공학박사학위논문

# Robust Edge-Point Visual Odometry and Monocular Scale Observer using Vehicle Kinematics

# 모서리와 점 특징을 이용하는 강인한 영상 항법과 차량 기구학 기반의 절대 스케일 감지 기법

2023년 2월

서울대학교 대학원

항공우주공학과

김창현

# Robust Edge-Point Visual Odometry and Monocular Scale Observer using Vehicle Kinematics

# 모서리와 점 특징을 이용하는 강인한 영상 항법과 차량 기구학 기반의 절대 스케일 감지 기법

지도교수 김 현 진

이 논문을 공학박사 학위논문으로 제출함

2022년 12월

서울대학교 대학원

항공우주공학과

김 창 현

김창현의 공학박사 학위논문을 인준함

2022년 12월

위 원 장: ____________________

부위원장: ____________________

위 원: ____________________

위 원: ____________________

위 원: ____________________

# Robust Edge-Point Visual Odometry and Monocular Scale Observer using Vehicle Kinematics

A Dissertation

by

Changhyeon 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 Aerospace Engineering

Seoul National University

Supervisor : Professor H. Jin Kim

FEBRUARY 2023

# Robust Edge-Point Visual Odometry and Monocular Scale Observer using Vehicle Kinematics

Changhyeon Kim

Department of Aerospace Engineering

Seoul National University

APPROVED:

Youdan Kim, Chair, Ph.D.

H. Jin Kim, Ph.D.

Chan Gook Park, Ph.D.

Hyeonbeom Lee, Ph.D.

Pyojin Kim, Ph.D.

*to my*

*FAMILY and JW*

*with love*

# Abstract

## Robust Edge-Point Visual Odometry and Monocular Scale Observer using Vehicle Kinematics

Changhyeon Kim
Department of Aerospace Engineering
The Graduate School
Seoul National University

Navigation is an essential functionality for the autonomy of robots, such as autonomous vehicles and drones. In particular, image-based navigation, visual odometry (VO), only uses small cameras to perform navigation and is an attractive alternative to indoor environments where external navigation, such as GPS, is unavailable. This dissertation proposes a robust VO that can operate reliably in low-textured and brightness-changing conditions frequently encountered in indoor environments. As a practical application for indoor VO, the indoor vehicle driving condition is considered, and a scale-aware monocular VO (MVO) method utilizing vehicle kinematics is proposed.

First, this research proposes edge and point-based VO systems robust to low-textured and brightness-varying conditions. To characterize edges, they are classified into eight orientation groups according to their image gradient directions. Using the edge groups, eight quadtrees are constructed, and overlapping areas are set belonging to adjacent quadtrees for robust and efficient matching. For further acceleration, previously visited tree nodes are stored and reused at the next iteration to warm-start. An edge culling method is proposed to extract prominent edgelets and prune redundant edges. The camera motion is estimated by minimizing point-to-edge distances within a re-weighted iterative closest points (ICP) framework, and simultaneously, 3-D structures are recovered by static and temporal stereo settings. To improve the accuracy of the edge-based VO, a hybrid VO is proposed by combining the conventional point features. In this extension, brightness changes between image frames are incorporated into the photometric error minimization problem. To analyze the effects of the proposed modules, extensive simulations are conducted in various settings. Quantitative results on public datasets confirm that the proposed approach has competitive performance with state-of-the-art stereo methods. In addition, the author demonstrates the practical values of the proposed system in author-collected modern building scenes with curved edges only.

Next, a scale-aware MVO using vehicle kinematics is addressed. To describe the motion of the camera attached to the vehicle, the unknown camera-vehicle relative pose is estimated by only using the monocular VO motions. An observer is designed to estimate the absolute scale of the MVO motions on turn regions, and turn regions are detected to stably observe scale. Using the observed scale, an absolute scale recovery is proposed to estimate the unknown scale between turn regions. By extensive simulations for each proposed module, appropriate conditions for stable scale estimation are investigated, and the effectiveness of the extrinsic calibration and the absolute scale recovery is statistically verified by Monte-Carlo simulations. To evaluate the overall performance, the proposed method and state-of-the-art VO methods are compared in public outdoor driving datasets. In addition, to show promising applicability, real-world driving datasets are collected in multi-floor underground parking lots and demonstrate the accurate absolute scale recovery performance of the proposed method in indoor driving situations.

**Keywords:** Robust visual odometry, monocular scale recovery, extrinsic calibration, image edges and points, vehicle kinematics

**Student Number:** 2018-31816

# Table of Contents

# List of Tables

# List of Figures

# 1

# Introduction

Camera motion estimation from consecutive images, known as visual odometry (VO) [6], is receiving increasing attention in areas of autonomous robots [7] and virtual and augmented reality (VR/AR) applications requiring self-localization abilities [8],[9]. A main interest of the VO research has been to improve the estimation accuracy while maintaining real-time applicability. Consequently, several algorithms are developed with competitive performance in real-time applications using various settings, such as mono [10], [11], stereo [12], and RGB-D cameras[13, 14].

