ROVIO: Robust Visual Inertial Odometry Using a Direct EKF-Based Approach [19] is direct visual-inertial odometry that tightly fuses inertial measurements with visual data by means of an extended Kalman filter framework. It directly integrates a photometric error into an innovation term in the filter update step by employing square image patches as landmark descriptors. The measurement model is to directly predict the appearance of a pixel patch (intensity values) of a reference view given the pixel values in the current camera view. The feature correspondence is an inherent part of the estimation process; thus no additional feature extraction or matching processes are required like [20].

LSD-SLAM: Large-scale Direct SLAM [21, 22] is a direct visual SLAM algorithm that is based on the nonlinear photometric error minimization called the direct image alignment used in DTAM [23]. It maintains and tracks on a global 3D map of the environment, which contains a pose-graph of keyframes with associated probabilistic semi-dense depth maps. Recently, a sparse hybrid VO approach called SVO [24, 25] is proposed to combine the advantages of indirect and direct methods. Furthermore, the author of LSD-SLAM proposes the DSO [26], which opti-

mizes all involved model parameters including 3D geometry, 6-DoF camera poses, illumination changes, and camera intrinsics in a joint, consistent Gauss-Newton manner.

1.3 Contribution and Outline

These thesis develop a novel direct VO and visual localization algorithms. In contrast to the current state-of-the-art direct VO and SLAM approaches, the proposed direct methods do not rely on the photo-consistency assumption: a world point observed by two cameras is assumed to yield the same brightness in both images, but rather model the light changes as an affine illumination model during the photometric error minimization. This thesis also presents a low-drift visual odometry algorithm that separately estimates rotational and translational motion from lines, planes, and points found in RGB-D images. Contrary to the typical VO and vSLAM methods that estimate the 6-DoF camera motion jointly without distinction between rotational and translational movement, we design a hybrid visual odometry algorithm which separately estimates the rotational and translational motion to achieve improved accuracy and low drift error. To improve the accuracy of rotational motion estimation, we exploit the environmental regularities such as lines and planes, common in man-made environments.

This cumulative thesis comprises seven full-length publications [8, 27, 28, 29, 30, 31, 32]. Five of these papers [8, 27, 29, 30, 31] were published in the international conferences. Two of these works [28, 32] have been submitted to the international conferences and journals, and are currently under review.

In this thesis, two different VO approaches are developed, focusing on the robustness and high accuracy of the VO and visual localization by explicitly modeling the light changes as a simple affine illumination model, and utilizing the indoor environmental structures such as lines and planes.

1.3.1 Robust Visual Odometry under Dynamic Lighting Changes

Sensitivity to light conditions poses a challenge when utilizing direct VO for autonomous navigation of small aerial vehicles in various applications. We propose an illumination-robust direct visual odometry for a stable autonomous flight of an aerial robot under unpredictable light condition. The proposed direct VO algorithm is robust to the light-changing environment by employing an affine illumination model to compensate abrupt, irregular illumination changes during the nonlinear photometric error minimization. We furthermore incorporate a motion prior from feature-based stereo visual odometry in the nonlinear optimization, resulting in higher accuracy and more stable motion estimate. Thorough analyses of convergence rate and linearity index for the feature-based and direct VO methods support the effectiveness of the usage of the motion prior knowledge.

The odometry and compensation components of the illumination-robust direct visual odometry are described in [8] (Chapter 3), the integration with the feature-based method as a motion prior and the autonomous flight experiments with an aerial robot in [28] (Chapter 5). We also investigate and quantify the effect of lighting variations on visual point-based localization systems, and extend the standard visual localization algorithm to make it more robust to changing-light environments in [27] (Chapter 4).

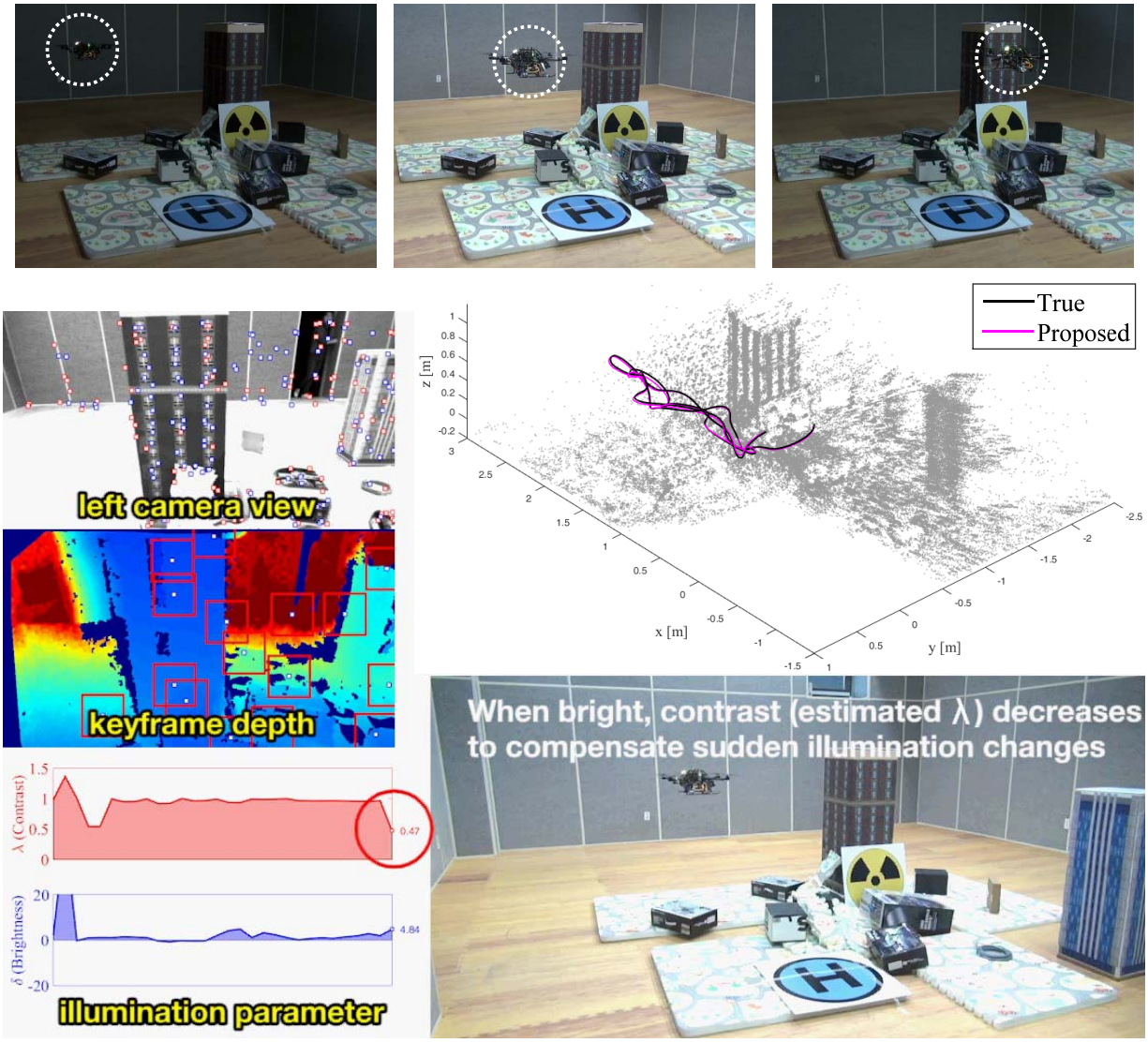


Figure 1.4: Hexacopter aerial robot used in our autonomous flight experiments with varying light conditions by turning on and off the lights repeatedly (top). The estimated (magenta) and true (black) trajectories with the proposed VO method overlap significantly, and the point cloud is consistently recovered despite sudden and severe illumination changes.

1.3.2 Low-Drift Visual Odometry in Structured Environments

Most existing typical VO and vSLAM approaches usually estimate the 6-DoF camera motion jointly without distinction between rotational and translational motion. However, inaccuracy in the rotation estimate is a primary source of drift in VO. Here, we propose a low-drift VO that separately estimates rotational and translational motion from lines, planes, and points found in RGB-D images. To estimate accurate and drift-free rotational motion, we exploit both lines and planes jointly from environmental regularities. We track the spatial regularities with an efficient SO(3)-manifold constrained mean shift algorithm. Once the absolute camera orientation for the environmental regularities is found, we recover the translational motion from all tracked points with and without depth by minimizing the de-rotated reprojection error.

The proposed visual odometry algorithm with drift-free rotation estimation is published in [29] (Chapter 6), and significantly surpasses the previous VO and vSLAM methods (direct approaches as well as indirect approaches) in tracking accuracy. Additional line features have been utilized to increase robustness and stability when inferring the absolute camera orientation in the indoor scene regularities in [30, 31] (Chapter 7 and 8). Furthermore, the proposed low-drift VO algorithm is extended to linear SLAM framework by mapping the planar landmarks in the planar environments, as well as the integration with a basic augmented reality (AR) engine is presented in [32] (Chapter 9).

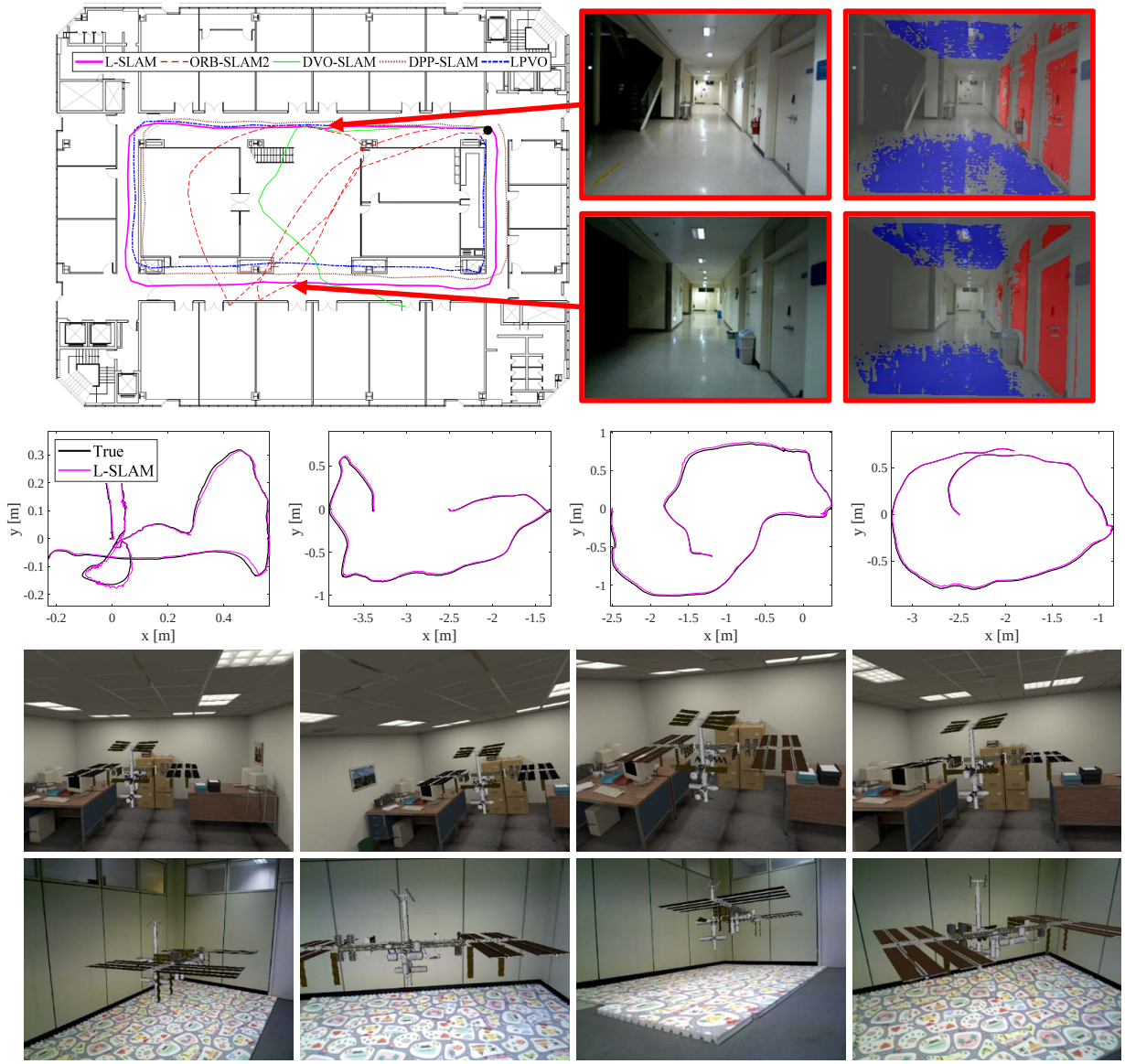


Figure 1.5: Estimated trajectories with the proposed and other VO methods in building-scale environments (top). Augmented reality (AR) rendering results from the proposed linear SLAM approach are shown with the ISS 3D model.

2

Background

In this chapter, we summarize the fundamental computer vision concepts, 3D geometry, and mathematical tools used throughout the thesis.

2.1 Camera Model

The camera model maps the points in the three-dimensional world to the two-dimensional image plane created by the camera sensor [33]. In this thesis, we use the simple pinhole camera model depicted in Figure 2.1. It simplifies the whole camera structures to an infinitely small hole, the pinhole, and the image plane. The real image plane in the camera is actually located behind the optical center of the camera, and not in front of the camera. This can be neglected without loss of generality by using the frontal perspective transformation.

The location of the pinhole is the optical center $\{C\}$ of the camera, and the focal distance between the optical center and image plane is the focal lengths f_x and f_y, respectively. Any 3D point $\mathbf{X} = [X, Y, Z]^\top \in S$ on the visible scene surface $S \subset \mathbb{R}^3$ expressed in the camera frame $\{C\}$ maps to the image pixel coordinates $\mathbf{x} = [x, y]^\top \in \Omega$ through the camera projection function

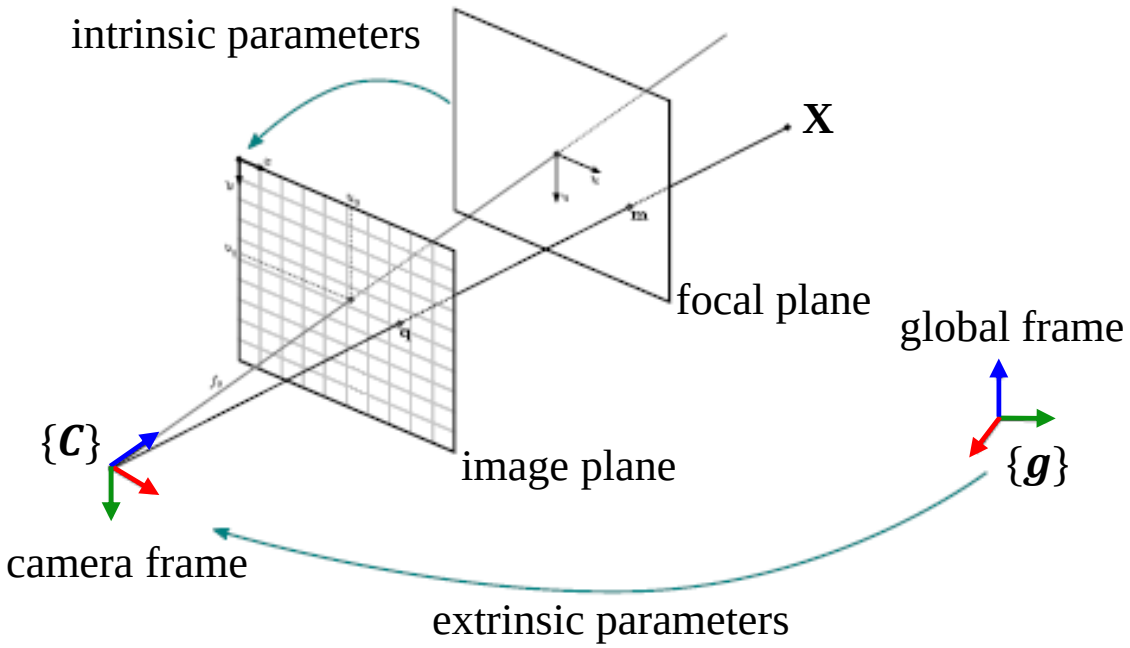


Figure 2.1: The frontal perspective pinhole camera model [5].

$\pi : \mathbb{R}^3 \mapsto \mathbb{R}^2$ as follows:

$$\mathbf{x} = \pi\left(\mathbf{X}\right) \tag{2.1}$$

$$x = f_x \frac{X}{Z} + c_x \tag{2.2}$$

$$y = f_y \frac{Y}{Z} + c_y \tag{2.3}$$

where f_x and f_y the two different focal lengths between focal and image plane since the pixels on the sensor chip do not have to be quadratic. The offsets c_x and c_y are the principal points, which would be ideally in the center point of the image.

We can also back-project the 3D point $\mathbf{X}$ with the pixel coordinates $\mathbf{x}$ and the depth value Z from the depth maps. These can be obtained from a stereo, RGB-D camera or by classical dense reconstruction approaches using standard cameras (e.g., DTAM [23] or REMODE [34]). The 3D point of the image point $\mathbf{x}$ can be reconstructed through the inverse projection function $\pi^{-1} : \mathbb{R}^2 \mapsto \mathbb{R}^3$ as follows:

$$\mathbf{X} = \pi^{-1}\left(\mathbf{x}, Z\right) \tag{2.4}$$

$$X = \frac{x - c_x}{f_x} Z \tag{2.5}$$

$$Y = \frac{y - c_y}{f_y} Z \tag{2.6}$$

$$Z = Z \tag{2.7}$$

The above intrinsic parameters (f_x, f_y, c_x, c_y) can be obtained by a camera calibration toolbox [35]. Further intrinsic parameters like skew and radial distortion coefficients can be determined during the camera calibration step. The sensor skew compensates for a sensor not mounted parallel to the camera lens, and the radial distortion coefficients describe the distortion in the image plane due to the lens. We remove these skew and radial distortions in the image before applying our VO algorithms to the camera images. Therefore, our camera follows the above pinhole camera projective model.

2.2 Rigid Body Motion

We represent the rigid body motion (i.e., the 6-DoF camera motion) as the 4×4 rigid body transformation matrix $T \in \mathrm{SE}(3)$ in Euclidean three-dimensional space [36]. This transformation represents the relative position and orientation movements between the two coordinate frames, so can be decomposed into a rotational and a translational part. The rotation denotes the orientation changes of the coordinate frame, and the translation means the linear movements in 3D space. A rigid body motion has six degrees of freedom in total, three degrees for rotation and three degrees for translation. It also allows us to map any 3D point from the world coordinate frame to the camera frame.

There are various mathematical representations for the rotation [37]. In this thesis, we represent the rotational motion of the camera as the 3×3 orthogonal rotation matrices $R \in \mathrm{SO}(3)$, which are belong to the special orientation group $\mathrm{SO}(3)$. Other frequently used representations are quaternions, Euler angles, and a combination of a rotation angle and axis. We represent the translational motion of the camera as the vector $\mathbf{t} \in \mathbb{R}^3$. The components of $\mathbf{t} = [t_x, t_y, t_z]^\top$ are

the translation along the x, y, and z axis. Rigid body motion is a combination of a rotation matrix from SO(3) and a translation vector from $\mathbb{R}^3$. It can be expressed as a 4×4 matrix $T \in$ SE(3) as follows:

$$T = \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \tag{2.8}$$

its inverse matrix can be written as follows:

$$T^{-1} = \begin{bmatrix} R^\top & -R^\top \mathbf{t} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \tag{2.9}$$

The rigid body transformation matrix also allows us to map the 3D point between two consecutive as follows:

$$\tilde{\mathbf{X}}^k = T_{k,k-1}\tilde{\mathbf{X}}^{k-1} \tag{2.10}$$

$$T_{k,k-1} = \begin{bmatrix} R_{k,k-1} & \mathbf{t}_{k,k-1} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \tag{2.11}$$

where $\tilde{\mathbf{X}}^k = [\mathbf{X}^k, 1]^\top$ is the homogeneous form of $\mathbf{X}^k$. $R_{k,k-1} \in$ SO(3) is the rotation matrix from $\{k-1\}$ to $\{k\}$, and $\mathbf{t}_{k,k-1}$ is the translation vector from the origin of frame $\{k\}$ to the origin of frame $\{k-1\}$ expressed in $\{k\}$. The transformation of the 3D point $\mathbf{X}^{k-1} = \left[X^{k-1}, Y^{k-1}, Z^{k-1}\right]^\top$ can be transformed with matrix multiplication as follows:

$$\begin{bmatrix} X^k \\ Y^k \\ Z^k \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X^{k-1} \\ Y^{k-1} \\ Z^{k-1} \\ 1 \end{bmatrix} \tag{2.12}$$

Multiple rigid body motions can be concatenated by left multiplying consecutive transformations. The identity transformation, meaning no motion, can be written as $R = I$ and $\mathbf{t} = \mathbf{0}$. The above defined notations and equations are illustrated in Figure 2.2.

We need a minimal representation of the rigid body transformation during the optimization,

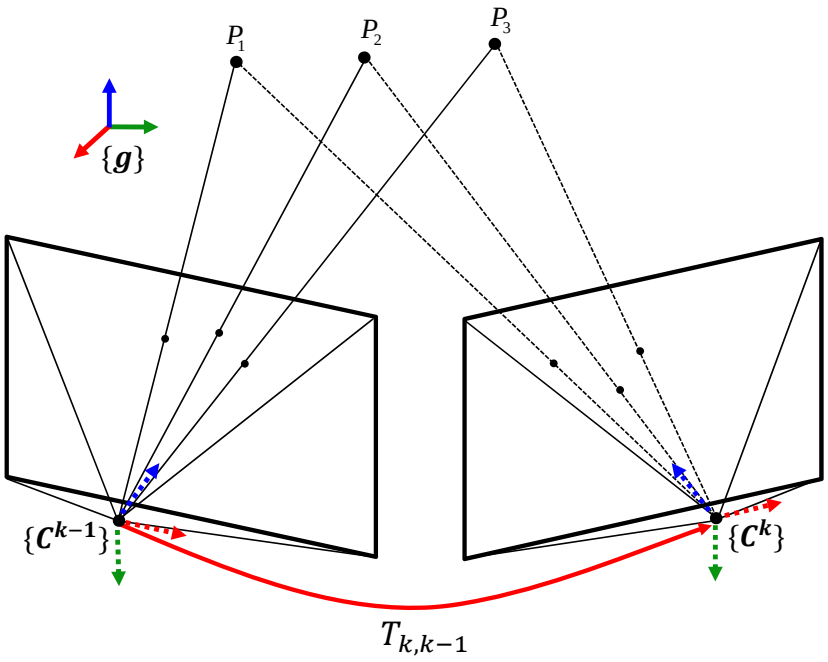


Figure 2.2: The relative transformation between two consecutive frames.

thus, use the Lie algebra $\mathrm{se}(3)$ parameters corresponding to the tangent space of $\mathrm{SE}(3)$ at the identity position. Such a minimal representation of Lie group $\mathrm{SE}(3)$ is beneficial when estimating the model parameters through a numerical optimization algorithm, and expressing the incremental displacements during a numerical optimization like Gauss-Newton method [38]. We represent the Lie algebra with a 6×1 vector $\xi = \left[\nu^\top, \omega^\top\right]^\top$ where $\nu = [\nu_1, \nu_2, \nu_3]^\top$ is the translational velocity and $\omega = [\omega_1, \omega_2, \omega_3]^\top$ is the rotational velocity. The rigid body motion can be computed from its Lie algebra ξ using the exponential map $\exp : \mathrm{se}(3) \mapsto \mathrm{SE}(3)$ as follows:

$$T(\xi) = \exp(\hat{\xi}) \tag{2.13}$$

where $\hat{\xi}$ is known as twist coordinates, and is the following 4×4 matrix as follows:

$$\hat{\xi} = \begin{bmatrix} [\omega]_\times & \nu \\ \mathbf{0}_{1\times 3} & 0 \end{bmatrix} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 & \nu_1 \\ \omega_3 & 0 & -\omega_1 & \nu_2 \\ -\omega_2 & \omega_1 & 0 & \nu_3 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.14}$$

The hat operator generates a 3×3 skew symmetric matrix from a 3×1 vector. The matrix exponential in Eq. (2.13) can be calculated easily using Taylor series expansion, and also has a closed form solution using Rodrigues' formula as follows:

$$\exp(\hat{\xi}) = \begin{bmatrix} \exp([\omega]_\times) & V\nu \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \tag{2.15}$$

$$\exp([\omega]_\times) = I + \frac{\sin(\|\omega\|)}{\|\omega\|}[\omega]_\times + \frac{1-\cos(\|\omega\|)}{\|\omega\|^2}[\omega]_\times^2 \tag{2.16}$$

$$V = I + \frac{1-\cos(\|\omega\|)}{\|\omega\|^2}[\omega]_\times + \frac{\|\omega\| - \sin(\|\omega\|)}{\|\omega\|^3}[\omega]_\times^2 \tag{2.17}$$

The inverse to the exponential map is the logarithm map $\log : \mathrm{SE}(3) \mapsto \mathrm{se}(3)$ as follows:

$$\xi = \log(T) \tag{2.18}$$

The identity transformation is $\xi = \mathbf{0}$.

2.3 Nonliear Least Squares Optimization

Many computer vision problems including VO and vSLAM can be formulated as solving the nonlinear optimization of an energy function E. All VO and vSLAM methods proposed in this thesis have at their core an energy function, which is minimized to find the best model parameters such as 6-DOF camera poses or 3D geometry parameters. They can be expressed in a nonlinear

least squares sense as follows:

$$E(\mathbf{x}) = \sum_i r_i^2(\mathbf{x}) \tag{2.19}$$

where $\mathbf{x}$ is the model parameters that we want to find like the 6-DoF camera motion, and the r_i is the nonlinear scalar function that can have arbitrary form. In most practical cases, the residual functions r_i are nonlinear and non-convex, making the overall energy function E non-convex as well. The optimality condition can be written as follows:

$$\sum_i r_i \frac{\partial r_i}{\partial \mathbf{x}_j} = 0 \tag{2.20}$$

Typically, we cannot directly solve these non-convex, nonlinear optimization problems. There exist iterative algorithms for computing approximate solutions, including Newton methods, the Gauss-Newton algorithm, and the Levenberg-Marquardt algorithm. We formulate the Newton methods for solving nonlinear least squares energy functions, as well as popular extensions such as Gauss-Newton, Levenberg-Marquardt algorithms.

2.3.1 Newton Optimization

In contrast to first-order methods like gradient descent algorithm, Newton methods are second-order methods, which make use of second derivatives. Geometrically, Newton method iteratively approximates the cost (objective or energy) function $E(\mathbf{x})$ quadratically and takes a step to the minimizer of this approximation. Let $\mathbf{x}_t$ be the estimated solution after t iterations. Then the Taylor series approximation of $E(\mathbf{x})$ in the vicinity of this estimate can be written as follows:

$$E(\mathbf{x}) \approx E(\mathbf{x}_t) + g^\top (\mathbf{x} - \mathbf{x}_t) + \frac{1}{2} (\mathbf{x} - \mathbf{x}_t)^\top H (\mathbf{x} - \mathbf{x}_t) \tag{2.21}$$

where g and H are the first and second derivatives called Jacobian as follows:

$$g = \frac{\mathrm{d}E}{\mathrm{d}\mathbf{x}(\mathbf{x}_t)} \tag{2.22}$$

$$H = \frac{\mathrm{d}^2 E}{\mathrm{d}\mathbf{x}^2(\mathbf{x}_t)} \tag{2.23}$$

For this second-order approximation of $E(\mathbf{x})$, the optimality condition can be rewritten as follows:

$$\frac{\mathrm{d}E}{\mathrm{d}\mathbf{x}} = g + H(\mathbf{x} - \mathbf{x}_t) = 0 \tag{2.24}$$

The above equation is called the normal equation. By setting the next iterate to the minimizer $\mathbf{x}$ leads to:

$$\mathbf{x}_{t+1} = \mathbf{x}_t - H^{-1} g \tag{2.25}$$

When applicable, second-order methods are often faster than first-order methods, at least when measured in the number of iterations. However, for massive optimization problems, computing and inverting the Hessian matrix H itself may be challenging. When H is not positive definite, there exist quasi-Newton methods which aim at approximating H or H^{-1} with a positive definite matrix.

2.3.2 Gauss-Newton Optimization

The Gauss-Newton algorithm is an approximation version of Newton methods to solve nonlinear least squares problems when the nonlinearity is small. It can be derived as an approximation to the Newton method. The first derivative g and the second derivative H can be written as follows:

$$g_j = 2 \sum_i r_i \frac{\partial r_i}{\partial \mathbf{x}_j} \tag{2.26}$$

$$H_{jk} = 2 \sum_i \left(\frac{\partial r_i}{\partial \mathbf{x}_j} \frac{\partial r_i}{\partial \mathbf{x}_k} + r_i \frac{\partial^2 r_i}{\partial \mathbf{x}_j \partial \mathbf{x}_k} \right) \tag{2.27}$$

Dropping the second order term leads to the approximation of Hessian matrix H:

$$H_{jk} \approx 2 \sum_i J_{ij} J_{ik} \qquad \text{with} \quad J_{ij} = \frac{\partial r_i}{\partial \mathbf{x}_j} \tag{2.28}$$

$$H \approx 2J^\top J \qquad g = 2J^\top r \tag{2.29}$$

The approximation together with g and H leads to the Gauss-Newton algorithm:

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \Delta \qquad \text{with} \quad \Delta = -\left(J^\top J\right)^{-1} J^\top r \tag{2.30}$$

In contrast to the Newton algorithm, the Gauss-Newton algorithm does not require the computation of second derivatives. Moreover, the above approximation of the Hessian matrix is by construction positive definite. Note that this approximation of the Hessian is only valid if:

$$\left| r_i \frac{\partial^2 r_i}{\partial \mathbf{x}_j \partial \mathbf{x}_k} \right| \ll \left| \frac{\partial r_i}{\partial \mathbf{x}_j} \frac{\partial r_i}{\partial \mathbf{x}_k} \right| \tag{2.31}$$

This is the case when the residual r_i is small or it is close to linear. In other words, Gauss-Newton method can be applied effectively when the second derivatives are small.

2.3.3 Levenberg-Marquardt Optimization

The Newton methods

$$\mathbf{x}_{t+1} = \mathbf{x}_t - H^{-1} g \tag{2.32}$$

can be modified as damped version:

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \left(H + \lambda I_n\right)^{-1} g \tag{2.33}$$

to create a hybrid between the Newton method ($\lambda = 0$) and a gradient descent with step size $1/\lambda$ ($\lambda \to \infty$). In the similar fashion, Levenberg suggested to damp the Gauss-Newton algorithm for nonlinear least squares:

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \Delta \qquad \text{with} \quad \Delta = -\left(J^\top J + \lambda I_n\right)^{-1} J^\top r \tag{2.34}$$

Marquardt suggested a more adaptive component-wise damping of the form:

$$\Delta = -\left(J^\top J + \lambda diag\left(J^\top J\right)\right)^{-1} J^\top r \tag{2.35}$$

which avoids slow convergence in directions of small gradient.

3

Robust Visual Odometry to Irregular Illumination Changes with RGB-D Camera

Authors	Pyojin Kim[1]	rlavywls@snu.ac.kr
	Hyon Lim[1]	hyonlim@snu.ac.kr
	H. Jin Kim[1]	hjinkim@snu.ac.kr
	[1] Seoul National University	

Publication	**Robust Visual Odometry to Irregular Illumination Changes with RGB-D Camera.** Kim, Pyojin, Hyon Lim, H. Jin Kim. In *Proceedings of IEEE International Conference on Intelligent Robots and Systems (IROS), 2015*. Copyright 2015 IEEE.

Contribution	Problem definition	*significantly contributed*
	Literature survey	*significantly contributed*
	Method development	*significantly contributed*
	Implementation	*significantly contributed*
	Experimental evaluation	*significantly contributed*
	Preparation of the manuscript	*significantly contributed*

Abstract Sensitivity to illumination conditions poses a challenge when utilizing visual odometry (VO) in various applications. To make VO robust with respect to illumination conditions, they need to be considered explicitly. In this paper, we propose a direct visual odometry method which can handle illumination changes by considering an affine illumination model to compensate abrupt, local light variations during direct motion estimation process. The core of our proposed method is to estimate the relative camera pose and the parameters of the illumination changes by minimizing the sum of squared photometric error with efficient second-order minimization. We evaluate the performance of the proposed algorithm on synthetic and real RGB-D datasets with ground-truth. Our result implies that the proposed method successfully estimates 6-DoF pose under significant illumination changes whereas existing direct visual odometry methods either fail or lose accuracy.

3.1 Introduction

Estimating egomotion of a robot with video sequences coming from the camera attached to it is called visual odometry (VO) [39]. Existing VO techniques can be broadly divided into two types, depending on pose estimation method: feature based methods [40] and direct methods [6]. Many studies have adopted feature-based methods with monocular [16, 41], stereo [40], [9], and RGB-D camera [42], [43]. However, direct methods are getting more interests recently [6], [23], [21]. In these direct methods, the core idea is to minimize the sum of squared photometric error between two images under the photo-consistency assumption [44]. The fundamental assumption of existing direct VO methods is that brightness constraint is valid only under sufficient and constant illumination in the environment [45], which is an impractical assumption in most real-world applications as illustrated in Fig. 3.1. Thus, it is difficult to directly apply the existing direct VO methods when the illumination change is not negligible.

To make robust VO algorithm to illumination changes, we propose a direct VO method which works well under sudden or local illumination changes during the direct motion estimation process by considering individual illumination changes in selected patches in an image. An affine

illumination change model [46] is applied to individual patches which are selected based on planarity test with RANSAC using depth map of patches. To the best of our knowledge, this is the first direct VO which takes into account irregular, local illumination changes in the patches that are selected based on the planarity condition with RANSAC to make patches have the same normal vector to satisfy the affine illumination model [46].

The proposed method is evaluated with synthetic RGB-D dataset by modifying the TUM RGB-D benchmark dataset [2] and carefully chosen sequences that have illumination changes in [2]. We evaluate the performance of the proposed algorithm compared to the other direct VO methods [6, 7].

This paper is organized as follows. Related works are discussed in Section 3.2. In Section 3.3, notation used throughout this paper and the problem we want to solve are stated in detail. Section 3.4 gives an overview of the proposed visual odometry algorithm and main pipeline of the proposed algorithm. After validation and evaluation results are presented in Section 3.5, conclusion is remarked in Section 3.6.

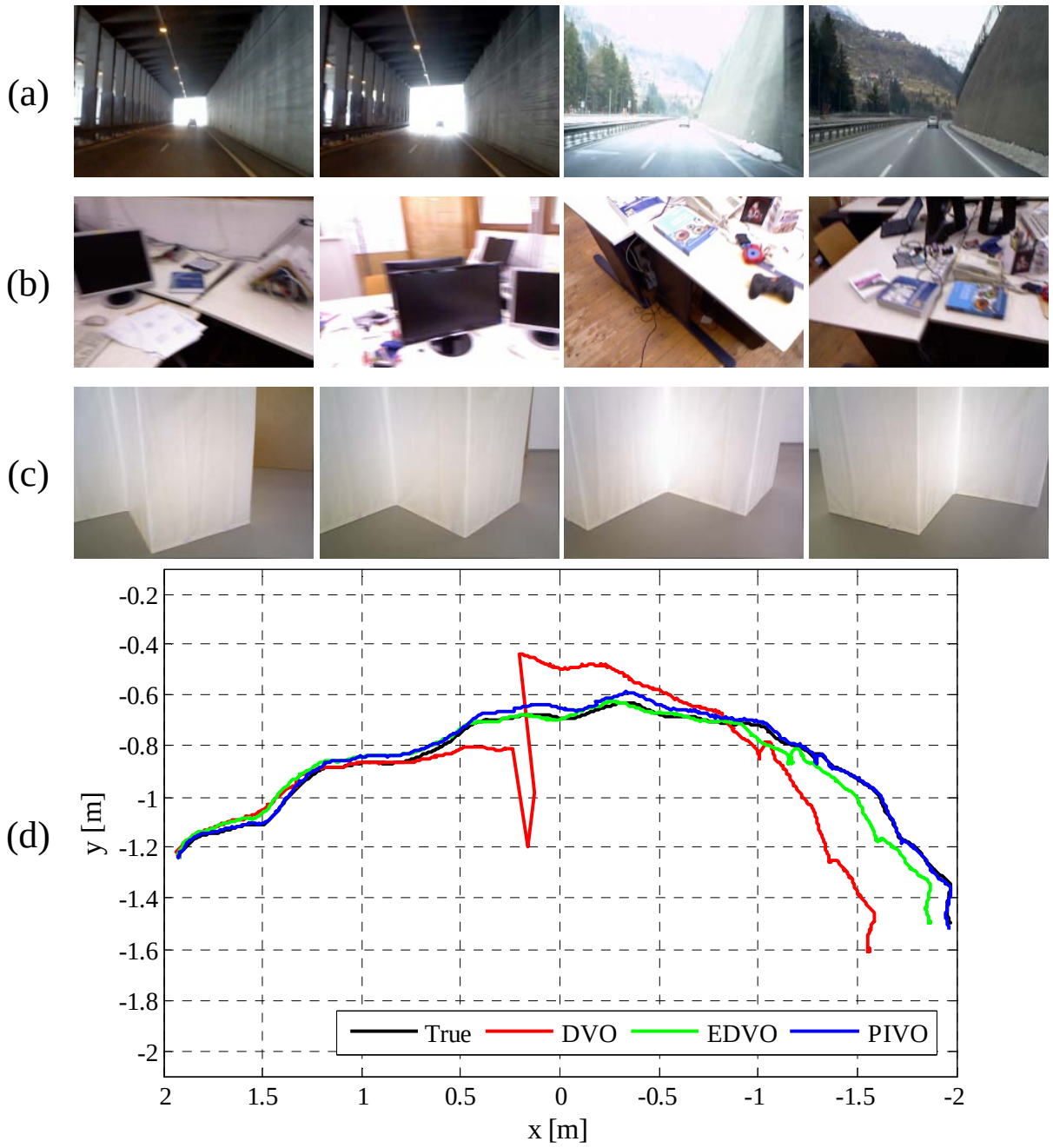


Figure 3.1: Illustrative examples of irregular illumination changes. (a) An automatic exposure control of camera makes the intensity of images change when the camera moves between different global illumination conditions. (b+c) Illumination variation in the TUM dataset 'fr1/room' and 'fr3/struc&notex'. (d) Comparison of the estimation results of DVO [6], EDVO [7], and the algorithm proposed in this paper (namely PIVO) with 'fr3/struc&notex'. A large drift error takes place under DVO when the illumination changes occur as illustrated in (c).

3.2 Related Work

For autonomous navigation of a robotic system, VO has been actively utilized both on the ground [40, 47] and in the air [42]. As discussed in Section 3.1, the VO methods can be categorized as two in terms of a kind of information used in pose estimation process: so called feature-based methods [40] and direct methods [6].

The feature-based methods encode an image to a list of keypoints (i.e. a list of image coordinates of distinctive points) and solve the geometric pose estimation problem on that list of coordinates and association table. Many keypoint extraction and matching algorithms are applied to those feature-based VO, however, they require enough brightness and textures to extract consistent keypoints from an image [48]. In varying illumination conditions that we consider in this paper, this requirement degrades the performance of feature-based VO. As a result, the feature-based methods cannot correctly estimate their own position in featureless or dark environments.

Therefore, direct methods which exploit the entire image information are receiving attention recently with the help of hardware progress. In [49] and [50], the camera pose tracking is performed based on alignment of 3D point clouds (ICP), which presents successful results in terms of robustness, computation time, and accuracy. The direct VO techniques ([6], [24], [51]) are proposed, which minimize the photometric error between image frames. They are fundamentally based on the photo-consistency assumption, which means that a point in the 3D world represents the same brightness intensity at different camera poses [45]. In [24], a semi-direct method was successfully implemented on an aerial vehicle with a single downward-looking camera. [6] estimates the relative RGB-D camera motion accurately with a robust error function which rejects the noise and outliers in the image. In [51], quadrifocal geometry constraints are used to track the trajectory of a stereo camera. Even though the outlier rejection algorithms exist in the above methods such as a robust error function, a large drift of the estimated trajectory caused by the abrupt illumination changes is inevitable since the photo-consistency assumption is no longer valid.

Only a few direct VO methods give consideration to illumination changes during the direct

motion estimation. It is assumed in [7] that the entire pixels in the image follow the same illumination change model [46]. A similar light variation model is also used in [52] and [53], which need the reconstructed 3D scene model for camera pose tracking and use a single global brightness (bias shift) parameter in the image. In order to ignore the illumination changes altogether between image frames, [54] estimate a pure albedo image of the texture. In contrast with the works mentioned above, the proposed direct VO method in this paper takes into account both global and local illumination changes. In particular, general illumination changes can be handled because each patch is allowed to have different model parameters of illumination changes. The proposed method is evaluated and compared with [6] and [7].

3.3 Notation and Problem Statement

We define and organize the notations used throughout this paper shortly, before the proposed algorithm is explained in detail. We assume that the camera model used in RGB-D sensor follows a pinhole camera model [33].

The superscript k is used to denote the index of an image frame. An intensity image obtained at time step k is denoted with I^k. In the intensity image I^k, i-th image patch is denoted with I_i^k. For an arbitrary 2D pixel point, pixel coordinates in I_i^k are denoted as $\mathbf{x}_{ij}^k = \left[x_{ij}^k, y_{ij}^k\right]^\top$, where the first subscript i is the patch index, and j is the pixel index. 3D points $\mathbf{X}_{ij}^k = \left[X_{ij}^k, Y_{ij}^k, Z_{ij}^k\right]^\top$ defined in camera coordinate $\{C^k\}$ are mapped to the pixel coordinates $\mathbf{x}_{ij}^k$ through the camera projection function $\pi : \mathbb{R}^3 \mapsto \mathbb{R}^2$

$$\mathbf{x}_{ij}^k = \pi\left(\mathbf{X}_{ij}^k\right) = \begin{bmatrix} \frac{X_{ij}^k \cdot f}{Z_{ij}^k} + p_x \\ \frac{Y_{ij}^k \cdot f}{Z_{ij}^k} + p_y \end{bmatrix} \tag{3.1}$$

The above projection function is determined uniquely with the camera intrinsic parameters f, p_x, p_y [33] (pg. 155).

Conversely, a 3D point $\mathbf{X}_{ij}^k$ can be computed with the depth value Z_{ij}^k (from depth map of

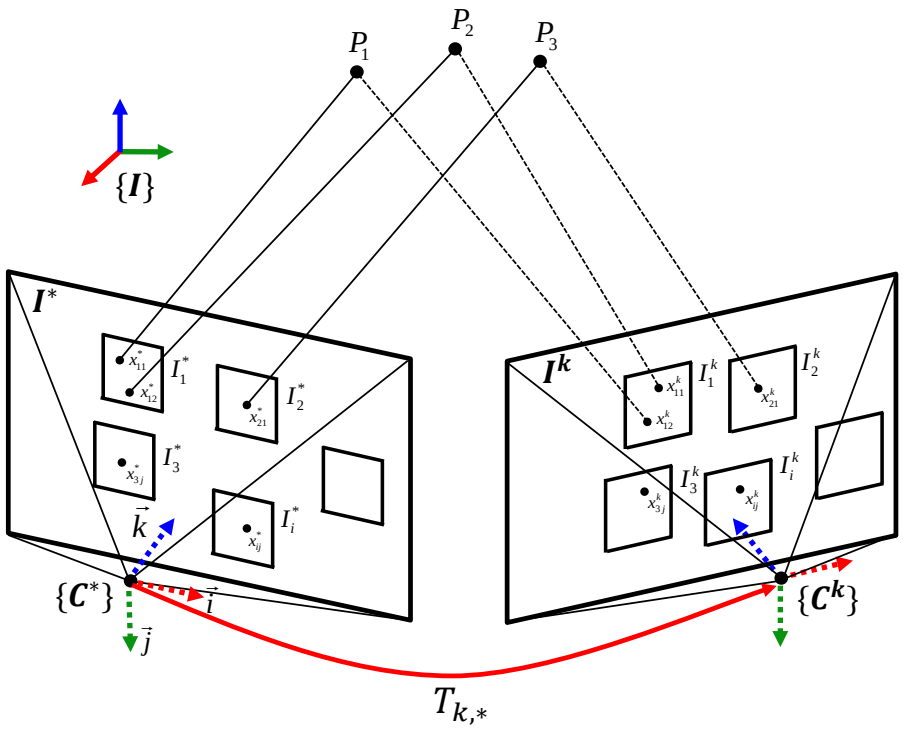


Figure 3.2: The notations and setting of the proposed visual odometry algorithm.

RGB-D sensor) and $\mathbf{x}_{ij}^k$ through the inverse projection function $\pi^{-1} : \mathbb{R}^2 \mapsto \mathbb{R}^3$

$$\mathbf{X}_{ij}^k = \pi^{-1}\left(\mathbf{x}_{ij}^k, Z_{ij}^k\right) = \begin{bmatrix} \frac{x_{ij}^k - p_x}{f} Z_{ij}^k \\ \frac{y_{ij}^k - p_y}{f} Z_{ij}^k \\ Z_{ij}^k \end{bmatrix} \tag{3.2}$$

The relative position and orientation between the current camera frame $\left\{C^k\right\}$ and the keyframe $\{C^*\}$ are represented with the rigid body transformation matrix $T_{k,*} \in SE(3)$:

$$\tilde{\mathbf{X}}^k = T_{k,*}\tilde{\mathbf{X}}^* \tag{3.3}$$

where $\tilde{\mathbf{X}}^k = [\mathbf{X}^k, 1]^\top$ is the homogeneous form of $\mathbf{X}^k$. In this paper, a minimal representation of Lie group $SE(3)$, i.e. Lie algebra $se(3)$ parameter ξ, is mainly used to express the incremental displacements during a numerical optimization algorithm. We can represent the Lie algebra

parameter with a 6×1 vector $\xi = \left[\nu^\top, \omega^\top\right]^\top$ where ν is the translational velocity and ω is the rotational velocity. The rigid body transformation matrix $T \in SE(3)$ can be calculated by the exponential map:

$$T(\xi) = \exp(\hat{\xi}) \tag{3.4}$$

where $\hat{\xi}$ is a 4×4 twist matrix from the Lie algebra parameter ξ [55]. The above defined notations and equations are illustrated in Fig. 3.2.

With the above notations, the problem we want to solve is to estimate the rigid body transformation matrix $T_{k,*}$ given a sequence of image frames and the corresponding depth maps from RGB-D sensor under arbitrary, abrupt, and partial illumination changes between consecutive image frames.

3.4 Proposed Visual Odometry Algorithm

The schematic overview of the proposed visual odometry algorithm and the data flow are represented in Fig. 3.3. First, RGB and depth images from RGB-D camera are used to initialize a keyframe with patches. The pixel points that are the center of the patches in RGB image are detected by the blob detector like LoG, DoG, or SURF [56]. Among them, the only valid patches are saved and utilized in motion estimation process until a next keyframe is re-initialized. After the keyframe is initialized, a residual image is obtained by using the keyframe with patches, current image frame, and the model parameters ξ. Each weight of the residual value is determined by t-distribution of all residuals.

Next, Jacobian matrix is calculated with the gradient images of the keyframe and the current image frame to minimize the newly proposed photometric error. The proposed visual odometry algorithm is based not on the photo-consistency assumption like [6], [24], and [57], but on the photo-consistency assumption with compensation of illumination changes between the two consecutive images. By combining the Jacobian matrix J, the residual vector r, and the diagonal weighting matrix W, the incremental displacements of the model parameter $\triangle z$ is calculated. The model parameter z is updated and the above procedure is repeated until convergence. If the Eu-

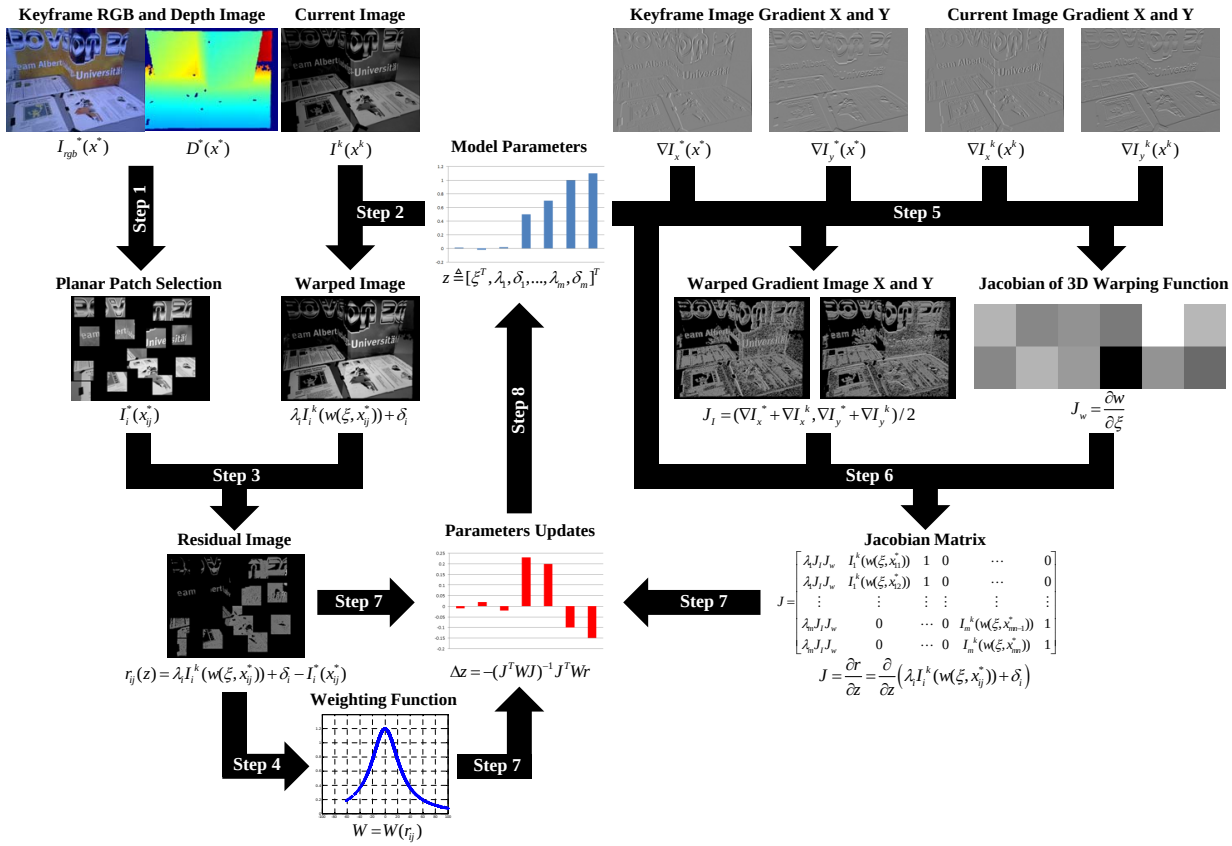


Figure 3.3: Overview of the proposed visual odometry algorithm.

clidian distance between the keyframe and the current image frame is too far, the next keyframe is newly initialized with the current RGB and depth image. Finally, the whole trajectory of RGB-D camera is obtained by concatenating the frame-to-frame motion estimation results.

In Section 3.4.1, we talk about the illumination change model which is utilized to define the proposed photometric cost function. To reduce the computational workload in the direct motion estimation process and take into account the irregular illumination changes in two consecutive images, the patch-based keyframe image generation technique is employed. Thus, Section 3.4.2 describes how the keyframe images are made with the useful patches extracted in the keyframe RGB image. The proposed photometric cost function with compensation of illumination changes is introduced in Section 3.4.3, and additional extensions to improve the robustness of the direct motion estimation method as well as the optimization process are also explained.

3.4.1 Illumination Change Model

The photo-consistency assumption employed in [6] and [24] is not always valid in real world because illumination changes such as highlights, shadows caused by the variation of the view-point of the camera, unpredictable changes of light source, and the camera automatic gain are unavoidable phenomena during the direct visual odometry. To reflect not only the global but also the local illumination changes between the keyframe and the current frame, we adopt the affine illumination change model [46] per patch as follows:

$$\lambda_i I_i^k + \delta_i = I_i^* \tag{3.5}$$

Here λ_i and δ_i are the model parameters to represent contrast and brightness changes of the i-th patch in the image. During the optimization process, these parameters per patch are estimated and utilized to compensate the irregular illumination changes between the keyframe and the current frame.

3.4.2 Planar Patch Selection

The patch-based keyframe image generation method is employed to solve two critical issues in direct visual odometry process: reducing computation time and taking into account both global and local illumination changes.

The computation time is proportional to the number of pixels in the direct visual odometry since every pixel should go through the 3D backward or forward image warping. For example, in [6], [7], the entire pixels in the image are used. We observe, however, that the number of pixels fewer than 50% of the entire pixels in the image is still enough to estimate the motion of the camera [24]. Thus, the patch-based keyframe image as depicted in Fig. 3.4(a) can be used to carry out the direct visual odometry process.