However, most VO algorithms still depend on two main assumptions; consistent brightness and feature-abundant scenes. The first one can be easily violated in the in-and-out movements and auto-exposure adjustments causing sudden brightness changes. Furthermore, texture-less scenes, such as monotonous walls and ceilings, make the second assumption invalid. If these assumptions are violated, the VO performance is significantly degraded, and the VO might even lose track of the motion estimation. Especially, those conditions are often observed in indoor scenes as seen in Fig. 1.1. In a modern building, interiors and office furniture have monotonous shapes, and there are many lighting factors,

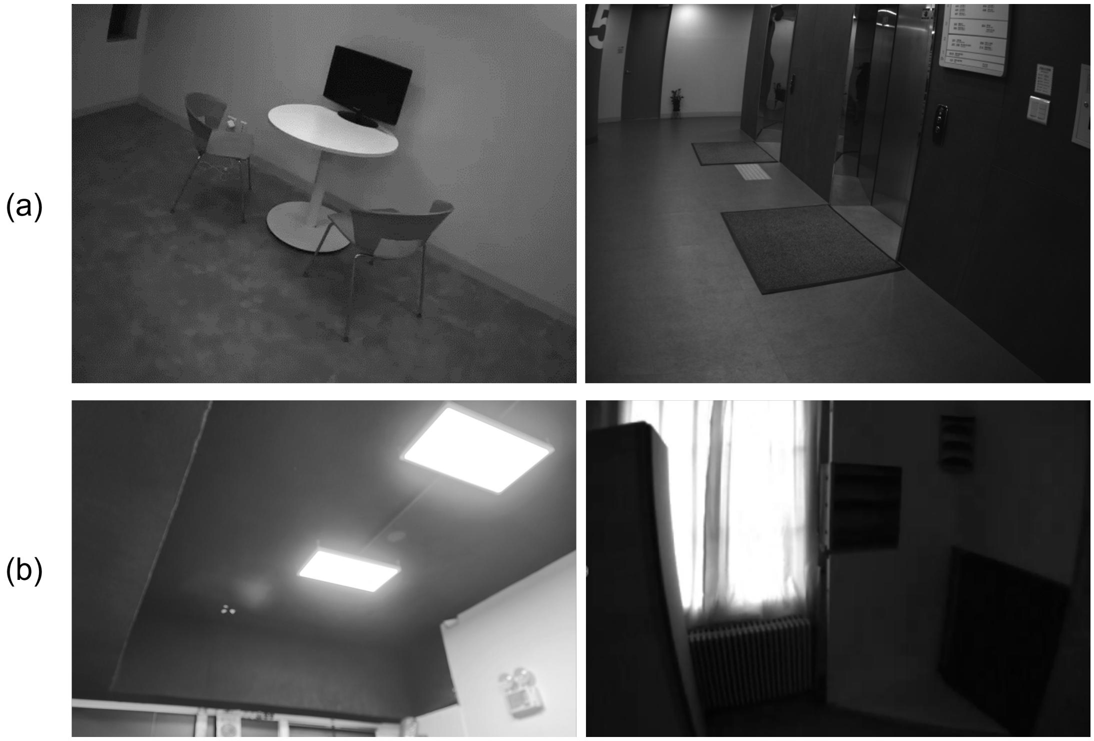
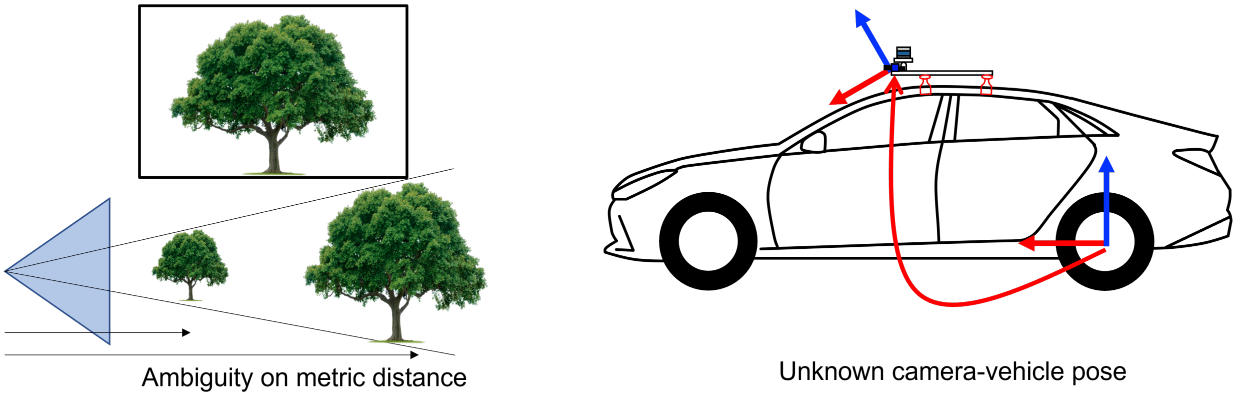


Figure 1.1: **Challenging environments for the conventional visual odometry** (a) low-textured office and corridors encountered in the modern indoor environments, (b) varying brightness conditions due to fluorescent lamps and sunlight from the windows

such as fluorescent lamps and sunlight from the windows. Thus, comprehensively considering robustness to the conditions mentioned above is required to use the VO in more general situations. The first part of this dissertation focuses on the robustness of VO against low-textured and varying brightness conditions, which are easily encountered in modern indoor situations.