For the issue of illumination changes, although the global changes have been considered in [7] and [53], the partial changes have not been concerned yet. On the other hand, to take

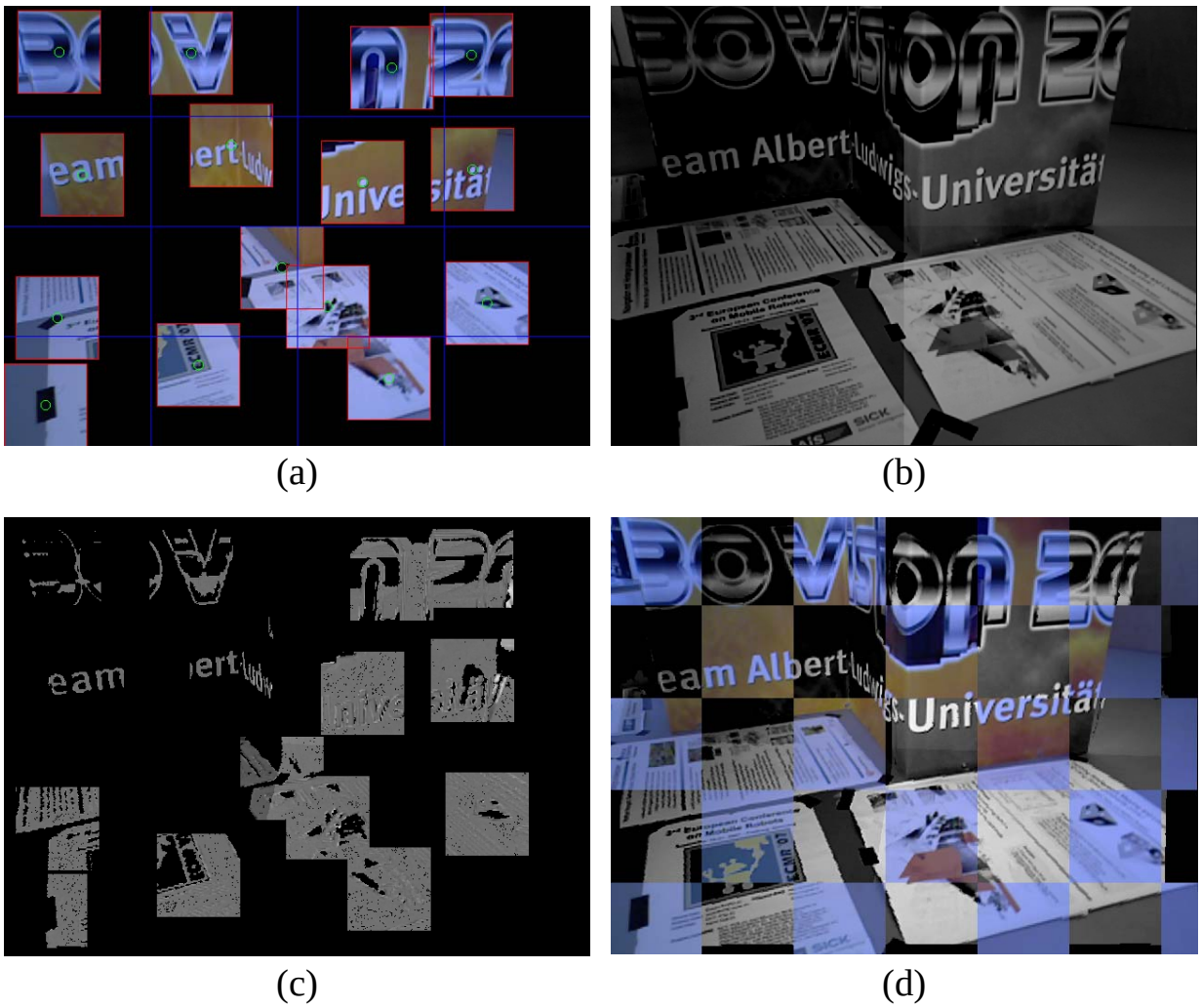


Figure 3.4: Input and output images of the proposed visual odometry algorithm. (c) shows residual image between the patch based keyframe and the current image frame.

into account the local illumination changes, we assume that each patch follows the different illumination changes individually as illustrated in Fig. 3.4(a), where each patch has its own unique λ_i and δ_i. And to make the above assumption valid, the patches only on the planar surface in the real world are selected because 3D points on the same plane undergo the similar illumination changes [46]. It can be achieved by utilizing the blob detector like LoG, DoG, SURF [56] and the plane model based RANSAC algorithm [58]. At first, the blob detector is used rather than the corner detector to extract the patches on the planar surface in the 3D space. After that, RANSAC

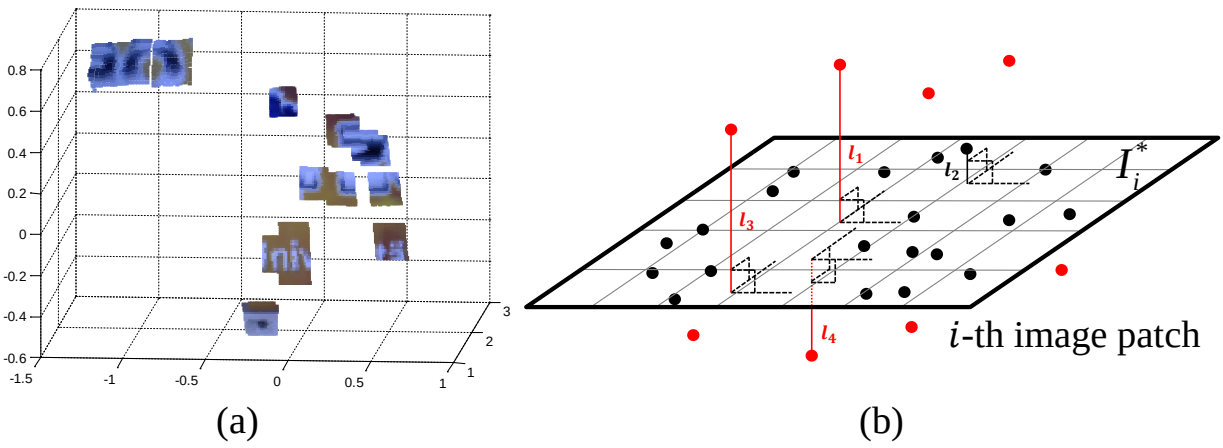


Figure 3.5: Planar patch selection results based on the plane RANSAC algorithm.

algorithm is applied to robustly fit the plane to a set of 3D data points from the extracted patch's pixel points and the depth image from RGB-D sensor. The plane of each patch is fitted with the following equation:

$$ax + by + cz + d = 0 \tag{3.6}$$

where a, b, c, and d are the model parameters of the plane and x, y, z is the 3D point on the plane. Based on Eq. (3.6), the error function of the plane RANSAC algorithm can be formulated as follows:

$$l_j = \frac{\left|aX_{ij}^* + bY_{ij}^* + cZ_{ij}^* + d\right|}{\sqrt{a^2 + b^2 + c^2}} \tag{3.7}$$

where l_j is the length between the plane (a, b, c, d) and the 3D point $(X_{ij}^*, Y_{ij}^*, Z_{ij}^*)$ expressed in the camera keyframe. Using RANSAC with the error function Eq. (3.7), we can determine how many points are out of the plane in each patch. Fig. 3.5(b) depicts the outlier points as red and inlier points as black. If more than half of the points of a patch are out of the plane, these kinds of patches are rejected and discarded. In this manner, with the blob detector and the plane RANSAC algorithm, only the valid patches that are on the plane in 3D space as described in Fig. 3.5(a) survive and pass through the direct motion estimation process.

3.4.3 Direct Motion Estimation

We generated the patch-based keyframe gray and depth image I^*, D^* in the previous step, and the current image frame I^k comes from the RGB-D camera. With I^*, D^*, and I^k, our goal is to estimate the relative camera pose $T_{k,*}$ and the illumination change model parameters per patch, i.e., $\lambda_1, \delta_1, ..., \lambda_m, \delta_m$ where m is the number of patches in the keyframe image I^*. In contrast with the existing photo-consistency assumption [6], our photo-consistency assumption that considers the illumination changes can be written as the following equation:

$$\lambda_i I_i^k(w(\xi, \mathbf{x}_{ij}^*)) + \delta_i = I_i^*(\mathbf{x}_{ij}^*) \tag{3.8}$$

$$w(\xi, \mathbf{x}_{ij}^*) = \pi(T(\xi) \cdot \pi^{-1}(\mathbf{x}_{ij}^*, Z_{ij}^*))) \tag{3.9}$$

where $\xi \in \mathbb{R}^6$ represents the relative motion of the camera and $w(\xi, \mathbf{x}_{ij}^*)$ is the 3D backward warping function which is a one-to-one mapping from a pixel point $\mathbf{x}_{ij}^*$ in the patch-based keyframe image to a pixel coordinate in the current image frame given the relative camera motion ξ. To simplify expression of the overall model parameters which we have to estimate, the integrated new model parameter $\mathbf{z}$ is defined as follows:

$$\mathbf{z} := \left[\xi^\top, \lambda_1, \delta_1, ..., \lambda_m, \delta_m\right]^\top \in \mathbb{R}^{6+2m} \tag{3.10}$$

Based on the defined notations and the modified photo-consistency assumption as written in Eq. (3.8), we define the residual of the j-th pixel in the i-th patch as the photometric difference with compensation of the illumination changes between pixels observed in the keyframe and the current frame:

$$r_{ij}(\mathbf{z}) = \lambda_i I_i^k(w(\xi, \mathbf{x}_{ij}^*)) + \delta_i - I_i^*(\mathbf{x}_{ij}^*) \tag{3.11}$$

We seek the optimal model parameter $\mathbf{z}^*$ that minimizes the weighted sum of squared residuals, which is the following non-linear weighted least square problem:

$$\mathbf{z}^* = \arg\min_{\mathbf{z}} \sum_{i=1}^{m} \sum_{j=1}^{n} W(r_{ij}) r_{ij}^2(\mathbf{z}) \tag{3.12}$$

where n is the number of pixels in each patch and $W(r_{ij})$ is the weighting function that gives the different weights to each residual value based on the residual distribution. We assume that the residual distribution follows the t-distribution by following [6]. We solve the iteratively re-weighted least square (IRLS) problem with weighting function determined by the t-distribution.

To find the optimal model parameter $\mathbf{z}^*$ written in Eq. (3.12), Gauss-Newton algorithm is selected. And there are several image alignment strategies in the direct methods: forward compositional (FC), inverse compositional (IC), and efficient second-order minimization (ESM). Among them, it is well known that ESM method outperforms the other methods [21], [7]. Thus, the Jacobian matrix is calculated with respect to the newly defined model parameter $\mathbf{z}$ based on the ESM algorithm [59]. By plugging and arranging the equations (3.8)$-$(3.12), the normal equation is obtained:

$$J^T W J \triangle \mathbf{z} = -J^T W r \tag{3.13}$$

$$J \in \mathbb{R}^{(mn)\times(6+2m)}, W \in \mathbb{R}^{(mn)\times(mn)}, r \in \mathbb{R}^{(mn)}$$

$$J = \begin{bmatrix} \lambda_1 J_I J_w & I_1^k(w(\xi, \mathbf{x}_{11}^*)) & 1 & 0 & \dots & 0 \\ \lambda_1 J_I J_w & I_1^k(w(\xi, \mathbf{x}_{12}^*)) & 1 & 0 & \dots & 0 \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ \lambda_m J_I J_w & 0 & \dots & 0 & I_m^k(w(\xi, \mathbf{x}_{mn-1}^*)) & 1 \\ \lambda_m J_I J_w & 0 & \dots & 0 & I_m^k(w(\xi, \mathbf{x}_{mn}^*)) & 1 \end{bmatrix} \tag{3.14}$$

$$J_I J_w = \frac{1}{2}\left(\frac{\partial I^*(\mathbf{x}^*)}{\partial \mathbf{x}} + \frac{\partial I^k(w(\xi, \mathbf{x}^*))}{\partial \mathbf{x}}\right)\frac{\partial w(\xi, \mathbf{x}^*)}{\partial \xi} \tag{3.15}$$

In Eq. (3.13), note that J is the stacked Jacobian matrix and W is the diagonal matrix that represents each residual's weight and r is the tall residual vector coming from Eq. (3.11). At every iteration in IRLS, we can compute the incremental value $\triangle\mathbf{z}$ and based on that incremental values, we update the relative camera pose $T_{k,*}$ and the illumination change model parameters λ_i, δ_i with $i = 1, ..., m$ until the integrated model parameter $\mathbf{z}$ converges. Additionally, similar to [6] and [7], a coarse-to-fine approach is employed with the image pyramid method for robustness and faster convergence. The gaussian pyramid is utilized to compute the image pyramid and run the above optimization process from the coarsest level to the finest. In this manner, we can compute the relative camera motion and the illumination change model parameters much faster and accurately.

3.5 Evaluation

The proposed *Patch-based Illumination invariant Visual Odometry* (PIVO) algorithm is tested with two types of datasets: synthetic RGB-D dataset which is based on TUM RGB-D benchmark [2] and author-collected RGB-D dataset. In RGB images of the synthetic RGB-D dataset, artificial illumination changes are applied to validate the proposed visual odometry method. Author-collected RGB-D dataset consists of the RGB and depth images captured under an actual illumination change with static pose. Three performance metrics are used to evaluate the performance of the proposed visual odometry algorithm: root mean square error (RMSE) of the relative pose error (RPE), and the absolute trajectory error (ATE) defined in [2] and the final drift error divided by the length of the entire trajectory. We compare the motion estimation results with author-implemented version of [6] and [7]. All calculations and processes are conducted on a desktop computer with Intel Core i5 with 3.2 Ghz with 8GB memory and the program is implemented in MATLAB. PIVO takes about 200-300 ms per frame in our current setting.

Figure 3.6: Example images from the synthetic RGB-D dataset.

3.5.1 Synthetic RGB-D Dataset

The TUM RGB-D dataset consists of calibrated RGB and depth images taken at full frame rate (30 Hz) and ground truth position, ground-truth pose of the RGB-D camera obtained from a motion capture system (100 Hz). But they do not involve abrupt, irregular illumination changes. Thus, we modify intensity values based on the illumination change model [46] to simulate the irregular illumination changes in TUM RGB-D dataset. A gray image, at first, is divided into four regions and the intensity values in each of the four region are modified with four different illumination change models to give partial lighting changes. We call the modified image sequences as the synthetic RGB-D dataset and some of the syntectic gray images are drawn in Fig. 3.6. The four sets of illumination change model parameters used to generate the synthetic RGB-D dataset and the corresponding parameter values estimated by the proposed visual odometry algorithm are compared at the end of this section.

We evaluate the motion estimation accuracy of PIVO in an environment where the irregular illumination changes occur compared to the Dense Visual Odometry (DVO) [6], and Efficient DVO (EDVO) [7]. The evaluations are performed with the eleven synthetic RGB-D image sequences and the motion estimation results for each visual odometry method are summarized in TABLE 3.1. In all cases, we observe that our method generates better results than DVO, EDVO.

Table 3.1: Estimation results with synthetic RGB-D dataset.

Name of Dataset	RPE [drift m/s]			Drift Error [%]		
	DVO	EDVO	PIVO	DVO	EDVO	PIVO
fr1/desk	0.257	0.076	0.057	3.23	1.65	1.15
fr1/desk2	0.616	0.252	0.183	15.34	6.18	3.42
fr1/floor	1.214	0.224	0.203	7.18	9.05	5.67
fr1/room	3.648	1.119	0.262	59.39	14.18	3.25
fr2/desk	0.288	0.232	0.231	13.88	1.19	0.79
fr2/largenoloop	5.300	3.624	2.268	74.24	48.06	12.32
fr3/longoffice	0.592	0.045	0.040	9.12	1.57	1.39
fr3/nostruc&notex	5.757	14.014	0.067	261.88	1325.86	15.62
fr3/nostruc&tex	1.615	0.228	0.107	106.63	13.16	8.49
fr3/struc&notex	10.062	1.483	0.021	372.98	312.43	5.03
fr3/struc&tex	0.105	0.034	0.023	31.56	4.54	2.34

In most cases except ‘fr1/desk’, DVO has failed to estimate pose with illumination changes. EDVO presents good performances on some datasets: ‘fr2/desk’, ‘fr1/desk’, and ‘fr3/longoffice’ due to the compensation factor of the global illumination changes. However, it shows poor results on ‘fr3/nostruc&notex’, ‘fr3/struc&notex’. Fig. 3.7 shows the 3D estimated trajectories of each visual odometry method with four different image sequences. Absolute trajectory error (ATE) of each method is also presented in Fig. 3.8 with the same datasets used in Fig. 3.7. Estimation error of the other two visual odometry methods except PIVO increases rapidly during the interval marked by the gray dotted lines where the illumination changes occur in the image sequences.

In particular, the dataset ‘fr3/struc&notex’ is selected to analyze the result in detail. During the period from 100 to 300 image frames where the illumination changes occur, we can find that PIVO estimates the position, pose of the camera accurately whereas the drift of DVO, EDVO

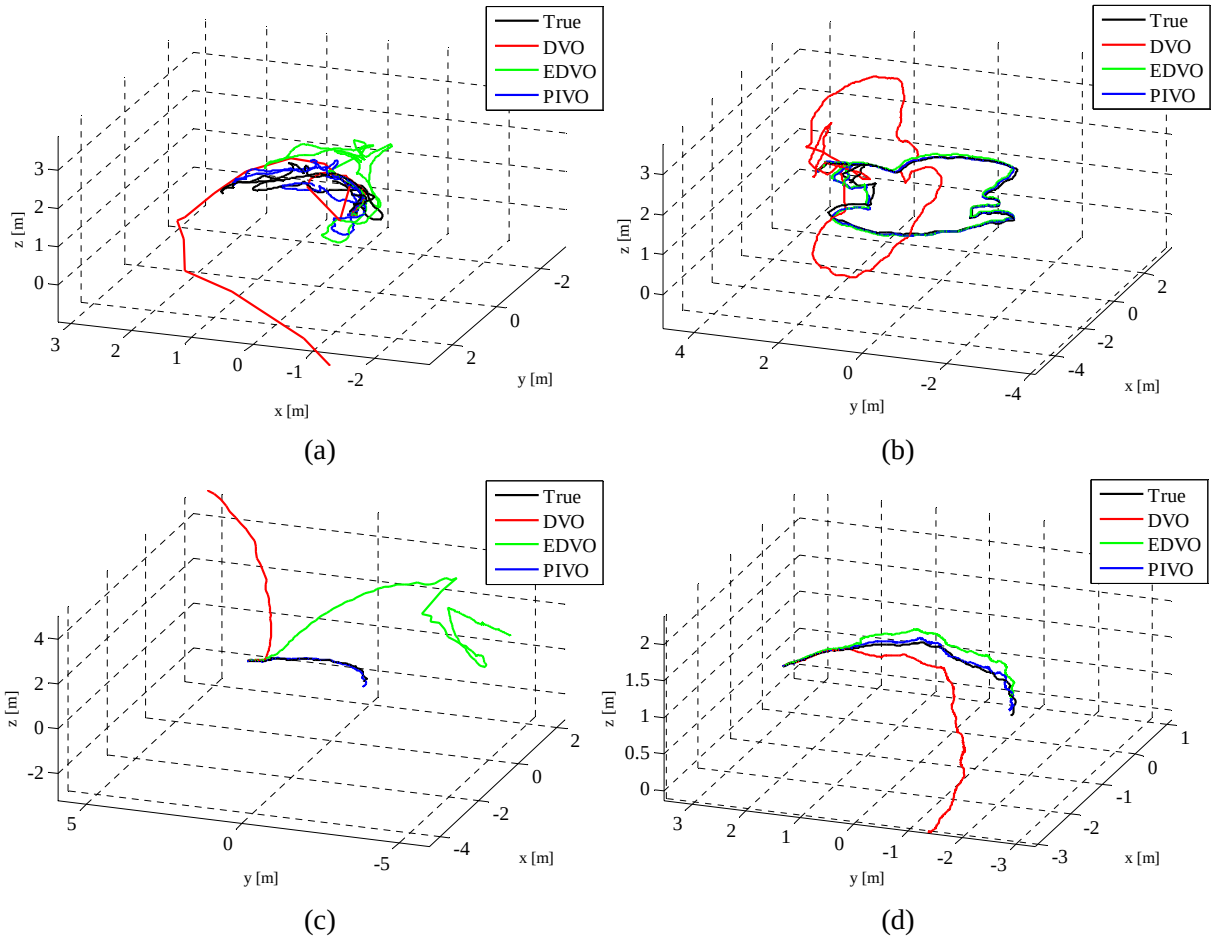


Figure 3.7: Comparison of VO results with the ground truth.

gradually increases as described in Fig. 3.8(c). The main reason for this difference is that the photo-consistency assumption is violated in this period. Although a robust weighting function is used for discarding outliers in DVO or a global affine illumination change concept is considered in EDVO, the cost functions in DVO, EDVO are not effective enough to take into account the sudden, partial lighting variations. PIVO efficiently copes with this kind of illumination changes by using the proposed cost function in Eq. (3.12). This can be confirmed in Fig. 3.9. DVO and EDVO assign high weight to large residual as the photo-consistency assumption breaks down in the cases of Figs. 3.9(b) and (c), which degrades the accuracy. On the other hand, under PIVO, the large weight remains only over small residual during the light variations, which means that the illumination changes are compensated properly. Fig. 3.10(a) also supports this. The percentage

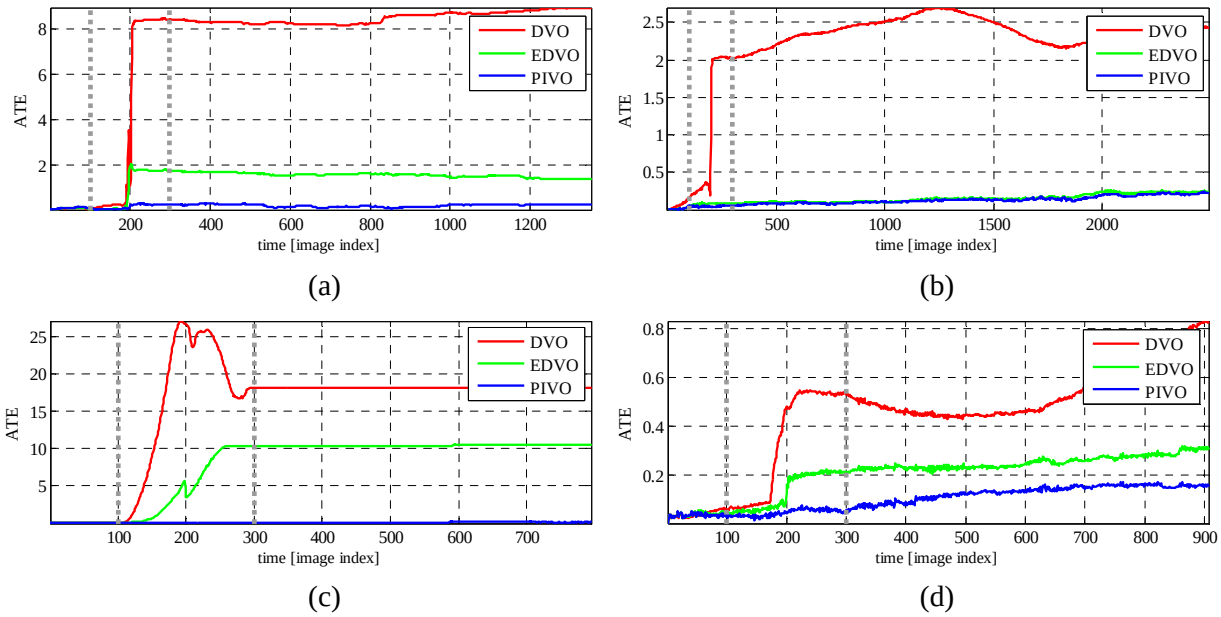


Figure 3.8: Absolute trajectory errors of each tested dataset.

of effective residuals which are close to 0 in PIVO is much higher than those in DVO and EDVO until the end of the sequences, and their difference becomes significant during irregular illumination changes. Furthermore, Fig. 3.10(b) shows the cost functions for the three methods over time. The cost values of DVO and EDVO increase noticeably, whereas PIVO value is maintained very small.

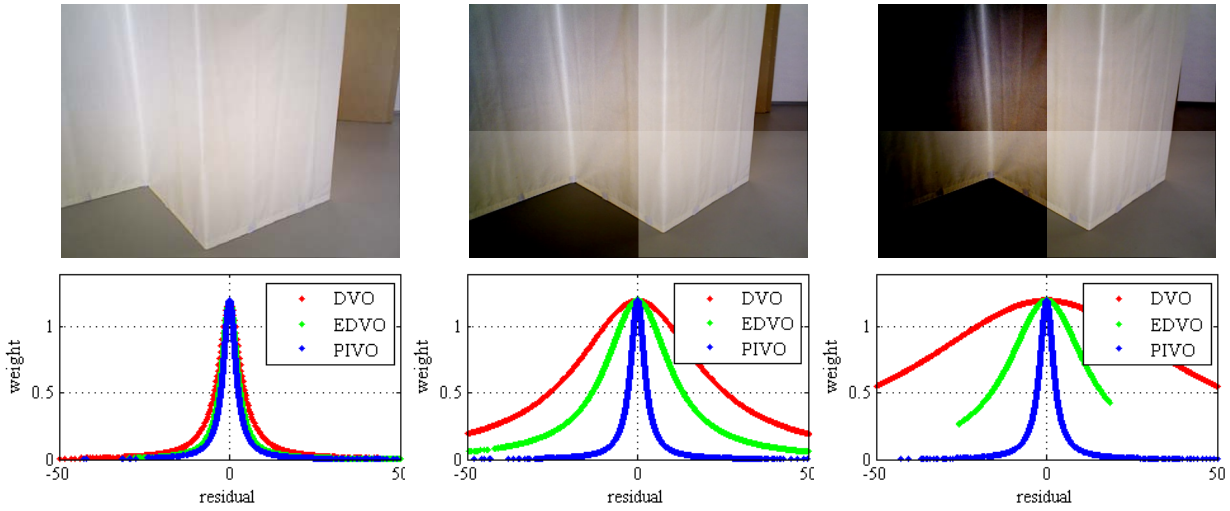


Figure 3.9: Weighting functions with respect to the residuals of each method.

(a)

(b)

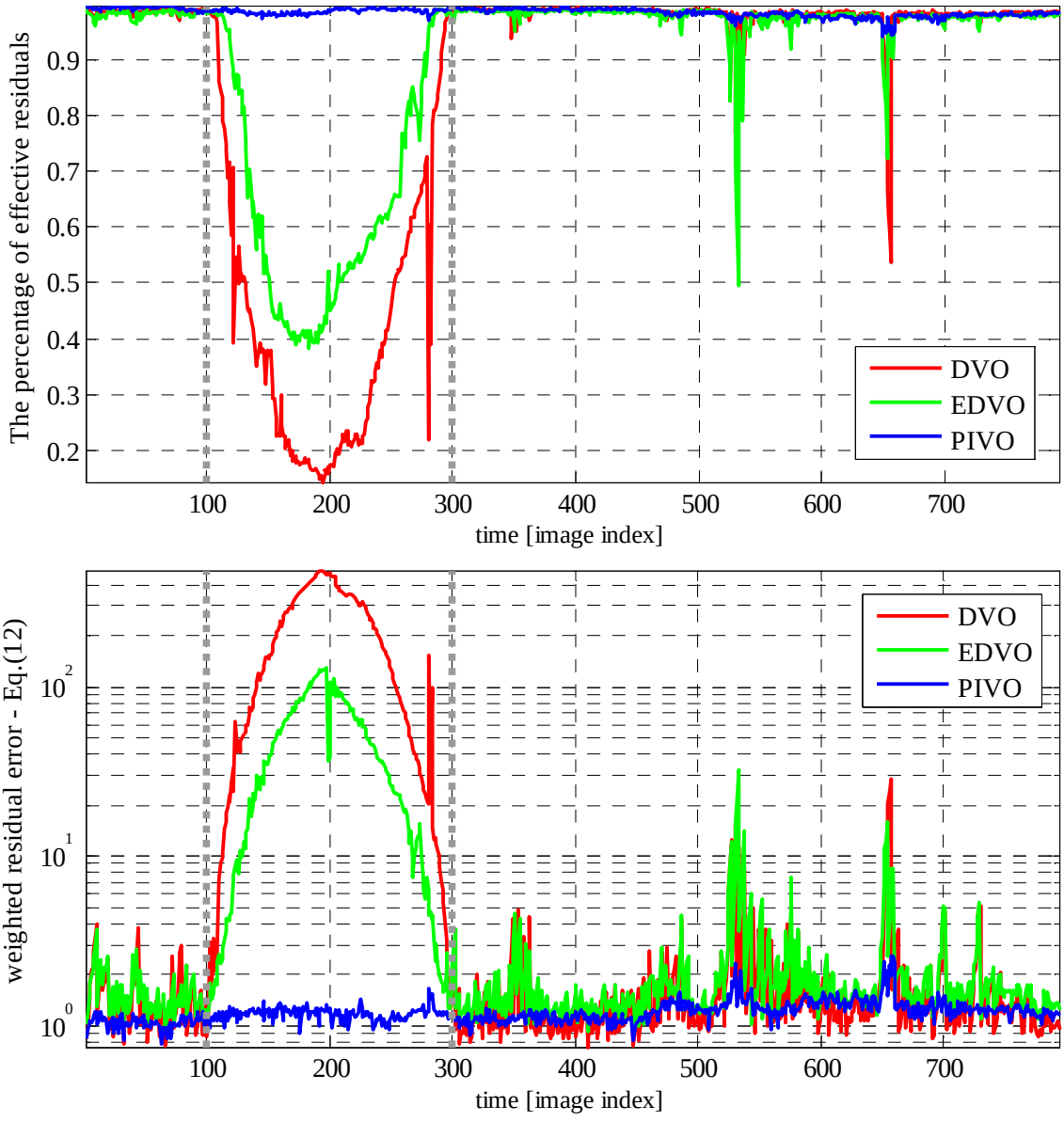


Figure 3.10: Residual distribution of each method.

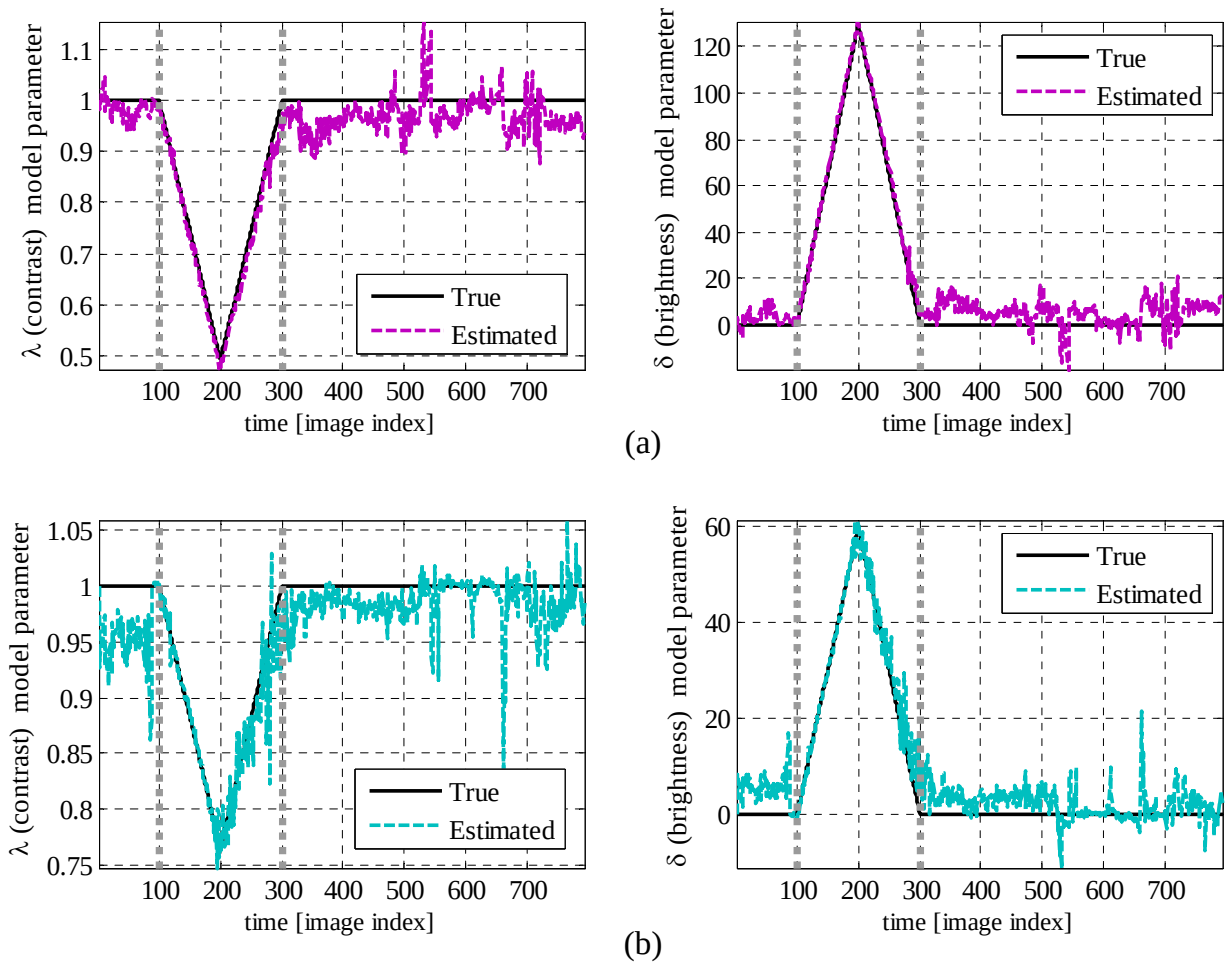


Figure 3.11: The true and estimated illumination change model parameters.

Fig. 3.11 shows the ground truth and the estimated illumination change model parameters defined in Eq. (3.5). It can be seen that PIVO estimates the illumination change model parameters for contrast and brightness changes correctly, which are used to generate the synthetic RGB-D dataset. Thanks to accurate estimation of the model parameters for each patch, the partial light variations are properly compensated during the direct motion estimation process in PIVO.

3.5.2 Author-collected Stationary RGB-D Dataset

RGB and depth images in the RGB-D dataset collected by the author are taken in a fixed position, which means that RGB-D camera does not move at all throughout the whole image sequences

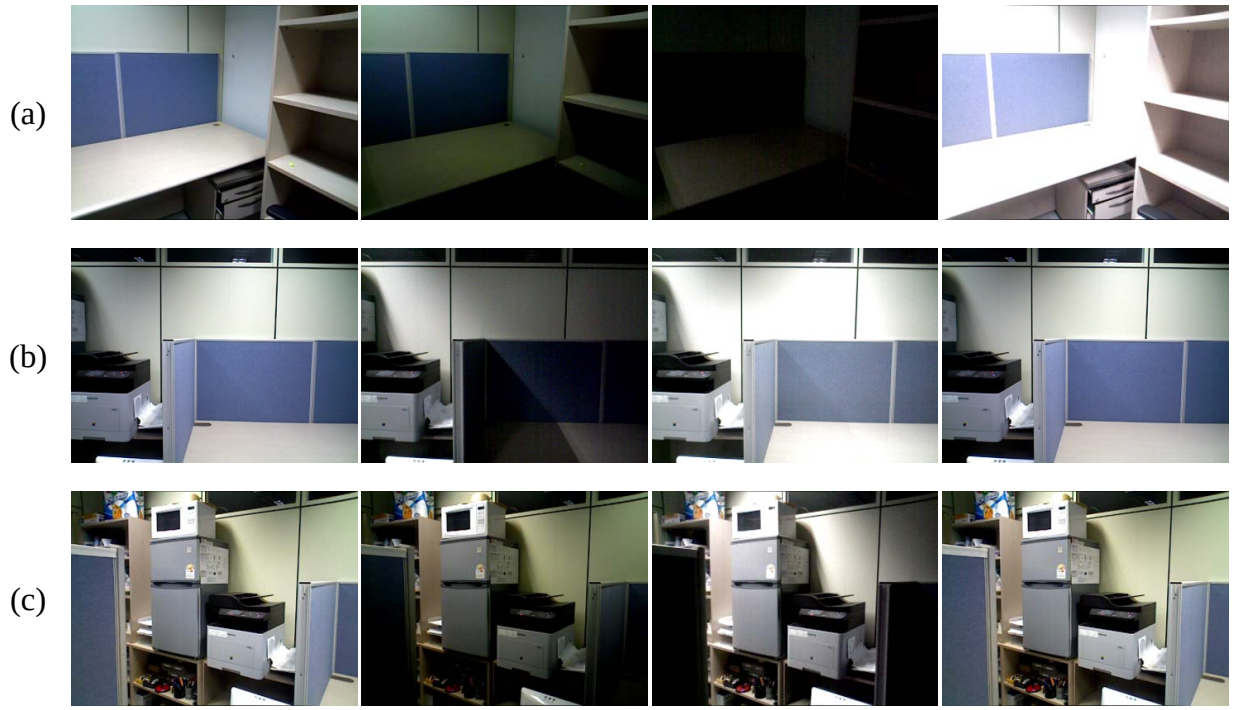


Figure 3.12: The RGB image sequences in the author-collected RGB-D dataset.

Table 3.2: Estimation results with author collected RGB-D dataset.

Name of Dataset	RPE [drift m/s]			ATE [m]		
	DVO	EDVO	PIVO	DVO	EDVO	PIVO
LAB1	1.320	1.036	0.044	2.484	1.868	0.247
LAB2	1.545	0.210	0.006	4.499	0.716	0.014
LAB3	0.222	0.009	0.011	0.942	0.014	0.023

as illustrated in Fig. 3.12. Instead, lights in the room are turned on and off repeatedly to test the robustness to illumination changes for the individual visual odometry methods. The evaluations are performed with three types of dataset. Sudden illumination changes take place in the entire images in 'LAB1'. Next, partial and irregular light variations occur in 'LAB2'. Lastly, lighting changes happen a little in 'LAB3'. The estimation results of each image sequence are summarized in TABLE 3.2.

As we expected, PIVO estimates the position of the stationary RGB-D sensor correctly in

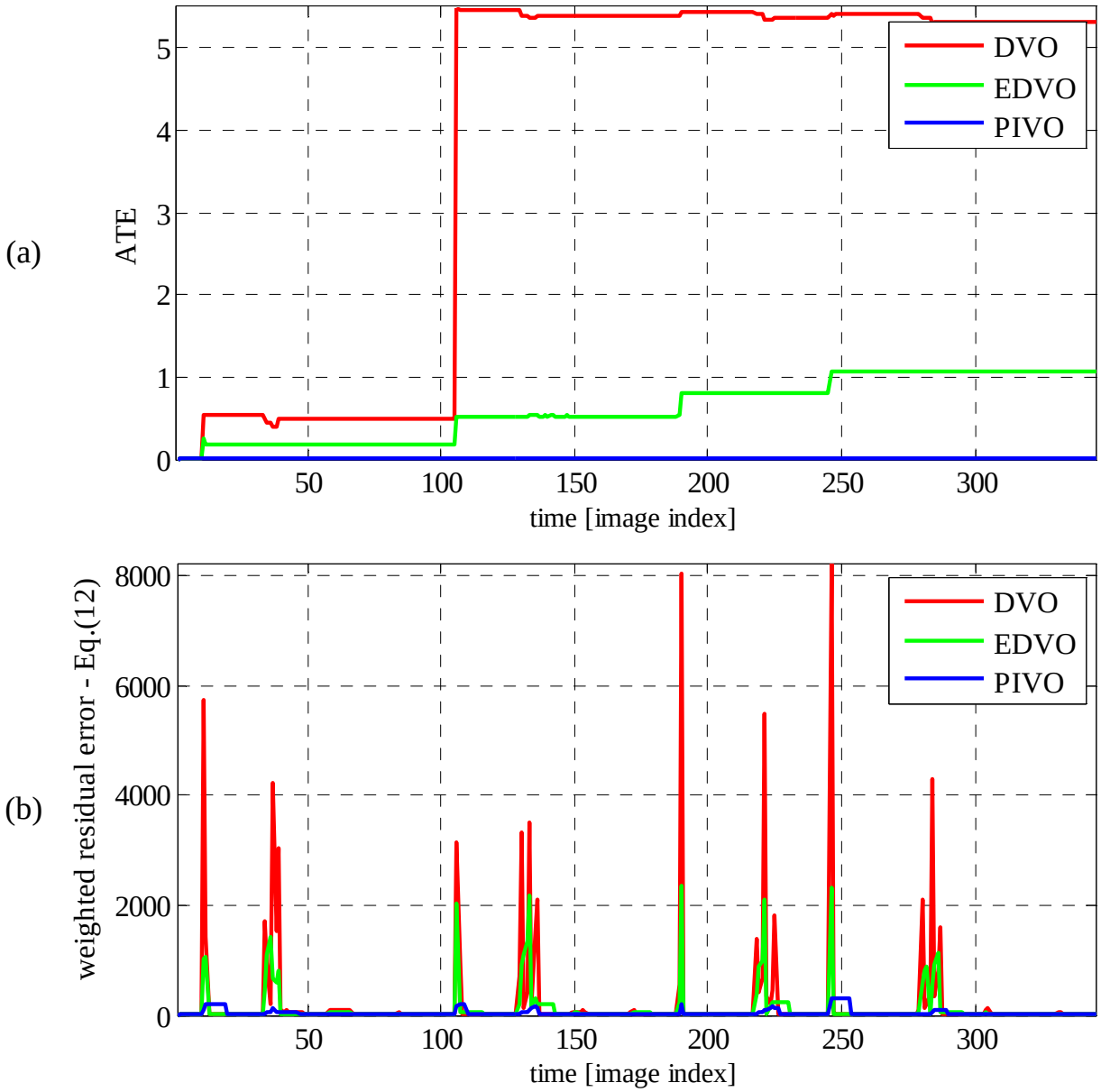


Figure 3.13: Comparison of ATE and weighted residual errors from 'LAB2'.

'LAB2' and 'LAB3'. On the other hand, incorrect movements of the camera as drawn in Fig. 3.13 are estimated by DVO and EDVO because of abrupt, irregular illumination changes in the images. DVO even produces a large drift error in 'LAB3', which means DVO is particularly sensitive to changes in light. From the above evaluation results and analysis, we can conclude that PIVO performs well not only in the ordinary image sequences, but also in the partial illumination change.

Please refer to the video clips submitted with this paper showing more details about the experiments.[1]

3.6 Conclusion

In this paper, we proposed a patch-based illumination invariant visual odometry pipeline, which works well in the irregular illumination change. To consider the partial light variations, the planar patch selection process is employed and the illumination change model is adopted in each extracted patch. The proposed cost function reflecting the illumination changes is minimized effectively by using the robust weighting function and the efficient second-order minimization (ESM) image alignment method. As a result, our method can accurately estimate the motion of the camera regardless of the partial lighting changes, also affine illumination parameters. Evaluation results with the synthetic RGB-D dataset and real experiment show that the accuracy of our algorithm is superior to the other direct visual odometry methods not only in the ordinary image sequences, but also in the illumination change.

[1] Video available at `https://youtu.be/sLKrVxGTbKQ`

4

Robust Visual Localization in Changing Lighting Conditions

Authors	Pyojin Kim[1]	rlavywls@snu.ac.kr
	Brian Coltin[2]	brian.j.coltin@nasa.gov
	Oleg Alexandrov[2]	oleg.alexandrov@nasa.gov
	H. Jin Kim[1]	hjinkim@snu.ac.kr
	[1] Seoul National University	
	[2] NASA Ames Research Center	

Publication **Robust Visual Localization in Changing Lighting Conditions.** Kim, Pyojin, Brian Coltin, Oleg Alexandrov, H. Jin Kim. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA), 2017*. Copyright 2017 IEEE.

Contribution	Problem definition	*significantly contributed*
	Literature survey	*significantly contributed*
	Method development	*significantly contributed*
	Implementation	*contributed*
	Experimental evaluation	*contributed*
	Preparation of the manuscript	*significantly contributed*

Abstract We present an illumination-robust visual localization algorithm for Astrobee, a free-flying robot designed to autonomously navigate on the International Space Station (ISS). Astrobee localizes with a monocular camera and a pre-built sparse map composed of natural visual features. Astrobee must perform tasks not only during the day, but also at night when the ISS lights are dimmed. However, the localization performance degrades when the observed lighting conditions differ from the conditions when the sparse map was built. We investigate and quantify the effect of lighting variations on visual feature-based localization systems, and discover that maps built in darker conditions can also be effective in bright conditions, but the reverse is not true. We extend Astrobee's localization algorithm to make it more robust to changing-light environments on the ISS by automatically recognizing the current illumination level, and selecting an appropriate map and camera exposure time. We extensively evaluate the proposed algorithm through experiments on Astrobee.

4.1 Introduction

Astrobee, a free-flying robot, is being built to autonomously navigate on the International Space Station (ISS), where it will assist astronauts, ground controllers, and researchers. It will replace the SPHERES robots, which currently operate on the ISS, but are limited to a two meter cube where they localize based on fixed ultrasonic beacons [60]. Astrobee will localize anywhere on the station through natural visual features (BRISK) stored in a pre-built sparse map [61]. See [62] for details about Astrobee's hardware. The lighting conditions on the ISS are controllable by the astronauts and change frequently. In particular, the lights are dimmed at night when the astronauts sleep. However, Astrobee's sparse maps are only constructed under a single lighting condition, and the effectiveness of this map under other lighting conditions is unknown. From the existing literature, it is unclear how the performance of visual localization systems are affected by illumination changes.

We analyze the effects of changing lighting conditions on our current visual localization algorithm, and then improve the algorithm to effectively localize regardless of lighting condition.

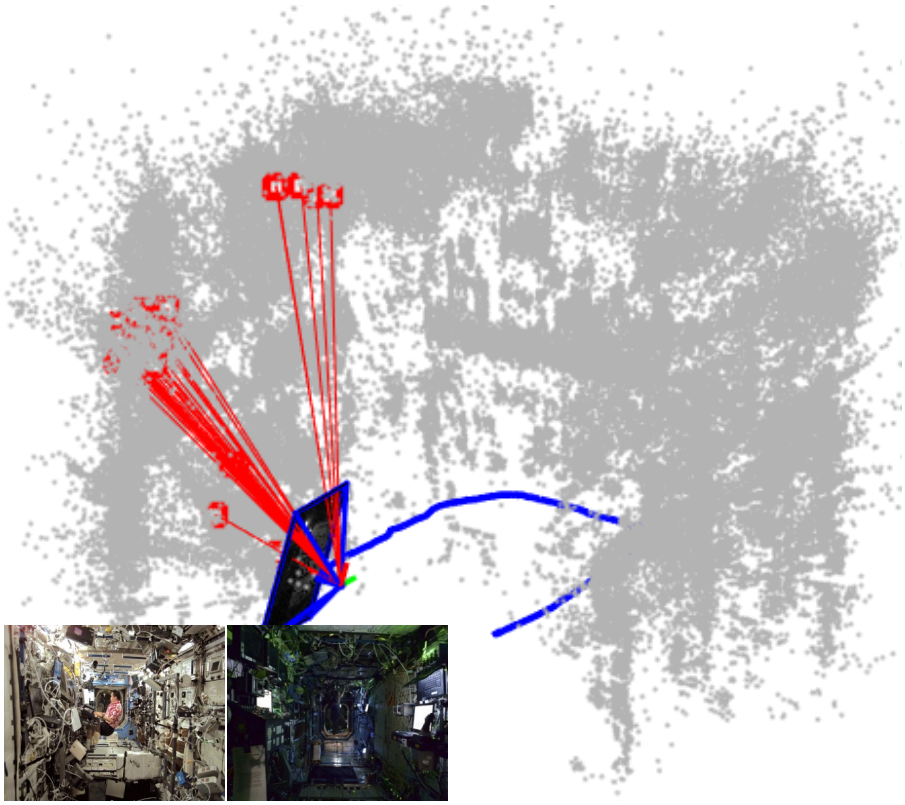


Figure 4.1: Astrobee, a free-flying robot designed to autonomously navigate on the International Space Station (ISS), can localize robustly under changing lighting conditions within multiple maps reconstructed offline using structure from motion (SfM). Two pictures show light conditions on the ISS during day (left) and night (right).

First, we perform extensive experiments to observe how Astrobee's localization algorithm performs under various lighting conditions. To the best of our knowledge, this is the first work to empirically investigate and analyze the performance of a visual localization system under changing lighting conditions. We learn which conditions are most effective for visual localization and map building.

Second, we present an algorithm for improved localization under changing lighting conditions. Instead of using a single pre-built map, we build multiple maps for various lighting conditions. We estimate the current brightness by comparing the image intensity distribution to the most similar images in each of the pre-built maps, and localize in the map with the closest brightness level. Furthermore, if the estimated brightness level is too dark or bright, we modify the

exposure time of the camera to improve feature matching. Our work focuses on Astrobee, but the challenge of illumination-robust visual localization and our proposed solution apply equally to other robots, such as self-driving cars and UAVs.

4.2 Related Work

In the past decade, image-based localization has been an active research area in robotics and computer vision. From the vast literature in visual localization, we review work related to illumination changes in image sequences.

Appearance-only SLAM localizes not in the metric space but in the appearance (image) space to find the most similar images in a map. It has proven successful at loop-closure and kidnapped robot problems. FAB-MAP [63], one of the most famous image retrieval approaches, shows impressive performance through a probabilistic model of natural features in a bag-of-words [64]. However, lighting variations drastically degrade FAB-MAP's place detection performance [65]. To deal with the weakness of image feature algorithms such as SIFT and SURF under lighting changes [66], the fine vocabulary method [67] was introduced to learn lighting invariant feature descriptors [68]. Illumination changes have also been addressed with a high dynamic range (HDR) camera, which is used to construct a set of SIFT descriptors captured at different exposure times [69]. Another approach, SeqSLAM [65], localizes under severe lighting changes by comparing sequences of normalized images rather than comparing images pairwise. However, SeqSLAM is extremely sensitive to the changes in the camera field of view, and its computational complexity ($O(nm)$, where n is the map size and m is the sequence size) is much higher than the vocabulary tree Astrobee uses ($O(\log n)$) [70].

Numerous metric visual SLAM algorithms are based on either features [13], [16], [71] or direct (dense) image comparisons [19], [24], [21]. Some have been successfully implemented on micro aerial vehicles (MAVs) [72], [24] and smartphones, and show promising results. However, most SLAM and visual odometry do not explicity address and have not been tested under changing lighting conditions. An affine model for lighting correction is taken into account in [8] and

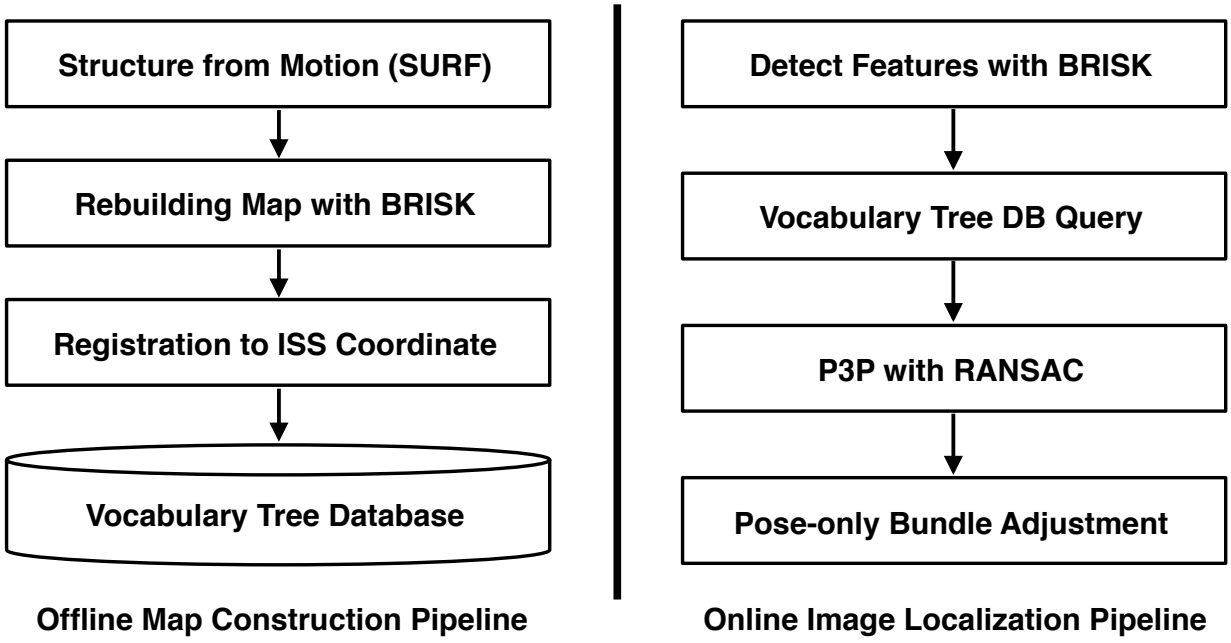


Figure 4.2: Astrobee's mapping and localization algorithms.

[22] to avoid performance degradation from unexpected illumination changes for direct visual odometry and SLAM approaches. In [73] and [74], an illumination invariant transform for images [75] is adopted for robust visual localization of autonomous vehicles, and one-dimensional greyscale images where intensity values primarily depend on the material property have been used successfully in a metric monocular localization application. However, it is not easy to apply many existing image feature algorithms like SIFT, SURF to the illumination-invariant color space directly, and a strong assumption is required to make use of the illumination invariant image transformation.

4.3 Astrobee's Current Localization System

We briefly explain Astrobee's localization and mapping system. For full details, see [61].

4.3.1 Offline Map Construction

We build an offline map because the operating region is fixed, and offline maps provide higher stability and accuracy than existing SLAM systems. An overview of the offline mapping algorithm is shown in Figure 4.2. First, a sequence of images for offline map construction is collected by Astrobee. We match features on the images with approximate nearest neighbors on SURF features [76]. From the features, we construct tracks of features seen across multiple images [77]. An initial map is formed from the estimated essential matrices between consecutive images. Incremental and then global bundle adjustment processes optimize the camera poses and the 3D positions of landmarks by minimizing reprojection error. The map with SURF features is then rebuilt with BRISK binary features [78], which are less accurate and robust than SURF features (critical for accurate offline map building) but also much faster (critical for online localization). The map is registered to the pre-defined ISS coordinate system. Finally, a hierarchical vocabulary tree database [79] is constructed for fast retrieval of similar images for localization. The final constructed map, composed of fast BRISK features and a vocabulary database, enables quick metric localization.

4.3.2 Online Image Localization

The algorithm for online image localization, which computes the 6 DoF camera pose, is shown in Figure 4.2. First, we detect BRISK features in the query image. We search for these features in the bag of words vocabulary tree database to find the most similar images in the map. After finding the candidate matching features in the map, the robot pose is estimated with RANSAC and the P3P algorithm [80].

4.4 Effect of Changing Lighting Conditions

We investigate the effect of changing lighting conditions by recording images under ten different conditions simulating day, night and intermediate lighting levels on the ISS. For each lighting

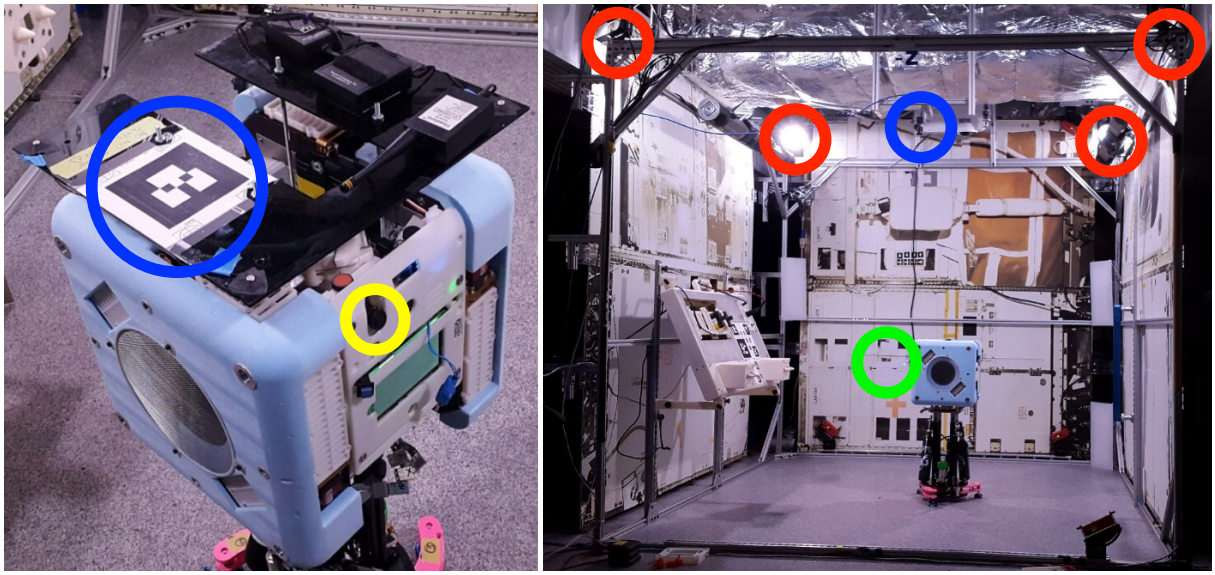


Figure 4.3: Astrobee (left) and experimental environment in the granite lab (right) simulating the interior of ISS. Red circles on the right denote the adjustable lights. The green circle is the place where the intensity of light is measured with a digital light meter. The blue circles indicate the AR tag and overhead camera for providing the ground-truth position. The yellow circle indicates Astrobee's navigation camera.

level (ranging from 5-135 lux), nearly 2500 images were recorded, and a digital light meter on a fixed point on the wall measured the luminance in lux. The images are captured on a granite table which simulates the indoor environment of the ISS (see Figure 4.3). The robot slides freely on the surface of the table, constrained to motion on the two dimensional plane, but the localization algorithm computes the full 6 DoF camera pose. An overhead camera and an AR tag on Astrobee measure the ground truth pose. Figure 4.4 shows example images recorded for each lighting level, and Figure 4.5 shows the corresponding brightness distributions.

Nine maps for different lighting conditions were produced with the offline map construction algorithm. The 5 lux lighting condition is too dark to detect features to build a map, and is excluded from the remainder of the analysis.

Figure 4.6 shows the success rate for a subset of lighting condition and map combinations. An image is labelled as a failure if the estimated pose is outside the plane of the granite table (a 1.5x1.5 m area) or if localization fails. The number of features detected is roughly constant

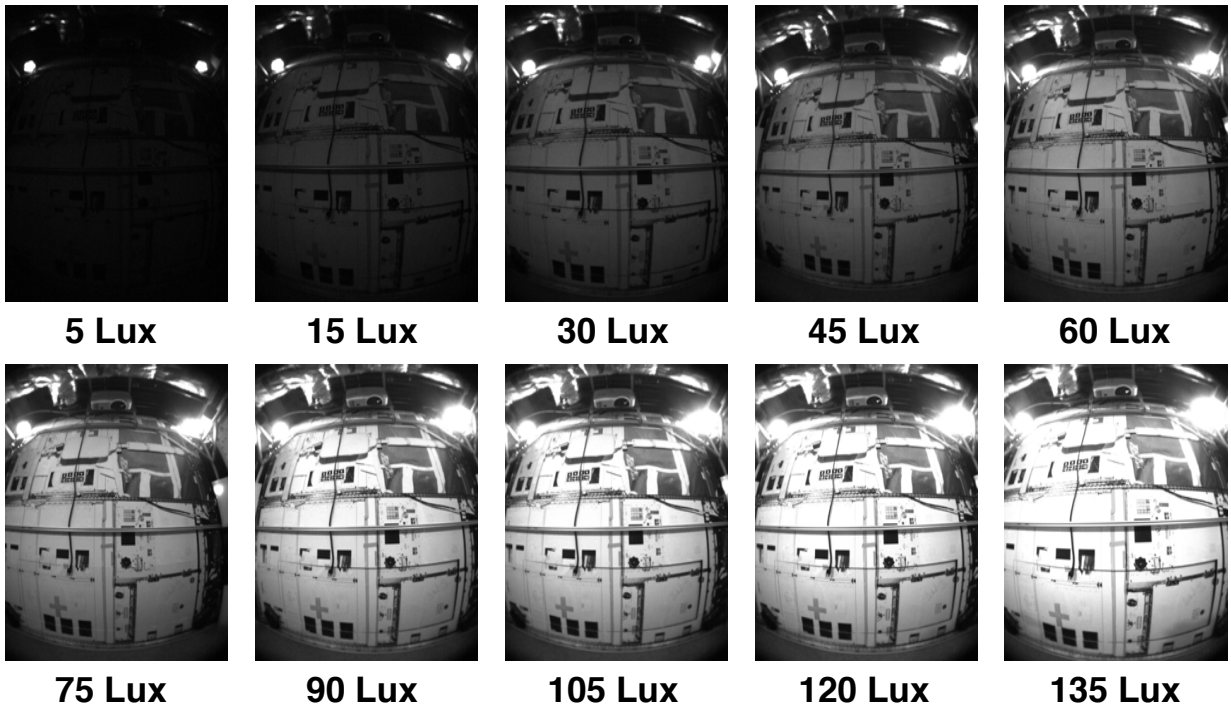


Figure 4.4: Images taken under different lighting conditions in the granite lab. The wall panels imitate the interior of the ISS.

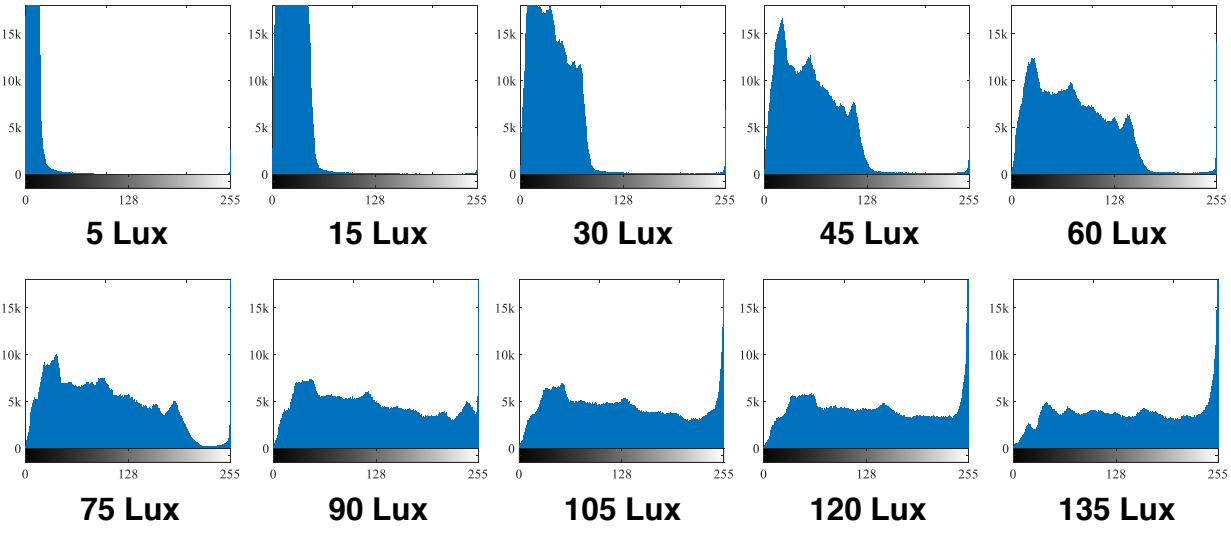


Figure 4.5: Brightness distributions of the images in Figure 4.4. Each pixel is represented as 8-bit grayscale from 0 (black) to 255 (white).

across all lighting levels, and the number of inlier features varies in a manner mirroring the success rate. Given that localization succeeds, the translational error varies little with map and

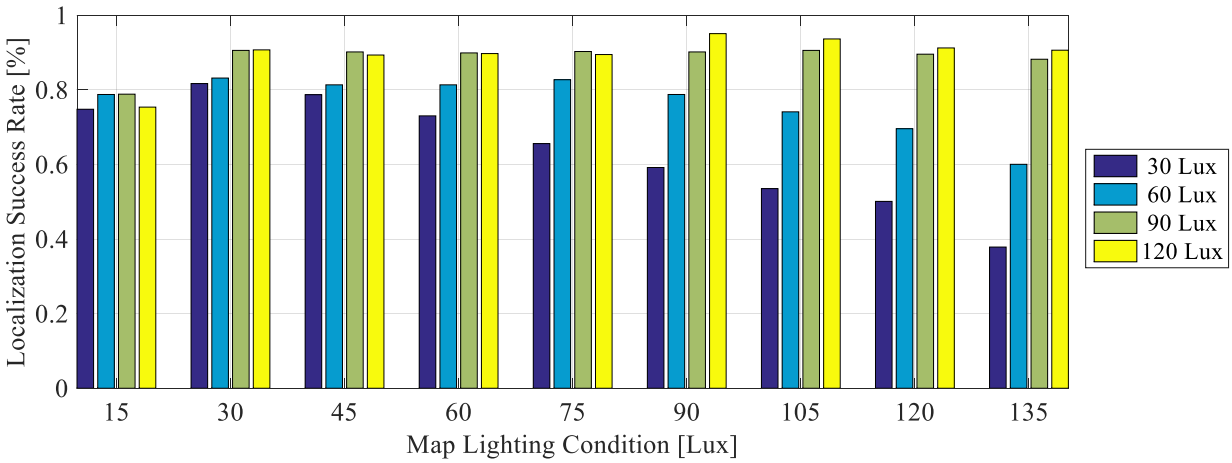


Figure 4.6: The success rate for various lighting / map combinations.

lighting conditions, averaging around 5 cm. Dark maps below 45 lux show over an 80% success rate regardless of lighting conditions. Bright maps over 60 lux work well only with bright images. The reason for this behavior is likely that the feature descriptors cannot describe features in bright lighting conditions well because many image intensity values are saturated. In dark conditions, saturation rarely occurs. Therefore, we expect that Astrobee will fail to localize at night if the pre-built map is constructed with images captured in the day. However, a map constructed from images collected at night will work well in all lighting conditions.

Still, as one would expect, the most effective combination for stable localization is to use a map constructed in the same lighting conditions as the test images. Therefore, to make Astrobee's localization succeed regardless of the current lighting conditions, we propose to automatically recognize the environmental brightness, and choose the best of multiple maps for different lighting conditions.

4.5 Illumination-Robust Visual Localization

We present an illumination-robust visual localization algorithm which estimates the current lighting condition and selects an appropriate sparse map. As shown in Figure 4.7, the proposed algorithm is easily inserted into the existing localization algorithm. The proposed approach can be

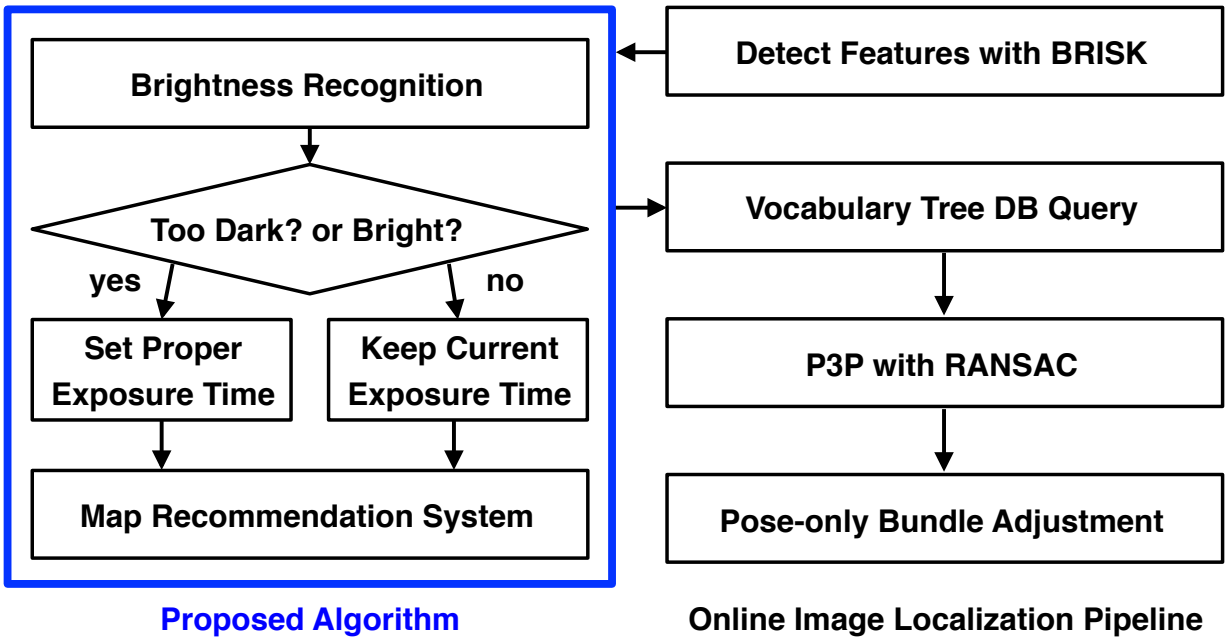


Figure 4.7: The illumination-robust visual localization pipeline. The proposed algorithm (blue box) is inserted into the original pipeline (Figure 4.2).

employed not only for Astrobee, but for any system that localizes in environments with changing lighting conditions.

4.5.1 Brightness Recognition

We design a brightness recognition algorithm on the premise that images taken at similar places under similar lighting will have similar intensity distributions. Figure 4.8 outlines the brightness recognition algorithm. We first detect BRISK features in the query image. Next, we query the bag of words vocabulary database for the images in each constructed map most similar to the detected BRISK descriptors. Then, we compute intensity distributions for each similar image, and compare them to the query image. We choose the most similar image with the symmetric KL-divergence:

$$D_{symKL}\left(P,Q\right) = \sum_{i=1}^{n} P(i) \log \frac{P(i)}{Q(i)} + \sum_{i=1}^{n} Q(i) \log \frac{Q(i)}{P(i)}$$

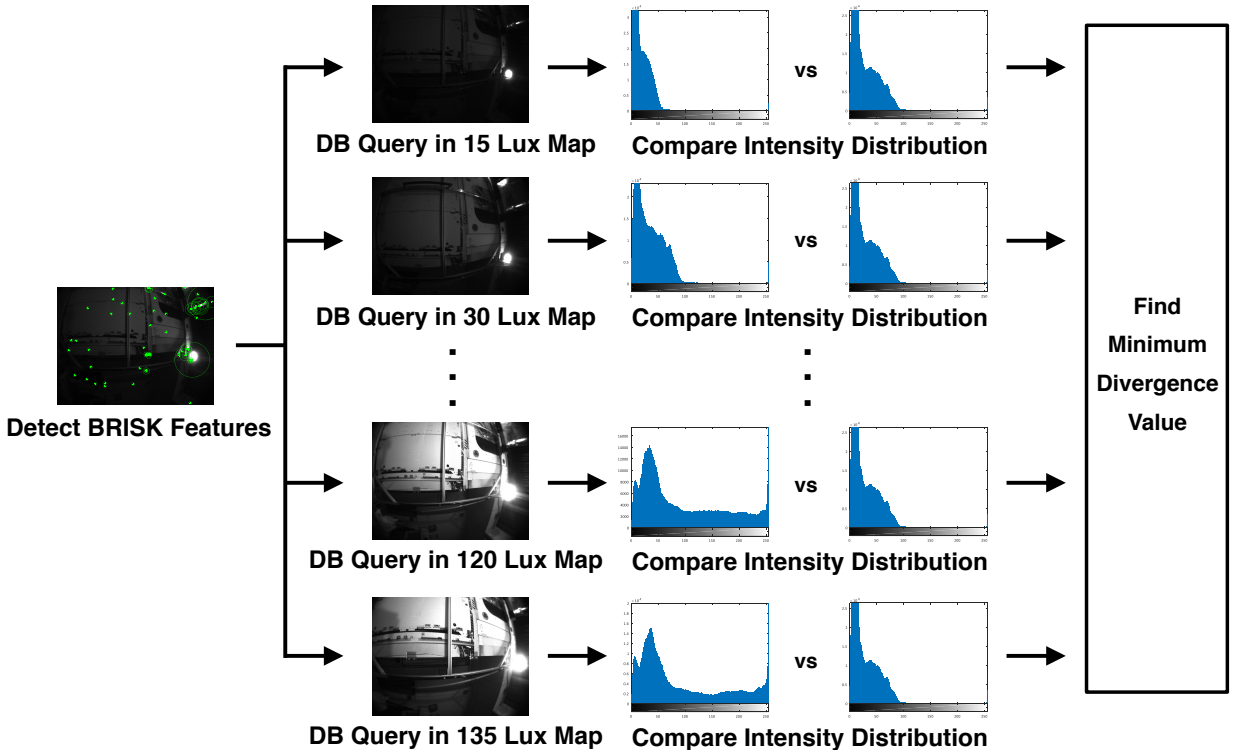


Figure 4.8: Illustration of the proposed brightness recognition algorithm.

where P is the intensity distribution of the query image and Q is the intensity distribution of the retrieved images from each map. The smaller the KL-divergence, the more similar the two images are, hence we pick the map with the nearest brightness. Additionally, the exposure time of the camera can be controlled to extract more valuable features based on the estimated brightness. For example, the exposure time can be increased if the estimated lighting condition is too dark to detect enough features, or can be decreased to remove intensity saturation in the image if the lighting is too bright.

Note that we develop the proposed algorithm for a camera with adjustable exposure time, not for a camera with automatic exposure control. Unpredictable light variations caused by automatic exposure control make many existing visual odometry and localization algorithms unstable and unreliable. However, by changing the exposure time of the camera in a predictable manner, visual localization can account for the changes and perform stably.

4.5.2 Map Recommendation System

To achieve the best localization, the constructed map closest to the estimated current lighting condition is selected. For Astrobee's case, using the lighting conditions and maps from Section 4.4, we match each estimated lighting condition with a map constructed under that same condition and the same exposure time, except for in the 15 lux case, where the 30 lux map is used with double the exposure time. This selection is based on the earlier analysis, which showed that it is most effective to use a map constructed under the same lighting conditions as the test images, and from the observation that there is a linear relationship between the estimated brightness and exposure time. Note that the number of maps used can easily be adjusted according to the robot's hardware, the environment, and user preference. This algorithm requires the construction of multiple sparse maps, but enables localization under changing lighting conditions.

4.6 Evaluation

We evaluate the effectiveness of the illumination-robust visual localization approach with experiments performed on the granite table (see Section 4.4 and Figure 4.3).

4.6.1 Constant Exposure Time

First, we captured images from Astrobee's navigation camera at 15 Hz with a constant exposure time to confirm the effectiveness of the brightness recognition algorithm. With a changing exposure time, we are not able to directly compare to the original algorithm on the same image sequence. We recorded test runs with three movements patterns: a circle facing outwards, side to side, and stationary. As the robot moved, we brightened and dimmed the lights repeatedly, simulating the range of expected ISS lighting conditions.

For every test run, we tested the localization system with and without the proposed brightness recognition and map recommendation algorithm. Figure 4.9 shows sample images, the estimated brightness, the number of inlier feature matches, and the success or failure of localization for the

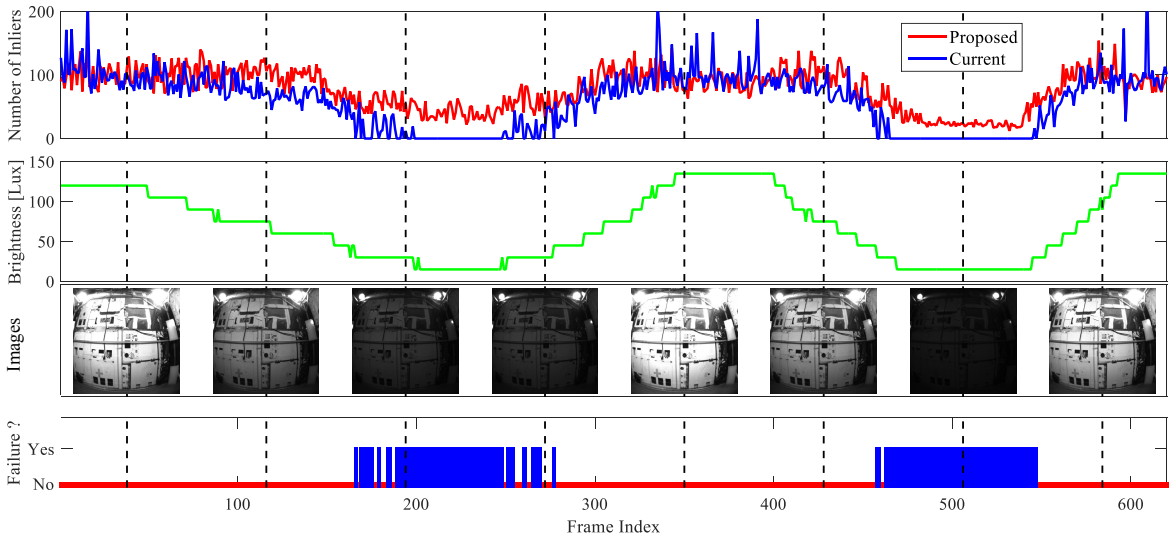


Figure 4.9: Evaluation results comparing the proposed and current localization methods with constant exposure time, a bright map, and no motion. The dotted vertical lines represent the time instants at which each snapshot is taken. The current estimated lighting condition (green line) shows similar behavior to the brightness level of the actual images in the third row. Although the lights dimmed from frames 200 to 250, the proposed algorithm (red line) shows no failure and maintains the proper number of inliers whereas the current method (blue line) cannot localize.

stationary experiment with a map built under bright conditions. The estimated brightness levels closely match the actual lighting conditions. The proposed algorithm increases the localization success rate, while the original algorithm without brightness recognition fails when it is dark.

We plotted estimated moving trajectories of Astrobee for the circle and sideways experiments with 3D landmarks and ground truth trajectories in Figure 4.10. The estimated trajectories with the proposed method are qualitatively similar to the true moving trajectories.

We also analyzed localization accuracy quantitatively by comparing the estimated pose with the ground-truth position provided by the overhead camera. The translational error with and without our proposed improvements is shown in Figure 4.11. As we showed in Section 4.4, given that visual localization succeeds, the translational accuracy depends little on the lighting conditions of either the current environment or the map. However, we can still observe a small improvement in positional accuracy from choosing a map through brightness recognition. All of the experimental results with the constant exposure time are summarized in Table 4.1.

Figure 4.10: ‘Circle’ (left) and ‘Sideways’ (right) trajectories estimated by our method (in red) are very similar to the ground truth trajectories (in black).

Table 4.1: Evaluation Results of Illumination-robust Visual Localization on Constant Exposure Time

Experiment	Algorithm	Success Rate	Mean Inliers / Matches	RMSE (cm)
Circle	Proposed	0.99	83 / 132	5.92
	Current	0.81	73 / 130	5.92
Sideways	Proposed	0.66	55 / 115	6.28
	Current	0.51	58 / 122	5.46
Stationary	Proposed	0.89	77 / 127	7.05
	Current	0.67	82 / 131	7.11

4.6.2 Dynamic Exposure Time

Additional experiments were conducted on Astrobee when the exposure time is changed based on the estimated lighting condition. The proposed illumination-robust algorithm is implemented on the robot (see [62] for platform details).

Figure 4.12 shows that the number of correct matches, an indicator of whether localization

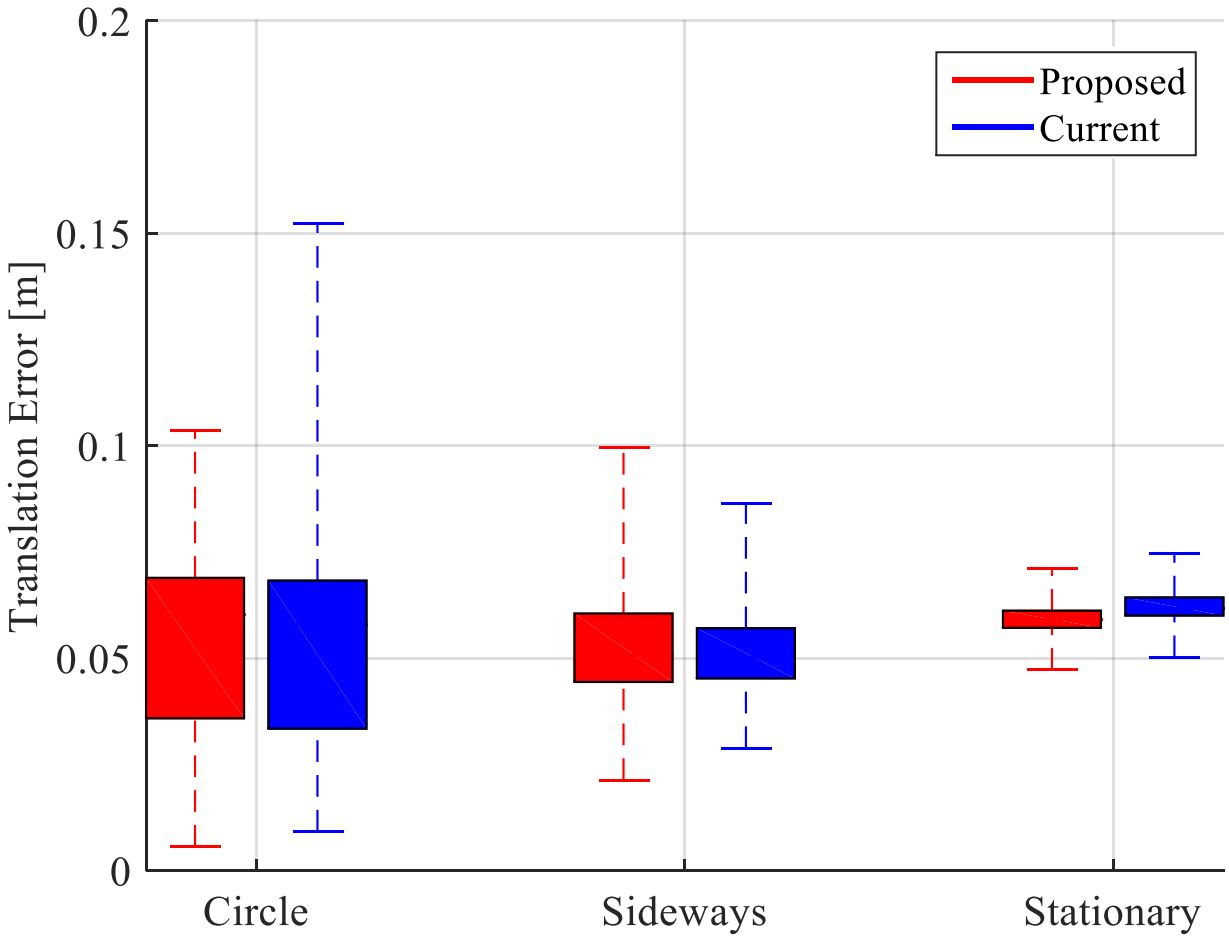


Figure 4.11: Translational error of each method in the experiments.

Table 4.2: Illumination-robust Localization with Dynamic Exposure

Experiment	Algorithm	Success Rate	Mean Inliers / Matches
Circle	Proposed	0.92	72 / 129
	Current	0.81	73 / 130
Sideways	Proposed	0.74	61 / 112
	Current	0.51	58 / 122
Stationary	Proposed	0.94	83 / 133
	Current	0.67	82 / 131

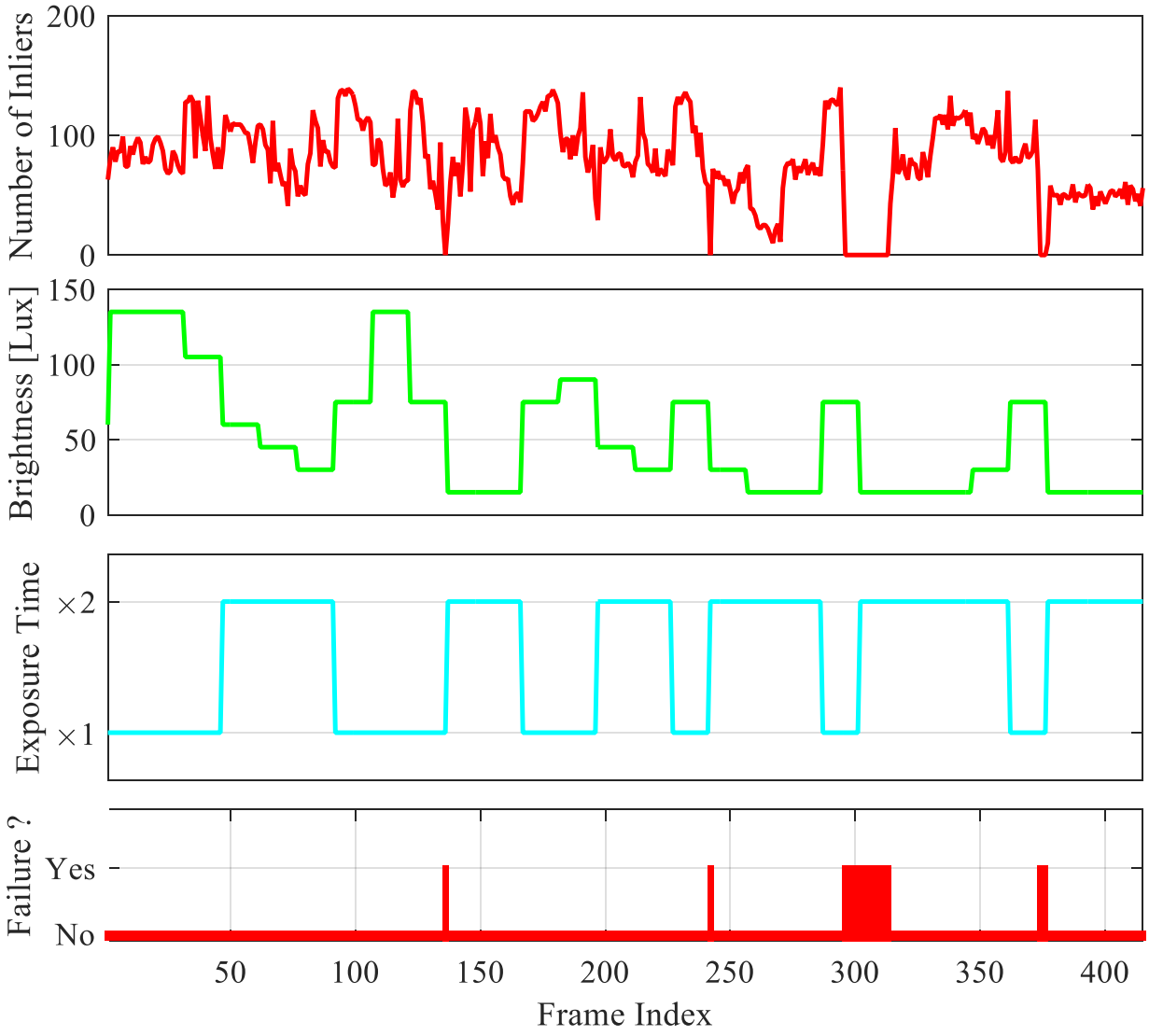


Figure 4.12: Results on Astrobee in the stationary case with dynamic exposure time. The exposure time setting (cyan line) changes if the estimated lighting condition (green line) is too dark or too bright. Failures occasionally occur when the lighting condition is too dark to detect features (almost black).

will succeed or fail, continues to remain high by adapting the exposure time if the current estimated lighting condition is too dark (below 30 lux) or too bright (over 135 lux). Unlike existing algorithms which are very sensitive to the light variations caused by unexpected automatic exposure control, we can also achieve high stability and robustness by adjusting the exposure time in a pre-defined predictable manner. The high number of correct matches and good success rate

compared to the original algorithm are shown in Table 4.2.

Please refer to the video clips submitted with this paper showing more details about the experiments.[1]

4.7 Conclusion

We have investigated the performance of Astrobee's visual localization algorithm under changing lighting conditions, and presented an illumination-robust visual localization algorithm that automatically recognizes the brightness level to select an appropriate camera exposure time and map. This approach enables Astrobee to localize robustly under changing lighting conditions at the cost of building multiple lighting-specific maps. Our approach assumes uniform lighting changes; future work should consider the effects of irregular lighting changes, such as bright spots and shadows caused by sunlight coming through the windows.

[1] Video available at `https://youtu.be/Nuyq74wbz7I`

5

Autonomous Flight with Robust Visual Odometry under Dynamic Lighting Conditions

Authors	Pyojin Kim[1]	rlavywls@snu.ac.kr
	Hyeonbeom Lee[1]	koreaner33@snu.ac.kr
	H. Jin Kim[1]	hjinkim@snu.ac.kr
	[1] Seoul National University	

Publication	**Autonomous Flight with Robust Visual Odometry under Dynamic Lighting Conditions.** Kim, Pyojin, Hyeonbeom Lee, H. Jin Kim. In *Autonomous Robots (AURO), 2018*. (under review)

Contribution	Problem definition	*significantly contributed*
	Literature survey	*significantly contributed*
	Method development	*significantly contributed*
	Implementation	*significantly contributed*
	Experimental evaluation	*contributed*
	Preparation of the manuscript	*significantly contributed*

Abstract Sensitivity to light conditions poses a challenge when utilizing visual odometry (VO) for autonomous navigation of small aerial vehicles in various applications. We present an illumination-robust direct visual odometry for a stable autonomous flight of an aerial robot under unpredictable light condition. The proposed stereo VO achieves robustness with respect to the light-changing environment by employing an affine illumination model to compensate abrupt, irregular illumination changes during direct motion estimation. We furthermore incorporate a motion prior from feature-based stereo visual odometry in the optimization, resulting in higher accuracy and more stable motion estimate. Thorough analyses of convergence rate and linearity index for the feature-based and direct VO methods support the effectiveness of the usage of the motion prior knowledge. We extensively evaluate the proposed algorithm on synthetic and real micro aerial vehicle (MAV) datasets with ground-truth. Autonomous flight experiments with an aerial robot show that the proposed method successfully estimates 6-DoF pose under significant illumination changes.

5.1 Introduction

Autonomous aerial robots that are designed to perform tasks without direct human remote control rely on accurate state information. Due to the limitations of GPS or motion capture system, investigations have been performed to combine multiple sensors such as laser scanner, sonar, barometer in order to localize the aerial robots. Alternatively, vision-based state estimation so-called visual odometry (VO) [39] can offer a less expensive solution with up to centimeter-level accuracy without sacrificing too much payload.

Unlike ground vehicle navigation [40, 47], however, small autonomous aerial robots pose a challenge in applying VO. VOs for the aerial robots have to compute sufficiently fast and accurate position estimates to maintain active control at high refresh rate and avoid failure. They also should be light enough to run on an onboard computer with limited processing power. Because of these difficulties, VO algorithms for aerial robots are still actively researched with RGB-D camera [42, 81], stereo camera [82, 83] and a single camera [24, 84].

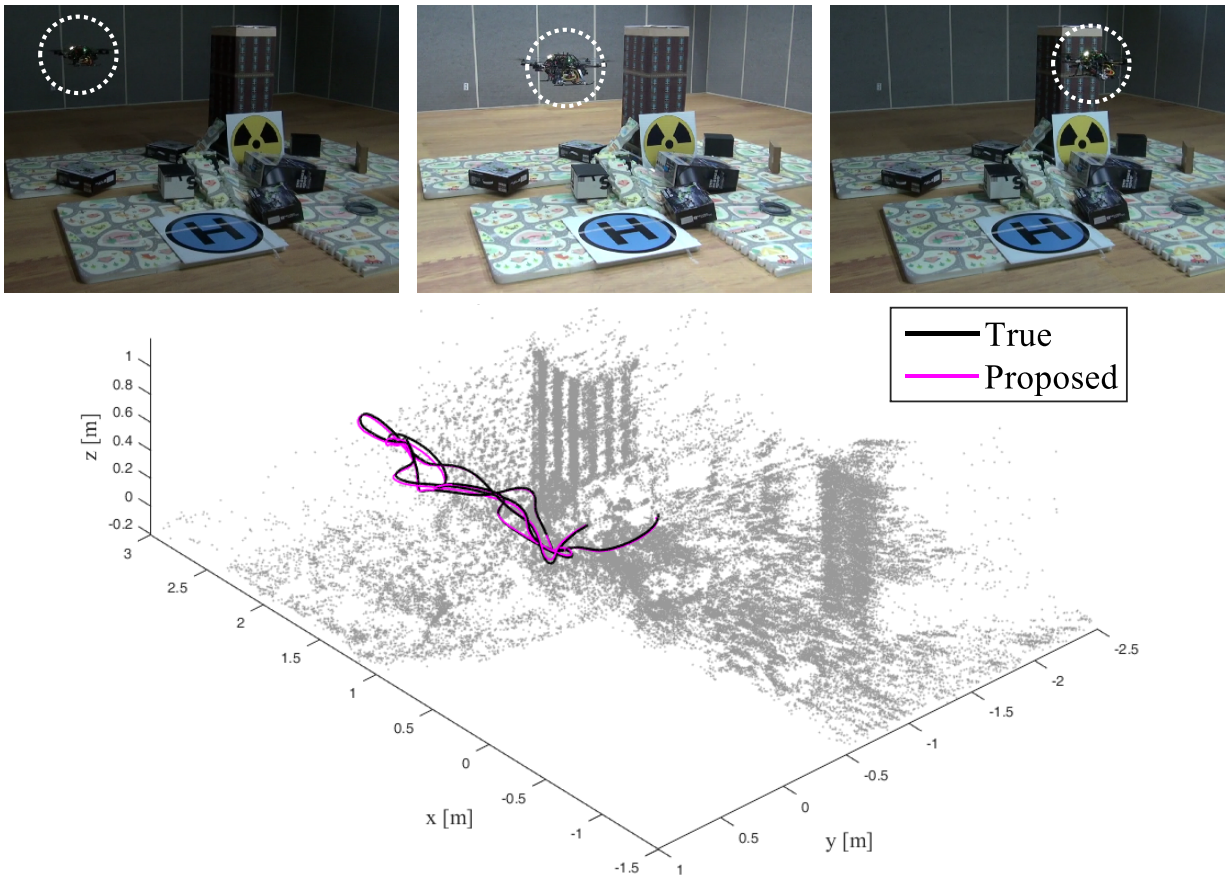


Figure 5.1: Hexacopter aerial robot in our autonomous flight experiments with varying light conditions by turning on and off the lights repeatedly.

In this work, we focus on the robustness of VO for an autonomous flight of the aerial robots. Although the accuracy and speed have been main objectives of many VO research [71, 25], the robustness to external environmental changes has not been addressed much. Among the various environmental factors, light changes in an image, including highlights, shadows caused by the changes of the camera viewing angle, unpredictable changes of a light source, and the automatic exposure control, are inevitable phenomena that the aerial robots must deal with in practice. For example, it is well known that direct VO methods [6], which perform state estimation using image brightness values, are very vulnerable to light variations, but the robustification to light changes still remains a challenge.

To address such issue, we propose a robust direct visual odometry algorithm that enables re-

liable autonomous flight of the aerial robots even in light-changing environments (see Fig. 5.1). The proposed stereo VO method *simultaneously* estimates the 6-DoF camera pose and the photometric parameters of the affine illumination change model [46] for individual patches in an image. Furthermore, we utilize a motion prior from feature-based VO to guide and stabilize direct motion estimation. Extensive evaluations show that the proposed algorithm can achieve more accurate state estimation than other state-of-the-art VO methods in light-changing environments while maintaining comparable performance in normal light conditions. The contributions of the paper can be summarized as follows.

- We present a novel direct VO algorithm that is robust under challenging lighting environments by including local affine parameters for estimating irregular illumination changes.
- We integrate a motion prior from the feature-based method into the direct approach for stable motion estimation, and analyze its usefulness in terms of convergence property.
- We demonstrate a real-time system enabling autonomous flights of the aerial robot in environments with irregular illumination changes.

Section 5.2 reviews related literature on direct motion estimation methods, particularly those involving light changes. Section 5.3 introduces some required notations and the problem. Section 5.4 provides an overview of the VO pipeline, and Section 5.5 explains the proposed motion estimation algorithm. Section 5.6 analyzes the convergence property of the proposed method. Section 5.7 provides evaluation results and demonstrates how the proposed method makes the aerial robot fly autonomously under challenging lighting conditions. We conclude in Section 5.8.

5.2 Related Work

In robotics and computer vision communities, various VO and visual simultaneous localization and mapping (V-SLAM) methods have been researched actively in the last decade. From the vast literature in the visual navigation field, we review related work in terms of illumination changes and implementation on aerial robots.

VO algorithms can be classified into indirect and direct methods depending on the type of visual information [85]. Indirect methods utilize an intermediate representation to track the camera pose rather than the direct measurements. Feature-based methods, the most widely used indirect methods, show successful 6-DoF camera motion estimation [9, 71, 17]. However, they require enough brightness and textures to extract consistent keypoints from an image [48]. This requirement is not satisfied in varying illumination conditions considered in this paper.

Direct VO methods [6, 21, 24] estimate 6-DoF camera motion by minimizing the photometric error between image frames, and they are receiving attention for their improved accuracy and robustness to little texture with the help of hardware progress. They heavily rely on the photo-consistency assumption that a scene point appears with constant brightness intensity across multiple images. [6] estimates the RGB-D camera motion accurately with a robust error function which rejects the noise and outliers in the photometric error. In [24], a semi-direct monocular VO is implemented on the onboard computer of a multirotor, showing precise and fast state estimation results by combining the advantages of feature-based and direct methods, and it is extended to multi-camera systems in [25]. Although these direct VO methods demonstrate impressive levels of accuracy, they have not been fully tested in challenging environments where the photo-consistency assumption does not hold (e.g., abrupt and irregular illumination changes occur).

Only a few direct VO methods give consideration to illumination changes during the direct motion estimation. It is assumed in [7] that the entire pixels follow the same affine illumination change model [46]. In order to ignore the illumination changes altogether between image frames, [54] estimates a pure albedo image of the texture. In [22], the modified photometric error based on the affine brightness change is employed. [86] addresses the vulnerability of light changes by using a binary descriptor, which is invariant to monotonic changes in intensity. [87] evaluates various direct image alignment methods for their accuracy and robustness under challenging lighting conditions. Recently, the direct sparse model [85, 88] is proposed, and photometric camera calibration is considered explicitly to mitigate photo-consistency assumption [89]. While these methods present superior motion estimation even in light-changing environments, they have

not been applied to the autonomous flight of an aerial robot in an environment with severe light changes.

The work which is the most similar to the proposed approach is [90, 25], which are the two-stage VO methods combining feature based and direct tracking approaches in a sequential manner. [90] extends LSD-SLAM to the stereo camera and employs a feature-based VO to estimate the motion between keyframes. The feasibility analysis of using the feature-based VO as a motion prior has not been addressed in detail while our manuscript provides in-depth analyses of convergence property of direct and feature-based functions. In [25], the photometric error is minimized at first, and then the estimated camera pose and the position of the observed 3D points are again optimized to reduce the reprojection residuals. SVO requires separate mapping thread additionally to minimize the reprojection residuals, and the minimization procedure of different cost functions is different from the proposed method. Both approaches have not performed sufficient performance evaluation with the aerial robot in a challenging environment where severe lighting changes occur.

Our algorithm builds on our previous work of [8], which is the patch-based illumination-robust direct visual odometry that estimates not only the 6-DoF camera pose but also the parameters of the affine illumination change model for individual patches. We newly integrate feature-based VO as a motion prior to the proposed direct VO method to guide the optimization by seeding it with an estimate closer to the true solution, resulting in more stable estimates. Importantly, we analyze and compare convergence rate and linearity index of each cost function used in feature-based and direct VO to support the usage of the feature-based VO as the motion prior. We validate the effectiveness and accuracy of our VO algorithm by recovering the 6-DoF camera motion and the photometric parameters of the author-collected dataset where irregular illumination changes exist in the stereo image sequences as well as on manually disturbed sequences of the EuRoC dataset [91]. Furthermore, we implement the proposed approach on an aerial robot with a stereo camera, achieving stable autonomous 3-D flight in light-changing environments.

5.3 Notation and Problem Statement

We organize the notations using a stereo camera model, but the setup can be transferred to the RGB-D camera model in [8]. The superscripts (l) and (r) denote the left and right camera respectively and k is used to represent the frame index. $I_i^{(l)k}$ is the i-th image patch in the left image at time step k. A pixel point is denoted with $\mathbf{x}_{ij}^{(l)k} = \left[x_{ij}^{(l)k}, y_{ij}^{(l)k}\right]^\top$, where the subscript ij represents the pixel index j in the i-th image patch. The center point $\mathbf{x}_{ic}^{(l)k}$ of the i-th image patch is the detected keypoint in the feature-based VO. The 3D points $\mathbf{X}_{ij}^{(l)k} = \left[X_{ij}^{(l)k}, Y_{ij}^{(l)k}, Z_{ij}^{(l)k}\right]^\top$ expressed in left camera coordinates $\{C^k\}$ are mapped to pixel coordinates $\mathbf{x}_{ij}^{(l)k}$ through the camera projection function $\pi : \mathbb{R}^3 \mapsto \mathbb{R}^2$:

$$\mathbf{x}_{ij}^{(l)k} = \pi\left(\mathbf{X}_{ij}^{(l)k}\right) = \begin{bmatrix} \frac{f \cdot X_{ij}^{(l)k}}{Z_{ij}^{(l)k}} + p_x \\ \frac{f \cdot Y_{ij}^{(l)k}}{Z_{ij}^{(l)k}} + p_y \end{bmatrix} \tag{5.1}$$

where f, p_x, p_y are the intrinsic calibration parameters of the rectified images. Conversely, we can compute a 3D point $\mathbf{X}_{ij}^{(l)k}$ with the depth value $Z_{ij}^{(l)k}$ and $\mathbf{x}_{ij}^{(l)k}$ through the inverse projection function $\pi^{-1} : \mathbb{R}^2 \mapsto \mathbb{R}^3$:

$$\mathbf{X}_{ij}^{(l)k} = \pi^{-1}\left(\mathbf{x}_{ij}^{(l)k}, Z_{ij}^{(l)k}\right) = \begin{bmatrix} \frac{x_{ij}^{(l)k} - p_x}{f} Z_{ij}^{(l)k} \\ \frac{y_{ij}^{(l)k} - p_y}{f} Z_{ij}^{(l)k} \\ Z_{ij}^{(l)k} \end{bmatrix} \tag{5.2}$$

For the proposed direct method, we compute a local dense depth map for each keyframe. We model the relative motion of the left camera frame between $\left\{C^k\right\}$ at time step k and $\{C^*\}$ at keyframe as rigid body transformation $T_{k,*} \in SE(3)$:

$$\tilde{\mathbf{X}}^{(l)k} = T_{k,*}\tilde{\mathbf{X}}^{(l)*} \tag{5.3}$$

where $\tilde{\mathbf{X}}^{(l)k} = [\mathbf{X}^{(l)k^\top}, 1]^\top$ is the homogeneous form of $\mathbf{X}^{(l)k}$. A minimal representation of Lie

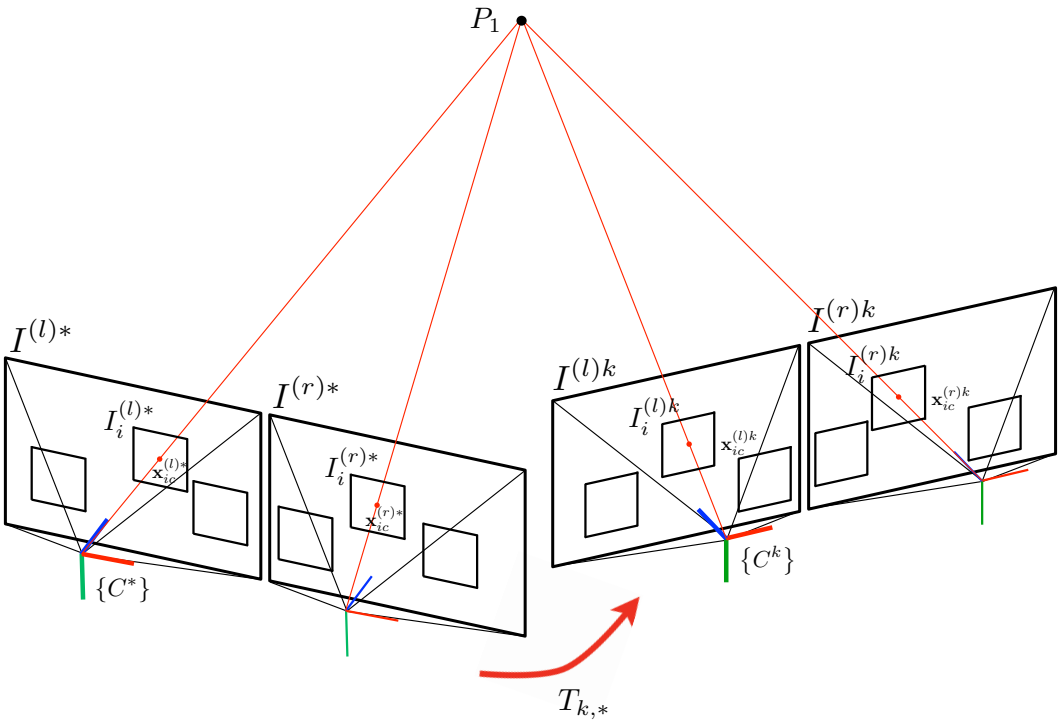


Figure 5.2: Stereo camera model and image coordinate systems.

group $SE(3)$, i.e. Lie algebra $se(3)$ parameter $\boldsymbol{\xi}$, is used to represent the incremental displacements during the numerical nonlinear optimization. We denote the Lie algebra with a 6×1 vector $\boldsymbol{\xi} = \left[\boldsymbol{\nu}^\top, \boldsymbol{\omega}^\top\right]^\top$ where $\boldsymbol{\nu}$ and $\boldsymbol{\omega}$ are infinitesimal translation and rotation in the tangent space of the matrix group $SE(3)$. The exponential map between the Lie algebra $se(3)$ and the rigid body transformation $T \in SE(3)$ can be written as follows:

$$T(\boldsymbol{\xi}) = \exp(\hat{\xi}) \tag{5.4}$$

where $\hat{\xi}$ is a 4×4 twist matrix from the Lie algebra $\boldsymbol{\xi}$ [36]. The above-defined notations and equations are illustrated in Fig. 5.2.

The problem we want to solve is to estimate the relative motion of the stereo camera $T_{k,*}$ given a sequence of image frames and the corresponding depth maps under arbitrary, abrupt, and partial illumination changes between the time step k and keyframe.

5.4 System Overview

Figure 5.3 provides an overview of the proposed stereo VO. The proposed method has two main steps: 1) feature-based VO for estimating initial camera pose as a motion prior; and 2) illumination-robust direct VO for refining the camera pose and photometric parameters to achieve higher accuracy. This sequential VO allows stable and accurate 6-DoF camera tracking in light-changing environments. We obtain the overall trajectory by concatenating the frame-to-keyframe motion estimation illustrated in Fig. 5.4.

Our feature-based VO method is largely based on [9]. We detect the salient feature points and obtain feature correspondence. With the matched features, we estimate a camera pose that minimizes the sum of the squared left and right reprojection error using three randomly selected correspondences in a RANSAC scheme (for full details, refer to [9]). If the feature-based VO prior fails due to low textured areas or light-changing environments, the proposed method performs the next direct VO approach without motion prior information.

For considering both global and local illumination changes in an image, we generate image patches around the matched feature points used in the feature-based VO. We initialize the camera pose from the feature-based estimation and the model parameters of individual patches following the affine illumination model [8]. We refine the camera pose and photometric parameters by minimizing the newly proposed photometric error, which is based on the modified photo-consistency assumption explained in Section 5.5.2.1, for compensation of illumination changes between image frames.

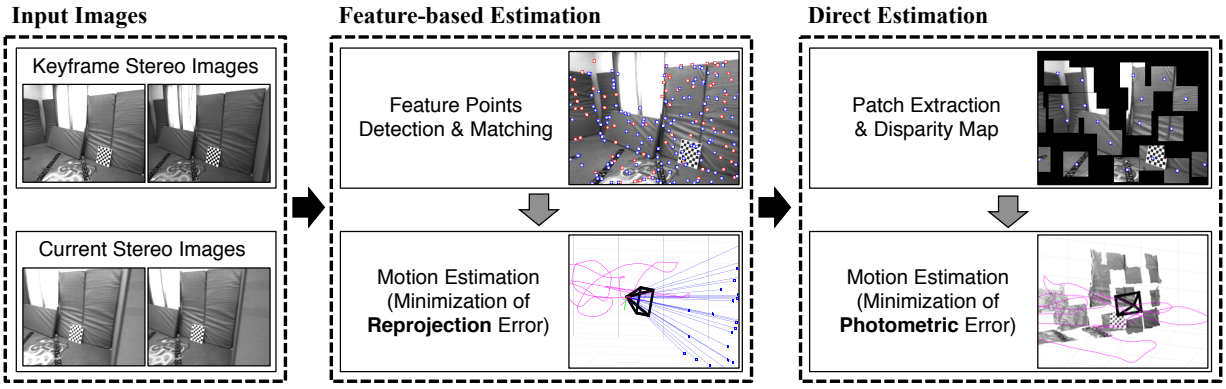


Figure 5.3: Overview of the proposed stereo visual odometry pipeline.

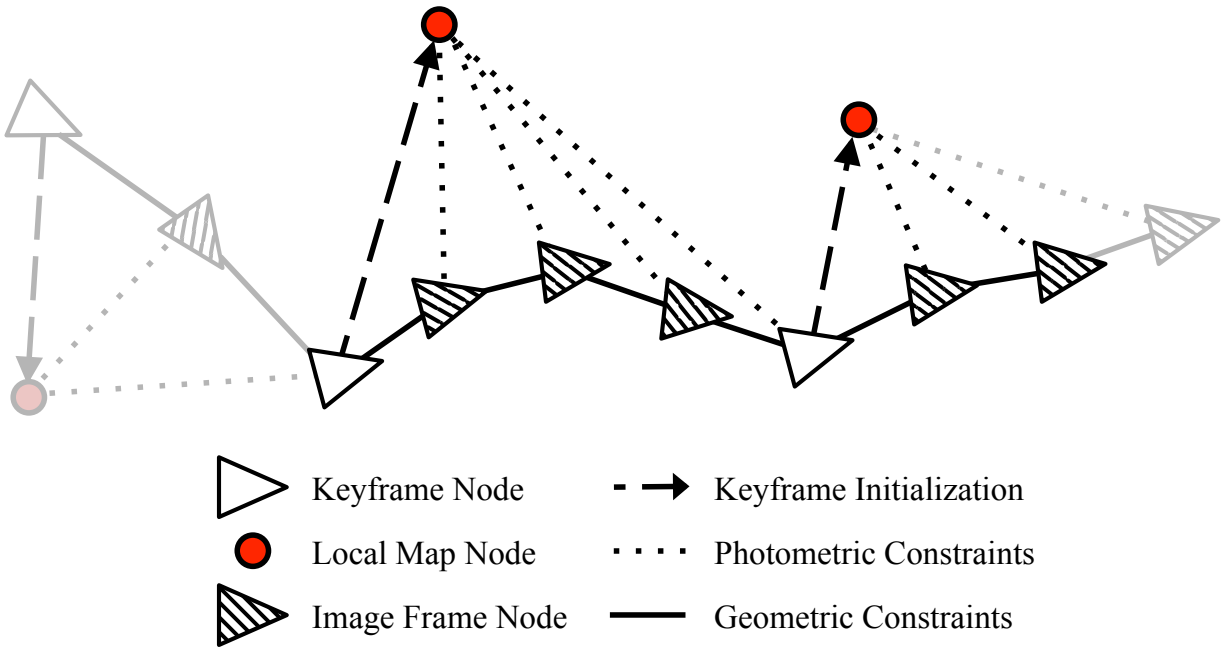


Figure 5.4: Topological representation of the proposed algorithm.

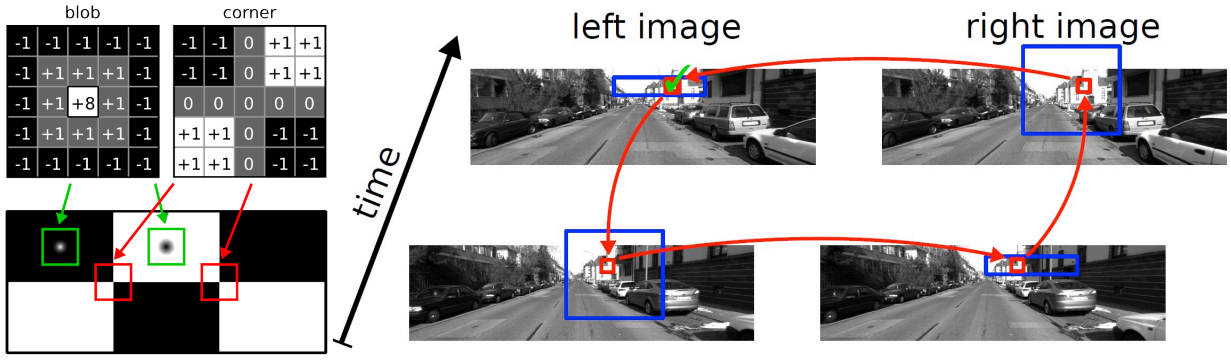


Figure 5.5: Kernels for feature detection and circular matching strategy. (figures courtesy of Andreas Geiger)

5.5 Visual Odometry Pipeline

5.5.1 Feature-based Estimation

5.5.1.1 Feature Detection & Matching

We detect the feature points by filtering the left and right images of the two consecutive image frames with the 5×5 kernels for finding blobs and corners as shown in the left of Fig. 5.5 (for full details, refer to [9]). We apply non-maximum and non-minimum suppression to the filtered four images for extracting the feature candidates.

The distribution of u- and v-directional image gradient around the feature candidates is employed as a descriptor for feature matching. To measure the similarity between the descriptors, we use the sum of absolute differences (SAD). We solve the feature correspondences by matching features with the two temporally consecutive stereo pairs and perform circular matching illustrated in the right of Fig. 5.5. We find the best match between the current and previous images within an 11×11 search window. When matching between the left and right images, we additionally utilize the epipolar constraint.

5.5.1.2 Reprojection Error Minimization

We compute the relative motion of the stereo camera $T_{k,k-1}(\boldsymbol{\xi})$ by minimizing the sum of squared left and right reprojection error given the matched feature points. The reprojection error of the matched features in the left camera can be written as follows:

$$r_i^{(l)}(\boldsymbol{\xi}) = \left\| \mathbf{x}_{ic}^{(l)k} - w^{(l)}(\boldsymbol{\xi}, \mathbf{x}_{ic}^{(l)k-1}) \right\| \tag{5.5}$$

$$w^{(l)}\big(\boldsymbol{\xi}, \mathbf{x}_{ic}^{(l)k-1}\big) = \pi^{(l)}\big(T_{k,k-1}\big(\boldsymbol{\xi}\big) \cdot \pi^{(l)-1}\big(\mathbf{x}_{ic}^{(l)k-1}, Z_i^{(l)k-1}\big)\big)\big) \tag{5.6}$$

where $\boldsymbol{\xi} \in \mathbb{R}^6$ represents the relative motion of the stereo camera and $w^{(l)}(\boldsymbol{\xi}, \mathbf{x}_{ic}^{(l)k-1})$ is the warping function of the left camera, which maps a center point of i-th patch $\mathbf{x}_{ic}^{(l)k-1}$ in the previous left image to its pixel coordinate in the current left image frame given the relative camera motion $\boldsymbol{\xi}$. The reprojection error of the features in the right camera can also be written in the same way with the superscript (r) instead of (l) in Eqs. (5.5) and (5.6). The objective energy function in the feature-based estimation is the sum of squared left and right reprojection error as follows:

$$\boldsymbol{\xi}^* = \arg\min_{\boldsymbol{\xi}} \sum_{i=1}^{N} \left[\left(r_i^{(l)}(\boldsymbol{\xi}) \right)^2 + \left(r_i^{(r)}(\boldsymbol{\xi}) \right)^2 \right] \tag{5.7}$$

where N is the number of the matched feature points. We use the relative motion of the camera $\boldsymbol{\xi}$ as a RANSAC model to reject outliers in feature matches. Given all inlier features from the RANSAC, we can obtain the optimal relative motion of the camera with the Gauss-Newton method for solving Eq. (5.7).

5.5.2 Direct Estimation

5.5.2.1 Affine Illumination Change Model

The traditional photo-consistency assumption commonly used in direct visual odometry [6, 24] denotes that the same 3D points should have the same intensity values across multiple images.

Unfortunately, this assumption almost never holds in real-world applications because light variations take place frequently. Thus, we employ the modified photo-consistency assumption which can make up for not only the global but also the local illumination changes between the current and keyframe time steps, proposed in [8]:

$$\lambda_i I_i^{(l)k} + \delta_i = I_i^{(l)*} \tag{5.8}$$

where λ_i and δ_i denote the photometric parameters for explaining contrast and brightness change of the i-th patch in the left image, which will have values close to one and zero, respectively in the normal environments without obvious illumination changes. We generate the image patches around the matched feature points in the feature-based VO, and select them with the planarity test [8], which determines whether the selected patches are on the plane in the 3D space or not because 3D points on the same plane undergo the similar illumination changes (for full details, refer to [8]). Patch size is one of the user-defined custom parameters, and we create the patches with a size of 91×91 pixels. We utilize at most 16 patches spread uniformly across the entire image if there is not enough number of the planar patches. We can compensate both global and local illumination changes because each patch can have different photometric parameters.

5.5.2.2 Modified Photometric Error Minimization

We simultaneously estimate the camera pose $T_{k,*}(\boldsymbol{\xi})$ and the photometric parameters per patch (e.g., $\{\lambda_1, \delta_1\}, ..., \{\lambda_m, \delta_m\}$ where m is the number of patches) by minimizing the sum of squared modified photometric error. The modified photometric error of the j-th pixel in the i-th patch can be written as follows:

$$r_{ij}(\mathbf{z}) = \lambda_i I_i^{(l)k}(w^{(l)}(\boldsymbol{\xi}, \mathbf{x}_{ij}^{(l)*})) + \delta_i - I_i^{(l)*}(\mathbf{x}_{ij}^{(l)*}) \tag{5.9}$$

$$\mathbf{z} := \left[\boldsymbol{\xi}^\top, \lambda_1, \delta_1, ..., \lambda_m, \delta_m\right]^\top \in \mathbb{R}^{6+2m} \tag{5.10}$$

where $\mathbf{z}$ is the integrated model parameter consisting of the relative motion of the camera and the photometric parameters per patch for the sake of simplicity. To perform the warping in Eq. (5.9), we generate the local map of the keyframe consisting of the dense depth map from [92] and the brightness information from the left keyframe image. The optimal model parameter $\mathbf{z}^*$ which minimizes the weighted sum of squared modified photometric error can be obtained by solving the following non-linear weighted least square problem:

$$\mathbf{z}^* = \arg\min_{\mathbf{z}} \sum_{i=1}^{m} \sum_{j=1}^{n} W(r_{ij}) r_{ij}^2(\mathbf{z}) \tag{5.11}$$

$$W_{\mathrm{t}}(r_{ij}) = \frac{\nu + 1}{\nu + \left(\frac{r_{ij}}{\sigma_{\mathrm{t}}}\right)^2}$$

where n is the number of pixels in each patch and $W(r_{ij})$ is the weighting function from the student t-distribution in order to achieve the robustness against outliers caused by occlusions, dynamic objects, and sensor noise [38]. Among the various weight functions like Tukey and Huber, we employ the student t-distribution for its effectiveness in the direct method (for full details, refer to [38]). We use the Gauss-Newton algorithm for solving the iteratively re-weighted nonlinear least square (IRLS) problem in Eq. (5.11). We compute the Jacobian matrix with the efficient second-order minimization (ESM) method [59] because it outperforms the other methods such as the forward compositional (FC) and inverse compositional (IC) approaches [7, 21]. We employ a coarse-to-fine approach with the image pyramid method for robustness and faster convergence. Note that there should be enough valid pixels in each image patch for accurate and stable correction of light changes.

5.5.3 Discussion

The motion prior from the feature-based VO (Section 5.5.1) for the proposed direct VO (Section 5.5.2) seems to be unnecessary and redundant, but we carefully design the proposed stereo visual odometry to solve two critical issues in the direct VO under light-changing environments:

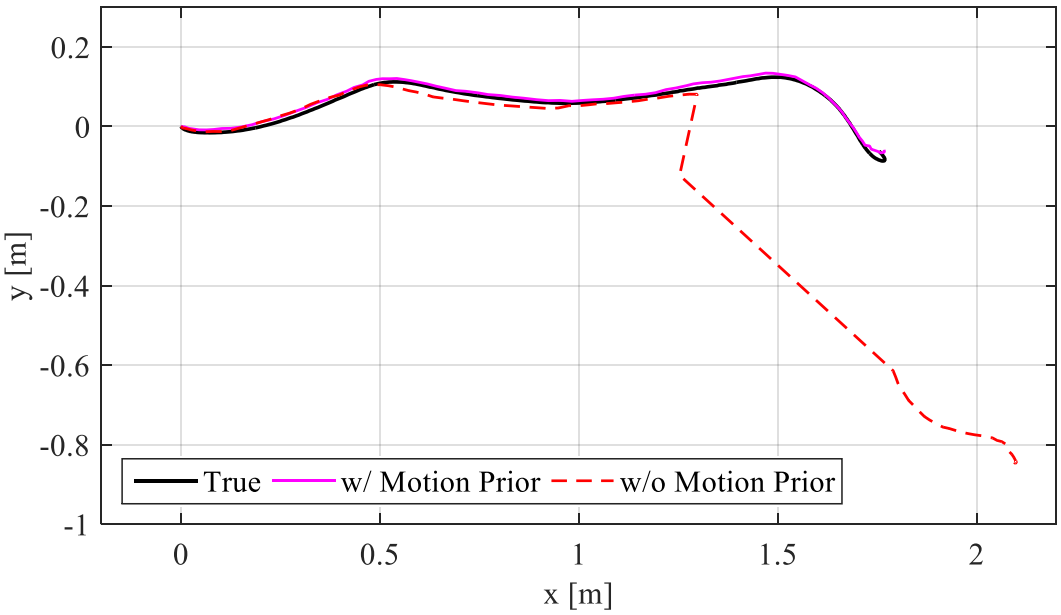


Figure 5.6: A motion prior from feature-based VO stabilizes the direct VO significantly.

more stable motion estimation and higher accuracy.

5.5.3.1 Stable Motion Estimation

We occasionally observe in the direct VO that a light-changing environment can lead to "jumps" in the motion estimate [6, 25]. Due to the nature of VO, jumps have a large impact on the estimated trajectories because VO drift continues to accumulate. We solve this problem by using a motion prior from the feature-based VO whose computation is less intensive than the direct method [24] and does not cause noticeable increases in the overall computational time as shown in the evaluation section later, resulting in stable motion estimation as illustrated in Fig. 5.6.

5.5.3.2 High Accuracy

It is well-known that a good initial pose can be very helpful for the direct VO during the nonlinear optimization. The presence of the motion prior knowledge improves not only stability, but also accuracy as shown in Fig. 5.7. Although both the proposed method and direct only method [8] employ the same nonlinear optimization formulation in Eq. (5.11), we can obtain the more accu-

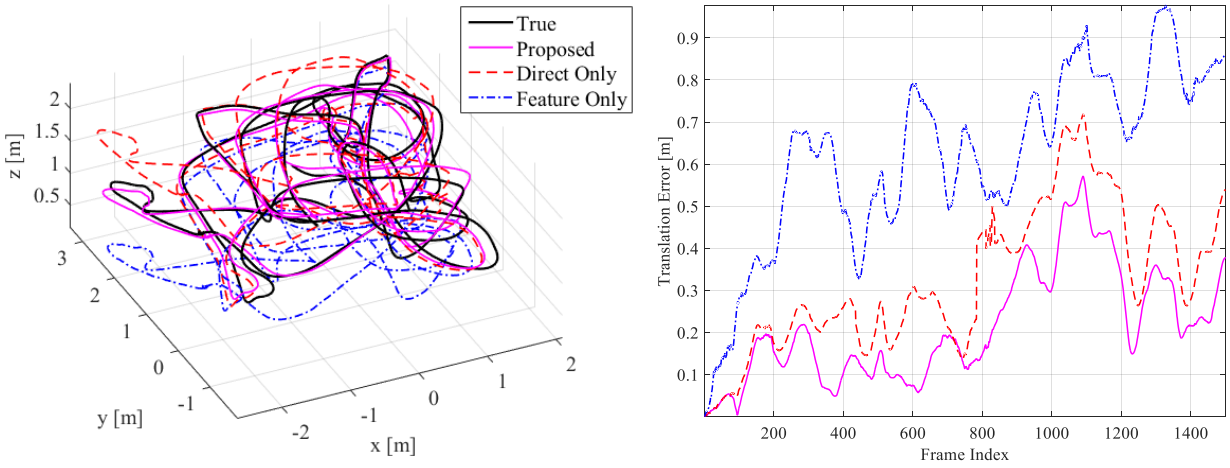


Figure 5.7: Our sequential VO shows the best accuracy among other direct only [8] or feature only VO methods [9].

rate camera pose with the proposed algorithm thanks to the motion prior.

5.6 Energy Function Analysis

We analyze the linearity of objective energy function and convergence rate used in feature-based and direct VO. These analyses theoretically support the usage of the feature-based VO as the motion prior.

5.6.1 Energy Function Convergence

The objective energy function (cost function) of the feature-based and direct estimation with respect to the camera pose can be written as follows:

$$E_f(\boldsymbol{\xi}) = \frac{1}{N}\sum_{i=1}^{N}\left[\left(r_i^{(l)}(\boldsymbol{\xi})\right)^2 + \left(r_i^{(r)}(\boldsymbol{\xi})\right)^2\right] \tag{5.12}$$

$$E_d(\boldsymbol{\xi}) = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} W(r_{ij}) r_{ij}^2(\boldsymbol{\xi}) \tag{5.13}$$

where the subscripts f and d denote the feature-based and direct estimation, respectively.

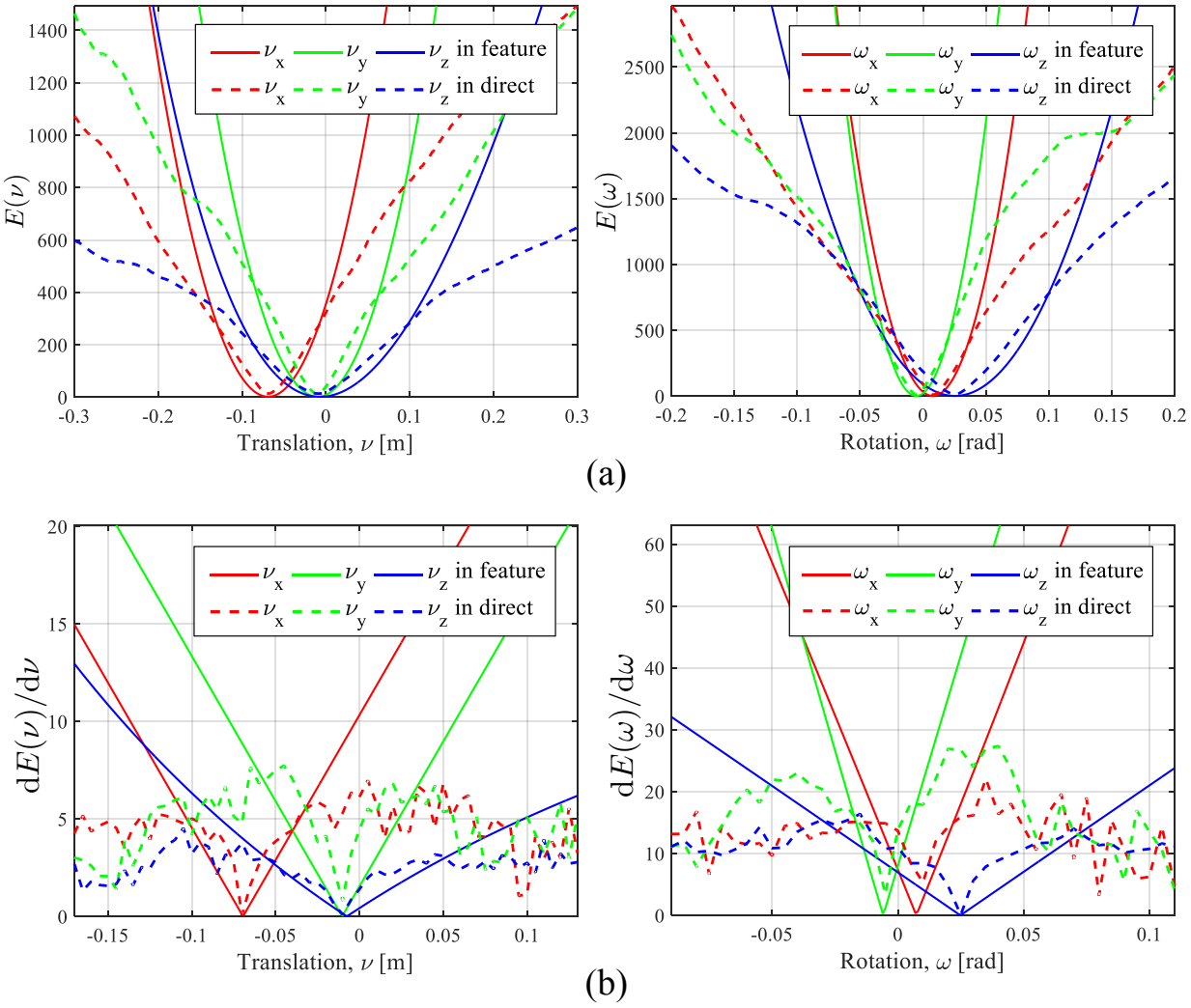


Figure 5.8: Tendency of (a) the reprojection and photometric error for transformation with respect to each translational and rotational direction, and (b) the first derivatives of the (a).

For comparing the convergence rate of each cost function, we plot the average reprojection and photometric error in Eqs. (5.12) and (5.13) with respect to the 6-DoF camera pose in the vicinity of the true camera pose in Fig. 5.8 (a). The range of translation and rotation error is in ± 0.3 m and ± 0.2 radian, respectively. We compute them by warping the feature points and images separately along each degree of freedom in the 6-DoF camera pose (i.e., the Lie algebra parameter $\boldsymbol{\xi} = [\nu_x, \nu_y, \nu_z, \omega_x, \omega_y, \omega_z]^\top$) while the other parameters in $\boldsymbol{\xi}$ are fixed to the true camera pose. Although the energy functions are highly nonlinear, both error plots have the distinct minimum values at similar places for each axis in (a).

In Fig. 5.8 (b), which shows the derivatives of the two error plots, a notable difference exists between the feature-based and direct estimation. When the camera pose is far from the true camera pose, the slope of error plots of the feature-based method is very steep compared to the direct method. Therefore, the farther the currently estimated camera pose is from the true camera pose, the faster the estimated camera pose approaches the distinct minimum, especially when the feature-based method is applied instead of the direct method. In the vicinity of the valley, however, the slope of the feature-based estimation error flattens gradually as the camera pose approaches the true camera pose. The direct method can be more efficient and accurate than the feature-based method especially when we start the optimization close to the true camera pose.

Fig. 5.9 shows the convergence property of the feature-based, direct, and the proposed VO approaches with respect to the 6-DoF camera pose. The numbers represent the error distance of each moment, and the subscript f and d denote the feature-based and direct estimation method, respectively. We write down the error value only for the x-axis (red) in the translation (left) and z-axis (blue) in the rotation (right) for readability. Each square or cross mark refers to the updated 6-DoF camera motion at each iteration during the optimization. In the feature-based estimation, the estimated camera motion rapidly converges to the neighborhood of the true camera pose within only two iterations, and the optimization stops in six iterations. In contrast, the direct estimation converges slowly near the true camera pose, and the total iteration in the optimization is 35, which is about 6 times more than the feature-based method. Although the direct method is slower especially from a distance, the finally estimated camera motion is a little closer to the true

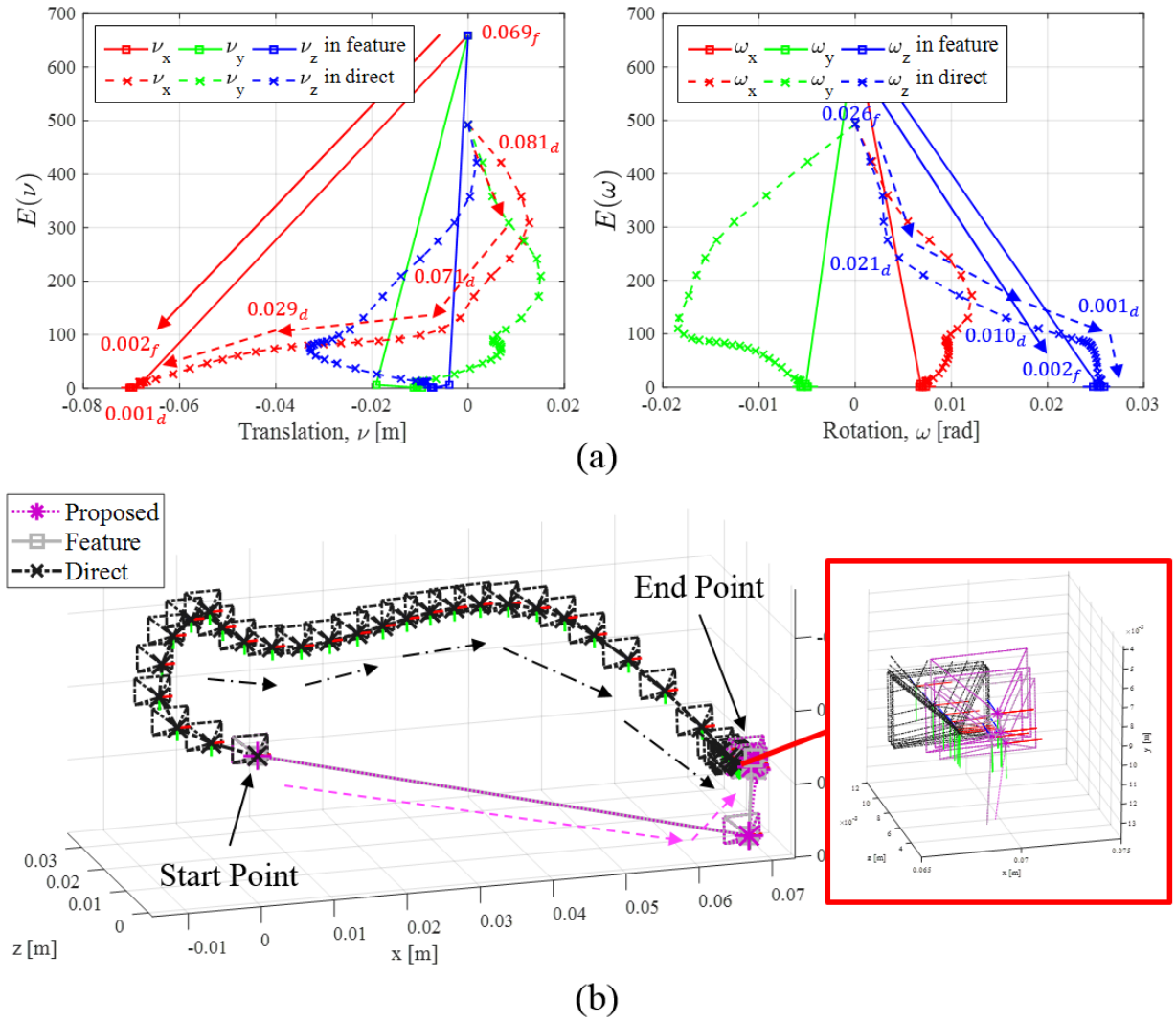


Figure 5.9: Convergence history of the feature-based and direct estimation with the error.

camera pose than the feature-based method due to the characteristics of this gradual approach. The final translation error of the feature-based and direct estimation is 4.6 mm and 3.7 mm, respectively.

We further analyze the convergence property of the proposed method compared to the two kinds of cost functions in Fig. 5.9 (b). The proposed method first jumps off the starting point (the origin) into the end point (true camera pose) very closely by performing a feature-based VO method. After that, with a motion prior from the feature-based VO, our two-stage method

performs the direct estimation to get closer to the true camera pose. The enlarged figure on the right shows convergence to a similar 3D point from the direct only and the proposed method. The total number of iteration in the direct only method (black) during the optimization is 35 whereas the proposed two-stage method takes six iterations from the feature-based and three iterations from the direct method, showing the effectiveness of the proposed sequential method design.

5.6.2 Measurement Equation Linearity

We validate the better convergence property of the feature-based method by analyzing the dimensionless linearity index [13], which represents the degree of linearity in the nonlinear measurement equation. The more linear the measurement equation is for the 6-DoF camera motion, the faster the nonlinear optimization converges. The dimensionless linearity index (DLI) of each measurement equation considering the Lie group $SE(3)$ can be written as follows:

$$L = \left| \frac{\left.\frac{\partial^2 h}{\partial \boldsymbol{\xi}^2}\right|_{\boldsymbol{\xi}=\boldsymbol{\xi}_0} \Delta\boldsymbol{\xi}}{\left.\frac{\partial h}{\partial \boldsymbol{\xi}}\right|_{\boldsymbol{\xi}=\boldsymbol{\xi}_0}} \right| \tag{5.14}$$

$$h_x(\boldsymbol{\xi}) = \left[\pi(T(\boldsymbol{\xi}) \cdot \pi^{-1}(\mathbf{x}, Z(\mathbf{x})))\right]_x$$

$$h_I(\boldsymbol{\xi}) = I(\pi(T(\boldsymbol{\xi}) \cdot \pi^{-1}(\mathbf{x}, Z(\mathbf{x}))))$$

where h in Eq. (5.14) denotes the observation model that h_x is the x component of the warping function in the feature-based method, and h_I is the image intensity observation model in the direct method. We omit the y component of the warping function, h_y, because its linearity index results are symmetric to the h_x. $\boldsymbol{\xi}_0$ is the center point of camera motion used in the first and second derivative, and $\Delta\boldsymbol{\xi} = \left[\Delta\nu^\top, \Delta\omega^\top\right]^\top$ is the transformation of 6-DoF camera motion from the center point $\boldsymbol{\xi}_0$. When $L \approx 0$, the observation model can be considered as a linear model in the interval $\Delta\boldsymbol{\xi}$, and vice versa. Unlike [13], we newly derive and calculate the DLI of each observation model with respect to the 6-DoF camera motion with the 3D projection model. The detailed derivation of the DLI and explanations of each component are provided in the Appendix.

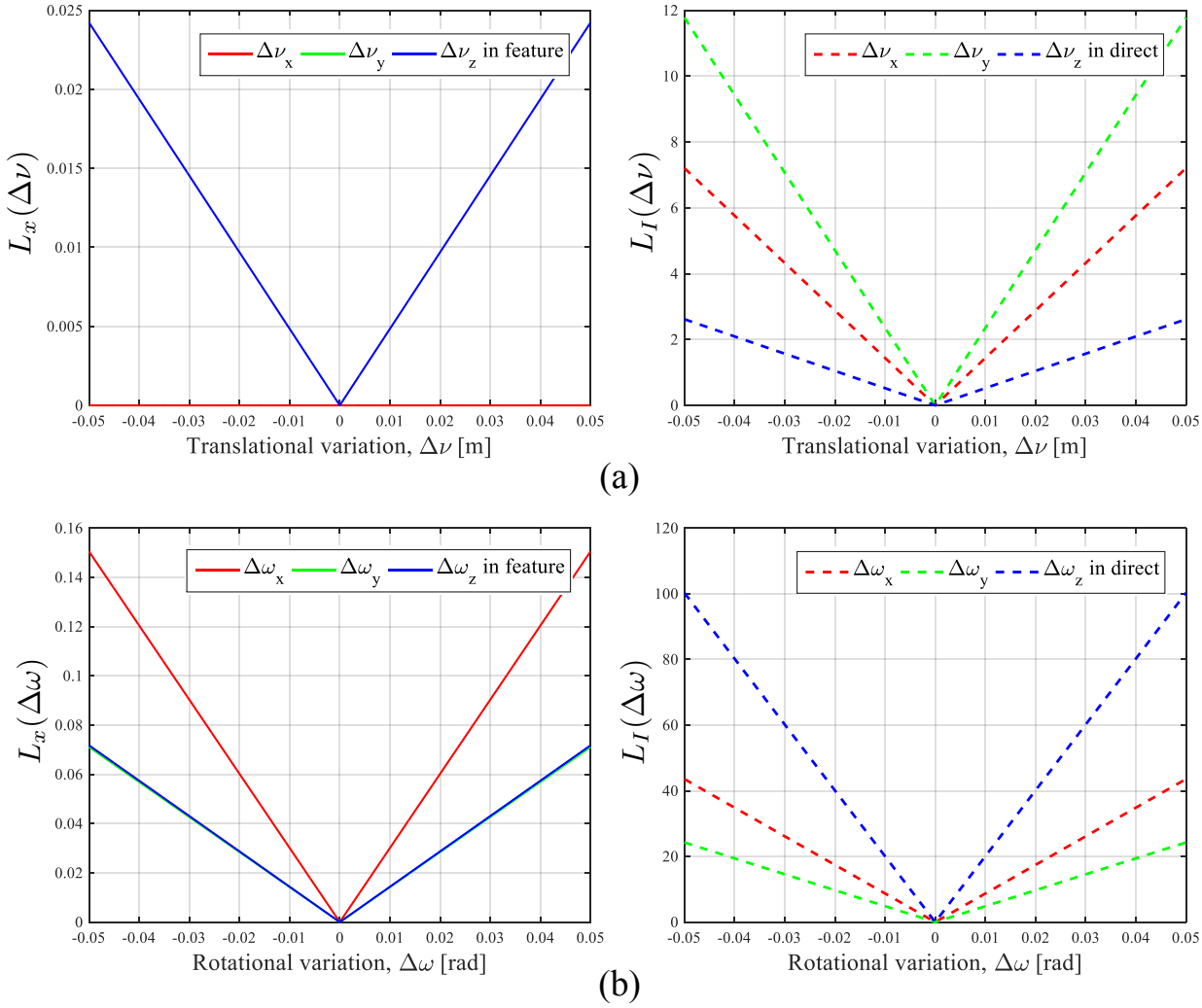


Figure 5.10: Dimensionless linearity index of the feature-based and direct estimation with respect to the (a) translational and (b) rotational motion of the camera.

We plot the DLI of each estimation method with respect to the translational and rotational transformation from the true camera pose in Fig. 5.10. The transformation of the translation and rotation from the true camera pose $\Delta\boldsymbol{\xi} = \left[\Delta\nu^\top, \Delta\omega^\top\right]^\top$ is in the range of $\pm$0.05 m and $\pm$0.05 radian, respectively. The main factors, which cause nonlinearity in $L_x\left(\Delta\nu\right)$, are the depth of the feature point and the camera movement along the forward direction shown in Fig. 5.10 (a) on the left. In the results of the feature-based method, the DLI in terms of x component in translation is exactly zero, which means that the warping function is linear to the translational movement of the x-axis. Back and forth motion (z component in translational motion) of the camera is nonlinear, and the DLI becomes larger as the camera pose is away from the center point. The direct method, however, has a high degree of nonlinearity for translational motion in all directions, and it is approximately 100 times larger than the feature-based method. In particular, the image intensity function $I(\cdot)$, which maps from the pixel coordinates to the image intensity in Eq. (5.14), causes such a severe nonlinearity.

Fig. 5.10 (b) shows the similar behavior of the DLI along the rotational motion of the camera. Both estimation methods have the nonlinearity for rotational motion in all directions, and it gets larger as the camera motion drifts farther away from the true camera pose. But the direct method has approximately 100 times more severe nonlinearity than the feature-based method.

In conclusion, the feature-based method, which minimizes the reprojection error in Eq. (5.7), converges more quickly to the vicinity of the true camera pose than the direct method as shown in Fig. 5.9 thanks to the high degree of linearity. But in the proximity of the true camera pose where the effect of nonlinearity is negligible, the direct method ultimately can approach the true camera pose a little closer. These analyses confirm the effectiveness of our decision to use the feature-based VO as the motion prior, followed by the proposed illumination-robust direct VO.

5.7 Experimental Results

We extensively evaluate the effectiveness of the proposed illumination-robust stereo visual odometry with two different experiments. We test the accuracy of the proposed algorithm under the

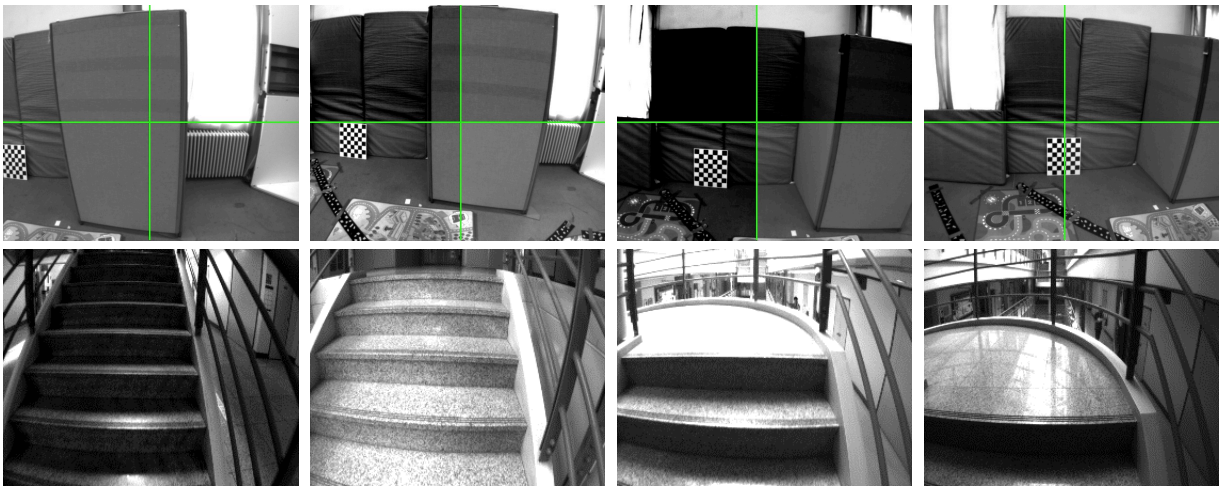


Figure 5.11: Extracts from the synthetic EuRoC dataset and author-collected dataset.

light-changing environments with the stereo image datasets that include irregular illumination changes. The other VO baselines are applied to these datasets to compare the performance of the motion estimation. Next, we construct a 3-D autonomous flight system with the proposed algorithm for online use. We perform flight experiments with the aerial robot to test the proposed algorithm in terms of accuracy and robustness to light variations, showing the short-term autonomous flight capability.

5.7.1 Experiments on the Datasets

We test the proposed algorithm on the synthetic EuRoC benchmark [91] and our own dataset that contains illumination changes through the stereo image sequences. We apply artificial illumination changes to RGB images in the EuRoC benchmark to validate the proposed algorithm under light-changing environments. We collect the stereo image datasets including actual light variations for evaluating the consistency of the proposed algorithm. Since many VO baselines accept only RGB-D input, we convert the stereo camera data to match the desired input format.

We compare the proposed VO method against other VO algorithms: the Dense Visual Odometry (DVO) [6], the Efficient DVO (EDVO) [7], and the depth enhanced monocular odometry (DEMO) [17]. DVO estimates the camera pose by minimizing the photometric error within the

overall images based on the photo-consistency assumption. DVO is a direct VO without the affine illumination change model. EDVO is an advanced direct tracking method which performs per-image brightness correction by considering a global affine illumination. Thus, EDVO estimates the photometric parameters per image whereas the proposed algorithm estimates them per patch. DEMO is one of the state-of-the-art feature-based VO algorithms, which estimates the motion of the camera by utilizing features with and without depth. The proposed method is implemented in Matlab/C++ and runs on a desktop computer with Intel Core i5 3.2 GHz and 8GB memory.

5.7.1.1 Synthetic EuRoC Datasets

The EuRoC micro aerial vehicle (MAV) datasets [91] consist of the stereo image pairs at 20 Hz mounted on an AscTec Firefly MAV and a ground-truth position from a motion capture system at 100 Hz. To test the robustness against abrupt and local lighting changes, we modify the intensity values in the stereo images in the four quadrants based on the affine illumination model in the Eq. (5.8), rather than the same affine illumination in the entire image. Each zone in the four quadrants follows a different affine model to simulate uneven lighting changes in the entire image. We call these stereo image sequences containing the artificial illumination changes as the synthetic EuRoC dataset and extracts are shown in the first row of Fig. 5.11.

To measure the accuracy of the proposed and other VO algorithms quantitatively, three types of error metrics are selected: root mean square error (RMSE) of the relative pose error (RPE), absolute trajectory error (ATE) in [2], and the final drift error divided by the total traveling distance of a recording platform.

We first evaluate the proposed two-stage design compared to feature only [9] and direct only [8] tracking methods while keeping other parts unchanged in Table 5.1. We measure the root mean squared error (RMSE) of the relative pose error, and present the improved accuracy of the proposed method. Thanks to a good initial pose from the feature-based approach, our direct VO can achieve better performance in terms of RPE value. In particular, the proposed method with motion prior shows more accurate results than the direct only method by preventing it from jumping in the motion estimates. The average RMSE of RPE is 0.151 drift m/s for the proposed

Table 5.1: Accuracy Improvement of the Proposed Algorithm

Experiment	Proposed	Feature Only	Direct Only	Length (m)
Vicon Room 1 01	**0.050**	0.061	0.055	57.97
Vicon Room 1 02	**0.050**	0.062	0.105	74.28
Vicon Room 1 03	0.609	**0.575**	1.017	78.70
Machine Hall 01	**0.014**	0.026	0.018	67.53
Machine Hall 02	**0.013**	0.026	0.028	63.00
Machine Hall 03	0.266	0.240	**0.062**	126.87
Machine Hall 04	**0.065**	0.100	1.172	88.39

Table 5.2: Experimental Results on Synthetic EuRoC Benchmark

Experiment	Relative Pose Error (m/s)				Absolute Trajectory Error (m)				Final Drift Error (%)				Length (m)	# of frame
	Proposed	DVO	EDVO	DEMO	Proposed	DVO	EDVO	DEMO	Proposed	DVO	EDVO	DEMO		
Vicon Room 1 01	**0.050**	0.166	0.059	0.136	**0.289**	1.906	0.668	1.653	**0.353**	3.863	1.543	3.359	57.97	2712
Vicon Room 1 02	**0.050**	0.356	0.088	0.294	**0.253**	6.474	0.975	2.528	**0.510**	9.911	1.714	4.309	74.28	1510
Vicon Room 1 03	**0.609**	1.592	0.771	0.862	6.583	14.346	7.879	**6.488**	7.895	21.400	11.393	**3.881**	78.70	1949
Vicon Room 2 01	**0.035**	0.371	0.089	0.091	**0.888**	5.450	1.475	1.202	4.550	**4.049**	7.300	4.421	35.85	2054
Vicon Room 2 02	**0.063**	0.140	0.079	0.284	1.285	1.485	**1.282**	3.635	2.433	**1.374**	2.300	7.229	83.34	2201
Vicon Room 2 03	**0.359**	1.631	0.734	0.634	**5.086**	16.361	8.888	5.927	12.394	28.150	12.878	**10.014**	86.56	1806
Machine Hall 01	**0.014**	0.040	0.029	0.116	**0.310**	0.770	0.429	1.273	**0.582**	1.182	0.840	2.612	67.53	2582
Machine Hall 02	**0.013**	0.053	0.029	0.101	**0.205**	0.882	0.312	0.953	**0.308**	1.697	1.053	1.750	63.00	2140
Machine Hall 03	0.266	0.600	**0.053**	0.250	4.342	8.781	**1.077**	2.723	5.333	7.799	**1.548**	5.137	126.87	2200
Machine Hall 04	**0.065**	0.365	0.167	0.227	**1.117**	7.591	2.077	5.005	**1.499**	9.536	2.566	6.199	88.39	1533
Machine Hall 05	**0.040**	0.372	0.101	0.147	**0.725**	8.913	0.992	3.260	**0.851**	9.952	1.365	3.468	93.97	1810

method, compared with 0.156, 0.351 drift m/s under feature only and direct only method, respectively.

We present the motion estimation results in Table 5.2. The smallest error for each dataset is bolded. The proposed algorithm shows better performance in terms of relative pose error for the synthetic EuRoC benchmark. The main reason for the improved results is that the proposed algorithm can cope with non-uniform light variations by integrating the affine illumination model per patch into the direct motion estimation. But other direct VO algorithms continue to perform

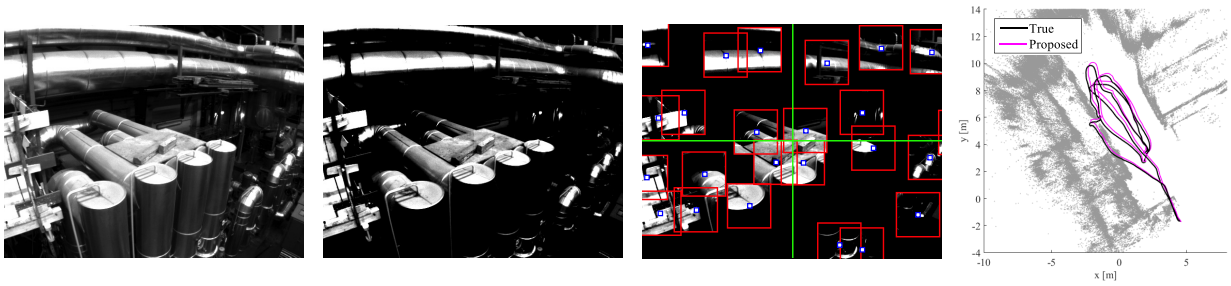


Figure 5.12: Red squares in (c) denote the image patches for compensating irregular illumination changes, and (d) shows the true and estimated trajectories.

the direct motion estimation under the photo-consistency assumption without considering such light changes. Excerpts from the Machine Hall 02, including synthetic image where irregular illumination changes occur and accurate motion tracking result, are shown in Fig. 5.12. Square patches distributed throughout the image marked red in Fig. 5.12 (c) help the proposed algorithm to cope with the irregular illumination changes present in the synthetic datasets.

In most cases, the DVO, which is a direct VO without the affine illumination model, greatly loses accuracy due to the light variations. EDVO, which compensates for the global illumination changes, shows good motion estimation results on some datasets: the Vicon Room 2 02 and Machine Hall 03. Since the EDVO performs per-image brightness correction, it cannot effectively deal with the partial light changes. On the other hand, the proposed method performs per-patch illumination correction, it can handle the irregular illumination changes, resulting in more accurate motion estimation results. If the features are well detected and tracked in the front-end, the DEMO is not sensitive to the light variations unlike the previous direct VO methods. DEMO presents better performance than the direct methods in terms of the final drift error on the Vicon Room 1 03 & 2 03, which have severe light variations. However, high drift error becomes more severe over time in most cases.

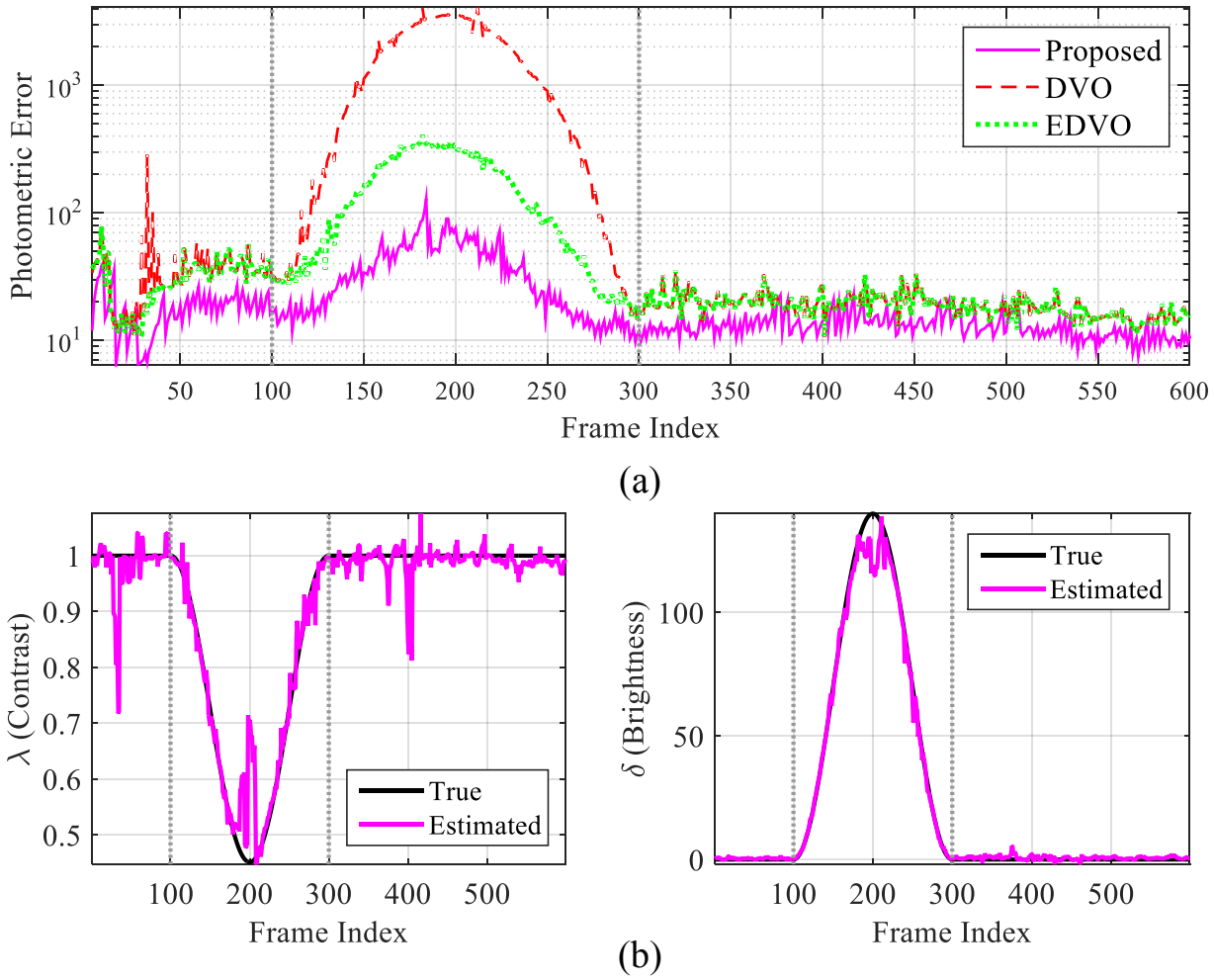


Figure 5.13: (a) The photometric error of the three direct VO methods is drawn in a logarithmic scale where photometric disturbances occur between the gray dotted lines. (b) The true and estimated photometric parameters with the proposed method overlap significantly.

The strength of the proposed algorithm becomes clear when analyzing the dataset Machine Hall 02 in detail. During the period from 100 to 300 image index where the irregular illumination changes occur, we can observe that the proposed method maintains the modified photometric error very small whereas the cost values of DVO and EDVO increase noticeably, which are reported in Fig. 5.13 (a). The main reason for this difference is that the photo-consistency assumption is severely violated in this period. Although a robust weighting function is employed for removing outliers in DVO or a global affine illumination model is considered in EDVO, these direct VO methods are not effective enough to take into account the sudden, partial lighting variations. The proposed algorithm efficiently handles this kind of illumination changes by using the proposed cost function in Eq. (5.11), resulting in accurate motion estimates as shown in Fig. 5.12 (d).

Fig. 5.13 (b) shows the true and estimated photometric parameters of a randomly selected patch. The proposed method estimates the photometric parameters for contrast and brightness changes correctly, which are used to compensate for the lighting changes in the synthetic EuRoC datasets as shown in Fig. 5.12 (b). Some jitters denote that our algorithm compensates for not only the artificial lighting changes we have made, but also the unmodeled and unpredictable light changes from sensor noise. Thanks to accurate estimation of the photometric parameters for each patch, the proposed method properly compensates the partial light variations during the direct motion estimation.

5.7.1.2 Author-collected Datasets

We want to demonstrate that the proposed algorithm works also well in the everyday indoor environments where the actual illumination changes occur due to various reasons such as sunlight entering through windows, automatic exposure control of the camera, etc. We collect stereo image datasets with a handheld VI sensor [93], capturing a multistoried stairway which includes actual and unknown illumination changes. Fig. 5.11 shows the example images where illumination changes are severe. For evaluating the consistency of the proposed and other VO baselines without the ground truth, we collect the stereo images along the carefully designed movements of the VI sensor in the stairway.

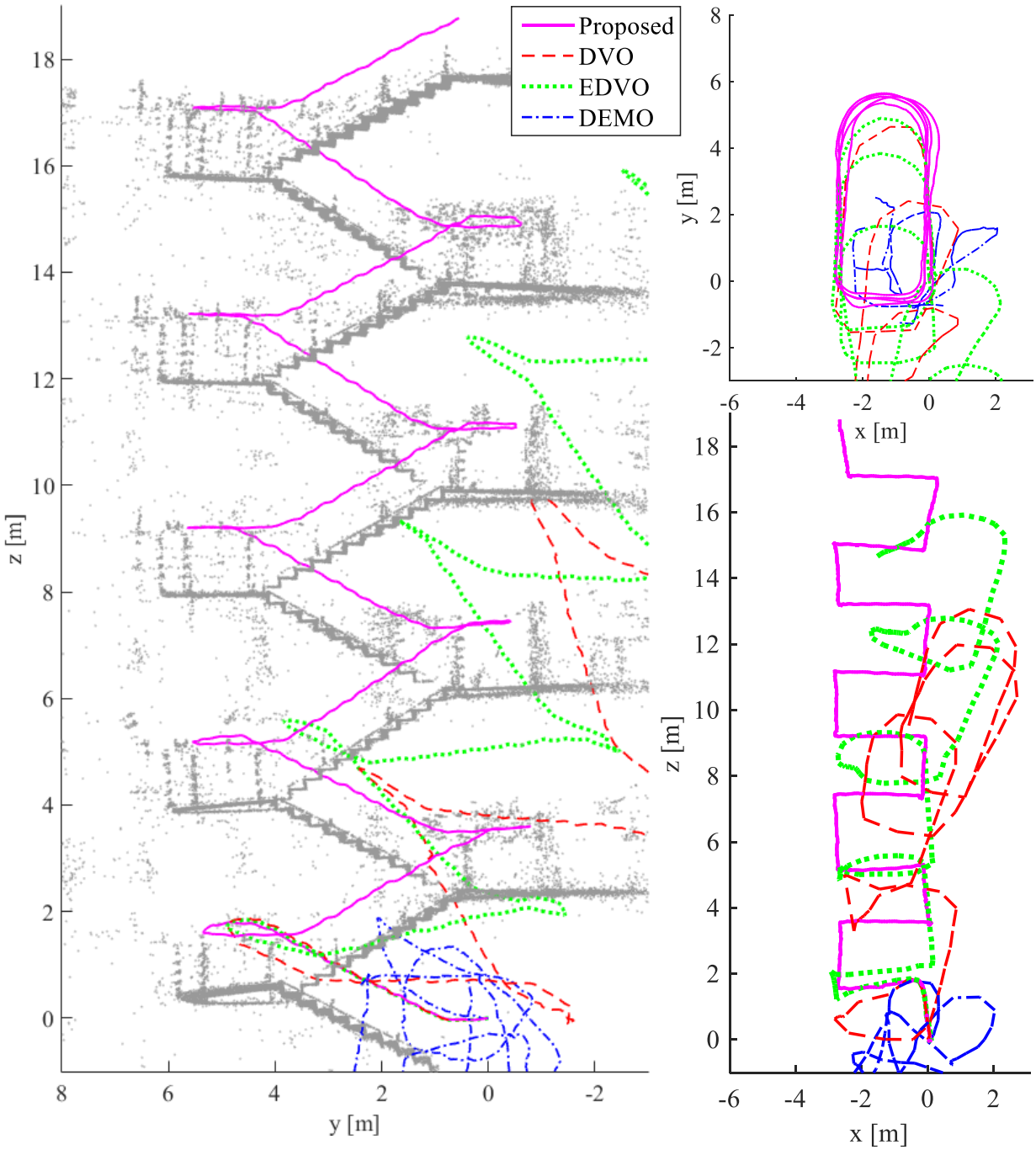


Figure 5.14: Comparison of the proposed and other VO methods on the multistoried stairway from the 1st to 6th floor.

In Fig. 5.14, the 80 m trajectory going up the stairs from the 1st to 6th floor of a building is visualized with three different views: top, front, and right side. The top view of the estimated trajectory shows the overlapped, consistent motion estimation result of the proposed method (magenta) while other estimated trajectories gradually diverge from the initially estimated loop. The side and front views of the stairway also support the high consistency of the proposed method compared with other VO methods.

5.7.2 Experiments on an Autonomous Aerial Robot

We build an aerial robot system capable of flying autonomously in a light-changing environment with only the onboard sensors and computer. In order to evaluate the accuracy of the autonomous flight when integrated with the proposed method, we perform trajectory tracking experiments. Through the flight experiments under actual light changes, we show the robustness and effectiveness of the proposed algorithm in the autonomous flight of the aerial robot. Although some research on the autonomous flight with VO exist [24], there has been no reported result in light-changing environments.

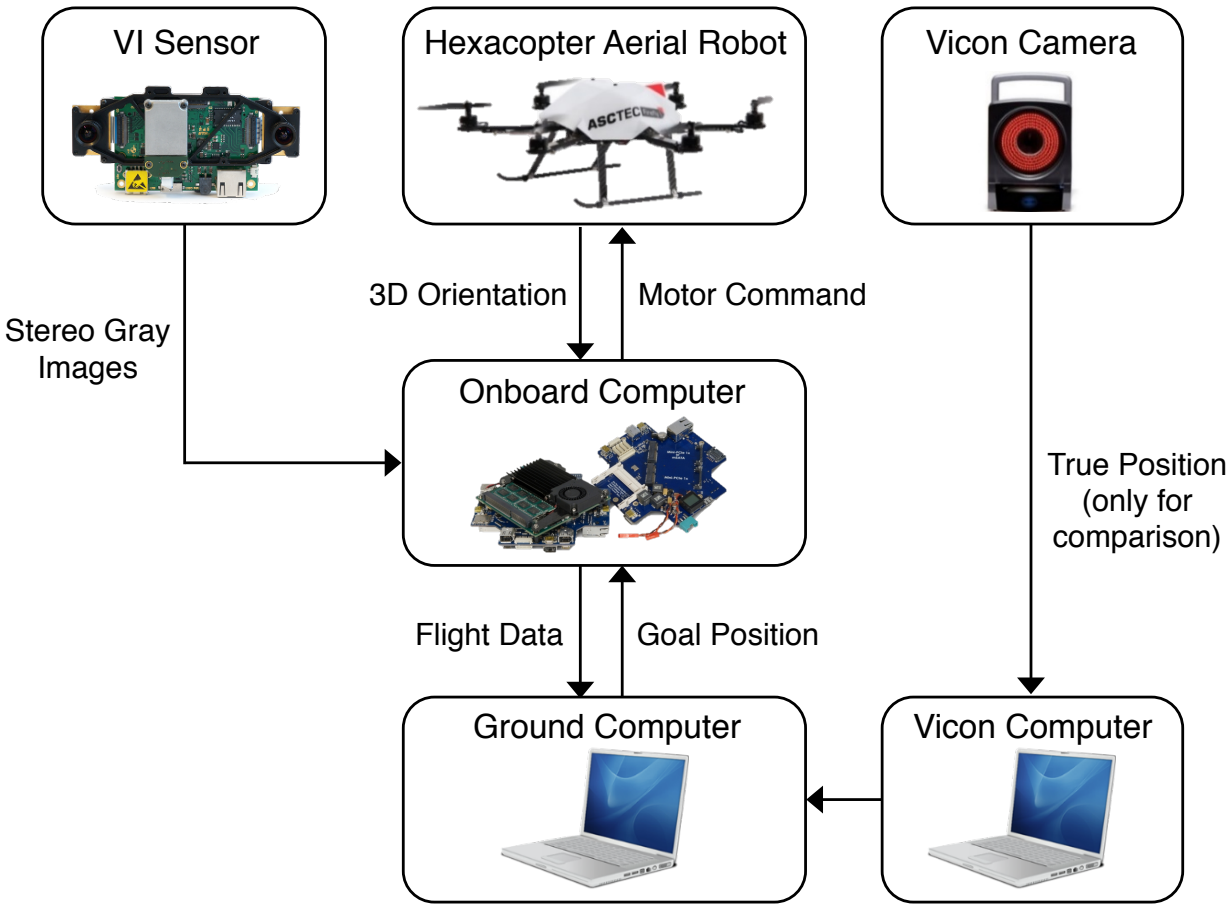


Figure 5.15: Schematic diagram of the data flow in our experimental setup.

5.7.2.1 Experimental Setup

We describe the hardware components of the aerial robot and our experimental setup as shown in Fig. 5.15. The VI sensor captures the stereo images at 752×480 pixel resolution at 20 Hz, and is mounted in a front, down-looking position of an AscTec Firefly aerial robot, equipped with an off-the-shelf inertial measurement unit (IMU). The proposed VO algorithm updates the current position at 15 Hz. We obtain the velocity estimates by differentiating the estimated position and gyro from IMU. We integrate the nonlinear sliding mode controller to generate the motor commands used in [94]. All state estimation and control algorithms run on the AscTec Mastermind onboard computer with 2.1 GHz cores and 4 GB memory. For performance comparison only, a Vicon motion capture system is used to obtain the ground truth pose of the aerial robot at 100 Hz. Desired position or trajectory determined by the user or path planning algorithm is sent to the aerial robot from the ground computer with Xbee at 40 Hz. All measured information from the onboard sensors is sent to the onboard computer to perform state estimation. Control inputs are calculated on the onboard computer with the estimated pose and the goal position given by the ground computer. All of the flight data and ground truth pose are sent to the ground computer through WiFi and TCP/IP communication.

5.7.2.2 Autonomous Flight with Light Variations

We evaluate the proposed algorithm in terms of accuracy and robustness through the autonomous flight experiments in an environment where sudden and partial light variations occur frequently as shown in Fig. 5.1 and 5.16. While the lights are turned on and off repeatedly and randomly for generating photometric disturbances, we command the aerial robot to follow the given trajectory.

The proposed method allows the aerial robot to fly autonomously along the trajectory even in such a light-changing environment as demonstrated in Fig. 5.1. The estimated trajectory is qualitatively similar to the ground-truth trajectory, and the average translational RMSE of the proposed method is 0.07 m. The point cloud is also reconstructed consistently with the trajectory estimates. Fig. 5.16 shows that per-patch illumination correction in the proposed method works

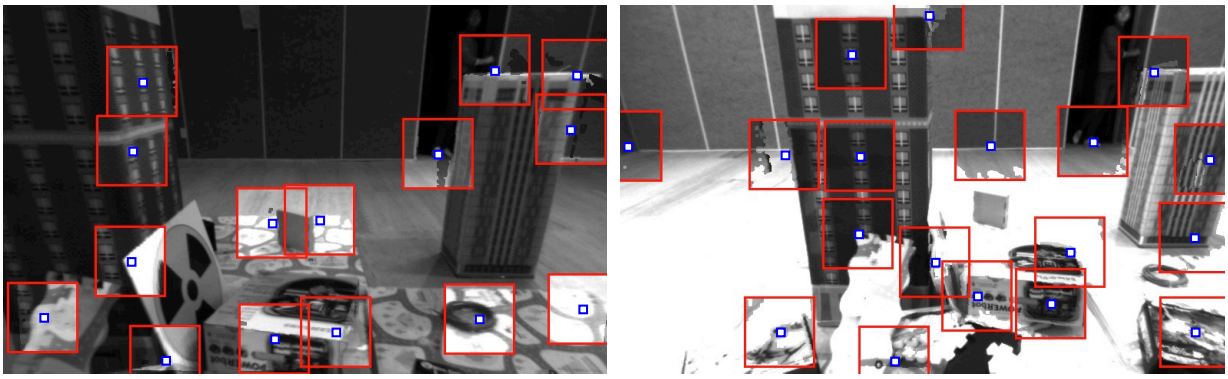


Figure 5.16: The image areas in the red patches show successful photometric compensation.

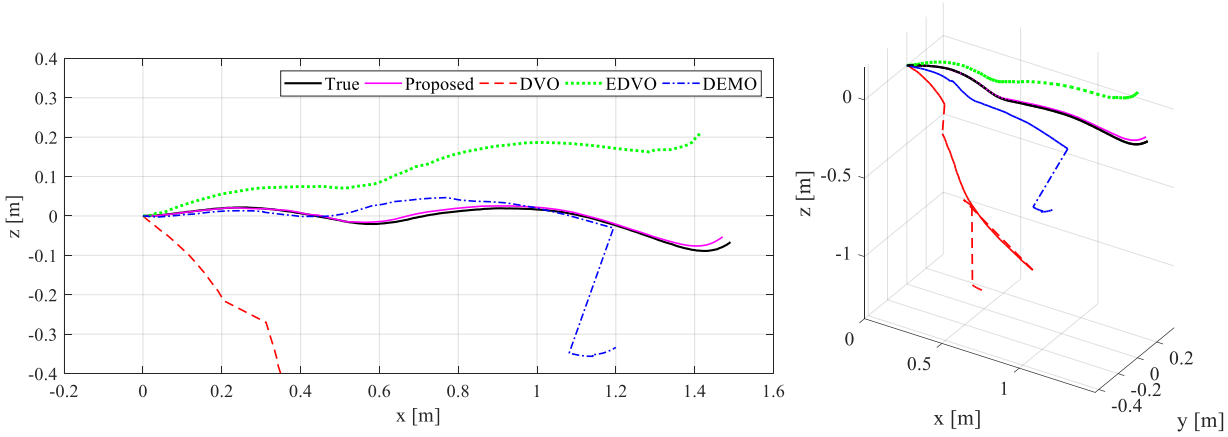


Figure 5.17: Comparison of the proposed and other VO methods in illumination changes.

successfully during the autonomous flight experiments. If the patch size is too small, we cannot estimate the photometric parameters of each patch stably and correctly. Conversely, if the patch size is too big, it will not be able to respond to local (irregular) lighting changes effectively. The patch size (91×91) we have currently used is the result of our own experimental evaluations. The autonomous flight experiments validate the accuracy and robustness of the proposed algorithm, which can estimate the accurate 6-DoF pose of the aerial robot using only onboard sensors and computer.

We also compare the proposed method to other VO methods with the stereo images obtained during the autonomous flight experiments under the irregular illumination changes as shown in Fig. 5.17. Existing VO approaches (DVO, EDVO, DEMO) cannot cope with the sudden and

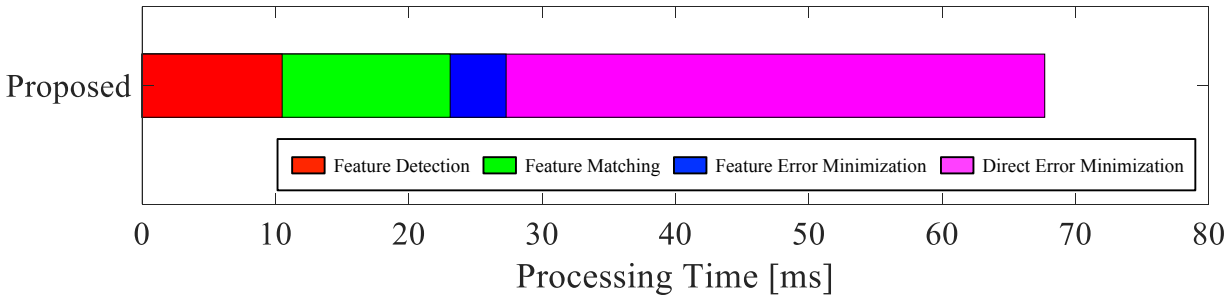


Figure 5.18: Runtime evaluation of the proposed algorithm on the aerial robot.

local lighting changes in the images, showing a large positional error compared to the ground-truth trajectory. Although EDVO, which compensates for the global illumination changes, shows good motion estimation results among other VO approaches, it cannot effectively deal with the local lighting changes. The proposed method can handle not only the global but also the local illumination changes with different affine models for different patches, showing the benefits of our proposals in challenging light environments.

Fig. 5.18 shows a break-up of the time required to compute the 6-DoF camera motion on the aerial robot. The computation time for the feature-based estimation and direct estimation is about 27.3 ms and 40.4 ms, respectively. Our direct VO approach can achieve stable and robust motion estimation performance without noticeable increases in the overall computational time. The proposed algorithm updates the current position of the aerial robot at 15 Hz, resulting in the stable autonomous flights under light-changing environments.

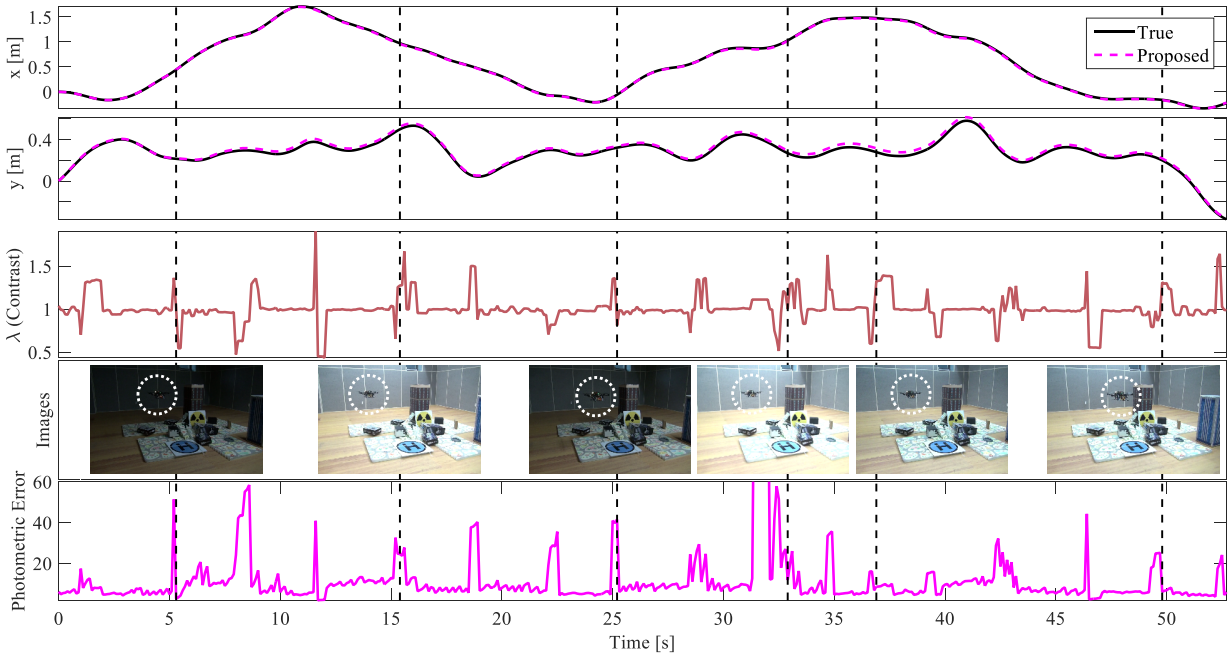


Figure 5.19: Flight experiment results in a light-changing environment.

Fig. 5.19 shows x and y position, the estimated photometric parameter (contrast), sample images, and photometric error of the proposed method under a light-changing environment for about one minute flight. The dotted vertical lines denote the time instants at which each snapshot is captured. We plot the changes of the estimated contrast in the third row, and all jumps correspond to light changes by switching the lights on and off during the sequence. When the lights dimmed near 5 seconds, the contrast increases and the illumination changes are compensated, and vice versa when the lights brightened. The estimated photometric parameters of the affine illumination model in the third row show similar behavior to the brightness level of the actual images in the fourth row. Although sudden and severe light changes continue to occur, the photometric error in the fifth row does not exceed 60, resulting in accurate motion tracking in the first and second rows. The estimated contrast parameters in the third row closely match the actual lighting conditions observed in the fourth row. When the lights are turned on and off, the photometric parameters of the affine illumination model are changed to compensate for the sudden and irregular illumination changes. Due to such compensation of the lighting changes, the photometric error in the fifth row does not exceed 60, resulting in accurate motion estimation results in the first and second rows.

Please refer to the video clips submitted with this paper showing more details about the experiments.[1]

5.8 Conclusion

We present an illumination-robust direct visual odometry for the autonomous flight of the aerial robot in a light-changing environment. The gain in robustness to irregular illumination changes is due to the fact that the affine illumination model is employed in each image patch and integrated into the direct motion estimation to *simultaneously* estimate the 6-DoF camera motion and the parameters of partial light changes. We further propose to utilize a motion prior from the feature-based visual odometry for stable and accurate motion estimation in a light-changing environment.

[1] Video available at `https://youtu.be/agOxpphFDfE`

Detailed analyses with the convergence rate and the degree of linearity of each cost function in feature-based and direct methods support such usage of the motion prior knowledge. The proposed VO algorithm enables the aerial robot to fly autonomously and robustly under changing lighting conditions at the cost of estimating the illumination change model parameters.

The results of this paper have many extensions and applications. Our work only focuses on the autonomous flight of the aerial robots, but the proposed illumination-robust visual odometry can be equally applied to various types of autonomous vehicles such as self-driving cars. Another interesting extension would be to use the proposed VO method to enhance other SLAM algorithms under changing lighting conditions. For example, our initial position estimates could be used as the cornerstone of a full SLAM system under irregular illumination changes. Our approach assumes that light changes will follow the affine illumination model; future work should consider various light change modes that vary more complexly such as bright spots and shadows caused by sunlight coming through the windows.

Appendix

In this section, we derive the analytic form of the dimensionless linearity index (DLI) for each observation model in feature-based and direct estimation in detail. The DLI of each measurement equation in terms of the 6-DoF camera motion can be written as follows:

$$L = \left| \frac{\left.\frac{\partial^2 h}{\partial \boldsymbol{\xi}^2}\right|_{\boldsymbol{\xi}=\boldsymbol{\xi}_0} \Delta\boldsymbol{\xi}}{\left.\frac{\partial h}{\partial \boldsymbol{\xi}}\right|_{\boldsymbol{\xi}=\boldsymbol{\xi}_0}} \right| \tag{5.14}$$

$$\hat{x} = h_x(\boldsymbol{\xi}) = \left[\pi(T(\boldsymbol{\xi}) \cdot \pi^{-1}(\mathbf{x}, Z(\mathbf{x})))\right]_x$$

$$\hat{I} = h_I(\boldsymbol{\xi}) = I(\pi(T(\boldsymbol{\xi}) \cdot \pi^{-1}(\mathbf{x}, Z(\mathbf{x}))))$$

where the h is the observation model in each estimation method. The division in the Eq. (5.14) denotes element-wise operation between two vectors with a slight abuse of notation. For the sake of simplicity, we can rewrite the x component of the warping function in the feature-based

method as $\hat{x}$, and the pixel intensity of the next image frame in the direct method as $\hat{I}$.

$$T = T(\boldsymbol{\xi}) = \exp(\hat{\xi}) = \exp\left(\sum_{i=1}^{6} \boldsymbol{\xi}_i E_i\right) \tag{5.4}$$

The 6-DoF camera motion, $T(\boldsymbol{\xi}) \in SE(3)$, is written as T in this section for simplicity. In the Eq. (5.4), $\boldsymbol{\xi}_i$ is a i-th component of the Lie algebra explained in Section 5.3, and E_i is one of the six Lie algebra $se(3)$ bases, each corresponding to either infinitesimal translations or rotations along each axis [95].

Feature-based Estimation

The first and second-order partial derivatives of $\hat{x}$ with respect to $\boldsymbol{\xi}$, i.e., the Jacobian and Hessian matrices, can be written as follows:

$$J_x = \frac{\partial \hat{x}}{\partial \boldsymbol{\xi}} \in \mathbb{R}^{1\times 6}, H_x = \frac{\partial^2 \hat{x}}{\partial \boldsymbol{\xi}^2} \in \mathbb{R}^{6\times 6} \tag{5.15}$$

By applying the chain rule, we can derive analytical Jacobian matrix as follows:

$$J_x = \frac{\partial \hat{x}}{\partial \mathbf{X}'} \cdot \frac{\partial \mathbf{X}'}{\partial T} \cdot \frac{\partial T}{\partial \boldsymbol{\xi}} \tag{5.16}$$

where $\mathbf{X}' = [X', Y', Z']^\top$ is a transformed 3D point from $\mathbf{X} = [X, Y, Z]^\top$ with respect to T. The analytical Jacobian matrix of warping function is:

$$J_x = \begin{bmatrix} f_x \frac{1}{Z'} & 0 & -f_x \frac{X'}{Z'^2} & -f_x \frac{X'Y'}{Z'^2} & f_x \left(1 + \frac{X'^2}{Z'^2}\right) & -f_x \frac{Y'}{Z'} \end{bmatrix}$$

For full details of each step, see [95] and [38].

In the following, the second order partial derivatives of warping function with respect to $\boldsymbol{\xi}$,

the Hessian matrix, can be obtained by applying the chain rule same as above:

$$\begin{aligned} H_x &= \frac{\partial}{\partial \boldsymbol{\xi}} \left(\frac{\partial \hat{x}}{\partial \boldsymbol{\xi}} \right) = \frac{\partial}{\partial \boldsymbol{\xi}} \left(\frac{\partial \hat{x}}{\partial T} \cdot \frac{\partial T}{\partial \boldsymbol{\xi}} \right) \\ &= \frac{\partial}{\partial \boldsymbol{\xi}} \left(\frac{\partial \hat{x}}{\partial T} \right) \cdot \frac{\partial T}{\partial \boldsymbol{\xi}} + \frac{\partial \hat{x}}{\partial T} \cdot \frac{\partial^2 T}{\partial \boldsymbol{\xi}^2} \end{aligned} \tag{5.17}$$

The (i, j)-th element of the Hessian matrix H_x can be written as follows:

$$\begin{aligned} H_{x(i,j)} &= \frac{\partial}{\partial T} \left(\frac{\partial \hat{x}}{\partial T} \right) \cdot \frac{\partial T}{\partial \boldsymbol{\xi}_j} \cdot \frac{\partial T}{\partial \boldsymbol{\xi}_i} + \frac{\partial \hat{x}}{\partial T} \cdot \frac{\partial^2 T}{\partial \boldsymbol{\xi}_i \partial \boldsymbol{\xi}_j} \\ &= \left(\frac{\partial T}{\partial \boldsymbol{\xi}_j} \right)^\top \cdot \frac{\partial^2 \hat{x}}{\partial T^2} \cdot \left(\frac{\partial T}{\partial \boldsymbol{\xi}_i} \right) + \frac{\partial \hat{x}}{\partial T} \cdot \frac{\partial^2 T}{\partial \boldsymbol{\xi}_i \partial \boldsymbol{\xi}_j} \end{aligned} \tag{5.18}$$

where the first and second-order derivatives of T with respect to $\boldsymbol{\xi}_i$ can be computed as follows:

$$\frac{\partial T}{\partial \boldsymbol{\xi}_i} = E_i \cdot T$$

$$\frac{\partial^2 T}{\partial \boldsymbol{\xi}_i \partial \boldsymbol{\xi}_j} = \frac{1}{2} \left(E_i E_j + E_j E_i \right) \cdot T$$

Note that, J_y and H_y, the Jacobian and Hessian matrices of y component of the warping function with respect to the camera motion, are omitted because they are symmetric to J_x and H_x. With the equations derived above, we obtain the Jacobian and Hessian matrices of $\hat{x}$ with respect to the $\boldsymbol{\xi}$, and compute the DLI of the observation model in the feature-based method in terms of the 6-DoF camera pose.

Direct Estimation

The first and second-order partial derivatives of $\hat{I}$ with respect to $\boldsymbol{\xi}$, the Jacobian and Hessian matrices of the observation model in the direct method, can be written as follows:

$$J_I = \frac{\partial \hat{I}}{\partial \boldsymbol{\xi}} \in \mathbb{R}^{1 \times 6}, H_I = \frac{\partial^2 \hat{I}}{\partial \boldsymbol{\xi}^2} \in \mathbb{R}^{6 \times 6} \tag{5.19}$$

To obtain the analytical Jacobian and Hessian matrices, we repeat similar calculation procedure again as above. Based on the analytical Jacobian and Hessian matrices of warping function calculated in the previous section and the chain rule, we can easily derive the analytical Jacobian matrix of direct method as follows:

$$J_I = \frac{\partial \hat{I}}{\partial \hat{\mathbf{x}}} \cdot \frac{\partial \hat{\mathbf{x}}}{\partial \boldsymbol{\xi}} = \frac{\partial \hat{I}}{\partial \hat{\mathbf{x}}} \cdot \frac{\partial \hat{\mathbf{x}}}{\partial \mathbf{X}'} \cdot \frac{\partial \mathbf{X}'}{\partial T} \cdot \frac{\partial T}{\partial \boldsymbol{\xi}} \tag{5.20}$$

where $\hat{\mathbf{x}} = [\hat{x}, \hat{y}]^\top$ is a 2D pixel point in the image $\hat{I}$. The first term $\frac{\partial \hat{I}}{\partial \hat{\mathbf{x}}}$ denotes the gradient of the image $\hat{I}$ given by the image derivatives in the horizontal and vertical directions and the latter terms are the Jacobian matrix of warping function written in Eq. (5.16). The analytical Jacobian matrix of the observation model in the direct method is:

$$\begin{aligned} J_I &= \begin{bmatrix} \frac{\partial \hat{I}}{\partial \hat{x}} & \frac{\partial \hat{I}}{\partial \hat{y}} \end{bmatrix} \cdot \begin{bmatrix} \frac{\partial \hat{x}}{\partial \mathbf{X}'} \cdot \frac{\partial \mathbf{X}'}{\partial T} \cdot \frac{\partial T}{\partial \boldsymbol{\xi}} \\ \frac{\partial \hat{y}}{\partial \mathbf{X}'} \cdot \frac{\partial \mathbf{X}'}{\partial T} \cdot \frac{\partial T}{\partial \boldsymbol{\xi}} \end{bmatrix} \\ &= \begin{bmatrix} \frac{\partial \hat{I}}{\partial \hat{x}} & \frac{\partial \hat{I}}{\partial \hat{y}} \end{bmatrix} \cdot \begin{bmatrix} \frac{\partial \hat{x}}{\partial \boldsymbol{\xi}} \\ \frac{\partial \hat{y}}{\partial \boldsymbol{\xi}} \end{bmatrix} \\ &= \begin{bmatrix} \nabla \hat{I}_x & \nabla \hat{I}_y \end{bmatrix} \cdot \begin{bmatrix} J_x \\ J_y \end{bmatrix} \end{aligned}$$

where $\nabla \hat{I}_x$ and $\nabla \hat{I}_y$ are image gradients of $\hat{I}$ along the x and y direction in the image plane.

The Hessian matrix used in the direct estimation can be also derived by applying the chain rule and the Eq. (5.20) as follows:

$$H_I = \frac{\partial}{\partial \boldsymbol{\xi}} \left(\frac{\partial \hat{I}}{\partial \boldsymbol{\xi}} \right) = \frac{\partial}{\partial \boldsymbol{\xi}} \left(\frac{\partial \hat{I}}{\partial \hat{\mathbf{x}}} \cdot \frac{\partial \hat{\mathbf{x}}}{\partial \boldsymbol{\xi}} \right) \tag{5.21}$$

Each element of the Hessian matrix, an element in the i-th row and j-th column of H_I, can be

written as follows:

$$
\begin{aligned}
H_{I(i,j)} &= \frac{\partial}{\partial \boldsymbol{\xi}_j}\left(\frac{\partial \hat{I}}{\partial \boldsymbol{\xi}_i}\right) \\
&= \frac{\partial}{\partial \boldsymbol{\xi}_j}\left(\frac{\partial \hat{I}}{\partial \hat{x}} \cdot \frac{\partial \hat{x}}{\partial \boldsymbol{\xi}_i} + \frac{\partial \hat{I}}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial \boldsymbol{\xi}_i}\right) \\
&= \frac{\partial}{\partial \boldsymbol{\xi}_j}\left(\frac{\partial \hat{I}}{\partial \hat{x}}\right) \cdot \frac{\partial \hat{x}}{\partial \boldsymbol{\xi}_i} + \frac{\partial \hat{I}}{\partial \hat{x}} \cdot \frac{\partial}{\partial \boldsymbol{\xi}_j}\left(\frac{\partial \hat{x}}{\partial \boldsymbol{\xi}_i}\right) \\
&+ \frac{\partial}{\partial \boldsymbol{\xi}_j}\left(\frac{\partial \hat{I}}{\partial \hat{y}}\right) \cdot \frac{\partial \hat{y}}{\partial \boldsymbol{\xi}_i} + \frac{\partial \hat{I}}{\partial \hat{y}} \cdot \frac{\partial}{\partial \boldsymbol{\xi}_j}\left(\frac{\partial \hat{y}}{\partial \boldsymbol{\xi}_i}\right) \\
&= \left(\frac{\partial^2 \hat{I}}{\partial \hat{x}^2} \cdot J_{x(1,j)} + \frac{\partial^2 \hat{I}}{\partial \hat{y} \partial \hat{x}} \cdot J_{y(1,j)}\right) \cdot J_{x(1,i)} \\
&+ \frac{\partial \hat{I}}{\partial \hat{x}} \cdot H_{x(i,j)} \\
&+ \left(\frac{\partial^2 \hat{I}}{\partial \hat{x} \partial \hat{y}} \cdot J_{x(1,j)} + \frac{\partial^2 \hat{I}}{\partial \hat{y}^2} \cdot J_{y(1,j)}\right) \cdot J_{y(1,i)} \\
&+ \frac{\partial \hat{I}}{\partial \hat{y}} \cdot H_{y(i,j)}
\end{aligned}
\tag{5.22}
$$

where J_x, J_y and H_x, H_y are the Jacobian and Hessian matrices of warping function derived in the previous section, and $\frac{\partial^2 \hat{I}}{\partial \hat{x}^2}, \frac{\partial^2 \hat{I}}{\partial \hat{x} \partial \hat{y}}, \frac{\partial^2 \hat{I}}{\partial \hat{y}^2}$ are the second image derivatives in the horizontal and vertical directions. With the above analytical Jacobian and Hessian matrices of the observation model in the direct method, we can compute the DLI of the image intensity observation model with respect to the 6-DoF camera motion.

6

Visual Odometry with Drift-Free Rotation Estimation Using Indoor Scene Regularities

Authors	Pyojin Kim[1]	rlavywls@snu.ac.kr
	Brian Coltin[2]	brian.j.coltin@nasa.gov
	H. Jin Kim[1]	hjinkim@snu.ac.kr
	[1] Seoul National University	
	[2] NASA Ames Research Center	

Publication	**Visual Odometry with Drift-Free Rotation Estimation Using Indoor Scene Regularities.** Kim, Pyojin, Brian Coltin, H. Jin Kim. In *Proceedings of British Machine Vision Conference (BMVC), 2017*. Copyright 2017 BMVA.

Contribution	Problem definition	*significantly contributed*
	Literature survey	*significantly contributed*
	Method development	*significantly contributed*
	Implementation	*significantly contributed*
	Experimental evaluation	*significantly contributed*
	Preparation of the manuscript	*significantly contributed*

Abstract We propose a hybrid visual odometry algorithm to achieve accurate and low-drift state estimation by separately estimating the rotational and translational camera motion. Previous methods usually estimate the six degrees of freedom camera motion jointly without distinction between rotational and translational motion. However, inaccuracy in the rotation estimate is a main source of drift in visual odometry. We design a hybrid visual odometry algorithm which separately estimates the rotational and translational motion to achieve improved accuracy and low drift error. To improve the accuracy of rotational motion estimation, we exploit orthogonal planar structures, such as walls, floors, and ceilings, common in man-made environments. We track orthogonal frames with an efficient SO(3)-constrained mean-shift algorithm, resulting in drift-free rotation estimates. Based on the absolute camera orientation, we newly propose a way to compute the translational motion by minimizing the de-rotated reprojection error with the tracked features. We compare the proposed algorithm with other state-of-the-art visual odometry methods and demonstrate an improved performance and lower drift error.

6.1 Introduction

Visual odometry (VO) and visual simultaneous localization and mapping (V-SLAM) estimate the motion of a camera from a sequence of images. While V-SLAM constructs a surrounding map and localizes the position of the camera within the constructed map simultaneously, VO only estimates the current position of the camera by accumulating the motions between each image frame. They are fundamental components for many emerging applications, from autonomous cars and unmanned aerial vehicles (UAVs) to augmented and virtual reality. Although VO has lower drift than conventional wheel odometry which is affected by wheel slip in uneven terrain [39], substantial research has focused on minimizing VO drift without any V-SLAM techniques (*i.e.*, loop closure, 3D mapping). VO techniques can be categorized into indirect [9, 43, 71] and direct [6, 24, 22, 85] methods. Direct methods use raw-image pixel values to estimate the six degrees of freedom (DoF) camera motion, while indirect methods use higher-level features detected from the images [85]. Although it is well-known that the main source of the VO drift is

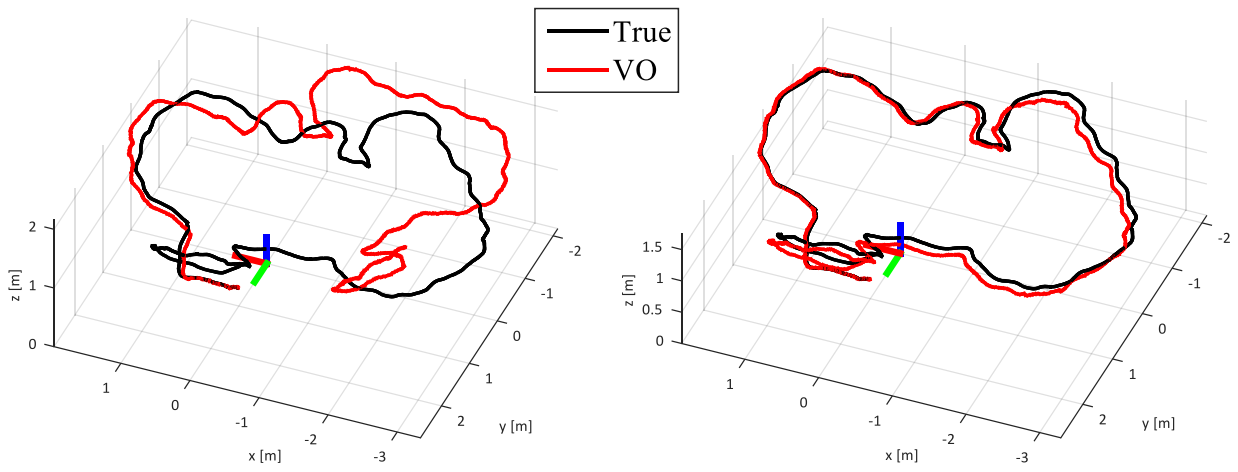


Figure 6.1: The drift of the rotation estimate is the main source of position inaccuracy in VO.

inaccurate rotation estimation [96, 97, 98] as shown in Figure 6.1, most VO approaches do not focus on rotational motion estimation and still estimate rotation and translation together.

We propose a new hybrid visual odometry method which separately estimates the rotation and translation for accurate state estimation in man-made environments. First, the proposed method estimates absolute, drift-free rotation by exploiting orthogonal structures in man-made environments with depth camera to eliminate the main source of positioning inaccuracy. Next, we find the optimal translation by minimizing the de-rotated reprojection error. This algorithm is drift-free in rotation, but requires an orthogonal structure (*i.e.*, inside a building). Extensive evaluation results show that the proposed method produces a low drift error and superior motion estimation to existing indirect and direct VO methods.

6.2 Related Work

In the last decade, VO and V-SLAM have been extensively studied in robotics and computer vision communities to lower the rate of VO drift. From the vast literature in VO and V-SLAM, we review the latest VO research showing high accuracy and some studies specifically aimed at

increasing the accuracy of rotational motion estimation.

VO research can be categorized into direct and indirect methods. Direct methods, which directly exploit the image brightness, have received recent attention for their improved accuracy and robustness to images with little texture with the help of improved hardware. Many direct VO algorithms [51, 6, 7] have been proposed to minimize the photometric error between image frames. But they also suffer from the accumulation of VO drift, for reasons such as irregular illumination changes [8]. In [85], the illumination problem is handled by considering a full photometric calibration model to reduce VO drift error due to light changes. [24] utilizes both direct and feature-based method showing high VO accuracy, and constructs temporary maps at keyframes to reduce VO drift error.

Visual feature-based methods, the most widely used indirect methods, show successful motion estimation results [9, 43, 99]. Using a stereo camera, [9] estimates motion on an autonomous car. In [99], low VO drift error is achieved with careful selection of stable features. [43] accurately estimates motion by jointly minimizing constraints from both an RGB and a depth image. Some researches [100, 101] have separately estimated the rotational motion, which contributes the most to drift, by using distant feature points and epipolar geometry, without assuming the environment is orthogonal (the "Manhattan World (MW)" assumption) [102]. Our approach requires the MW assumption, but drastically reduces drift thanks to drift-free rotation estimation.

Some recent studies have focused on accurate rotation estimation in structured environments. From MW surface normal vectors, [98] estimates rotational motion based on the maximum a posteriori (MAP) inference of the local Manhattan frame in real-time on a GPU. [103] decouples rotation and translation to estimate absolute orientation by tracking the Manhattan frame (MF) with a mean shift algorithm. However, this method suffers from a translation error that increases rapidly over time, as the translational motion is computed by aligning 1D density distribution of the point cloud. In [104], the absolute attitude (roll and pitch angles) is estimated based on vanishing points (VP) detection, and translation estimation is performed with a 2-point algorithm for catadioptric vision. While [105] jointly estimates accurate camera orientation and VPs without the Manhattan world assumption, it only estimates rotation, not translation. We propose a new

hybrid visual odometry algorithm which achieves low drift and high accuracy by first estimating drift-free rotational motion and then computing translational motion separately. This approach relies on the MW assumption, and only applies to orthogonal environments such as buildings.

6.3 Proposed Method

We present a new hybrid visual odometry algorithm which separately estimates rotational and translation motion to achieve low drift error and higher accuracy. The proposed method is shown in Algorithm 1 in detail. To minimize the effect of the drift in the rotation estimates, which is the main source of the VO drift error [96, 98], drift-free rotational motion is estimated by tracking the Manhattan frame with a SO(3)-manifold constrained mean shift algorithm using density of surface normal vectors as shown in Figure 6.2. Given the absolute camera orientation, we estimate the translational motion by minimizing the de-rotated reprojection error with the tracked features. By combining the estimated 3-DoF rotational and 3-DoF translational motion, the entire 6-DoF camera motion can be tracked. The overall moving trajectory can be obtained by concatenating the frame-to-frame motion estimation results incrementally.

Algorithm 1 Rotation and Translation Estimation From Frame $k-1$ to Frame k

Input: Greyscale Images I_k, I_{k-1}; Depth Images D_k, D_{k-1}
Output: Rotation $\mathbf{R}_{k,k-1}$; Translation $\mathbf{T}_{k,k-1}$

1: extract surface normal vectors $\mathbf{n}_k$ from D_k
2: **repeat**
3: project $\mathbf{n}_k$ into each tangential plane of MF axes using Eqs. (6.1) and (6.2)
4: perform Gaussian mean shift algorithm using Eq. (6.3)
5: back onto the unit sphere using Eqs. (6.4) and (6.5)
6: project $\hat{\mathbf{R}}_{\mathrm{c}_k\mathrm{M}}$ onto the SO(3) manifold using Eq. (6.7)
7: **until** $\mathbf{R}_{\mathrm{c}_k\mathrm{M}}$ converges
8: $\mathbf{R}_{k,k-1} \leftarrow \mathbf{R}_{\mathrm{c}_k\mathrm{M}} \cdot \mathbf{R}_{\mathrm{Mc}_{k-1}}$
9: track feature points from I_k
10: derive residual vectors of all tracked features using Eq. (6.9)
11: find optimal $\mathbf{T}^*$ using Eq. (6.10)
12: $\mathbf{T}_{k,k-1} \leftarrow \mathbf{T}^*$

6.3.1 Rotational Motion Estimation

6.3.1.1 Surface Normal Vector Extraction

We pre-process the depth image D_k with a simple box filter to remove noise in the raw depth data. The unit surface normal vectors $\mathbf{n}_k$ on the unit sphere $\mathbb{S}^2$ can be computed using the cross product from the two tangential vectors, which are tangential to the local surface at the 3D points in the point cloud. They can be easily calculated between the left and right neighboring pixels for the u-direction (horizontal), and between the up and down neighboring pixels for the v-direction (vertical) in the point cloud. To reduce noise data in the two maps of tangential vectors, we compute the average u- and v-tangential vectors within a certain neighborhood. To perform this smoothing process efficiently, we generate integral images of the two tangential vectors and calculate the average u- and v-tangential vectors with only $2\times4\times3$ memory access regardless of the size of the smoothing area [106]. It is very important to extract accurate and reliable surface normal vectors since the density distribution of surface normal vectors obtained from the depth image directly affects the accuracy of rotational motion estimation in MW.

6.3.1.2 Tracking Manhattan Frame

The core of the proposed MF tracking is that we compare and track orthogonality of planar structures, which is called the Manhattan frame, observed from the current camera viewpoint. We track the orthogonal Manhattan frames with a SO(3)-manifold constrained mean-shift algorithm, an approach for finding the mode, for a dominant axis given a set of surface normal vectors on the unit sphere $\mathbb{S}^2$ under the assumption that the MF does not change too much between the frame-to-frame motion.

We express the Manhattan world frame with respect to the camera frame as a 3D rotation matrix $\mathbf{R}_{\mathrm{cM}} = \begin{bmatrix}\mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3\end{bmatrix} \in \mathrm{SO}(3)$ where each column $\mathbf{r}_j$ denotes the x-, y-, and z-axis of the dominant MF expressed in the camera frame. We assume that the previous rotation of MF $\mathbf{R}_{\mathrm{c}_{k-1}\mathrm{M}}$ is known, and it is used as initialization point to find the current unknown orientation of MF $\mathbf{R}_{\mathrm{c}_k\mathrm{M}}$. We transform the current surface normal vectors $\mathbf{n}_k$ expressed in the camera frame

into $\mathbf{n}'_k$ expressed in the MF:

$$\mathbf{n}'_k = \mathbf{R}^{\top}_{\mathrm{c}_k\mathrm{M}}\, \mathbf{n}_k \tag{6.1}$$

We alternately project the surface normal vectors $\mathbf{n}'_{k_j}$ into the tangential planes of the x-, y-, and z-axis of the MF to compute a mean shift. The index k_j indicates relevant normal vectors inside a conic section of the j-th dominant axis in the MF. We apply the Riemann logarithmic map to represent proper distances in the tangential plane given by:

$$\mathbf{m}'_{k_j} = \frac{\sin^{-1}(\lambda)\,\mathrm{sign}(\mathbf{n}'_{k_j,z})}{\lambda} \begin{bmatrix} \mathbf{n}'_{k_j,x} \\ \mathbf{n}'_{k_j,y} \end{bmatrix} \tag{6.2}$$

$$\text{where} \;\; \lambda = \sqrt{\mathbf{n}'^2_{k_j,x} + \mathbf{n}'^2_{k_j,y}}$$

where $\mathbf{m}'_{k_j}$ means the two-dimensional coordinate position in the tangential plane. We perform the mean shift algorithm with a Gaussian kernel in the tangential plane:

$$\mathbf{s}'_j = \frac{\sum e^{-c\|\mathbf{m}'_{k_j}\|^2} \mathbf{m}'_{k_j}}{\sum e^{-c\|\mathbf{m}'_{k_j}\|^2}} \tag{6.3}$$

where c is the width of the kernel, defined by the user. We apply the Riemann exponential map to transform the mean shift result back to the unit sphere from the tangential plane:

$$\mathbf{s}_j = \overline{\begin{bmatrix} \frac{\tan(\|\mathbf{s}'_j\|)}{\|\mathbf{s}'_j\|}\mathbf{s}'^{\top}_j & 1 \end{bmatrix}}^{\top} \tag{6.4}$$

where the $\overline{x}$ means x normalized. The estimated j-th dominant axis expressed in the MF is converted with respect to the camera frame to obtain the updated direction vector:

$$\hat{\mathbf{r}}_j = \mathbf{R}_{\mathrm{c}_k\mathrm{M}}\, \mathbf{s}_j \tag{6.5}$$

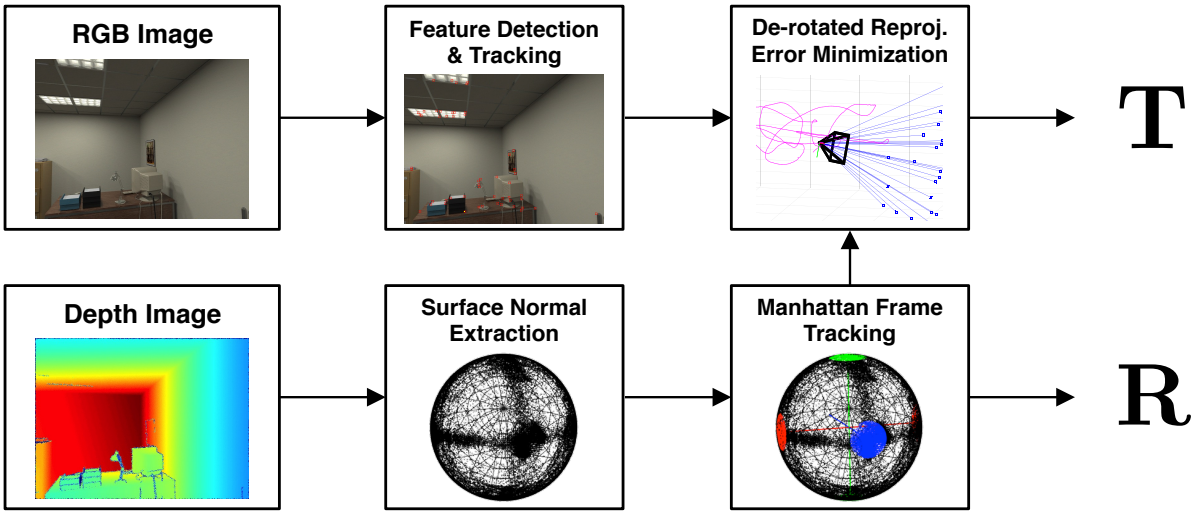


Figure 6.2: Overview of the algorithm that separately estimates rotation and translation.

Repeating for all dominant axes in the MF, we obtain the updated rotation matrix:

$$\hat{\mathbf{R}}_{\mathrm{c}_k\mathrm{M}} = \begin{bmatrix} \hat{\mathbf{r}}_1 & \hat{\mathbf{r}}_2 & \hat{\mathbf{r}}_3 \end{bmatrix} \tag{6.6}$$

However, $\hat{\mathbf{R}}_{\mathrm{c}_k\mathrm{M}}$ violates the orthogonality constraint because each axis in the MF is updated independently by the mean shift algorithm in Eq. (6.3). Thus, we project $\hat{\mathbf{R}}_{\mathrm{c}_k\mathrm{M}}$ onto the SO(3) manifold to satisfy the orthogonality constraint using singular value decomposition (SVD):

$$\mathbf{R}_{\mathrm{c}_k\mathrm{M}} = \mathbf{U}\mathbf{V}^\top \tag{6.7}$$

$$[\mathbf{U}, \mathbf{D}, \mathbf{V}] = \mathrm{SVD}(\begin{bmatrix} \lambda_1\hat{\mathbf{r}}_1 & \lambda_2\hat{\mathbf{r}}_2 & \lambda_3\hat{\mathbf{r}}_3 \end{bmatrix})$$

where λ is a weighting factor of how certain the observation of a direction is [103]. The above procedure (lines 2 to 7 of Algorithm 1) is repeated until the change in the estimated rotation of MF is very small. In this manner, we obtain an absolute and drift-free estimate of the MF orientation. Note that at least two orthogonal planes must be visible to track the MF.

6.3.1.3 Dominant Manhattan Frame

We find the most dominant MF, *i.e.*, the three dominant planar axes in an environment, to estimate drift-free rotational motion. We employ the mean shift clustering algorithm to initialize the estimation of the rotational motion. We first perform the MF tracking 100 times from a random initial rotation, in the manner explained previously. We cluster these 100 MF tracking results and perform histogram-based non-maximum suppression. The most frequent MF is selected as the dominant initial MF. For more details, see [103]. The dominant MF only needs to be found on the first frame.

6.3.2 Translational Motion Estimation

6.3.2.1 Feature Extraction

Feature points used in translational motion estimation are extracted using the Good Features to Track corner detector [107]. Bucketing is utilized to spread them uniformly across the entire image domain and reduce the number of features. We maintain the number of features between 150 and 200 in practice. The points are tracked in the next image frame using KLT feature tracker [108].

6.3.2.2 De-rotated Reprojection Error Minimization

The core of the estimation of the translational motion is that the translational movement of the camera can be obtained by minimizing de-rotated reprojection error given the absolute camera orientation. Previous approaches [9, 43] jointly minimized the original reprojection error, which leads to higher drift over time.

We model the mathematical relationship between the camera motion and i-th tracked 3D feature point as follows:

$$\mathbf{X}_i^k = Z_i^k \bar{\mathbf{X}}_i^k = \mathbf{R}\mathbf{X}_i^{k-1} + \mathbf{T} \tag{6.8}$$

where $\mathbf{X}_i^k = [X_i^k, Y_i^k, Z_i^k]^\top$ is the coordinates of the feature in the camera frame at time step k,

and $\bar{\mathbf{X}}_i^k = [\bar{X}_i^k, \bar{Y}_i^k, 1]^\top$ is the normalized term of $\mathbf{X}_i^k$ where the z component is one. The rotation matrix $\mathbf{R}$ and the translation vector $\mathbf{T}$ form an SE(3) rigid body transformation [36]. Eq. (6.8) contains three rows. Substituting the expression Z_i^k in the third row into the first and second rows, we can derive two equations for the tracked feature point as follows:

$$\begin{aligned} r_{i,1}(\mathbf{T}) &= (\mathbf{R}_1 - \bar{X}_i^k \mathbf{R}_3)\mathbf{X}_i^{k-1} + \mathbf{T}_1 - \bar{X}_i^k \mathbf{T}_3 = 0 \\ r_{i,2}(\mathbf{T}) &= (\mathbf{R}_2 - \bar{Y}_i^k \mathbf{R}_3)\mathbf{X}_i^{k-1} + \mathbf{T}_2 - \bar{Y}_i^k \mathbf{T}_3 = 0 \end{aligned} \tag{6.9}$$

where $\mathbf{R}_h$ and $\mathbf{T}_h$, $h \in \{1, 2, 3\}$ are h-th rows of $\mathbf{R}$ and $\mathbf{T}$ respectively. There are two residual terms per feature. Because we already know the absolute camera orientation by tracking MF in the previous section, the residual terms in Eq. (6.9) are only a function of the translational camera motion. The optimal translational camera motion that minimizes the residual vectors of all tracked feature points can be obtained by solving the following optimization problem:

$$\mathbf{T}^* = \arg\min_{\mathbf{T}} \sum_{i=1}^{M} \left[(r_{i,1}(\mathbf{T}))^2 + (r_{i,2}(\mathbf{T}))^2 \right] \tag{6.10}$$

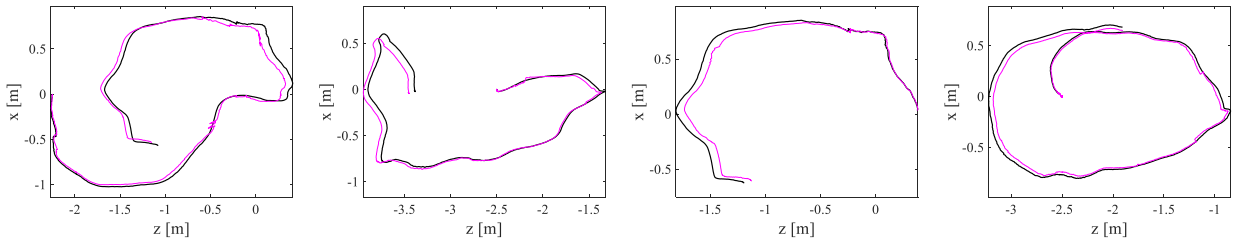
where M is the number of tracked features. Eq. (6.10) can be solved by the Levenberg–Marquardt (LM) algorithm [33]. Note that textures and brightness in the images should be sufficient for reliable feature detection and tracking. Otherwise, the inaccurate translational motion estimation caused by wrong correspondence can degrade the overall 6-DoF camera motion estimation.

6.4 Evaluation

We evaluate the effectiveness of the proposed VO algorithm on the publicly available ICL-NUIM benchmark [1] and author-collected RGB-D datasets in man-made environments. For the performance comparison, we provide the motion estimation results of the proposed and other VO algorithms: indirect, direct, and hybrid VO methods, namely DEMO [43], DVO [6], and MWO [103] respectively. DEMO and DVO are the visual odometry methods which estimate the rotational and translational motion jointly, while MWO decouples the estimation of the rotational and transla-

Experiment	Relative Pose Error (m/s)				Absolute Trajectory Error (m)				Final Drift Error (%)				Length (m)	# of frame
	Proposed	DEMO	DVO	MWO	Proposed	DEMO	DVO	MWO	Proposed	DEMO	DVO	MWO		
lr kt0	**0.031**	0.077	0.048	0.084	**0.052**	0.145	0.237	0.317	**4.546**	13.293	7.084	32.359	2.74	502
lr kt1	0.021	**0.020**	0.023	0.100	**0.042**	0.143	0.065	0.589	**1.744**	13.231	3.151	30.158	2.05	951
lr kt2	**0.031**	0.090	0.084	0.052	**0.064**	0.616	0.502	0.130	**0.934**	11.813	8.819	1.363	8.42	881
lr kt3	**0.052**	0.076	0.068	0.090	**0.096**	0.282	0.409	0.373	**1.798**	5.327	4.629	3.826	5.47	554
of kt0	**0.014**	0.063	0.106	×	**0.048**	0.338	0.371	×	**1.295**	5.947	10.090	×	6.53	1507
of kt1	**0.014**	0.054	0.045	0.263	**0.052**	0.371	0.357	1.092	**1.103**	9.197	8.912	25.250	6.72	965
of kt2	**0.015**	0.079	0.065	0.047	**0.061**	0.311	0.229	0.087	**1.577**	7.586	4.696	2.776	4.47	467
of kt3	**0.009**	0.030	0.052	0.155	**0.030**	0.176	0.304	1.312	**0.425**	2.514	5.358	24.161	7.82	1240

Table 6.1: Evaluation Results on ICL-NUIM Benchmark



lr kt2 of kt1 of kt2 of kt3

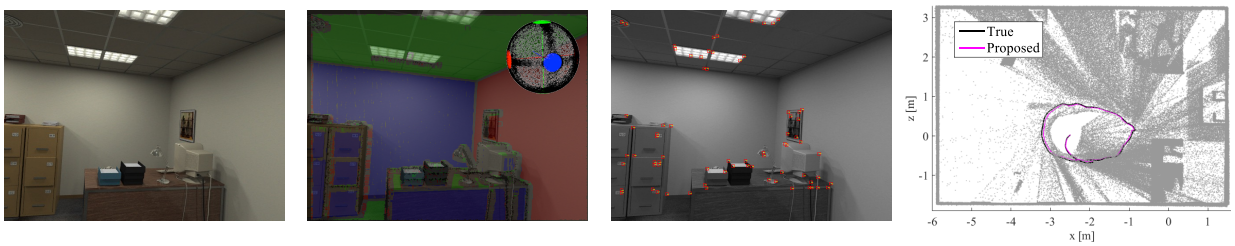
Figure 6.3: Some motion estimation results of the proposed algorithm in the ICL-NUIM dataset.

tional motion like the proposed VO method.

6.4.1 Tests with ICL-NUIM Datasets

We first test the proposed algorithm with the ICL-NUIM dataset [1]. It consists of a collection of handheld RGB-D camera sequences within synthetically generated living room and office. Although the synthetic RGB and depth images are captured at 30 Hz in virtual environments, typically observed noise existing in the actual camera image is reproduced well by modeling sensor noise in both RGB and depth data. The proposed method and other VO baselines are applied to all living room and office datasets. To perform quantitative analysis, three types of error metrics are selected: root mean square error (RMSE) of the relative pose error (RPE), absolute trajectory error (ATE) as in [2], and the final drift error divided by the total traveling distance.

Table 9.1 shows the performances of the proposed and other VO algorithms. The smallest error value in each tested dataset for each error metric is highlighted in bold type. Figure 6.3



(a) (b) (c) (d)

Figure 6.4: The inferred MF orientation is drawn in the top right corner of (b), and (d) shows the reconstructed trajectory and consistent point cloud.

shows some motion estimation results of the proposed algorithm compared to the ground truth trajectory. In most cases, the proposed algorithm shows better performance in terms of relative pose error compared to other VO algorithms. In all cases, we observe that the proposed method generates the lowest absolute trajectory error and final drift error compared to other VO baselines. The final drift error of the proposed method is 1.68% on the average, while DEMO, DVO, and MWO are 8.61%, 6.59%, and 17.13%, respectively. We also evaluate the proposed algorithm with other VO methods on TUM RGB-D dataset [2], resulting in similar experimental values: 4.59% of the proposed VO, while DEMO, DVO, and MWO are 16.82%, 9.61%, and 22.45% averagely. The main reason for the improved results is that the proposed algorithm can estimate drift-free rotational motion over time. But other VO algorithms are cumulative in the drift of the rotation estimates, which is the main source of position inaccuracy, showing a big difference from the ground truth trajectory at the end.

Excerpts from ‘of kt3’ in the ICL-NUIM dataset, including Manhattan world scene segmentation result, the inferred MF orientation, and the tracked features are shown in Figure 6.4. The Manhattan frame in the office room is successfully tracked in Figure 6.4 (b), resulting in the accurate estimation of the drift-free camera orientation. The tracked features marked red in Figure 6.4 (c) are used to minimize de-rotated reprojection error for translational motion estimation. In Figure 6.4 (d), the estimated and the ground truth trajectory overlap significantly, thus, consistent point cloud can be reconstructed based on the motion estimation results of the proposed method.

The strength of the proposed method becomes clear when analyzing the dataset ‘of kt3’ in

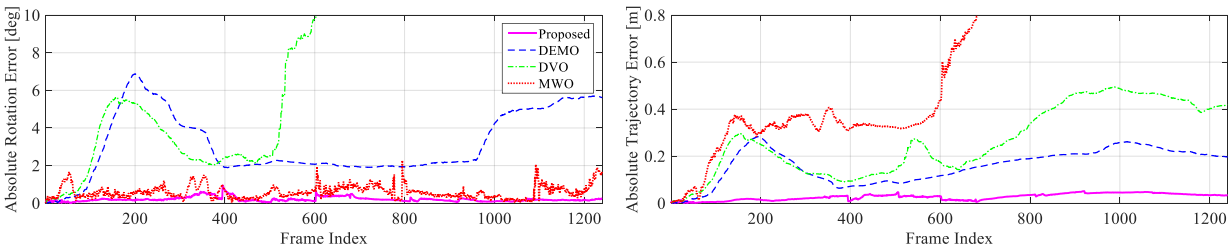


Figure 6.5: The rotation matrix errors for the proposed and other VO algorithms.

terms of the absolute rotation error as shown in Figure 6.5 (a). The absolute rotation error of the proposed method and MWO does not increase over time due to drift-free Manhattan frame tracking method. The absolute rotation error in the DEMO and DVO methods, however, continues to increase over time, resulting in positioning inaccuracy at the end. The average rotation error of the proposed method is 0.21 degree whereas DEMO, DVO, and MWO are 3.15, 8.12, 0.59 degree respectively. Although the proposed method and MWO employ the similar Manhattan frame tracking method for rotation estimation, different results are obtained due to the different algorithms for extracting surface normal vectors from a depth image.

Figure 6.5 (b) shows the superior performance of the proposed method in terms of absolute trajectory error (ATE), which includes not only rotational but also translational motion error. Although the drift-free rotation estimation is performed in MWO, increases of absolute trajectory error are shown in the graph due to incorrect estimation of translational motion in MWO. While the accumulated rotational error in DEMO and DVO makes the overall position estimation inaccurate, the proposed method shows the lowest growth rate of the absolute trajectory error given the accurate *drift-free* rotation estimates.

6.4.2 Tests with Author-collected RGB-D Datasets

We want to demonstrate that the proposed algorithm works well in the everyday indoor environments which generally satisfy Manhattan world assumption. So, we recorded our own datasets with an Asus Xtion Pro Live RGB-D camera capable of providing RGB and depth images at 30 Hz with 640×480 resolution. Figure 6.6 shows the example RGB and depth images captured

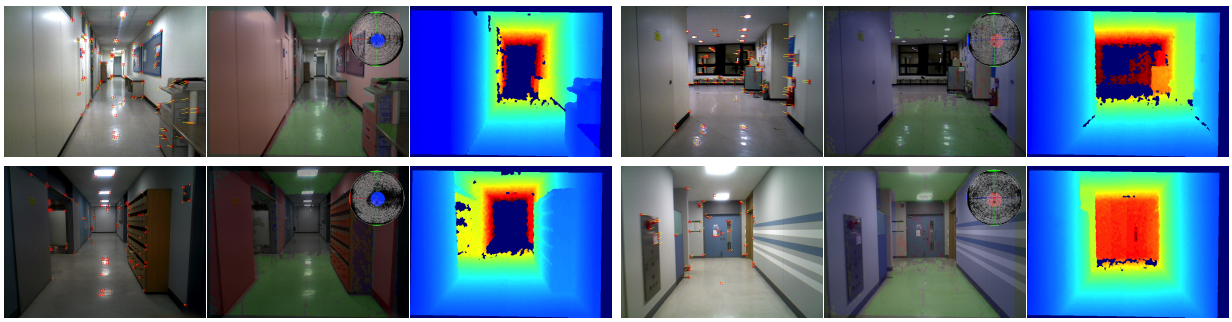
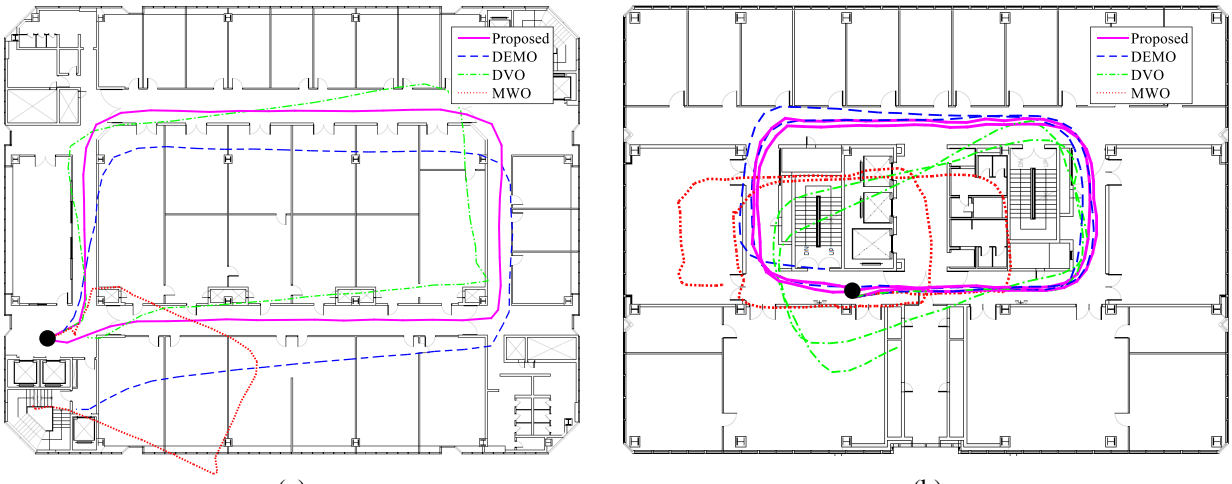


Figure 6.6: Example images from the author-collected RGB-D dataset.



(a) (b)

Figure 6.7: Motion estimation results with the proposed algorithm compared to other VO methods on the author-collected RGB-D dataset in a single-loop (a) and multiple-loop (b) sequences.

in corridors of the buildings satisfying Manhattan world constraint. For evaluating the final drift error of the proposed and other VO algorithms without the ground truth data, the images are collected along the carefully designed trajectories where the beginning and end points are at the same place. We also overlap the estimated trajectories on the floorplan of the buildings to check the consistency of the proposed and other VO methods.

Figure 6.7 (a) shows the evaluation results along about 105 meters long single loop trajectory where we start and end at the same place. DEMO and DVO methods cannot meet the starting and end points due to drift of the rotation estimates accumulated at the corners of the loop. The

starting and end points of the trajectory estimated with the proposed method coincide at the black circle. The high consistency of the proposed method is also observed in Figure 6.7 (b), which is about 127 meter long trajectory consisting of multiple loops in the same place and left turns of 90 degrees. Although MWO can estimate the drift-free rotational motion, inaccuracy in translational motion estimation causes inconsistent results. The overlapping estimated trajectory with the proposed method shows the most consistent path while other estimated trajectories gradually diverge from the initially estimated loop.

Please refer to the video clips submitted with this paper showing more details about the experiments.[1]

6.5 Conclusion

We have presented a low-drift visual odometry algorithm that separately estimates the rotational and translational motion. For reducing drift of the rotation estimate, which is the main source of position inaccuracy in visual odometry algorithms, the Manhattan frame tracking is performed to estimate the absolute camera orientation. Given drift-free rotation estimates in MW, translational motion is estimated by minimizing de-rotated reprojection with the tracked features. This approach enables accurate and low-drift motion estimation results of the proposed VO algorithm in man-made indoor environments. Our approach assumes Manhattan world environments; future work should consider more general and relaxed environments, such as Atlanta World (AW) [109] and Mixture of Manhattan Frames (MMF) [110].

[1] Video available at `https://youtu.be/sC3iiaxBhdw`

7

Low-Drift Visual Odometry in Structured Environments by Decoupling Rotational and Translational Motion

Authors	Pyojin Kim[1]	rlavywls@snu.ac.kr
	Brian Coltin[2]	brian.j.coltin@nasa.gov
	H. Jin Kim[1]	hjinkim@snu.ac.kr
	[1] Seoul National University	
	[2] NASA Ames Research Center	

Publication	**Low-Drift Visual Odometry in Structured Environments by Decoupling Rotational and Translational Motion.** Kim, Pyojin, Brian Coltin, H. Jin Kim. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA), 2018*. Copyright 2018 IEEE.

Contribution		
	Problem definition	*significantly contributed*
	Literature survey	*significantly contributed*
	Method development	*significantly contributed*
	Implementation	*significantly contributed*
	Experimental evaluation	*significantly contributed*
	Preparation of the manuscript	*significantly contributed*

Abstract We present a low-drift visual odometry algorithm that separately estimates rotational and translational motion from lines, planes, and points found in RGB-D images. Previous methods estimate drift-free rotational motion from structural regularities to reduce drift in the rotation estimate, which is the primary source of positioning inaccuracy in visual odometry. However, multiple orthogonal planes are required to be visible throughout the entire motion estimation process; otherwise, these VO approaches fail. We propose a new approach to estimate drift-free rotational motion jointly from both lines and planes by exploiting environmental regularities. We track the spatial regularities with an efficient $\mathrm{SO}(3)$-manifold constrained mean shift algorithm. Once the drift-free rotation is found, we recover the translational motion from all tracked points with and without depth by minimizing the de-rotated reprojection error. We compare the proposed algorithm to other state-of-the-art visual odometry methods on a variety of RGB-D datasets (including especially challenging pure rotations) and demonstrate improved accuracy and lower drift error.

7.1 Introduction

Visual odometry (VO) algorithms estimate the six degrees of freedom (DoF) rotational and translational camera motion from a sequence of images. They are a fundamental tool for applications from augmented reality to autonomous robots.

Many VO and Visual Simultaneous Localization and Mapping (V-SLAM) approaches, which jointly estimate rotational and translational motion, have shown promising results. However, these approaches cannot avoid drift in the rotation estimate without SLAM techniques (loop closure, global 3D map construction), resulting in large drift errors because the main source of positional inaccuracy in VO is rotation estimation error [96, 98, 97]. Many visual navigation methods are also unstable for pure, on the spot rotations [26, 71].

Our previous work [29] introduced *Orthogonal Plane-based Visual Odometry* (OPVO) to address these issues. OPVO exploits orthogonal planar structures to determine the absolute, drift-free orientation of an RGB-D camera. Based on the absolute camera orientation, it finds the

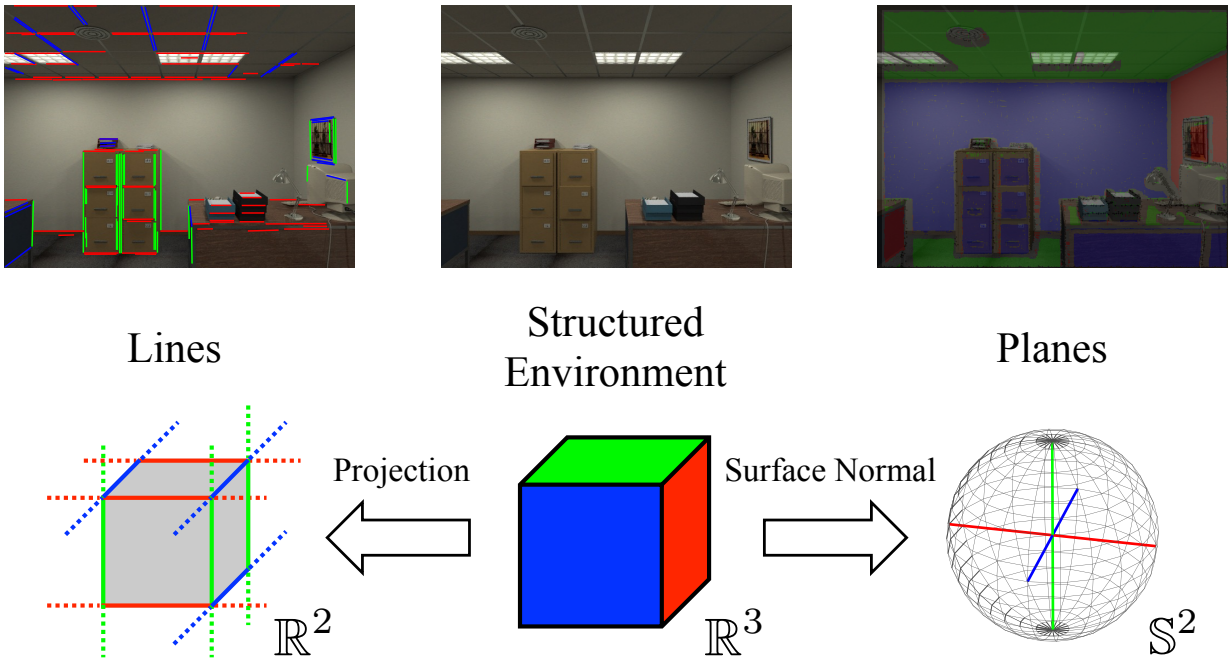


Figure 7.1: Example of a structured environment exhibiting strong orthogonal spatial regularities.

optimal translation by minimizing the de-rotated reprojection error from tracked points with depth information. Although OPVO drastically reduces the drift error, there are still two key limitations: OPVO requires at least two orthogonal planes to be visible at all times, and point features with depth information. Also, the experimental evaluation in [29] was performed mainly on synthetic RGB-D datasets.

To address these issues, we propose *Line and Plane based Visual Odometry* (LPVO), a novel VO algorithm that exploits line and plane primitives jointly to recognize the spatial regularities of orthogonal structured environments (see Fig. 7.1). Lines from RGB images and surface normal vectors from depth images are simultaneously used to perceive environmental regularities accurately and stably. LPVO can track drift-free rotational motion while at least a single plane and a pair of lines parallel to the Manhattan world (MW) axes are visible. Furthermore, we utilize point features without depth information when we recover the optimal translational motion. Extensive evaluations show that LPVO produces the lowest drift error compared to other state-of-the-art VO methods, including OPVO [29]. The main contributions of this paper are:

- We propose a novel approach to estimate absolute and drift-free rotational motion jointly from both lines and planes by utilizing environmental regularities.
- We newly use tracked points with no depth information to recover the 3-DoF translation.
- We evaluate the VO algorithms on the TUM [2] and TAMU [111] RGB-D datasets, as well as a new dataset traversing a large building, showing low drift for LPVO.

7.2 Related Work

VO and V-SLAM methods are being actively researched to lower the rate of VO drift. From the vast literature in VO and V-SLAM, we review a subset of state-of-the-art contributions, existing VO methods with motion decoupling, and studies specifically focusing on accurate rotation estimation.

VO algorithms can be classified into indirect, direct, and hybrid methods depending on the type of visual information used [26]. The most widely used indirect methods, point feature-based methods, have proven successful for 6-DoF motion estimation [9, 17, 15]. In [17], low drift error is achieved using salient feature points both with and without depth information. The recent ORB-SLAM2 [15] shows outstanding motion estimation performance with monocular, stereo, and RGB-D cameras using the same ORB features for all SLAM tasks. To reduce drift error, however, ORB-SLAM2 relies heavily on SLAM techniques (loop closing, relocalization, local 3D map reuse), which require substantial memory and computation. Direct VO methods [51, 6, 26] estimate 6-DoF camera motion by minimizing the photometric error between image frames. But they suffer from drift caused by unmodeled visual effects such as irregular illumination changes, and fare poorly at tracking on-the-spot rotations.

Some research has estimated rotational and translational motion separately. Rotation is estimated using epipolar geometry, and the translation is recovered with triangulated 3D points [100]. [101] splits camera motion into the separate rotation and translation estimates using distant and close points with a disparity map and the camera speed, while [99] estimates rotation and trans-

lation separately with carefully selected features. These VO methods, however, cannot estimate drift-free rotation in structured environments because it is difficult to recognize environmental regularities using point primitives. To utilize structural information, [104] detects dominant bundles of parallel lines for rotation and estimates translation from a 2-point algorithm up to a scale. OPVO [29] tracks a Manhattan frame (MF) for absolute camera orientation from surface normal vectors, and recovers translation by minimizing de-rotated reprojection error with available depth points. Although these approaches use structural features, no existing approach uses both lines and planes.

Several studies have more focused on accurate rotation estimation in structured environments due to the importance of rotational motion [112]. From the line segments in the image, [113] estimates the rotational motion by finding orthogonal vanishing points (VPs) with a 3-line RANSAC algorithm. While this method can estimate drift-free rotation using RGB images only, the performance is sensitive to the quality of visible lines. [98] derives MF inference algorithms based on the distribution of the surface normal vectors from depth images. In [103] and [29], drift-free rotation estimation is performed with a mean-shift algorithm based on the surface normal vector distribution. While these methods demonstrate superior rotation estimation in structured environments, at least two orthogonal planes must always be visible.

7.3 Background 3D Geometry

7.3.1 Gaussian Sphere

A Gaussian sphere is a unit sphere centered on the center of projection (COP) of a camera, and is a convenient method to represent geometric elements such as lines and normal vectors when the camera intrinsic parameters are known. A line in the image is projected onto the Gaussian sphere as a great circle (the intersection of the unit sphere and the plane defined by the line and the COP, see Fig. 7.2). The great circle of each line can be expressed as a unit vector in the Gaussian sphere. Great circles representing parallel lines in the image intersect at two antipodal points on

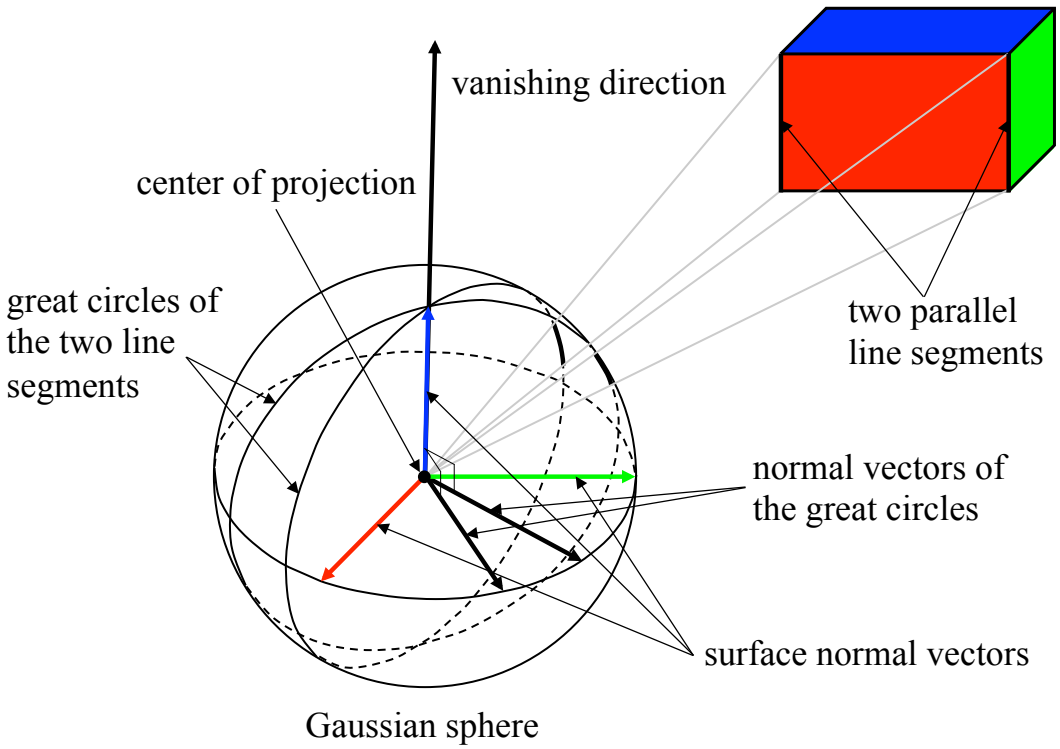


Figure 7.2: Geometric relationship between the lines, planes, and the Gaussian sphere.

the Gaussian sphere. A unit vector from the COP to the intersection point is a vanishing direction (VD), calculated as the cross product of the normal vectors of two great circles representing parallel lines in the image. The three orthogonal VDs defined by the parallel lines match the orthogonal surface normal vectors of the planes in a perfect Manhattan world. These vectors form the basis of a Manhattan frame.

7.3.2 Rotation Motion with Vanishing Directions

In Euclidean 3D space, we represent the 6-DoF camera motion as the 4×4 rigid body transformation matrix $T \in \mathrm{SE}(3)$, composed of the 3-DoF rotational motion $R \in \mathrm{SO}(3)$ and the 3-DoF translational motion $\mathbf{t} \in \mathbb{R}^3$. A vanishing direction $\mathbf{d} \in \mathbb{R}^3$ on the Gaussian sphere can be

transformed into $\mathbf{d}'$ by the 6-DoF camera motion as:

$$\tilde{\mathbf{d}}' = T\tilde{\mathbf{d}} = \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{d} \\ 0 \end{bmatrix} = \begin{bmatrix} R\mathbf{d} \\ 0 \end{bmatrix} \tag{7.1}$$

where $\tilde{\mathbf{d}} = \begin{bmatrix} \mathbf{d}^\top & 0 \end{bmatrix}^\top \in \mathbb{P}^3$ denotes the VD in homogeneous coordinates. From Eq. (9.1), we can observe that the VD is only dependent on the rotational motion of the camera.

7.4 Proposed Method

7.4.1 Orthogonal Plane-based Visual Odometry

The proposed *Line and Plane based Visual Odometry* (LPVO) method builds on our previous *Orthogonal Plane-based Visual Odometry* (OPVO) algorithm [29], which we summarize briefly (for full details, refer to [29]). OPVO has two main steps: 1) structural regularities (Manhattan frame) are tracked to estimate drift-free rotation with a $\mathrm{SO}(3)$-manifold constrained mean shift algorithm; and 2) translational motion is recovered by minimizing the de-rotated reprojection error from tracked points.

The core of the OPVO rotation estimation is tracking the Manhattan frame with a $\mathrm{SO}(3)$-manifold constrained mean shift algorithm based on the tangent space Gaussian MF (TG-MF) model [114] under the assumption that the MF does not change too much between the frame-to-frame motion. Given the density distribution of surface normal vectors on the Gaussian sphere $\mathbb{S}^2$, OPVO infers the mean of the surface normal vector distribution around each dominant Manhattan frame axis through a mean shift algorithm in the tangent plane $\mathbb{R}^2$ with a Gaussian kernel. The modes found by the mean shift algorithm are projected onto the $\mathrm{SO}(3)$ manifold to maintain orthogonality, resulting in the absolute orientation estimate of the camera.

For the translation estimation, OPVO transforms feature correspondences between consecutive frames into a pure translation by taking advantage of the drift-free rotation estimation in the previous step. OPVO recovers the 3-DoF translational motion of the camera by minimizing

de-rotated reprojection error from the tracked points, which is only dependent on the translational movement.

Next, we present LPVO, a new approach to exploit line and plane information jointly for stable and accurate drift-free rotation estimation even when only a single plane is visible. For more accurate translation estimation, we additionally use tracked points without depth information. Fig. 7.4 shows an overview of the LPVO algorithm.

7.4.2 Drift-Free Rotation Estimation with Lines and Planes

We extract the vanishing directions from lines in the RGB images and surface normal vectors from planes in the depth images to determine the camera orientation relative to the Manhattan frame. To extract the vanishing direction vectors [104], line features over a fixed length (in our experiments, 25 pixels) are detected using LSD [115]. Given the N detected line features, we compute their corresponding great circle unit normal vectors. From the N associated normal vectors, we calculate $\binom{N}{2}$ vanishing directions (one for every possible pair of lines) by taking the cross product of each pair of normal vectors.

We also extract surface normal vectors for every pixel point from the depth image with an RGB-D camera [106]. We pre-process the depth image with a simple box filter to remove noise in the raw depth data. Unit surface normal vectors on the Gaussian sphere $\mathbb{S}^2$ are computed by the cross product of two tangential vectors, which are tangential to the local surface at the 3D points in the point cloud. In order to remove noise from the tangential vectors, we average the surrounding tangential vectors within a certain neighborhood, which can be done efficiently and quickly using integral images. For further details, see [106].

We represent the extracted vanishing directions and surface normal vectors as 3D points on the concentric spheres $\mathbb{S}^2$ in Fig. 7.3. Purple points on the inner sphere denote the VDs from lines, and grey points on the outer sphere are the surface normal vectors from planes, showing that the two types of directional vectors from lines and planes gather together around the Manhattan frame axes. The number of the VDs and surface normals is constantly changing by various factors such as the number of detected lines, invalid per-pixel depth, and an environmental condition.

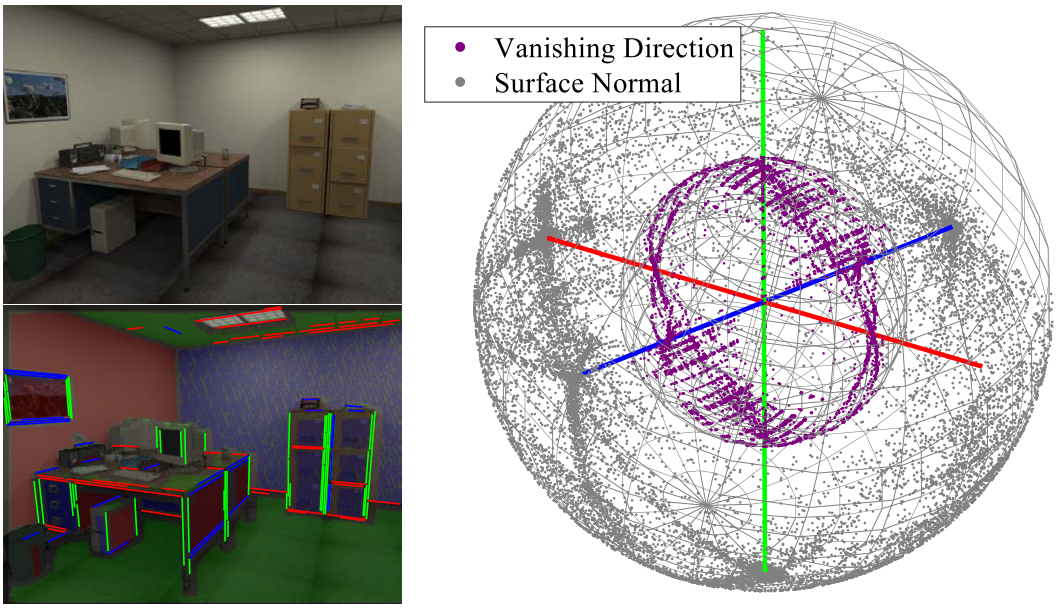


Figure 7.3: Clustered lines and segmented planes are overlaid on the RGB image.

Although most of the directional vectors in MW are distributed around the basis axes of the Manhattan frame, some other points are not near the Manhattan frame because the real 3D world is not a perfect and noise-free Manhattan world. Therefore, it is very important to extract accurate and reliable VDs and surface normal vectors since the density distribution of the directional vectors directly affects the accuracy of rotational motion estimation. LPVO enables MF tracking even when viewing only a single plane with the help of the lines, unlike the previous MF tracking methods [103, 114, 29].

7.4.3 Translation Estimation with All Tracked Points

We detect and track the feature points with the Good Features to Track [107] and KLT tracker [108] (for further details, see [29]). We recover the 3-DoF translational motion of the camera by minimizing the de-rotated reprojection error based on tracked points with and without depth information, which is only dependent on the translational movement. We start with the mathematical relationship between the frame-to-frame 6-DoF camera motion and the i-th tracked point fea-

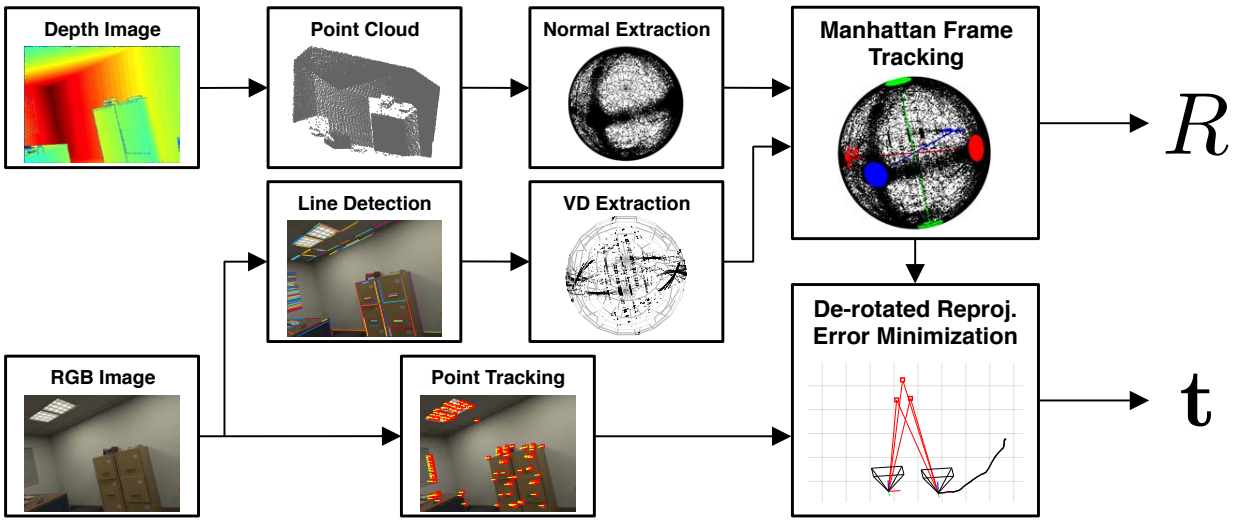


Figure 7.4: Overview of the proposed LPVO algorithm.

ture [17]:

$$\begin{aligned} \mathbf{X}_i^k = Z_i^k \bar{\mathbf{X}}_i^k &= R\mathbf{X}_i^{k-1} + \mathbf{t} \\ &= Z_i^{k-1} R\bar{\mathbf{X}}_i^{k-1} + \mathbf{t} \end{aligned} \tag{7.2}$$

where $\mathbf{X}_i^k = [X_i^k, Y_i^k, Z_i^k]^\top$ is the 3D coordinates of the point feature in the camera frame at time k, and $\bar{\mathbf{X}}_i^k = [\bar{X}_i^k, \bar{Y}_i^k, 1]^\top$ is the normalized $\mathbf{X}_i^k$ divided by the depth. The rotation R and the translation $\mathbf{t}$ form a rigid body transformation as explained in Section 7.3.2. For a feature with known depth at time $k-1$, we derive two constraint equations from the first row of Eq. (9.2) by substituting Z_i^k in the third row into the first and second rows, respectively:

$$\begin{aligned} r_{i_1}(\mathbf{t}) &= (R_1 - \bar{X}_i^k R_3)\mathbf{X}_i^{k-1} + \mathbf{t}_1 - \bar{X}_i^k \mathbf{t}_3 = 0 \\ r_{i_2}(\mathbf{t}) &= (R_2 - \bar{Y}_i^k R_3)\mathbf{X}_i^{k-1} + \mathbf{t}_2 - \bar{Y}_i^k \mathbf{t}_3 = 0 \end{aligned} \tag{7.3}$$

where R_h and $\mathbf{t}_h$, $h \in \{1, 2, 3\}$ are h-th rows of R and $\mathbf{t}$ respectively. For a feature with unknown depth, we can derive one constraint equation from the second row of Eq. (9.2) by combining all rows to eliminate both Z_i^k and Z_i^{k-1}:

$$r_i'(\mathbf{t}) = \mathbf{p} R\bar{\mathbf{X}}_i^{k-1} = 0 \tag{7.4}$$

$$\text{where } \mathbf{p} = \begin{bmatrix} -\bar{Y}_i^k \mathbf{t}_3 + \mathbf{t}_2 & \bar{X}_i^k \mathbf{t}_3 - \mathbf{t}_1 & -\bar{X}_i^k \mathbf{t}_2 + \bar{Y}_i^k \mathbf{t}_1 \end{bmatrix}$$

There are two residual equations for features with depth, and one residual equation for those without depth. Since we already estimated the drift-free rotational motion R, the residual terms in Eqs. (9.3) and (9.4) are only a function of the translational camera motion $\mathbf{t}$. The optimal 3-DoF translation motion, which minimizes the residual vectors of all tracked feature points with and without depth, can be obtained by solving the following optimization problem:

$$\mathbf{t}^* = \arg\min_{\mathbf{t}} \sum_{i=1}^{M} \left(r_{i_1}(\mathbf{t})\right)^2 + \left(r_{i_2}(\mathbf{t})\right)^2 + \sum_{i=1}^{N} \left(r'_i(\mathbf{t})\right)^2 \tag{7.5}$$

where M and N are the number of tracked features with known and unknown depth, respectively. We use the Levenberg–Marquardt (LM) algorithm for solving Eq. (9.5). By additionally constraining the 3-DoF translation from tracked points without depth, we can estimate more accurate translational motion compared to our previous approach. Note that the proposed method is less sensitive to the existence of enough textures and brightness in the image than the typical feature-based VO methods [15, 17] since the minimum number of points for estimating translation only is smaller than the number of feature points to determine both rotation and translation.

7.5 Evaluation

We evaluate LPVO on a variety of RGB-D datasets in man-made structured environments:

- *ICL-NUIM* [1] is a synthetic dataset consisting of a collection of RGB and depth images at 30 Hz captured in a living room and office with ground-truth camera poses. The synthesized RGB and depth images are corrupted by the modeled sensor noise to simulate typically observed real world artifacts. It is challenging to estimate the camera trajectory accurately due to low texture and frequent on-the-spot rotations.

- *TUM RGB-D* [2] is a famous dataset for VO evaluation, containing RGB-D images from a Microsoft Kinect RGB-D camera in various indoor environments. It is recorded in room-

scale environments with ground-truth trajectories provided by a motion capture system.

- *TAMU RGB-D* [111] contains RGB-D images at 30 Hz recorded in larger scale man-made environments like corridors and stairs inside a building.

- *Author-collected RGB-D dataset* consists of RGB and depth images at 30 Hz with an Asus Xtion Pro Live RGB-D camera in large building-scale indoor environments over 100 m traveling distance.

We compare the proposed LPVO method against other state-of-the-art VO algorithms, including indirect, direct, and hybrid methods, namely ORB [71], DEMO [17], DVO [6], MWO [103], and OPVO [29]. ORB, DEMO, and DVO estimate the rotational and translational motion jointly, while MWO and OPVO decouple the estimation of the rotational and translational motion like LPVO. Recall that the proposed LPVO builds on our previous work OPVO [29]. We deactivate the capability to detect loop closures via image retrieval in ORB for a fair comparison.

The proposed LPVO written in unoptimized MATLAB codes is able to run at 13.5 Hz on a desktop computer with an Intel Core i5 (3.20 GHz) and 8 GB memory, suggesting potential when implemented in C/C++ in the near future.

7.5.1 ICL-NUIM Dataset

Table 7.1: Evaluation Results on ICL-NUIM Benchmark

Experiment	LPVO	ORB	DEMO	DVO	MWO	OPVO	Length (m)
Living Room 0	**0.01**	0.02	0.14	0.22	×	×	4.14
Living Room 1	0.04	**0.03**	0.15	0.07	0.32	0.04	2.05
Living Room 2	**0.03**	0.07	0.62	0.50	0.11	0.06	8.42
Living Room 3	0.10	**0.07**	0.33	0.43	0.40	0.10	5.95
Office Room 0	**0.06**	0.20	0.34	0.37	0.31	0.06	6.53
Office Room 1	**0.05**	0.60	0.37	0.36	1.10	0.05	6.72
Office Room 2	**0.04**	0.30	0.76	0.58	×	×	9.01
Office Room 3	**0.03**	0.46	0.18	0.30	1.38	0.04	7.82

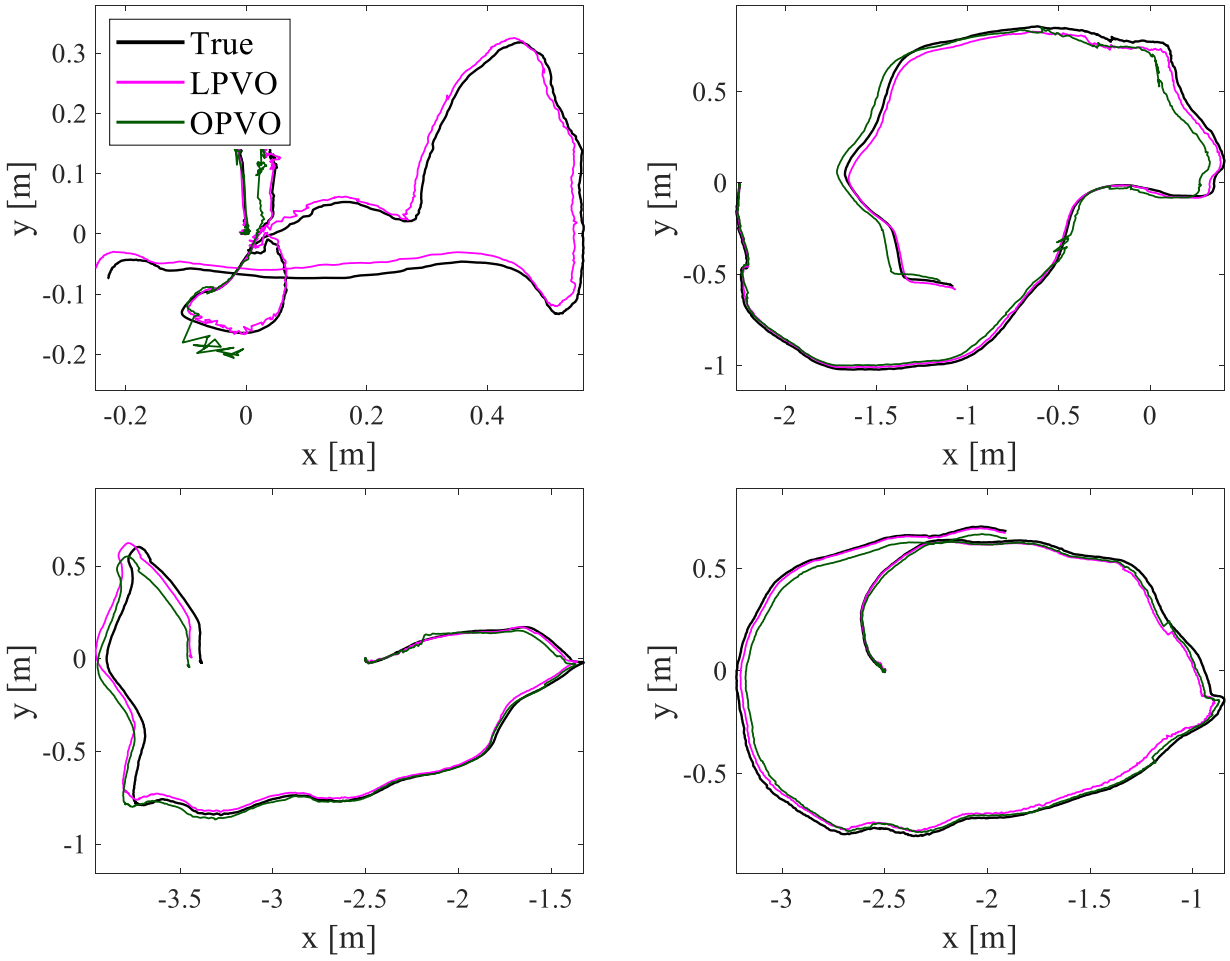


Figure 7.5: Estimated trajectories with LPVO (magenta), OPVO (dark green), and ground-truth (black) in the ICL-NUIM dataset Living Room 0, 2 and Office Room 1, 3.

We measure the root mean squared error (RMSE) of the absolute translational error and present the results in Table 9.1. The smallest error for each dataset is bolded. ORB results are from [25]. MWO and OPVO sometimes fail to track the camera (marked as $\times$ in Table 9.1) due to multiple orthogonal planes not always being visible. For example, in 'Living Room 0', at one point OPVO sees only a single plane, leading to failure. LPVO can continue estimating the motion stably as shown in the top left of Fig. 7.5. Our method outperforms the other VO algorithms for most test cases. In two cases, ORB performs better thanks to sufficient texture and local map construction, but the proposed algorithm performs nearly as well. The average translational RMSE of the proposed LPVO is 0.04, while ORB, DEMO, DVO, MWO, and OPVO are 0.21, 0.36, 0.35, 0.60, and 0.06, respectively. The main reason for the improved performance is that LPVO accurately tracks rotations even when the camera rotates in a place by exploiting both lines and surface normal vectors to recognize structural regularities. Although OPVO also estimates accurate camera rotation, it is unstable and fails when only a single plane is visible.

The strength of LPVO becomes clear when plotting the rotation and translation errors for the dataset 'Office Room 3' in Fig. 7.6. While the rotation error of ORB, DEMO, and DVO gradually increase over time, LPVO, MWO, and OPVO drift less than 0.5 degrees thanks to drift-free rotation estimation. The average rotation error of the proposed method is 0.22 degrees whereas ORB, DEMO, DVO, MWO, and OPVO are 1.63, 3.15, 8.12, 0.22, 0.21 degrees respectively. We can also observe that the translational error mainly occurs due to the drift of rotation estimate in the bottom of Fig. 7.6. Because there are sufficient orthogonal planes, MWO and OPVO can also estimate accurate and drift-free rotational motion. However, LPVO estimates more accurate translational motion by minimizing the de-rotated reprojection error from tracked points with and without depth information.

7.5.2 TUM RGB-D Dataset

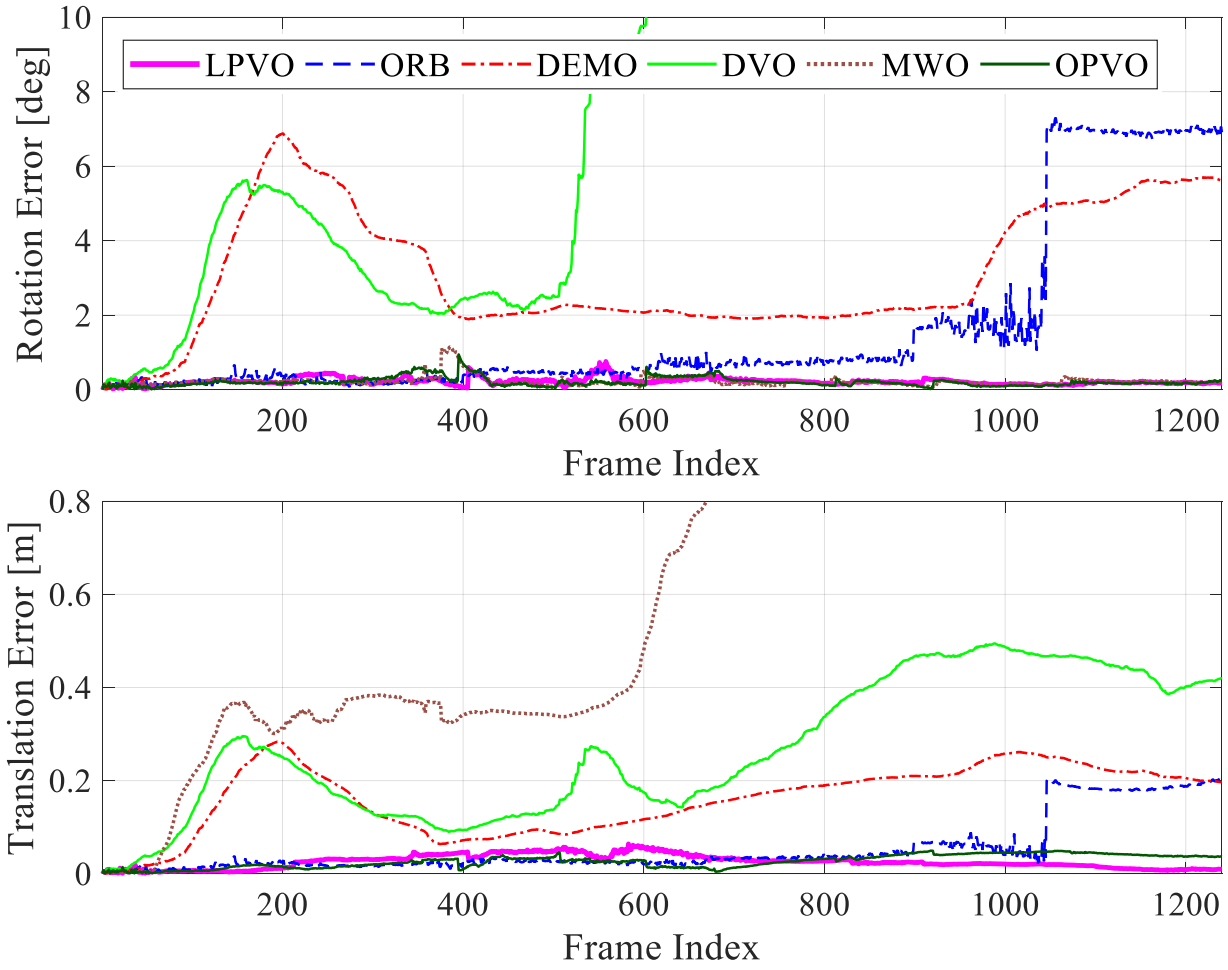


Figure 7.6: Absolute rotational error (top) and translational error (bottom) for the proposed and other VO algorithms are plotted.

Table 7.2: Evaluation Results on TUM RGB-D Benchmark

Experiment	LPVO	ORB	DEMO	DVO	MWO	OPVO	Length (m)
fr3_longoffice	0.19	**0.02**	1.50	0.61	×	×	22.14
fr3_struc_notex_far	**0.07**	0.28	0.40	0.59	0.47	0.13	1.66
fr3_struc_notex_near	**0.08**	0.63	2.59	0.73	0.95	0.16	2.05
fr3_struc_tex_far	0.17	**0.03**	0.06	0.13	1.57	0.18	6.04
fr3_struc_tex_near	0.11	**0.03**	0.20	0.08	0.62	0.19	5.21
fr3_large_cabinet	**0.28**	0.47	0.96	0.97	0.83	0.51	12.37

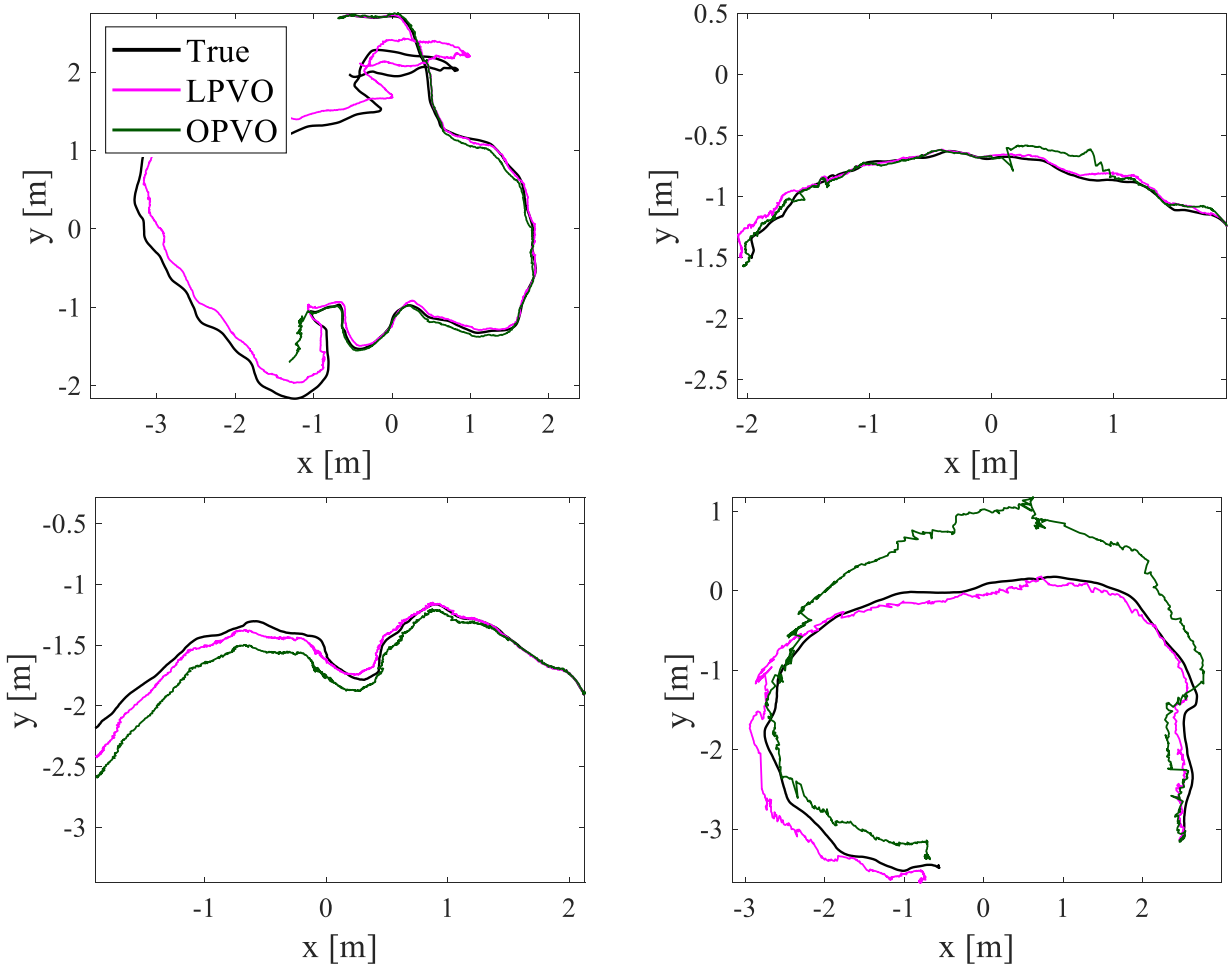


Figure 7.7: Estimated trajectories with LPVO (magenta), OPVO (dark green), and ground-truth (black) in the TUM fr3_longoffice, fr3_struc_notex_far, fr3_struc_tex_near, and fr3_large_cabinet.

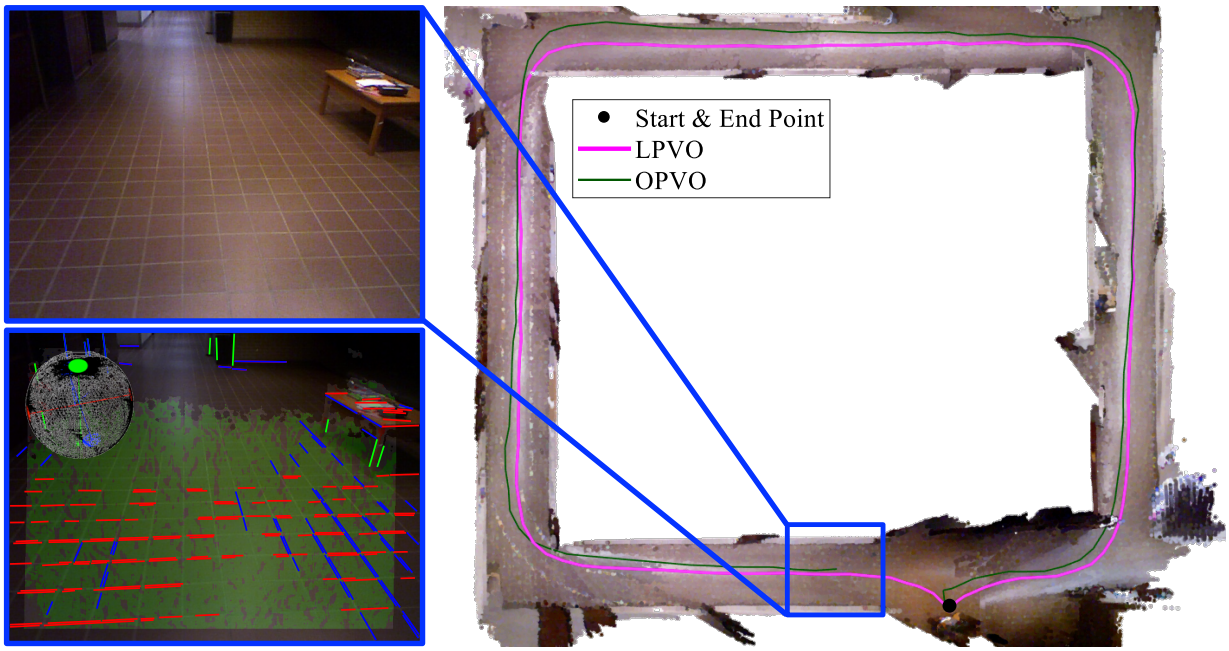


Figure 7.8: Example image from 'Corridor-A-const', the clustered lines/planes, and the inferred MF orientation are shown on the left.

We evaluate the motion estimation results on a subset of image sequences which contain sufficient structural regularities (lines and planes) in the observed scenes. Table 9.2 compares results of the VO methods. Estimated camera trajectories with the ground-truth, LPVO, and OPVO are shown in Fig. 7.7. We observe that ORB outperforms the proposed algorithm in incomplete (ambiguous) structured environments such as 'fr3_longoffice'. However, LPVO shows better performance in very low texture environments with the help of structural information. LPVO can also work in imperfect structural environments like 'fr3_longoffice' whereas MWO and OPVO require at least two orthogonal planes throughout the entire motion estimation process. When there is only a single plane visible, OPVO fails but the proposed method does not as shown on the top left of Fig. 7.7.

7.5.3 TAMU RGB-D Dataset

We present a 3D reconstruction result of 'Corridor-A-const' in the TAMU dataset based on the motion estimation of LPVO in Fig. 7.8. The trajectory is about 88 meters long, and includes four

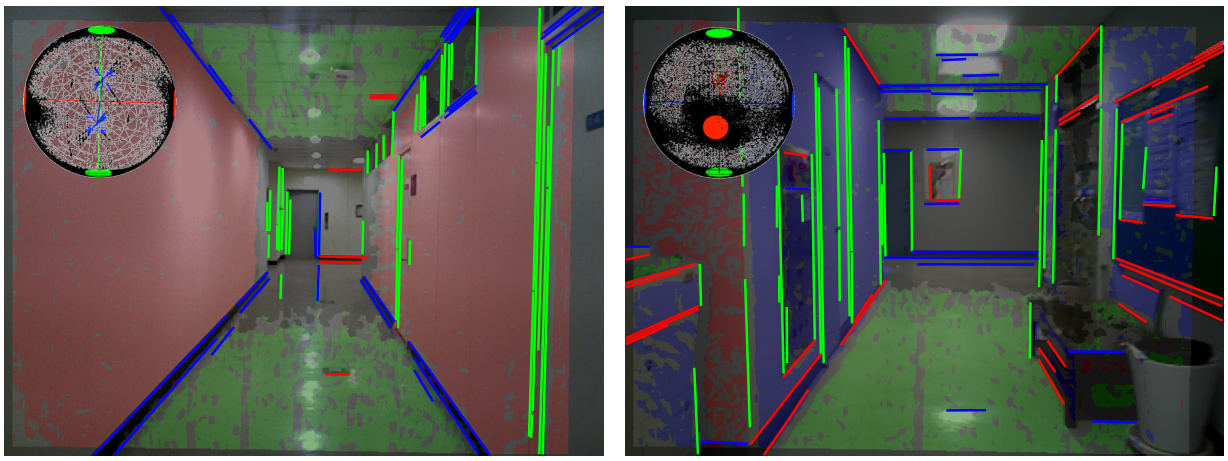


Figure 7.9: Example images from the author-collected RGB-D dataset.

pure rotational movements, difficult textures, and a segment where the camera looks at only a single plane as shown on the left of Fig. 7.8. LPVO stably tracks the 6-DoF camera motion even when looking at only a single plane while OPVO fails to estimate rotational motion of a camera, resulting in overall motion estimation failure. Therefore, LPVO accurately estimates the entire camera motion, and achieves final drift error lower than 0.3%, which is the final positioning error divided by the total traveling distance. The start and end points of the estimated camera trajectory accurately meet. The drift-free rotation estimates act like an indoor 3-DoF compass in the long square corridor, resulting in consistent and low-drift motion estimation. Our method preserves the orthogonality of the reconstructed 3D point cloud well, which is rendered by back-projecting the depth image from the estimated camera poses. Note that we do not perform any additional SLAM techniques like 3D local map fusion, loop detection & closure, and relocalization, but the consistent 3D reconstruction result indicates the high accuracy of our VO approach.

7.5.4 Author-collected RGB-D Dataset

Finally, we demonstrate that the proposed VO method can work in building-scale indoor environments like long corridors. Fig. 7.9 shows excerpts from the 'single-loop' (left) and 'multiple-loop' (right) datasets, with trajectory lengths of 93 m and 120 m respectively. The dataset was taken on long square corridors of two different buildings, and is very challenging due to frequent on-

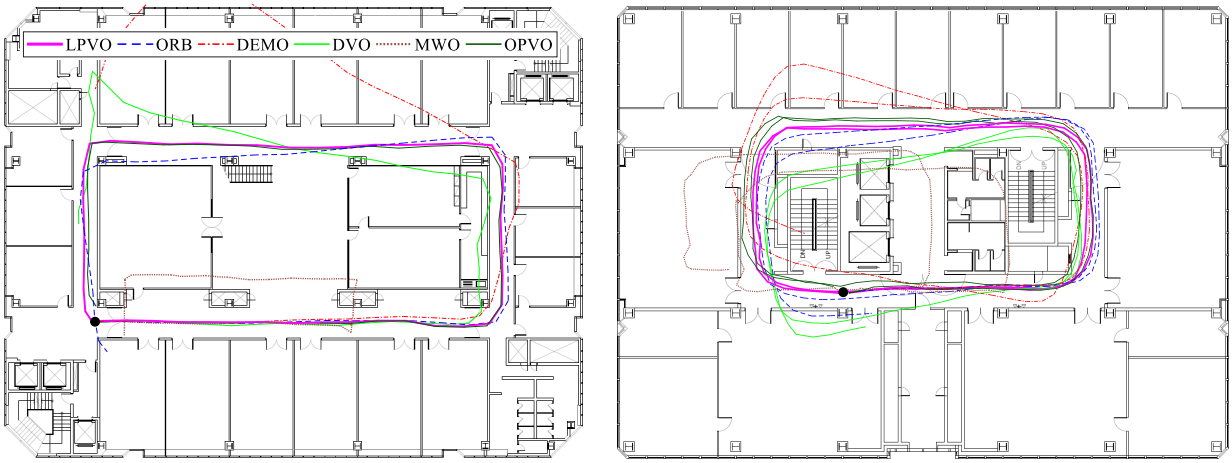


Figure 7.10: Estimated trajectories with the proposed and other VO methods on the author-collected dataset in a single-loop (left) and multiple-loop (right) sequences.

the-spot rotations and difficult textures. For evaluating the VO algorithms without a ground-truth trajectory, we collect the dataset on closed-loop trajectories where the starting and end points coincide.

The resulting trajectories for all algorithms are shown in Fig. 7.10. With LPVO, the starting and ending points nearly match; for the others, they do not. LPVO's final drift error is under 0.2%. LPVO robustly and accurately tracks the 6-DoF camera motion, preserving the orthogonality of the estimated corridor trajectory in the square building.

Similarly, for the 'multiple-loop' dataset (see right of Fig. 7.10) the start and end points meet only with the proposed algorithm, with final drift error under 0.7%. Although MWO and OPVO can perform drift-free rotation estimation, inaccuracies in translational motion estimation as discussed in Section 9.4.1 cause errors to accumulate. The reconstructed trajectory with the proposed method preserves the orthogonality of the corridors in the square building, demonstrating the high quality of the motion estimation. Please refer to the video clips submitted with this paper showing more details about the experiments.

Please refer to the video clips submitted with this paper showing more details about the experiments.[1]

[1]Video available at `https://youtu.be/mt3kbv2TJZw`

7.6 Conclusion

We propose a new visual odometry algorithm that is able to perform accurate and low-drift motion estimation in structured environments by decoupling the camera motion into a separate rotation and translation estimation. We newly exploit line and plane primitives together to deal with the degenerate case in the previous drift-free rotation estimation methods, resulting in stable and accurate zero-drift rotation estimation. Given the absolute camera orientation, we recover the optimal translational motion, which minimizes de-rotated reprojection error based on all tracked points with and without depth. The proposed algorithm is tested thoroughly with a large number of datasets, and shows accurate and low-drift motion estimation results in structural environments. Our method is currently tested with an RGB-D camera in indoor environments. In the future, we will try to implement the proposed algorithm with a stereo camera and possibly extend to outdoor urban environments.

8

Indoor RGB-D Compass from a Single Line and Plane

Authors	Pyojin Kim[1]	rlavywls@snu.ac.kr
	Brian Coltin[2]	brian.j.coltin@nasa.gov
	H. Jin Kim[1]	hjinkim@snu.ac.kr
	[1] Seoul National University	
	[2] NASA Ames Research Center	

Publication **Indoor RGB-D Compass from a Single Line and Plane.** Kim, Pyojin, Brian Coltin, H. Jin Kim. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2018*.

Contribution	Problem definition	*significantly contributed*
	Literature survey	*significantly contributed*
	Method development	*significantly contributed*
	Implementation	*significantly contributed*
	Experimental evaluation	*significantly contributed*
	Preparation of the manuscript	*significantly contributed*

Abstract We propose a novel approach to estimate the three degrees of freedom (DoF) drift-free rotational motion of an RGB-D camera from only a single line and plane in the Manhattan world (MW). Previous approaches exploit the surface normal vectors and vanishing points to achieve accurate 3-DoF rotation estimation. However, they require multiple orthogonal planes or many consistent lines to be visible throughout the entire rotation estimation process; otherwise, these approaches fail. To overcome these limitations, we present a new method that estimates absolute camera orientation from only a single line and a single plane in RANSAC, which corresponds to the theoretical minimal sampling for 3-DoF rotation estimation. Once we find an initial rotation estimate, we refine the camera orientation by minimizing the average orthogonal distance from the endpoints of the lines parallel to the MW axes. We demonstrate the effectiveness of the proposed algorithm through an extensive evaluation on a variety of RGB-D datasets and compare with other state-of-the-art methods.

8.1 Introduction

Camera orientation estimation from a sequence of images is a fundamental problem for many applications in computer vision [116, 117] and robotics [96, 98]. Recent visual odometry (VO) and visual simultaneous localization and mapping (V-SLAM) methods [26, 25, 15] have shown promising results in estimating camera orientation from a variety of video sequences. However, these approaches cannot avoid drift error in the rotation estimate without computationally expensive SLAM techniques (loop closure, global 3D map construction).

Several studies [113, 118, 98, 119] have focused on accurate and drift-free rotation estimation in urban and indoor scenes consisting of parallel and orthogonal lines and planes, called the Manhattan world (MW) [102]. They exploit structural regularities to achieve accurate 3-DoF rotation estimation by using the distribution of surface normal vectors and points at infinity, i.e., vanishing points (VPs). The accuracy of VO has been improved dramatically in [103, 29, 30] by using the MW assumption in rotation estimation. Although they can estimate the rotational motion of the camera accurately by exploiting significant structural organization, there are still some

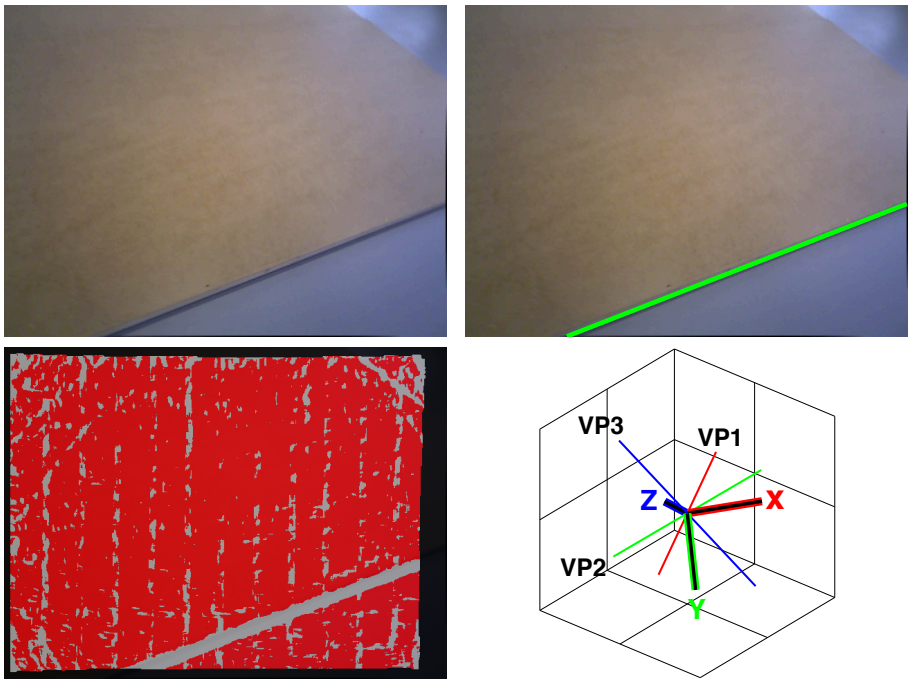


Figure 8.1: A single line and a single plane from RGB-D images.

problems: multiple orthogonal planes or many consistent lines must remain visible throughout the entire video sequence. In practice, robots often encounter harsh environments where there are insufficient structural regularities (see Fig. 8.1 and Fig. 8.7), resulting in the failure of rotation estimation or a loss of accuracy.

To address these issues, we propose a novel approach that estimates absolute 3-DoF camera orientation from only a single line and plane to recognize the spatial regularities of structural environments as shown in Fig. 8.1. We detect and track the normal vector of a plane from the depth image in order to determine two orientation angles of the 3-DoF rotation. The remaining orientation angle is computed with a line from the RGB image, which lies on the plane and is parallel to the MW axes. We incorporate the 3-DoF rotation estimation from only a single line and plane into the model estimation step of the RANSAC, which is the minimal solution for rotation estimation [113]. Furthermore, we refine the initial rotation estimate by minimizing the average orthogonal distance from the multiple lines, which are parallel to the MW axes. Our algorithm

requires a plane and a line on the plane aligned with the MW to be visible, which is typically the case in most indoor environments.

Extensive evaluations show that the proposed method produces accurate and drift-free camera orientation on a variety of video sequences compared to other state-of-the-art approaches. The contributions of this work are as follows:

- We present a novel approach to estimate accurate and drift-free 3-DoF camera orientation from only a single line and plane in the RANSAC framework.
- We refine the initial rotation estimate with the parallel and orthogonal lines to obtain a more accurate 3-DoF camera orientation.
- We evaluate the proposed algorithm on the ICL-NUIM [1] and TUM [2] RGB-D datasets, showing robust, stable, and accurate performance.

8.2 Related Work

The use of the Manhattan world (MW) estimation for determining the orientation of a camera has been studied previously due to its importance in high-level vision applications such as 3D reconstruction and scene understanding. The approaches for understanding the structural regularities in man-made environments can be classified into either estimating the VPs from the intersection of multiple parallel lines in the image or estimating the principal normal vectors of the surface with 3D information in depth image.

A VP, which is invariant to camera translation movements, has been widely used for tracking the rotational motion of the camera accurately [104, 120, 113]. In [118, 113], Manhattan frame (MF) estimation is performed based on three lines with two of the lines parallel and orthogonal to the third in RANSAC [121], which is the minimal sampling for rotation estimation. The method in [122] finds a triplet of orthogonal vanishing points with RANSAC-based line clustering to track the camera orientation along a video sequence in real-time. [105] jointly estimates the VPs and camera orientation based on sequential Bayesian filtering without the MW assumption.

These VP-based methods, however, are not robust and stable in the presence of spurious or noisy line segments. A sufficient number of parallel and orthogonal lines should exist in the image for accurate and reliable rotation estimation.

Recent studies have utilized 3D information to estimate dominant orthogonal directions in a MW from a depth sensor like a Kinect camera. [98, 123] propose real-time maximum a posteriori (MAP) inference algorithms for estimating MW in the surface normal distribution of a scene on a GPU. The method in [103, 29] estimates drift-free camera orientation with an efficient SO(3)-constrained mean shift algorithm given the surface normal vector distribution. In [124, 119], a branch-and-bound (BnB) strategy is employed to guarantee the globally optimal Manhattan frame estimation. While these approaches based on the surface normals demonstrate more stable and accurate rotation estimation results than VP-based methods, at least two orthogonal planes must be observable in the depth images.

Prior research has used the connection between VPs in the RGB image and 3D information from depth image to perform MW estimation. Given an a priori known normal vector of the horizon plane, an estimate of the camera orientation is performed with additional line segments in RANSAC [113]. The method in [30] tracks the MW utilizing both lines and planes together, but it requires a sufficient number of lines in the image. [125] estimates global geometry of indoor MW environments by integrating RGB images with associated depth data.

8.3 Proposed Method

We propose a new method for estimating a drift-free 3-DoF rotational motion of an RGB-D camera from the RGB and depth image pairs. For each pair of RGB-D images, we perform two steps: 1) estimate the absolute camera orientation with respect to the MF using only a single line and plane in RANSAC [121]; and 2) refine this initial rotation estimate with parallel and orthogonal lines from inliers. An overview of the proposed algorithm is shown in Fig. 8.3.

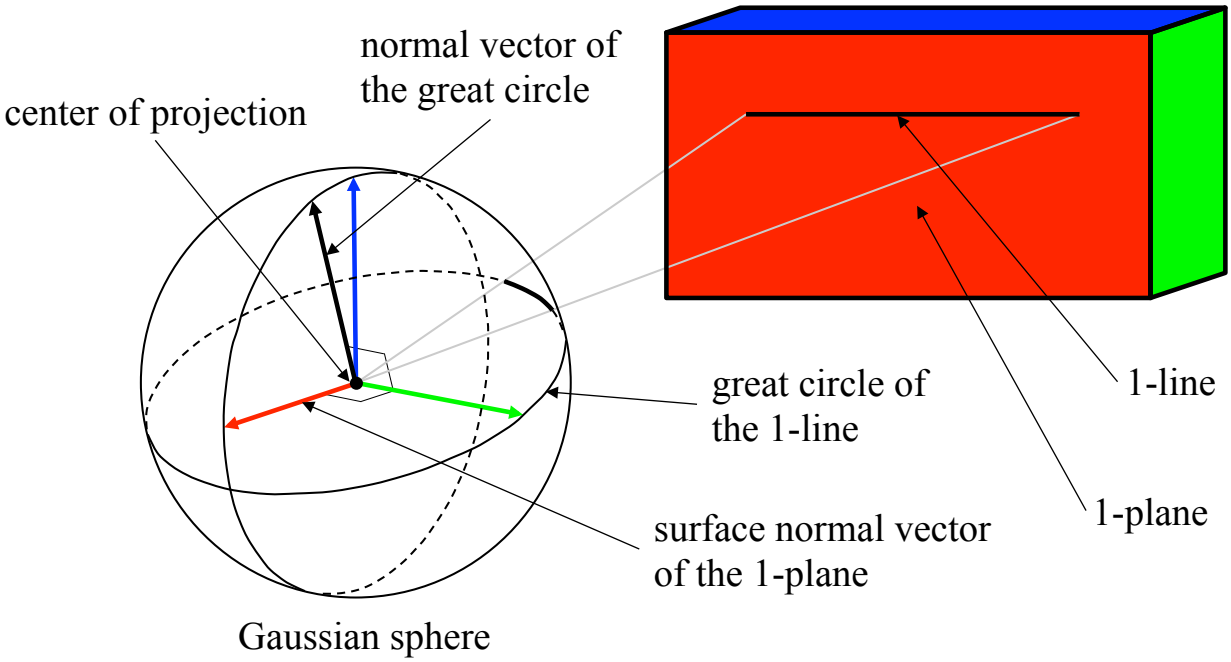


Figure 8.2: Geometric relationships between the line, plane, and the Gaussian sphere in the MW.

8.3.1 Dominant Plane Detection and Tracking

We first detect a dominant plane in the current environment from a depth image's 3D point cloud with a RANSAC algorithm [126]. The algorithm first randomly selects three points and computes the model parameters (normal vector) of the corresponding plane. It then checks the number of inliers exceed a given threshold by calculating the distance between the 3D points and the plane. It repeats these first two steps until it finds the best (dominant) 3D plane supported by the largest number of inliers.

We track the normal vector of the dominant plane with a mean shift algorithm based on the tangent space Gaussian MF (TG-MF) model [123] given the density distribution of surface normal vectors on the Gaussian sphere $\mathbb{S}^2$ [103] in Fig. 8.4. The unit surface normal vector of each pixel is calculated by taking the cross product of two tangential vectors at the 3D points in the point cloud. To obtain the noiseless tangential vectors for stable surface normal vectors, we average the surrounding tangential vectors within a certain neighborhood, which can be done

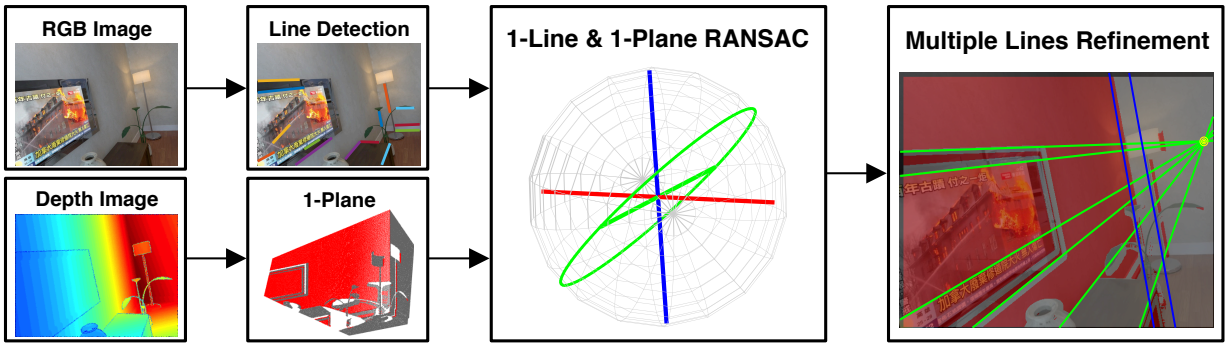


Figure 8.3: Overview of the proposed algorithm.

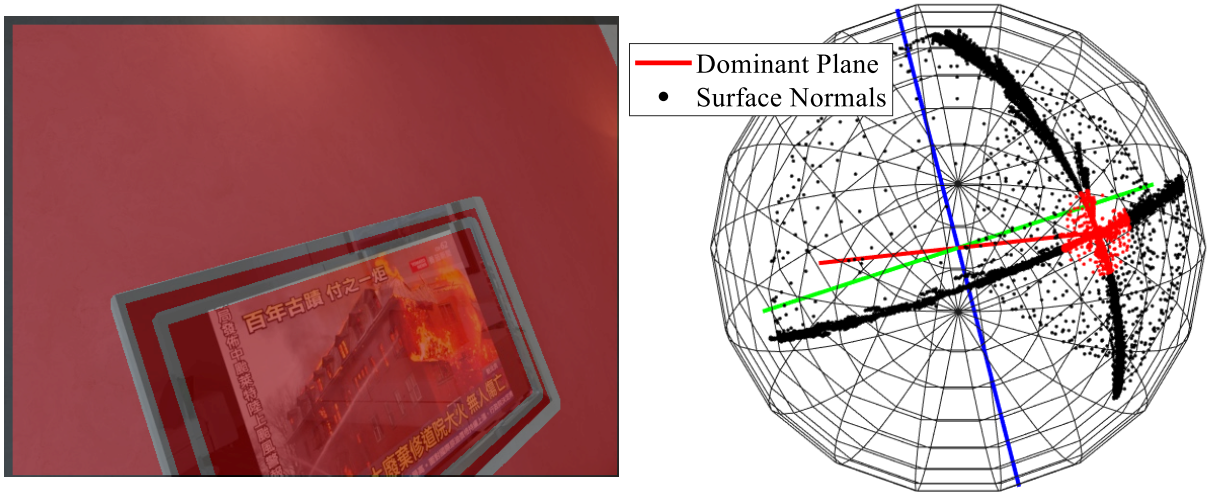


Figure 8.4: The normal vector of a dominant plane from the distribution of the surface normals.

efficiently and quickly using integral images [106]. Unlike the previous approaches [103, 29], we only track a single normal vector of the dominant plane as shown in Fig. 8.4. By using the tracked (detected) normal vector from the previous frame as an initial value, we perform the mean shift algorithm in the tangent plane $\mathbb{R}^2$ of the Gaussian sphere $\mathbb{S}^2$ with a Gaussian kernel (for full details, refer to [103]). Although we could use RANSAC to discover a dominant plane for every frame, tracking is less expensive and makes a smoother estimate.

If the density distribution of the surface normal vectors around the currently tracked normal vector is too low, we re-initialize and detect a new dominant plane again with plane model-based

RANSAC. We assign the normal vector of the new dominant plane to the closest axis in the MF under the assumption that the MF does not change too much between subsequent frames. There are 24 possible representations for the same MF orientation; we convert the matrices representing the MF into a unique canonical form [103] for consistent tracking.

8.3.2 One Line and One Plane RANSAC

Our approach utilizes line and plane geometric features, which provide the theoretical minimal solution for 3-DoF rotation estimation [113] as illustrated in Fig. 8.2. A plane provides two constraints on the two orientation angles, and the remaining orientation angle θ is constrained by a line. Once a dominant plane and a parallel line lying on the corresponding plane are found in the MW, they can determine the orientation of the Manhattan structure uniquely. By using this geometric feature, we estimate the 3-DoF drift-free rotational motion of the camera with respect to the MW in the RANSAC framework.

We detect N line segments using LSD [115], and calculate their corresponding unit normal vectors of great circles on the Gaussian sphere $\mathbb{S}^2$. Each RANSAC iteration starts by randomly selecting one great circle among N line segments. Given the tracked normal vector (the first VP v_1) of the dominant plane from the previous Section 8.3.1, we take cross product between the first VP and the normal vector of the selected great circle to define the second VP v_2. The third VP v_3 (blue) is automatically determined by the cross product of the first VP (red) and the second VP (green) as shown in Fig. 8.2.

To evaluate the currently estimated orientation in RANSAC, we use the average orthogonal distance in the image plane as illustrated in Fig. 8.5 [33], which is a function of lines and camera orientation (VPs). The average orthogonal distance can be computed from the endpoints of the line l to an auxiliary line $\hat{l}$, which passes through the closest VP and the middle point of the line l as follows:

$$d_{i,k} = \left(d_{i1,k} + d_{i2,k}\right)/2 \tag{8.1}$$

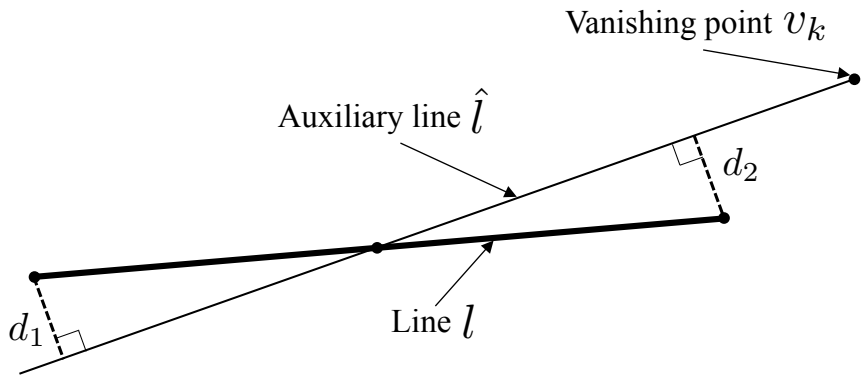


Figure 8.5: Orthogonal distance metric from the endpoints of the line l to an auxiliary line $\hat{l}$.

$$\text{where} \quad d_{i1,k} = \frac{|A_{i,k}u_{i1} + B_{i,k}v_{i1} + C_{i,k}|}{\sqrt{A_{i,k}^2 + B_{i,k}^2}}$$

where $A_{i,k}, B_{i,k}, C_{i,k}$ are the auxiliary line parameters of the i-th line segment with the k-th VP, and u_{i1}, v_{i1} is the first endpoint of the i-th line segment in the image plane.

Unlike the typical RANSAC algorithm in [113] which uses only the number of inliers, we find the largest consensus line set utilizing not only the average orthogonal distance $d_{i,k}$ but also the length of a line segment [127]:

$$vote\left(v_k\right) = \sum_{i=1}^{M_k} w_1 \left(1 - \frac{d_{i,k}}{t_a}\right) + w_2 \left(\frac{length(l_i)}{max(length(l))}\right) \quad (8.2)$$

where M_k, $k \in \{2, 3\}$ is the number of associated line segments for each VP v_2 and v_3, respectively. $d_{i,k}$ and t_a are the average orthogonal distance of the i-th line segment with the k-th closest VP and a certain threshold defined by user (in our experiments, 1 pixel). The i-th line length relative to the maximum line length is also considered in the second term in Eq. (9.2) because the longer the lines are, the more reliable they are. The weights w_1 and w_2 denote the importance of each term, the orthogonal distance and the line length, respectively (in our experiments, 0.7 and 0.3). When we calculate the vote value in Eq. (9.2), we do not use the line segments parallel to the tracked normal vector of the dominant plane (the first VP v_1) because the plane normal tracking on the surface normal vector distribution is quite accurate [98, 29]. We find the lines and the cor-

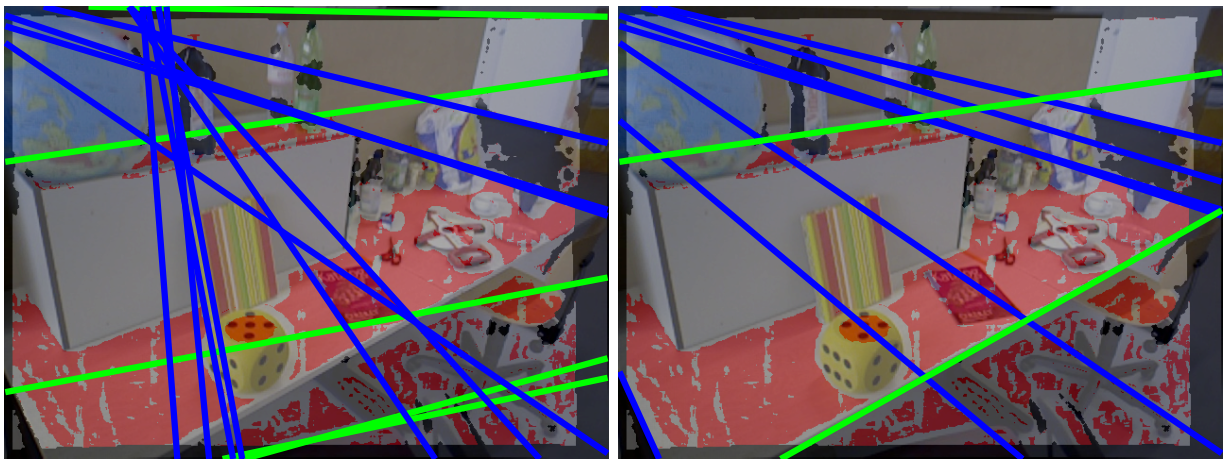


Figure 8.6: Improved performance to recognize the regularities of structural space.

responding camera orientation (VPs) leading to the highest total vote sum value from Eq. (9.2). It is noteworthy that the proposed one line and one plane RANSAC is computationally efficient since the number of required line sample to perform model estimation is only one similar to [39]. As the number of RANSAC iterations (computational complexity) exponentially increases depending on the number of required samples [33], using only one line sample to estimate the model makes our algorithm computationally inexpensive.

Fig. 8.6 shows the effectiveness of the continuous criteria in Eq. (9.2) compared to the standard RANSAC. When we try the standard RANSAC, it sometimes fails because it only considers whether the average orthogonal distance is smaller or larger than a certain threshold dichotomically. If there are many spurious or noisy lines, it cannot recognize the structural regularities correctly in the left of Fig. 8.6. We can find the correct inlier line set by using the continuous criteria written in Eq. (9.2) in the right of Fig. 8.6.

Our approach can fail when there is not any line that is parallel to the MW axes, or we cannot find any valid lines because of extreme motion blur. In other words, for our algorithm to succeed, we must have at least: 1) a plane from the depth image; and 2) a line from the RGB image, which lies on the plane and is parallel to the MW axes. While these geometric conditions may seem restrictive, our extensive experiments on multiple datasets in the Section 8.4 show that they

often hold in most structural indoor environments, and our approach achieves better accuracy and demonstrates the effectiveness.

8.3.3 Multiple Lines Refinement

The initial rotation estimate from only a single line and plane in the previous RANSAC step can be affected by noise in the line segments, resulting in suboptimal rotation estimation. To estimate more accurate and optimal camera orientation, we further refine the initial rotation estimation from the single line and plane RANSAC by minimizing the average orthogonal distance with parallel and orthogonal lines in inliers.

Since the tracked normal vector of the dominant plane on the surface normal vector distribution is relatively accurate [98, 29], the cost function, which is the average orthogonal distance written in Eq. (9.1), is only a function of the remaining one orientation angle θ constrained by multiple inlier lines. We express the 3-DoF camera orientation (VPs) as the axis-angle representation where the direction of an axis of rotation is the tracked unit normal vector of the dominant plane, and the magnitude of the rotation about the axis is the remaining orientation angle θ. The optimal drift-free camera orientation, which minimizes the orthogonal distance of all parallel and orthogonal inlier lines found in the RANSAC, can be obtained by solving the following optimization problem:

$$\theta^* = \arg\min_{\theta} \sum_{k=2}^{3} \sum_{i=1}^{M_k} \left(d_{i,k}\left(\theta\right)\right)^2 \tag{8.3}$$

where M_k, $k \in \{2, 3\}$ is the number of parallel or orthogonal lines related to the k-th VP counted in the RANSAC as inliers. $d_{i,k}(\theta)$ denotes the orthogonal distance of the i-th line segment with the k-th VP in the image space. We use the Levenberg–Marquardt (LM) algorithm for solving Eq. (9.3). By additionally constraining the remaining orientation angle θ from the parallel and orthogonal lines found in RANSAC, we can estimate more accurate and consistent rotational motion compared to the initial rotation estimate directly from the RANSAC process.

Note that the first RANSAC step (Section 8.3.2) and the second (Section 8.3.3) optimization of the algorithm seem to be redundant as both estimate the rotational motion of the camera.

Figure 8.7: Examples of the MW from the ICL-NUIM [1] and TUM RGB-D [2] datasets.

The additional refinement step, however, makes the estimated camera orientation more accurate and consistent by utilizing multiple lines. We validate the effect of the refinement in the next evaluation section.

8.4 Evaluation

We evaluate the proposed approach on a variety of RGB-D video sequences in man-made structural environments:

- *ICL-NUIM* [1] is a synthetic dataset consisting of a collection of RGB and depth images at 30 Hz captured in a living room and office with ground-truth camera orientation. The synthesized RGB and depth images are corrupted by the modeled sensor noise to simulate typically observed real-world artifacts. It is challenging to estimate the accurate 3-DoF camera rotation throughout the entire video sequences due to very low texture and a single

Experiment	Proposed	GOME	OLRE	OPRE	ROVE	# of frame
Living Room 0	**0.31**	×	×	×	×	1507
Living Room 1	**0.38**	8.56	3.72	0.97	26.74	965
Living Room 2	**0.34**	8.15	4.21	0.49	39.71	880
Living Room 3	**0.35**	×	×	1.34	×	1240
Office Room 0	0.37	5.12	6.71	**0.18**	29.11	1507
Office Room 1	0.37	×	×	**0.32**	34.98	965
Office Room 2	0.38	6.67	10.91	**0.33**	60.54	880
Office Room 3	0.38	5.57	3.41	**0.21**	10.67	1240

Table 8.1: Comparison of the absolute rotation error (degrees) on ICL-NUIM benchmark [1].

plane as shown in Fig. 8.7.

- *TUM RGB-D* [2] is a famous dataset for VO/V-SLAM evaluation, containing RGB-D images from a Microsoft Kinect RGB-D camera in various indoor environments as shown in Fig. 8.7. It is recorded in room-scale environments with ground-truth camera trajectories provided by a motion capture system.

We compare the proposed algorithm against other state-of-the-art 3-DoF camera orientation estimation methods using lines and planes, namely GOME [119], OLRE [113], OPRE [103], and ROVE [105]. GOME and OPRE estimate the drift-free rotational motion of the camera by tracking the distribution of the surface normal vectors from the depth images, while OLRE and ROVE utilize many consistent line features from the RGB images to estimate the camera orientation. The proposed method, GOME, OLRE, and OPRE rely on the MW assumption whereas ROVE does not require the MW in the scene.

8.4.1 ICL-NUIM Dataset

We measure the mean value of the absolute rotation error (ARE) [103] in degrees, and present the evaluation results in Table 9.1. The smallest rotation error for each dataset is bolded. Other methods using only multiple lines or planes sometimes fail to track the camera orientation (marked as

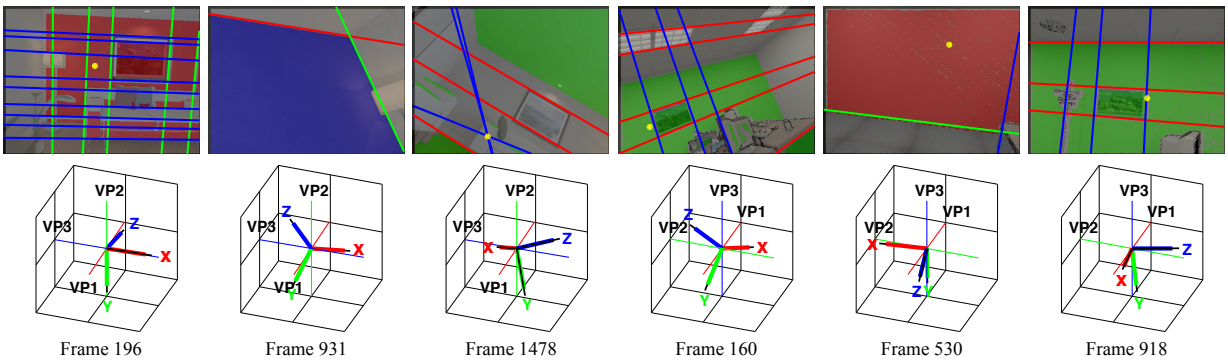


Figure 8.8: Experimental results in the ICL-NUIM dataset (a) 'Living Room 0' and (b) 'Office Room 1'.

$\times$ in Table 9.1) due to multiple lines or orthogonal planes not always being visible throughout the entire video sequences. In 'Living Room 0', at one point the camera sees only a single plane with very low texture, leading to failure of other approaches. The proposed method can continue tracking the absolute camera orientation stably and accurately as shown in Fig. 8.8. The tracked normal vector of the dominant plane is changed depending on the current situation.

Our approach outperforms the other methods for most cases. In 'Office Room' environments, OPRE performs slightly better thanks to sufficient surface normals distribution throughout the estimation period, but the proposed algorithm performs nearly as well. The average ARE of the proposed method is 0.36 degrees, while GOME, OLRE, OPRE, and ROVE are 6.82, 5.79, 0.55, and 33.63 degrees respectively. Since ROVE does not utilize the MW assumption, ROVE cannot estimate the drift-free camera orientation, resulting in accumulation of ARE over time. The main reason for the improved performance is that the proposed method can stably track the absolute rotations even when the camera sees only a planar surface with little texture by exploiting the minimal sampling (one line and one plane) to recognize structural regularities.

The advantage of the additional refinement step in the proposed method described in Section 8.3.3 becomes clear when plotting the ARE statistics from the dataset 'Living Room 1' in Fig. 8.9. We can observe that there are some large ARE (marked as a red cross) from the proposed method when the refinement step is not performed. The optimization with parallel and orthogo-

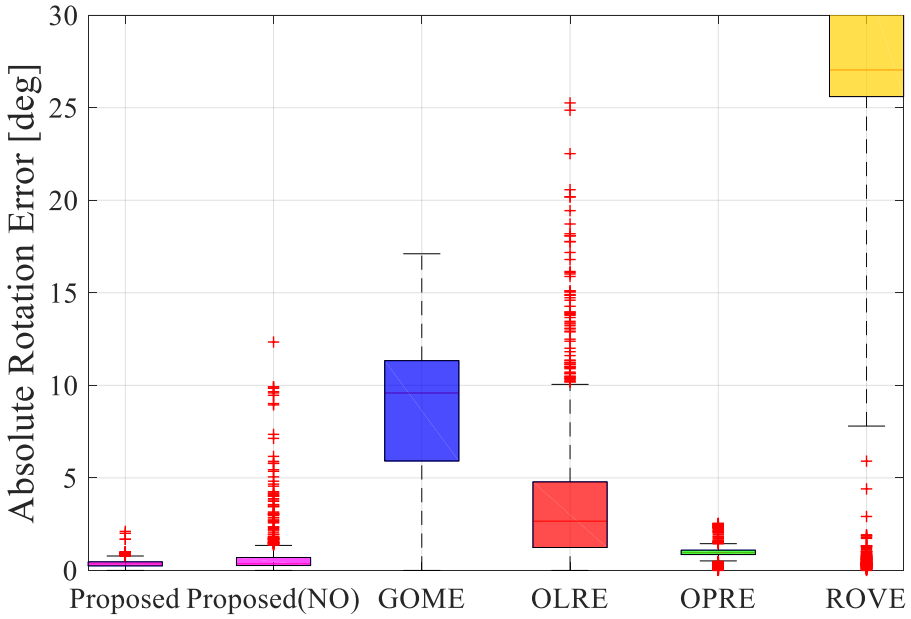


Figure 8.9: Comparison with and without refinement step (NO) versus the other algorithms.

nal lines found in the RANSAC as inliers enables to estimate the drift-free camera rotation more consistently and accurately.

8.4.2 TUM RGB-D Dataset

We evaluate the proposed and other algorithms on the video sequences of the TUM RGB-D dataset, which contain structural regularities (lines or planes) in the observed scenes. We also investigate the effect of the existence of structure and texture components in the scenes on the camera rotation estimation. Table 9.2 compares the average ARE results of the proposed and other methods. Our method can track accurate and drift-free camera rotational motion even in insufficient (imperfect) structural environments like 'fr3_longoffice' or 'fr3_nostruc_notex' as shown in Fig. 8.10. However, other approaches require at least two orthogonal planes (GOME, OPRE) or many consistent line segments (OLRE, ROVE) throughout the entire motion estimation process. While other methods are significantly affected by the presence or absence of the structure and texture components in the scenes, the proposed method shows accurate MW estimation not only

Experiment	Proposed	GOME	OLRE	OPRE	ROVE	# of frame
fr3_longoffice	**1.75**	×	×	4.99	×	2488
fr3_nostruc_notex	**1.51**	×	×	×	×	239
fr3_nostruc_tex	**2.15**	×	46.18	×	16.45	1639
fr3_struc_notex	**1.96**	4.07	11.22	3.01	×	794
fr3_struc_tex	**2.92**	4.71	8.21	3.81	13.73	907
fr3_cabinet	2.48	2.59	×	**2.42**	×	1112
fr3_large_cabinet	**2.04**	3.74	38.12	36.34	28.41	984

Table 8.2: Comparison of the absolute rotation error (degrees) on TUM RGB-D dataset [2].

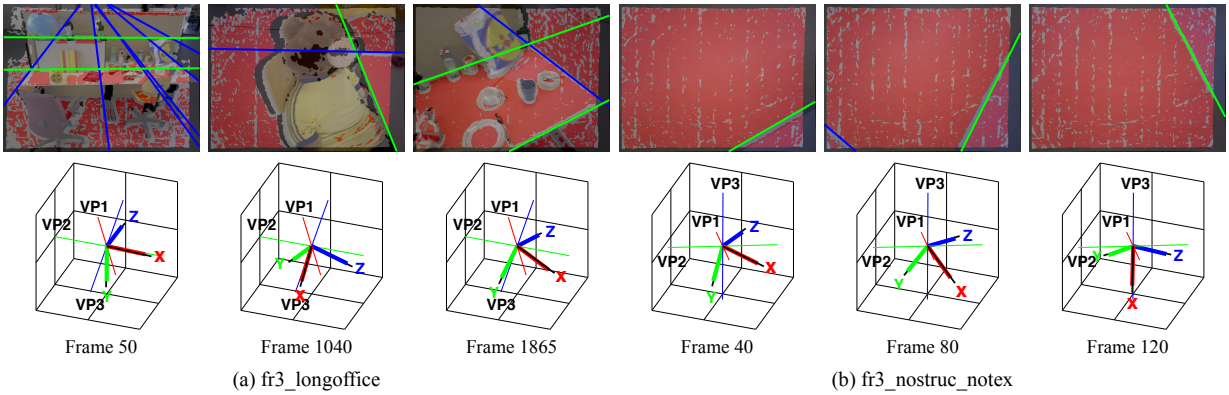


Figure 8.10: Inferred MW (VPs) orientation, the clustered lines and plane with the proposed method are overlaid on top of the RGB images in the TUM RGB-D dataset (a) 'fr3_longoffice' and (b) 'fr3_nostruc_notex'.

in abundant but also in very low structure and texture environments with the help of the minimal solution (one line and one plane).

We can also observe the effect of the refinement step in the proposed method by drawing the boxplot of the ARE from the dataset 'fr3_struc_tex' in Fig. 8.11. Outliers marked as red cross are removed, and the average ARE of the proposed method decreases thanks to the proposed additional refinement step.

Please refer to the video clips submitted with this paper showing more details about the experiments.[1]

[1] Video available at `https://youtu.be/qusvgMequqM`

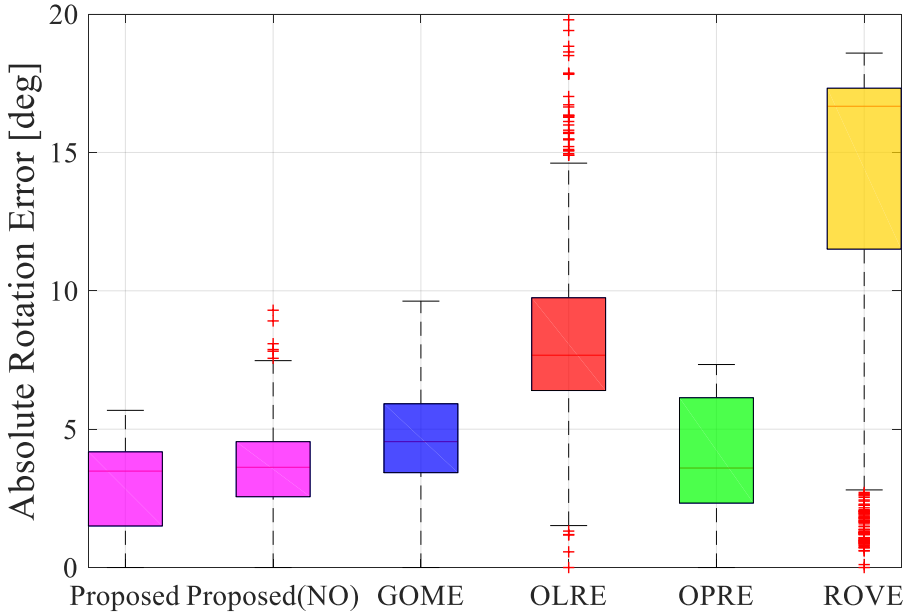


Figure 8.11: The statistical distribution of the absolute rotation error from the 'fr3_struc_tex'.

8.5 Conclusion

We propose a new method that is able to perform accurate and drift-free camera orientation estimation under insufficient structural environments by exploiting a single line and plane in RANSAC, which are the minimal solution for 3-DoF rotation estimation. We refine the initial rotation estimate by minimizing the average orthogonal distance from the endpoints of the parallel and orthogonal lines found in the RANSAC as inliers. The proposed algorithm is tested thoroughly with a large number of RGB-D datasets on the video sequences, and shows accurate and drift-free rotation estimation results in the environments where the structural regularities are challenging to find. Our method is currently tested with an RGB-D camera in indoor environments. In the future, we will try to implement the proposed algorithm with a stereo camera and possibly extend to outdoor urban environments.

9

Linear RGB-D SLAM for Planar Environments

Authors	Pyojin Kim[1]	rlavywls@snu.ac.kr
	Brian Coltin[2]	brian.j.coltin@nasa.gov
	H. Jin Kim[1]	hjinkim@snu.ac.kr
	[1] Seoul National University	
	[2] NASA Ames Research Center	

Publication	**Linear RGB-D SLAM for Planar Environments.** Kim, Pyojin, Brian Coltin, H. Jin Kim. In *Proceedings of European Conference on Computer Vision (ECCV), 2018*.

Contribution	Problem definition	*significantly contributed*
	Literature survey	*significantly contributed*
	Method development	*significantly contributed*
	Implementation	*significantly contributed*
	Experimental evaluation	*significantly contributed*
	Preparation of the manuscript	*significantly contributed*

Abstract We propose a new formulation for including orthogonal planar features as a global model into a linear SLAM approach based on sequential Bayesian filtering. Previous planar SLAM algorithms estimate the camera poses and multiple landmark planes in a pose graph optimization. However, since it is formulated as a high dimensional nonlinear optimization problem, there is no guarantee the algorithm will converge to the global optimum. To overcome these limitations, we present a new SLAM method that jointly estimates camera position and planar landmarks in the map within a linear Kalman filter framework. It is rotations that make the SLAM problem highly nonlinear. Therefore, we solve for the rotational motion of the camera using structural regularities in the Manhattan world (MW), resulting in a linear SLAM formulation. We test our algorithm on standard RGB-D benchmarks as well as additional large indoor environments, demonstrating comparable performance to other state-of-the-art SLAM methods *without* the use of expensive nonlinear optimization.

9.1 Introduction

Visual simultaneous localization and mapping (vSLAM) is the problem of estimating the six degrees of freedom (DoF) rotational and translational camera motion while simultaneously building a map of a surrounding unknown environment from a sequence of images. They are fundamental building blocks for various applications from autonomous robots to virtual and augmented reality (VR/AR).

Many typical visual RGB-D SLAM approaches such as DVO-SLAM [128] and ORB-SLAM2 [15], which are based on the pose graph optimization [129], have shown promising results in the environments with rich texture. However, they fare poorly in textureless scenes, which are commonly encountered in indoor environments with large planar structures [130]. They also rely on pose graph optimization methods, which are computationally expensive, and sometimes fail.

For working well in low-texture environments, recent visual SLAM methods [131, 132, 130] utilize additional geometric information like planar features. They combine plane measurements and scene layout with graph-based SLAM approaches [133, 134] to improve robustness and

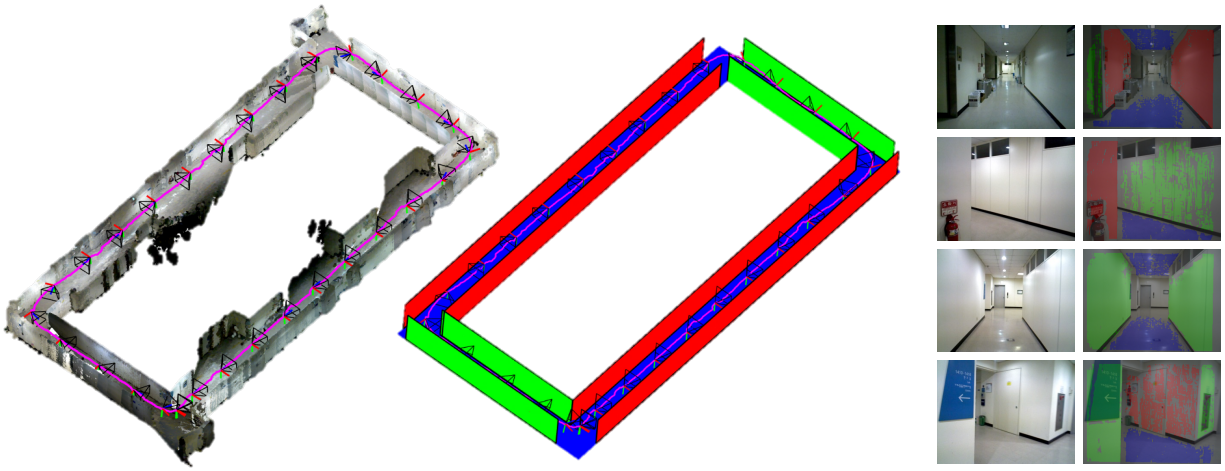


Figure 9.1: Linear RGB-D SLAM generates a consistent global planar map using a linear KF.

accuracy. Although these SLAM approaches show better accuracy for low-texture environments, there are some limitations: they are still dependent on the pose graph optimization, which is the non-convex and nonlinear optimization problem [112]. Since their SLAM is formulated as a high dimensional nonlinear optimization problem for jointly refining 6-DoF camera poses and multiple landmarks, there is no guarantee that the algorithm can converge to the global optimum [135]. Also, if the nonlinearity of pose graph optimization is too high due to the rotational components of the camera and the landmarks, they will fail to find the true solution.

To address these issues, we propose *Linear RGB-D SLAM* (L-SLAM), a novel method that jointly estimates camera position and planar landmarks in the map within a linear Bayesian filter as shown in Fig. 9.1. To separate the need for rotational motion estimation, which is a main source of nonlinearity in SLAM formulation, from the SLAM problem, we first track drift-free 3-DoF rotation and initial 3-DoF translational movement separately using Manhattan world (MW) assumption from VO algorithm [30]. Given the absolute camera orientation, L-SLAM identifies the horizontal and vertical planes in structured environments, and measures the distance to these orthogonal planes from the current camera pose at every frame. With the distance measurements from the orthogonal planes, we simultaneously update the 3-DoF camera translation and the 1-D distance of the associated global planes in the map within a linear Kalman filter (KF) framework. We present a simple, linear KF SLAM formulation by fully compensating for the 3-DoF rota-

tional camera motion obtained from [30], resulting in very low computational complexity while working well in textureless regions.

Extensive evaluations show that L-SLAM produces comparable estimation results compared to other state-of-the-art SLAM methods without expensive SLAM techniques (loop detection, pose graph optimization). Furthermore, we apply L-SLAM to augmented reality (AR) without any external infrastructure. We highlight our main contributions below:

– We develop an orthogonal plane detection method in structured environments when the absolute camera orientation is given.

– We propose a new, linear KF SLAM formulation for localizing the camera translation and mapping the global infinite planes.

– We evaluate L-SLAM on the RGB-D benchmark datasets from room-size to building-size with other state-of-the-art SLAM methods.

– We implement augmented reality (AR) using L-SLAM.

9.2 Related Work

Visual SLAM methods have been actively studied in the robotics and computer vision communities for the past two decades due to its importance in various applications such as autonomous UAV to augmented reality (AR). From the vast literature in the visual SLAM, we provide a brief overview of state-of-the-art typical approaches and some SLAM methods utilizing planar structures.

Many successful SLAM algorithms have been developed using either point features (indirect) or high gradient pixels (direct). Representatives of them are direct LSD-SLAM [21], DSO [88], and feature-based ORB-SLAM2 [15]. But their SLAM performance can be severely degraded in challenging low-texture environments.

Some research in early years of SLAM exploits planes within an extended Kalman filter (EKF) based SLAM approaches [12]. In [136, 137], tracked points lying on the same plane are

reformulated as a planar feature to reduce the state size in EKF-SLAM. [138] includes planar features in the EKF state vector with a priori structural information. [139] use planar features extracted from 2D laser scanner in an EKF-based SLAM. However, these EKF-SLAM methods utilizing planar features have some problems. They cannot avoid local linearization error [140] because the estimation of camera rotation and translation together results in non-linearity of measurement model. Also, since both distance and orientation are used to represent the planar features, the state vector and covariance matrix size (computational complexity) grows rapidly over time, which limits applications to a small room-size space.

Several recent planar SLAM studies apply graph-based SLAM [133, 134, 129], which is a nonlinear and non-convex optimization problem [112]. To avoid singularities in pose graph optimization, [141] presents a minimal plane representation of infinite planes. With the help of the GPU, [142] tracks keyframe camera pose and global plane model by performing direct image alignment and global graph optimization. [131] performs graph-based SLAM with the plane measurements coming from scene layout understanding using convolutional neural networks (CNN). In [130], a keyframe-based factor graph optimization is performed to achieve real-time operation on a CPU only. Although these approaches demonstrate superior estimation results in structured environments, they require expensive and difficult pose graph optimization since they estimate the camera rotation and translation together [112].

The most relevant planar SLAM approach to the proposed L-SLAM is [132], which first estimates the 3-DoF camera rotation by recognizing the piecewise planar models, and utilizes graph SLAM optimization to recover the 2-DoF camera translation. However, unlike the proposed L-SLAM which estimates full 6-DoF camera motion, there is an assumption that the translational motion of the camera is always planar.

9.3 Proposed Method

Our proposed L-SLAM method builds on the previous *Line and Plane based Visual Odometry* (LPVO) algorithm [30]. However, while LPVO cannot avoid drift over time due to the nature of

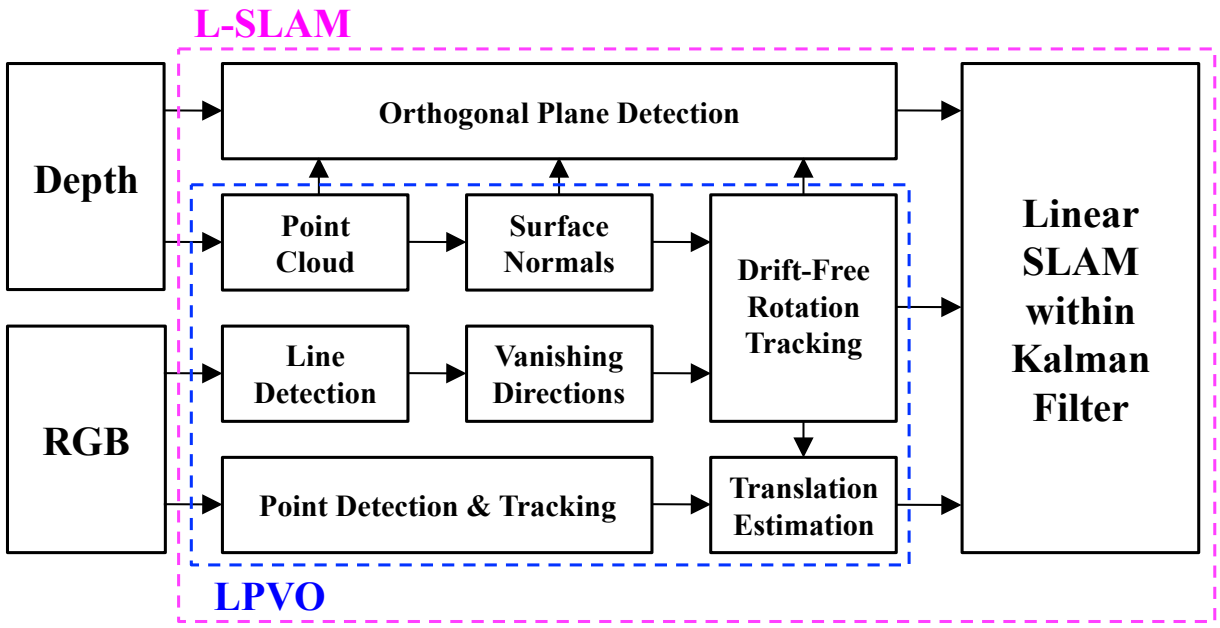


Figure 9.2: Overview over the complete L-SLAM algorithm.

VO, we extend it to the SLAM formulation in which the planar features are directly modeled as landmarks in order to further constrain the camera motion and significantly reduce drift in translation.

We start by giving a brief description of the previous LPVO algorithm in Sec. 9.3.1. As a first contribution, we present a method of detecting orthogonal planes in structured environments in Sec. 9.3.2, which plays an important role in our SLAM method. Next, we introduce L-SLAM, a novel SLAM approach using orthogonal planar features within a linear Kalman filter (KF) framework in Sec. 9.3.3. Fig. 9.2 shows an overview of the L-SLAM.

9.3.1 Line and Plane based Visual Odometry

We summarize the LPVO algorithm briefly (for full details, refer to [30]). LPVO has two main steps: 1) structural regularities (Manhattan frame) are tracked to obtain the drift-free rotation with a $\mathrm{SO}(3)$-manifold constrained mean shift algorithm; and 2) it estimates translation by minimizing a de-rotated reprojection error from tracked points.

The core of the drift-free rotation estimation in LPVO is to track the Manhattan frame (MF)

jointly from both lines and planes by exploiting environmental regularities. Given the density distribution of vanishing directions from lines and surface normals from planes on the Gaussian sphere $\mathbb{S}^2$, LPVO infers the mean of the directional vector distribution around each dominant Manhattan frame axis through a mean shift algorithm in the tangent plane $\mathbb{R}^2$ with a Gaussian kernel. The modes found by the mean shift are projected onto the $\mathrm{SO}(3)$ manifold to maintain orthogonality, resulting in the absolute orientation estimate of the camera with respect to the Manhattan world.

For the translation estimation, LPVO transforms feature correspondences between consecutive frames into a pure translation by making use of the drift-free rotation estimation in the previous step. LPVO estimates the 3-DoF translational motion of the camera by minimizing the de-rotated reprojection error from the tracked points, which is only a function of the translational camera motion.

9.3.2 Orthogonal Plane Detection

Once the Manhattan world orientation of the scene with respect to the camera pose has been established from LPVO, we can easily identify the dominant orthogonal planes in current structured environments. Given the surface normals for each pixel used when we track the Manhattan frame in LPVO, we find the relevant normal vectors inside a conic section of each Manhattan frame axis. We perform the plane RANSAC [126] with the pixels corresponding to the surface normals near each axis of the tracked Manhattan frame. We model the plane [143] as:

$$n_x u + n_y v + n_z = w \qquad (u = \frac{X}{Z}, v = \frac{Y}{Z}, w = \frac{1}{Z}) \tag{9.1}$$

where X, Y, Z denote the 3D coordinates, u, v, w correspond to the normalized image coordinates and the measured disparity at that coordinate. n_x, n_y, n_z are the model parameters representing the distance and orientation of the plane. The error function of the plane RANSAC is the distance between the 3D point and the plane. We fit the plane to the given inlier 3D points from the plane RANSAC in the least-squares sense.

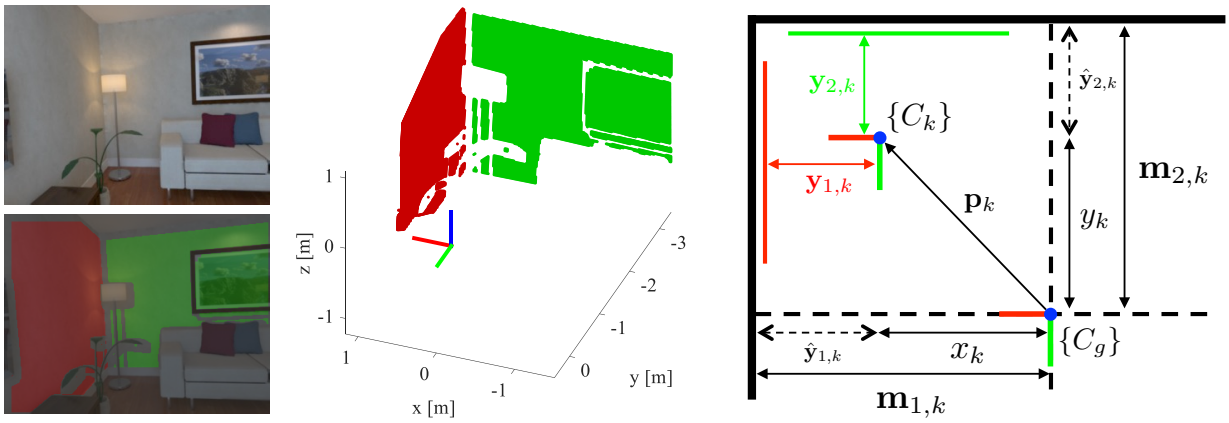


Figure 9.3: Results of orthogonal plane detection are overlaid on top of the RGB images.

If the angle difference between the normal vector of the plane and one of the three Manhattan frame axes is less than 5 degrees, we refit this plane again to a set of disparity values (w) subject to the constraint that it must be parallel to the corresponding Manhattan frame axis. We compute the optimal scale factor in the least-squares sense that minimizes:

$$s^* = \arg\min_s \|s\,(r_x u + r_y v + r_z) - w\| \tag{9.2}$$

where s is the scale factor representing the reciprocal of the distance (offset) from the plane to the origin, and r_x, r_y, r_z denote the unit vector of the corresponding Manhattan frame axis. In this way, we can find the orthogonal planar features in the scene whose normals are aligned with the tracked Manhattan frame as shown in Fig. 9.3

9.3.3 Linear RGB-D SLAM

9.3.3.1 KF State Vector Definition.

The state vector in the KF consists of the current 3-DoF translational motion of the camera and a 1-D representation of the orthogonal planar features in the map. We denote the state vector by $\mathbf{X}$

with its associated covariance $\mathbf{P}$:

$$\mathbf{X} = \begin{bmatrix} \mathbf{p}^\top & \mathbf{m}_1 & \cdots & \mathbf{m}_n \end{bmatrix}^\top \in \mathbb{R}^{3+n} \qquad \mathbf{P} = \begin{bmatrix} \mathbf{P_{pp}} & \mathbf{P_{pm}} \\ \mathbf{P_{mp}} & \mathbf{P_{mm}} \end{bmatrix} \in \mathbb{R}^{(3+n)\times(3+n)} \tag{9.3}$$

where $\mathbf{p} = \begin{bmatrix} x & y & z \end{bmatrix}^\top \in \mathbb{R}^3$ denotes the 3-DoF camera translation in the global Manhattan map frame where the rotation of the camera is completely compensated. Unlike the previous planar SLAM approaches, we do not include the camera orientation in the state vector, which is the main factor that increases the nonlinearity in the SLAM problem [112] because we already obtain accurate and drift-free camera rotation from LPVO in Sec. 9.3.1. The map $\mathbf{m}_i = [o_i] \in \mathbb{R}^1$ denotes the 1-D distance (offset) of the orthogonal planar feature from the origin in the global Manhattan map frame, and n is the number of orthogonal planes in the global map. Although the each orthogonal planar feature in Sec. 9.3.2 consists of the 1-D distance and the alignment for the Manhattan frame, we only track and update the distance since the alignment of the orthogonal planes does not change over time. A newly detected orthogonal planar feature $\mathbf{m}_{new}$ is additionally augmented after the last map component of the state vector. Note that there are no variables related to the camera or plane orientation in the state vector, resulting in a linear KF formulation.

9.3.3.2 Process Model.

We predict the next state based on the 3-DoF translational movement estimated from LPVO between the consecutive frames. We propagate the 3-DoF camera translation, and assume the map does not change. Our process model can be written as follows:

$$\mathbf{X}_k = \mathbf{F}\mathbf{X}_{k-1} + \begin{bmatrix} \triangle\mathbf{p}_{k,k-1}^\top & \mathbf{0}_{1\times n} \end{bmatrix}^\top \tag{9.4}$$

where $\mathbf{F}$ denotes the identity matrix, and $\triangle\mathbf{p}_{k,k-1}$ is the estimated 3-DoF translational movement between the k and $k-1$ image frame from LPVO.

9.3.3.3 Measurement Model.

We update the state vector in the KF by observing the distance between the currently detected orthogonal planar features and the current camera pose. A measurement model $\mathbf{y}$ for the $\mathbf{m}_i$ is defined by:

$$\mathbf{y} = \begin{bmatrix} \mathbf{m}_1 - x \\ \mathbf{m}_2 - y \\ \mathbf{m}_3 - z \\ \vdots \end{bmatrix} = \mathbf{HX} \in \mathbb{R}^m \qquad \mathbf{H} = \begin{bmatrix} -1 & 0 & 0 & 1 & 0 & 0 & \cdots \\ 0 & -1 & 0 & 0 & 1 & 0 & \cdots \\ 0 & 0 & -1 & 0 & 0 & 1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \in \mathbb{R}^{(m)\times(3+n)} \quad (9.5)$$

where $\mathbf{H}$ is the observation model which maps the state space into the observed space, and m is the number of matched orthogonal planar features. For the sake of presentation, we assume that the each orthogonal planar feature corresponds to the x or y or z axis of the Manhattan frame in the Eq. (9.5). A value of the measurement model $\mathbf{y}$ is the observed distance from the orthogonal planar features computed with the current state vector. We perform the KF update (SLAM) for all associated orthogonal planes with the global planes in the map. Since all formulas and calculations are perfectly linear from the Eqs. (9.3) to (9.5), there is no local linearization error, and we can easily calculate the optimal Kalman gain. In this manner, we can consistently track the 3-DoF camera translation and 1-D planar map position efficiently and reliably.

9.3.3.4 Planar Map Management.

At the beginning of L-SLAM, we initialize a state vector and its covariance with the orthogonal planar features detected at the first frame. When constructing a global planar map, we only utilize the orthogonal planes that have a sufficiently large area in order to accurately recognize the dominant structural characteristics such as walls, floor, and ceiling in the current structured environments. We perform plane matching using the distance (offset) and alignment from the currently detected orthogonal planar features and the global plane map in the state vector. If the metric distance between the two planes is less than a certain length (in our experiments, 10 cm),

and they have the same alignment, the detected planar feature is associated with an existing global planar map to update the state vector. The global planar map can be extended incrementally as new orthogonal planes are detected.

9.4 Evaluation

We evaluate the proposed L-SLAM on various RGB-D datasets from room-size ($\sim$10 m) to building-size ($\sim$100 m) for planar environments:

- *ICL-NUIM* [1] is a room-size RGB-D dataset providing RGB and depth images rendered in a synthetic living room and office with ground-truth camera trajectories. It is challenging to accurately estimate the camera pose due to the low-texture and artificial noise in the depth images.

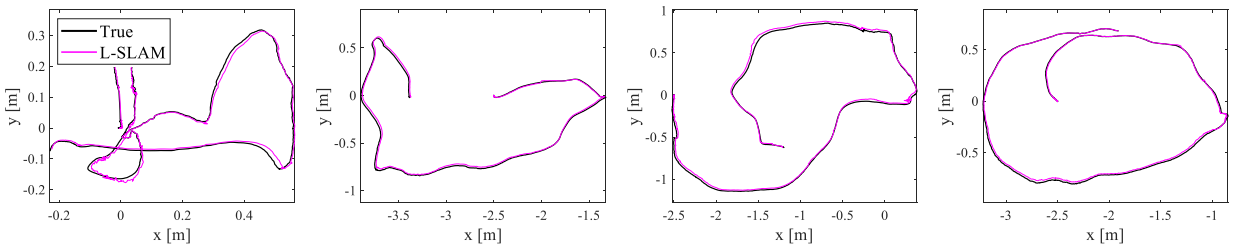
- *TUM RGB-D* [2] is the de facto standard RGB-D dataset for VO/vSLAM evaluation consisting of ground-truth camera poses and RGB-D images captured in room-scale environments with various objects.

- *Author-collected RGB-D dataset* contains RGB and depth images at 30 Hz in large building-scale planar environments with an Asus Xtion RGB-D camera. We start and end at the same position to evaluate loop closing and consistency since ground-truth trajectories and maps are not available.

We compare our L-SLAM to other state-of-the-art RGB-D SLAM and planar SLAM approaches, namely ORB-SLAM2 [15], DVO-SLAM [128], CPA-SLAM [142], KDP-SLAM [130], and DPP-SLAM [132]. Unlike the proposed L-SLAM, which is based on a linear formulation, they all perform a high dimensional nonlinear pose graph optimization. We also show an improvement compared to LPVO [30], which our new SLAM approach builds on. Note that we test each SLAM method with the original source code provided by the authors while we include the result of CPA-SLAM and KDP-SLAM taken directly from [130].

Table 9.1: Evaluation Results of ATE RMSE (unit: m) on ICL-NUIM Benchmark

Sequence	lr-kt0n	lr-kt1n	lr-kt2n	lr-kt3n	of-kt0n	of-kt1n	of-kt2n	of-kt3n
ORB-SLAM2	0.010	0.185	**0.028**	0.014	0.049	0.079	**0.025**	0.065
DVO-SLAM	0.108	0.059	0.375	0.433	0.244	0.178	0.099	0.079
CPA-SLAM	**0.007**	**0.006**	0.089	**0.009**	–	–	–	–
KDP-SLAM	0.009	0.019	0.029	0.153	–	–	–	–
LPVO	0.015	0.039	0.034	0.102	0.061	0.052	0.039	0.030
L-SLAM (Ours)	0.012	0.027	0.053	0.143	**0.020**	**0.015**	0.026	**0.011**



lr-kt0n of-kt1n of-kt2n of-kt3n

Figure 9.4: Selected motion estimation results in the ICL-NUIM dataset.

9.4.1 ICL-NUIM Dataset

We report the root mean square error (RMSE) of the absolute trajectory error (ATE) [2] for the resulting camera trajectories of all living room and office sequences with noise in Table 9.1. The smallest error for each sequence is highlighted. The results of the CPA-SLAM and KDP-SLAM for the office are not available. Although CPA-SLAM, which requires GPU for expensive computation, shows the best quantitative results in most living room sequences, L-SLAM presents comparable estimation results. We plot the estimated camera trajectories using L-SLAM in Fig. 9.4, showing that L-SLAM is comparable to other state-of-the-art SLAM approaches without a non-linear pose graph optimization.

In the office sequences, L-SLAM achieves more accurate or similar performance to other SLAM methods since the office environments consist of sufficient orthogonal planar features. Although ORB-SLAM2 performs the best thanks to sufficient texture in 'of-kt2n', L-SLAM also performs nearly as well. The average ATE RMSE of L-SLAM is 0.038, while ORB-SLAM2,

Table 9.2: Evaluation Results of ATE RMSE (unit: m) on TUM RGB-D Benchmark

Sequence	fr3/str_notex_far	fr3/str_notex_near	fr3/str_tex_far	fr3/str_tex_near	fr3/cabinet	fr3/large_cabinet
ORB-SLAM2	0.276	0.652	**0.024**	**0.019**	×	0.179
DVO-SLAM	0.213	0.076	0.048	0.031	0.690	0.979
LPVO	**0.075**	0.080	0.174	0.115	0.520	0.279
L-SLAM (Ours)	0.141	**0.066**	0.212	0.156	**0.291**	**0.140**

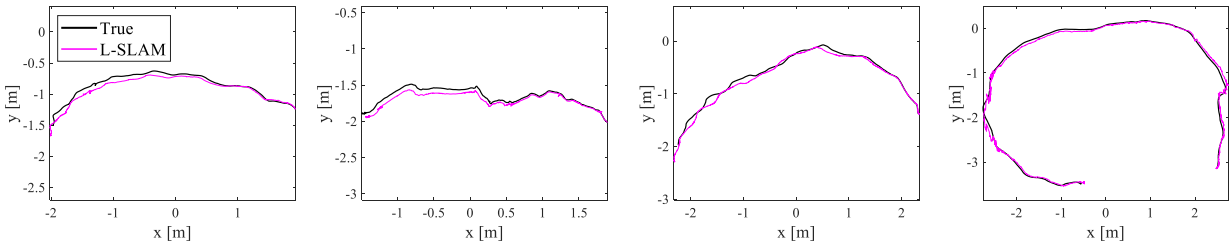


Figure 9.5: Trajectories with L-SLAM (magenta) and true (black) for the TUM RGB-D dataset.

DVO-SLAM, CPA-SLAM, KDP-SLAM, and LPVO are 0.057, 0.197, 0.028, 0.053, and 0.046, respectively. Among the CPU-only RGB-D and planar SLAM methods (except for CPA-SLAM, which requires a GPU), L-SLAM presents the lowest average trajectory error. The resulting camera trajectories with L-SLAM are plotted in Fig. 9.4, showing that L-SLAM, with an efficient and linear KF, is comparable to other recent SLAM approaches especially for highly-planar environments.

9.4.2 TUM RGB-D Dataset

We choose several RGB-D sequences in the environments where the planar features are sufficiently present in the TUM RGB-D dataset. Table 9.2 compares estimation results of the SLAM approaches. ORB-SLAM2 outperforms the proposed and other SLAM methods in texture-rich scenes such as 'fr3/str_tex_far', which is entirely expected as L-SLAM utilizes a much cheaper method. While L-SLAM shows comparable performance even in poorly-featured environments, the accuracy of ORB-SLAM2 drops drastically, and the trajectory estimation fails (marked as × in Table 9.2). Although inaccurate planar distance measurements in L-SLAM sometimes cause slight performance degradation of LPVO, L-SLAM is generally more accurate than LPVO on

Figure 9.6: Estimated trajectories with the proposed and other SLAM methods on the author-collected RGB-D dataset in a square corridor sequence.

average. The average ATE RMSE of L-SLAM is 0.168, while ORB-SLAM2, DVO-SLAM, and LPVO are 0.230, 0.340, and 0.205, respectively. Fig. 9.5 presents the estimated trajectories using L-SLAM from 'fr3/large_cabinet', showing that other SLAM methods perform poorly in low-texture scenes but the proposed method does not.

9.4.3 Author-collected RGB-D Dataset

We provide the qualitative 3D reconstruction results generated by L-SLAM with other SLAM methods' trajectories of square corridor sequence, with trajectory lengths of 90 m as shown in Fig. 9.6. L-SLAM maintains the orthogonal planar structure and significantly reduces the drift error in the final position compared to DVO-SLAM and LPVO. ORB-SLAM2 performs a wrong loop closing in pose graph optimization, resulting in the entire estimated camera trajectory breaking. Although DPP-SLAM [132] shows the second best trajectory estimation results, it only works well in such a 2-D environment with little change in camera height; otherwise, it fails in all sequences from ICL-NUIM and TUM RGB-D dataset. With L-SLAM, the starting and ending points nearly match without loop closure detection; for the others, they do not. Our final drift error is under 0.1 %. We demonstrate that L-SLAM can accurately track the camera pose and the global infinite planes in the map by preserving the planar geometric structure of indoor

Figure 9.7: Augmented reality (AR) implementation results.

environments in a much more efficient and cheaper way within a linear KF framework.

9.4.4 Augmented Reality with Linear RGB-D SLAM

We further apply the proposed L-SLAM to augmented reality (AR) to effectively demonstrate its usefulness in a practical application. Current commercial VR/AR products like Oculus Rift and HTC Vive must use external devices to track the 3-DoF translational movements of the head. However, the AR implemented by the proposed method enables full 6-DoF head tracking only with the onboard RGB-D sensor especially for highly-planar environments. Such geometric characteristics can be found easily in most structured indoor environments. For AR application, we obtain the international space station (ISS) 3D model from the 3D Warehouse website, and render the 3D object as an image with the Open Scene Graph [144]. Fig. 9.7 shows a consistent view of the 3D ISS model no matter where we look thanks to the accurate 6-DoF camera motion tracking with respect to the current structured environments from the proposed SLAM method, suggesting potential in VR/AR applications.

9.5 Conclusion

We present a new, linear KF SLAM formulation that jointly estimates the camera position and the global infinite planes in the map by compensating the rotational motion of the camera from structural regularities in the Manhattan world. By measuring the distance from the orthogonal planar features, we update the 3-DoF camera translation and the position of associated global planes in the map. Extensive evaluation demonstrates the superior performance of the proposed SLAM algorithm in a variety of planar environments, especially in keeping its efficiency without the use of expensive nonlinear optimization. Future work will further consider more general and relaxed planar environments including multiple groups of Manhattan frames such as a mixture of Manhattan frames (MMF) and Atlanta world (AW).

10

Conclusion

This thesis has developed a robust and high accurate visual odometry and localization approaches for one of the most fundamental tasks in computer vision, estimating 6-DoF camera motion from a sequence of images. (i) We have developed a robust VO against unexpected light variation by modeling the light changes as a simple affine illumination model, which can make the aerial robot autonomously fly in an environment where irregular illumination changes occur. (ii) We have proposed a low-drift VO that exploits line and plane primitives jointly to recognize the indoor scene regularities of orthogonal structured environments. Each part and step in the proposed VO and localization algorithms is based on the nonlinear optimization framework. Through a variety of experiments including the challenging scenarios for motion estimation algorithms, we compare the proposed algorithms to other state-of-the-art visual odometry methods on and demonstrate improved accuracy and much lower drift error at the end. In the following, we give a concise summary of the main contribution of each publication included in this thesis.

- **Chapter 3: Robust Visual Odometry to Irregular Illumination Changes with RGB-D Camera.** We proposed a patch-based illumination-invariant visual odometry (PIVO) pipeline, which works well in the irregular illumination change. To consider both global

and local light variations, we employed the planar patch selection process, and applied the affine illumination change model in each patch. PIVO minimized the proposed energy function reflecting the illumination changes with the robust weighting function and the efficient second-order minimization. As a result, our method could accurately estimate the camera motion and illumination parameters regardless of the partial lighting changes.

- **Chapter 4: Robust Visual Localization in Changing Lighting Conditions.** We investigated the performance of Astrobee's visual localization algorithm under changing lighting conditions. Furthermore, we presented an illumination-robust visual localization algorithm that automatically recognizes the brightness level to select an appropriate camera exposure time and map. This approach enables Astrobee to localize robustly under changing lighting conditions at the cost of building multiple lighting-specific maps.

- **Chapter 5: Autonomous Flight with Robust Visual Odometry under Dynamic Lighting Conditions.** We presented novel visual odometry for the autonomous flight of the aerial robot in a light-changing environment. The gain in robustness to irregular illumination changes is since the affine illumination model is employed in each image patch and integrated into the direct motion estimation. We proposed to utilize a motion prior from the feature-based visual odometry for stable and accurate motion estimation in a light-changing environment. Detailed analyses with the convergence rate and the degree of linearity supported such usage of the motion prior knowledge. The proposed VO algorithm enables the aerial robot to fly autonomously and robustly under changing lighting conditions at the cost of estimating the illumination change model parameters.

- **Chapter 6: Visual Odometry with Drift-Free Rotation Estimation Using Indoor Scene Regularities.** We presented a low-drift visual odometry algorithm that separately estimates the rotational and translational motion. For reducing drift of the rotation estimate, which is the primary source of position inaccuracy in visual odometry algorithms, we performed the Manhattan frame tracking to estimate the absolute camera orientation. Given drift-free rotation estimates in MW, translational motion is estimated by minimizing de-rotated

reprojection with the tracked features. This approach enables accurate and low-drift motion estimation results of the proposed VO algorithm in man-made indoor environments.

- **Chapter 7: Low-Drift Visual Odometry in Structured Environments.** We proposed a novel visual odometry algorithm that can perform accurate and low-drift motion estimation in structured environments by decoupling the camera motion into a separate rotation and translation estimation. We exploited line and plane primitives together to deal with the degenerate case in the previous drift-free rotation estimation methods, resulting in stable and accurate zero-drift rotation estimation. Given the absolute camera orientation, we recovered the optimal translational motion, which minimizes de-rotated reprojection error. We tested the proposed algorithm thoroughly with a large number of datasets, and showed accurate and low-drift motion estimation results in structural environments.

- **Chapter 8: Indoor RGB-D Compass from a Single Line and Plane.** We proposed a new method that can perform accurate and drift-free camera orientation estimation under insufficient structural environments by exploiting a single line and plane in RANSAC, which are the minimal solution for 3-DoF rotation estimation. We refined the initial rotation estimate by minimizing the average orthogonal distance from the endpoints of the parallel and orthogonal lines. The proposed algorithm is tested thoroughly with a large number of RGB-D datasets on the video sequences, and shows accurate, and drift-free rotation estimation results in the environments where the structural regularities are challenging to find.

- **Chapter 9: Linear RGB-D SLAM for Planar Environments.** We presented a new, linear KF SLAM formulation that jointly estimates the camera position and the global infinite planes in the map by compensating the rotational motion of the camera from structural regularities in the Manhattan world. By measuring the distance from the orthogonal planar features, we update the 3-DoF camera translation and the position of associated global planes in the map. The extensive evaluation demonstrates the superior performance of the proposed SLAM algorithm in a variety of planar environments, especially in keeping its efficiency without the use of expensive nonlinear optimization.

References

[1] A. Handa, T. Whelan, J. McDonald, and A. J. Davison, "A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM," in *Robotics and automation (ICRA), 2014 IEEE international conference on*.

[2] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*.

[3] "MIT Media Lab," http://news.mit.edu/2017/drones-relay-rfid-signals-inventory-control-0825.

[4] "Microsoft HoloLens," https://www.microsoft.com/en-us/hololens.

[5] P. Moulon, P. Monasse, R. Marlet *et al.*, "Openmvg. an open multiple view geometry library," 2014.

[6] C. Kerl, J. Sturm, and D. Cremers, "Robust odometry estimation for RGB-D cameras," in *IEEE ICRA*, 2013.

[7] S. Klose, P. Heise, and A. Knoll, "Efficient compositional approaches for real-time robust direct visual odometry from RGB-D data," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013.

[8] P. Kim, H. Lim, and H. J. Kim, "Robust visual odometry to irregular illumination changes with RGB-D camera," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015.

[9] A. Geiger, J. Ziegler, and C. Stiller, "Stereoscan: Dense 3D reconstruction in real-time," in *Intelligent Vehicles Symposium (IV)*. IEEE, 2011.

[10] J. Engel, "Large-scale direct slam and 3d reconstruction in real-time," PhD Thesis, 2017, technical University Munich.

[11] H. Strasdat, J. Montiel, and A. J. Davison, "Real-time monocular slam: Why filter?" in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 2657–2664.

[12] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *PAMI*, vol. 29, no. 6, pp. 1052–1067, 2007.

[13] J. Civera, A. J. Davison, and J. M. Montiel, "Inverse depth parametrization for monocular SLAM," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 932–945, 2008.

[14] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint kalman filter for vision-aided inertial navigation," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3565–3572, 2007.

[15] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Transactions on Robotics*, 2017.

[16] G. Klein and D. Murray, "Parallel tracking and mapping on a camera phone," in *Proc. of Int. Symp. on Mixed and Augmented Reality*, 2009.

[17] J. Zhang, M. Kaess, and S. Singh, "A real-time method for depth enhanced visual odometry," *Autonomous Robots*, vol. 41, no. 1, pp. 31–43, 2017.

[18] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, 2015.

[19] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust visual inertial odometry using a direct EKF-based approach," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 298–304, 2015.

[20] P. Tanskanen, T. Naegeli, M. Pollefeys, and O. Hilliges, "Semi-direct EKF-based monocular visual-inertial odometry," in *Proc. of IROS*, 2015.

[21] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," *European Conference on Computer Vision (ECCV)*, pp. 834–849, 2014.

[22] J. Engel, J. Stückler, and D. Cremers, "Large-scale direct SLAM with stereo cameras," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015.

[23] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," in *2011 international conference on computer vision*. IEEE, 2011, pp. 2320–2327.

[24] C. Forster, M. Pizzoli, and D. Scaramuzza, "Svo: Fast semi-direct monocular visual odometry," in *ICRA*. IEEE, 2014, pp. 15–22.

[25] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, "SVO: Semidirect visual odometry for monocular and multicamera systems," *IEEE Transactions on Robotics*, 2017.

[26] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[27] P. Kim, B. Coltin, O. Alexandrov, and H. J. Kim, "Robust visual localization in changing lighting conditions," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 5447–5452.

[28] P. Kim, H. Lee, and H. J. Kim, "Autonomous flight with robust visual odometry under dynamic lighting conditions (under review)," *Autonomous Robots*, 2018.

[29] P. Kim, B. Coltin, and H. J. Kim, "Visual odometry with drift-free rotation estimation using indoor scene regularities," in *2017 British Machine Vision Conference*.

[30] ——, "Low-drift visual odometry in structured environments by decoupling rotational and translational motion," in *Robotics and automation (ICRA), 2018 IEEE international conference on*.

[31] ——, "Indoor rgb-d compass from a single line and plane," in *Computer Vision and Pattern Recognition (CVPR), 2018 IEEE Conference on*, 2018.

[32] ——, "Linear RGB-D SLAM for planar environments (under review)," *European Conference on Computer Vision (ECCV)*, 2018.

[33] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[34] M. Pizzoli, C. Forster, and D. Scaramuzza, "REMODE: Probabilistic, monocular dense reconstruction in real time," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 2609–2616.

[35] J.-Y. Bouguet, "Camera calibration toolbox for matlab," 2004.

[36] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry, *An invitation to 3-D vision: from images to geometric models*. Springer Science & Business Media, 2012, vol. 26.

[37] J. Diebel, "Representing attitude: Euler angles, unit quaternions, and rotation vectors," *Matrix*, vol. 58, no. 15-16, pp. 1–35, 2006.

[38] C. Kerl, "Odometry from rgb-d cameras for autonomous quadrocopters," Master's thesis, Technical University Munich, Germany, Nov. 2012.

[39] D. Scaramuzza and F. Fraundorfer, "Visual odometry [tutorial]," *Robotics & Automation Magazine, IEEE*, vol. 18, no. 4, pp. 80–92, 2011.

[40] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. I–I, 2004.

[41] H. Lim, J. Lim, and H. J. Kim, "Real-time 6-dof monocular visual slam in a large-scale environment," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1532–1539.

[42] A. S. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, and N. Roy, "Visual odometry and mapping for autonomous flight using an RGB-D camera," in *ISRR*, 2011, pp. 1–16.

[43] J. Zhang, M. Kaess, and S. Singh, "Real-time depth enhanced monocular odometry," in *IROS*. IEEE, 2014, pp. 4973–4980.

[44] S. Baker and I. Matthews, "Lucas-kanade 20 years on: A unifying framework," *International journal of computer vision*, vol. 56, no. 3, pp. 221–255, 2004.

[45] M. Irani and P. Anandan, "About direct methods," in *Vision Algorithms: Theory and Practice*. Springer, 2000, pp. 267–277.

[46] H. Jin, P. Favaro, and S. Soatto, "Real-time feature tracking and outlier rejection with changes in illumination," in *Computer Vision, IEEE International Conference on*, vol. 1. IEEE Computer Society, 2001, pp. 684–684.

[47] M. Maimone, Y. Cheng, and L. Matthies, "Two years of visual odometry on the mars exploration rovers," *Journal of Field Robotics*, vol. 24, no. 3, pp. 169–186, 2007.

[48] Z. Fang and S. Scherer, "Experimental study of odometry estimation methods using rgb-d cameras," in *IROS*. IEEE, 2014, pp. 680–687.

[49] S. Rusinkiewicz and M. Levoy, "Efficient variants of the icp algorithm," in *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*. IEEE, 2001, pp. 145–152.

[50] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *ISMAR*. IEEE, 2011, pp. 127–136.

[51] A. I. Comport, E. Malis, and P. Rives, "Real-time quadrifocal visual odometry," *IJRR*, 2010.

[52] G. Silveira, E. Malis, and P. Rives, "An efficient direct approach to visual SLAM," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 969–979, 2008.

[53] M. Meilland, A. Comport, P. Rives, and I. S. A. Méditerranée, "Real-time dense visual tracking under large lighting variations," in *British Machine Vision Conference, University of Dundee*, vol. 29, 2011.

[54] C. Kerl, M. Souiai, J. Sturm, and D. Cremers, "Towards illumination-invariant 3d reconstruction using tof rgb-d cameras," in *3DV*, 2014.

[55] Y. Ma, *An invitation to 3-d vision: from images to geometric models*. springer, 2004, vol. 26.

[56] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," *European Conference on Computer Vision (ECCV)*, pp. 404–417, 2006.

[57] F. Steinbrücker, J. Sturm, and D. Cremers, "Real-time visual odometry from dense rgb-d images," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*. IEEE, 2011, pp. 719–722.

[58] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[59] S. Benhimane and E. Malis, "Real-time image-based tracking of planes using efficient second-order minimization," in *IEEE IROS*, 2004.

[60] S. Nolet, "The SPHERES navigation system: from early development to on-orbit testing," in *Proc. of AIAA Guidance, Navigation and Control Conf.*, 2007, p. 6354.

[61] B. Coltin, J. Fusco, Z. Moratto, O. Alexandrov, and R. Nakamura, "Localization from visual landmarks on a free-flying robot," in *Proc. of IROS*, 2016.

[62] J. Barlow, E. Smith, T. Smith, M. Bualat, T. Fong, C. Provencher, and H. Sanchez, "Astrobee: A new platform for free-flying robotics on the international space station," in *Proc. of i-SAIRAS*, 2016.

[63] M. Cummins and P. Newman, "FAB-MAP: Probabilistic localization and mapping in the space of appearance," *IJRR*, vol. 27, no. 6, pp. 647–665, 2008.

[64] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proc. of ICCV*. IEEE, 2003, pp. 1470–1477.

[65] M. J. Milford and G. F. Wyeth, "SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights," in *Proc. of ICRA*, 2012.

[66] C. Valgren and A. J. Lilienthal, "SIFT, SURF and seasons: Long-term outdoor localization using local features." in *Proc. of European Conf. on Mobile Robotics*, 2007.

[67] A. Mikulík, M. Perdoch, O. Chum, and J. Matas, "Learning a fine vocabulary," in *ECCV*. Springer, 2010, pp. 1–14.

[68] A. Ranganathan, S. Matsumoto, and D. Ilstrup, "Towards illumination invariance for visual localization," in *Proc. of ICRA*, 2013.

[69] K. Irie, T. Yoshida, and M. Tomono, "A high dynamic range vision approach to outdoor localization," in *IEEE ICRA*, 2011.

[70] N. Sünderhauf, P. Neubert, and P. Protzel, "Are we there yet? challenging SeqSLAM on a 3000 km journey across all four seasons," in *Proc. of ICRA*, 2013.

[71] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, vol. 31, no. 5, 2015.

[72] S. Weiss, M. W. Achtelik, S. Lynen, M. C. Achtelik, L. Kneip, M. Chli, and R. Siegwart, "Monocular vision for long-term micro aerial vehicle state estimation: A compendium," *Journal of Field Robotics*, vol. 30, no. 5, pp. 803–831, 2013.

[73] P. Corke, R. Paul, W. Churchill, and P. Newman, "Dealing with shadows: Capturing intrinsic scene appearance for image-based outdoor localisation," in *Proc. of IROS*, 2013.

[74] W. Maddern, A. Stewart, C. McManus, B. Upcroft, W. Churchill, and P. Newman, "Illumination invariant imaging: Applications in robust vision-based localisation, mapping and classification for autonomous vehicles," in *Proc. of ICRA*, 2014.

[75] G. D. Finlayson, S. D. Hordley, C. Lu, and M. S. Drew, "On the removal of shadows from images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 1, pp. 59–68, 2006.

[76] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[77] P. Moulon and P. Monasse, "Unordered feature tracking made fast and easy," in *CVMP*, 2012, p. 1.

[78] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: Binary robust invariant scalable keypoints," in *Proc. of ICCV*, 2011.

[79] D. Gálvez-López and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, 2012.

[80] X.-S. Gao, X.-R. Hou, J. Tang, and H.-F. Cheng, "Complete solution classification for the perspective-three-point problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 8, pp. 930–943, 2003.

[81] R. G. Valenti *et al.*, "Autonomous quadrotor flight using onboard rgb-d visual odometry," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 5233–5238.

[82] M. Achtelik, M. Achtelik *et al.*, "Sfly: Swarm of micro flying robots," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 2649–2650.

[83] D. Scaramuzza, M. Achtelik *et al.*, "Vision-controlled micro flying robots: from system design to autonomous navigation and mapping in gps-denied environments," *IEEE Robotics & Automation Magazine*, vol. 21, no. 3, pp. 26–40, 2014.

[84] L. von Stumberg, V. Usenko, J. Engel, J. Stückler, and D. Cremers, "Autonomous exploration with a low-cost quadrocopter using semi-dense monocular slam," *arXiv preprint arXiv:1609.07835*, 2016.

[85] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[86] H. Alismail, M. Kaess, B. Browning, and S. Lucey, "Direct visual odometry in low light using binary descriptors," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 444–451, 2017.

[87] S. Park, T. Schöps, and M. Pollefeys, "Illumination change robustness in direct visual slam," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 4523–4530.

[88] R. Wang, M. Schwörer, and D. Cremers, "Stereo DSO: Large-scale direct sparse visual odometry with stereo cameras," in *International Conference on Computer Vision (ICCV), Venice, Italy*, 2017.

[89] P. Bergmann, R. Wang, and D. Cremers, "Online photometric calibration of auto exposure video for realtime visual odometry and slam," *IEEE Robotics and Automation Letters*, 2017.

[90] N. Krombach, D. Droeschel, and S. Behnke, "Combining feature-based and direct methods for semi-dense real-time stereo visual odometry," in *International Conference on Intelligent Autonomous Systems*. Springer, 2016, pp. 855–868.

[91] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The EuRoC micro aerial vehicle datasets," *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1157–1163, 2016.

[92] A. Geiger, M. Roser, and R. Urtasun, "Efficient large-scale stereo matching," in *ACCV*. Springer, 2010, pp. 25–38.

[93] J. Nikolic, J. Rehder, M. Burri, P. Gohl, S. Leutenegger, P. T. Furgale, and R. Siegwart, "A synchronized visual-inertial sensor system with FPGA pre-processing for accurate real-time SLAM," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 431–437, 2014.

[94] H. Lee, H. Kim, and H. J. Kim, "Planning and control for collision-free cooperative aerial transportation," *IEEE Transactions on Automation Science and Engineering*, 2016.

[95] J.-L. Blanco, "A tutorial on se (3) transformation parameterizations and on-manifold optimization," *University of Malaga, Tech. Rep*, vol. 3, 2010.

[96] C. F. Olson, L. H. Matthies, M. Schoppers, and M. W. Maimone, "Stereo ego-motion improvements for robust rover navigation," in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*.

[97] Y. Zhou, L. Kneip, and H. Li, "Real-time rotation estimation for dense depth sensors in piece-wise planar environments," in *IEEE IROS*, 2016.

[98] J. Straub, N. Bhandari, J. J. Leonard, and J. W. Fisher, "Real-time Manhattan world rotation estimation in 3D," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*.

[99] I. Cvišić and I. Petrović, "Stereo odometry based on careful feature selection and tracking," in *Mobile Robots (ECMR), European Conference on*. IEEE, 2015.

[100] J.-P. Tardif, Y. Pavlidis, and K. Daniilidis, "Monocular visual odometry in urban environments using an omnidirectional camera," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*. IEEE, 2008.

[101] M. Kaess, K. Ni, and F. Dellaert, "Flow separation for fast and robust stereo odometry," in *IEEE ICRA*, 2009.

[102] J. M. Coughlan and A. L. Yuille, "Manhattan world: Compass direction from a single image by bayesian inference," in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*.

[103] Y. Zhou, L. Kneip, C. Rodriguez, and H. Li, "Divide and conquer: Efficient density-based tracking of 3D sensors in Manhattan worlds," in *Asian Conference on Computer Vision*. Springer, 2016.

[104] J. C. Bazin, C. Demonceaux, P. Vasseur, and I. Kweon, "Motion estimation by decoupling rotation and translation in catadioptric vision," *Computer Vision and Image Understanding*, vol. 114, no. 2, 2010.

[105] J.-K. Lee, K.-J. Yoon *et al.*, "Real-time joint estimation of camera orientation and vanishing points." in *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*.

[106] D. Holz, S. Holzer, R. B. Rusu, and S. Behnke, "Real-time plane segmentation using RGB-D cameras," in *Robot Soccer World Cup*. Springer, 2011.

[107] J. Shi and C. Tomasi, "Good features to track," in *IEEE CVPR*, 1994.

[108] J.-Y. Bouguet, "Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm," *Intel Corporation*, 2001.

[109] G. Schindler and F. Dellaert, "Atlanta world: An expectation maximization framework for simultaneous low-level edge grouping and camera calibration in complex man-made environments," in *IEEE CVPR*, 2004.

[110] J. Straub, G. Rosman, O. Freifeld, J. J. Leonard, and J. W. Fisher, "A mixture of Manhattan frames: Beyond the Manhattan world," in *IEEE CVPR*, 2014.

[111] Y. Lu and D. Song, "Robustness to lighting variations: An rgb-d indoor visual odometry using line segments," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 688–694.

[112] L. Carlone, R. Tron, K. Daniilidis, and F. Dellaert, "Initialization techniques for 3d slam: a survey on rotation estimation and its use in pose graph optimization," in *IEEE IROS*, 2015.

[113] J.-C. Bazin and M. Pollefeys, "3-line RANSAC for orthogonal vanishing point detection," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*.

[114] J. Straub, O. Freifeld, G. Rosman, J. J. Leonard, and J. W. Fisher, "The Manhattan frame model–Manhattan world inference in the space of surface normals," *IEEE T-PAMI*, 2017.

[115] R. G. Von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "LSD: A fast line segment detector with a false detection control," *IEEE transactions on pattern analysis and machine intelligence*, 2010.

[116] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *European Conference on Computer Vision*. Springer, 2012.

[117] S. Gupta, P. Arbelaez, and J. Malik, "Perceptual organization and recognition of indoor scenes from RGB-D images," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*.

[118] A. Elqursh and A. Elgammal, "Line-based relative pose estimation," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*.

[119] K. Joo, T.-H. Oh, J. Kim, and I. So Kweon, "Globally optimal Manhattan frame estimation in real-time," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[120] S. N. Sinha, D. Steedly, and R. Szeliski, "A multi-stage linear approach to structure from motion," in *European Conference on Computer Vision*. Springer, 2010.

[121] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," in *Readings in computer vision*. Elsevier, 1987.

[122] W. Elloumi, S. Treuillet, and R. Leconge, "Real-time camera orientation estimation based on vanishing point tracking under Manhattan world assumption," *Journal of Real-Time Image Processing*, 2017.

[123] J. Straub, O. Freifeld, G. Rosman, J. J. Leonard, and J. W. Fisher, "The Manhattan frame model–Manhattan world inference in the space of surface normals," *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[124] J.-C. Bazin, Y. Seo, C. Demonceaux, P. Vasseur, K. Ikeuchi, I. Kweon, and M. Pollefeys, "Globally optimal line clustering and vanishing point estimation in Manhattan world," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*.

[125] N. Neverova, D. Muselet, and A. Trémeau, "21/2 d scene reconstruction of indoor scenes from single rgb-d images." in *CCIW*, 2013, pp. 281–295.

[126] M. Y. Yang and W. Förstner, "Plane detection in point cloud data," in *Proceedings of the 2nd int conf on machine control guidance, Bonn*, 2010.

[127] C. Rother, "A new approach to vanishing point detection in architectural environments," *Image and Vision Computing*, 2002.

[128] C. Kerl, J. Sturm, and D. Cremers, "Dense visual slam for rgb-d cameras," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 2100–2106.

[129] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g 2 o: A general framework for graph optimization," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 2011.

[130] M. Hsiao, E. Westman, G. Zhang, and M. Kaess, "Keyframe-based dense planar slam," in *IEEE ICRA*, 2017.

[131] S. Yang, Y. Song, M. Kaess, and S. Scherer, "Pop-up slam: semantic monocular plane slam for low-texture environments," in *IEEE IROS*, 2016.

[132] P.-H. Le and J. Kosecka, "Dense piecewise planar rgb-d slam for indoor environments," in *IEEE IROS*, 2017.

[133] M. Kaess, A. Ranganathan, and F. Dellaert, "isam: Incremental smoothing and mapping," *IEEE Transactions on Robotics*, 2008.

[134] G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based slam," *IEEE Intelligent Transportation Systems Magazine*, 2010.

[135] L. Zhao, S. Huang, and G. Dissanayake, "Linear slam: A linear solution to the feature-based and pose graph slam based on submap joining," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, 2013.

[136] A. P. Gee, D. Chekhlov, W. W. Mayol-Cuevas, and A. Calway, "Discovering planes and collapsing the state space in visual slam." in *BMVC*, 2007.

[137] A. P. Gee, D. Chekhlov, A. Calway, and W. Mayol-Cuevas, "Discovering higher level structure in visual slam," *IEEE Transactions on Robotics*, 2008.

[138] F. Servant, E. Marchand, P. Houlier, and I. Marchal, "Visual planes-based simultaneous localization and model refinement for augmented reality," in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*. IEEE, 2008.

[139] J. Weingarten and R. Siegwart, "3d slam using planar segments," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, 2006.

[140] T. Bailey, J. Nieto, J. Guivant, M. Stevens, and E. Nebot, "Consistency of the ekf-slam algorithm," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, 2006.

[141] M. Kaess, "Simultaneous localization and mapping with infinite planes," in *IEEE ICRA*, 2015.

[142] L. Ma, C. Kerl, J. Stückler, and D. Cremers, "Cpa-slam: Consistent plane-model alignment for direct rgb-d slam," in *IEEE ICRA*, 2016.

[143] C. J. Taylor and A. Cowley, "Parsing indoor scenes using rgb-d imagery," in *Robotics: Science and Systems*, vol. 8, 2013, pp. 401–408.

[144] T. Hassner, S. Harel, E. Paz, and R. Enbar, "Effective face frontalization in unconstrained images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

국문초록

본 논문에서는 카메라로부터 촬영되는 일련의 연속적인 이미지들로부터, 3차원 공간상에서 자기 자신의 6 자유도 움직임을 추정하는 기법인 영상 기반 주행 거리 기록계 (VO) 그리고 동시적 위치 인식 및 지도 작성 (vSLAM) 기법들에 대해 탐구하였고, 특히 주변 환경 조건에 대해 강건하고, 정확한 위치 추정 기법들을 새롭게 제안하였다. 영상 내 갑작스럽게 발생하는 빛 변화를 간단한 아핀 변화 모델로 모사함으로써 조도 변화에도 강건한 직접적 방식 기반의 위치 추정 알고리즘을 새롭게 제안하였다. 또한 선, 면과 같은 실내 구조적 특징들을 효과적으로 동시에 이용함으로써 매우 정확한 위치 인식 기법을 새롭게 연구하였다. 이렇게 새롭게 제안한 기법들은 갑작스러운 조도 변화가 일어나는 실내 및 실외, 그리고 카메라의 순수한 제자리 회전 운동과 같은 도전적인 환경과 움직임에서도 정확도를 잃지 않는다는 것이 가장 중요한 특징이다.

첫 번째로, 영상 내 조도 변화에 대해 강건하게 대응하는 직접적 방식 기반의 영상 기반 주행 거리 기록계 알고리즘을 새롭게 제안하였다. 일반적인 직접적 방식 기반의 알고리즘은 영상 내 동일 물체는 동일 밝기를 가진다는 밝기 불변량 가정을 기반으로 위치 추정을 수행하기 때문에 조그마한 빛 변화에도 매우 취약하고 불안정한 성능을 보여준다. 제안한 직접적 방식 기반의 알고리즘은 영상 내 여러 개의 패치를 생성한 뒤, 각 패치별로 독립적인 아핀 빛 변화 모델을 적용하고 이를 실시간으로 추정, 보상함으로써 전역적 및 부분적 조도 변화에 매우 강건한 성능을 보여준다. 추가적으로 특징점 기반 알고리즘을 사전 움직임으로 결합하고 이를 위치 추정 최적화 시에 사용하여, 기존 알고리즘 대비 더 높은 정확도와 더 안정적인 자가 위치 추정 성능을 얻을 수 있었다. 다양하고 도전적인 환경에서 촬영된 영상 데이터셋을 이용하여, 타 알고리즘 대비 제안한 방법의 효율성 및 위치 추정 성능을 정량적으로 평가해보았으며, 추가적으로 실내 비행 드론에 탑재하여 도전적인 조도 변화 환경 내에서도 영상 기반 자율 비행이 가능함을 제시하였다.

두 번째로는, 컬러 및 깊이 영상에서 발견되는 선 및 면과 같은 공간 구조적 특징을 적극적으로 활용한 고정밀 영상 기반 주행 기록계 알고리즘을 새롭게 연구하였다. 대부분의 영상

기반 위치 추정 알고리즘들은 시간이 지남이 따라 누적되는 큰 위치 추정 오차를 피할 수 없는데, 이는 대부분 부정확한 회전 운동 추정으로 인해 유발된다고 알려져 있다. 새롭게 제안한 알고리즘은 실내 구조적 특징인 선과 면을 추적하여, 이러한 위치 추정 오차의 주요 원인인 회전운동을 매우 정확하고 드리프트 없이 추정할 수 있었다. 우리는 효율적인 SO(3) 공간상에서 제한된 평균 이동 알고리즘을 사용하여 환경의 구조적 규칙성을 인지하고 추적하였다. 이렇게 정확한 카메라의 회전 움직임이 얻어진 후, 이를 이용하여 회전 성분이 제거된 재투영 오차를 최소화하여 카메라의 병진 움직임을 추정한다. 제안된 위치 추정 알고리즘은 순수 회전 움직임과 같이 추정하기 어려운 카메라 움직임이 포함된 다양한 영상 데이터셋에서 평가되었으며, 특히 타 알고리즘 대비 매우 높은 정확성과 강건성, 그리고 낮은 드리프트 오차를 실험적으로 보여주었다.

주요어**:** 영상 기반 주행 거리 기록계, 영상 기반 동시적 위치 인식 및 지도 작성, 환경 내 조도 변화, 실내 구조적 규칙성, 맨해튼 월드, 실내 자율 비행 드론

학 번**:** 2013-20663