VO using a single camera, monocular VO (MVO), is an attractive solution for automobile navigation due to its minimum setting. Furthermore, because one or more cameras can be easily found in most vehicles as forms of driving assistant systems and user-mounted dashboard cameras as seen in Fig. 1.2, the MVO implementation targeted for mobile vehicles could be highly valuable. Especially in indoor driving circumstances where the external navigation module such as GPS is not available, the MVO for vehicular settings can be a promising application.

Figure 1.2: **Image sensors installed in the automotive vehicles.**



Ambiguity on metric distance

Unknown camera-vehicle pose

Figure 1.3: **Monocular metric scale ambiguity problem and the unknown camera-vehicle pose**

However, due to the monocular projective nature, absolute metric information disappears from an image as seen in Fig. 1.3, and the MVO can only yield up-to-scale translation motion, which makes the MVO-only setup more challenging without additional metric measurements. It is called the scale ambiguity problem [15]. To solve this problem, vehicle speed or other sensor information can be used; however, except for experimental settings, it can be difficult to use various sensors. Another problem when utilizing the VO in the vehicle is that the camera pose relative to the vehicle cannot be known in general. In this case, even if the VO information is obtained, it cannot be used for vehicle reference navigation.

In the second part of this dissertation, the aforementioned problems when realizing the

MVO for indoor driving situations are addressed, and the scale-aware MVO system for the vehicular application is proposed. In addition, a self-contained camera-vehicle extrinsic pose calibration method is also developed for completeness.

An overall flowchart of the proposed methods shows the relationships between the robust edge and point-based VO and the scale-aware MVO for vehicular application as seen in Fig. 1.4.

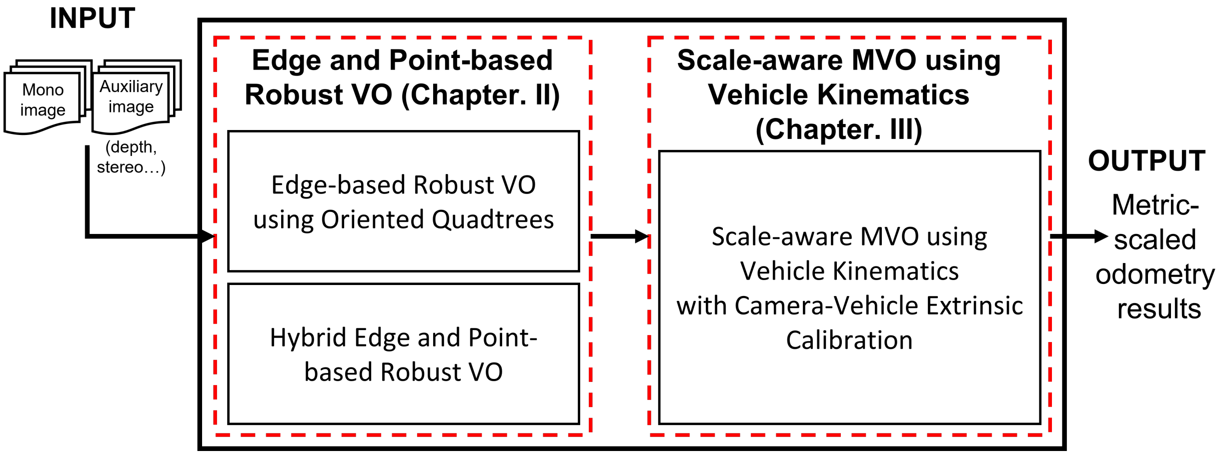

INPUT
Mono image
Auxiliary image
(depth, stereo…)
Edge and Point-based Robust VO (Chapter. II)
Edge-based Robust VO using Oriented Quadtrees
Hybrid Edge and Point-based Robust VO
Scale-aware MVO using Vehicle Kinematics (Chapter. III)
Scale-aware MVO using Vehicle Kinematics with Camera-Vehicle Extrinsic Calibration
OUTPUT
Metric-scaled odometry results

Figure 1.4: **Overall flowchart of the proposed methods in the dissertation**

## 1.1 Contributions and Outline of the Dissertation

The outline of the dissertation is as follows. Chapter 2 addresses edge and point-based visual odometry systems robustly operating in illumination-changing and point features-less indoor environments. Chapter 3 presents a scale-aware monocular visual odometry system using vehicle kinematics. In Chapter 4, experimental results of the proposed modules are presented using the various public and real-world author-collected datasets. Chapter 5 ends the dissertation with concluding remarks and suggestions on further works.

The main contributions of this dissertation are summarized as follows.

**Chapter 2: Robust VO Systems using Edge and Point Features**

In this chapter, in-depth consideration is addressed to solving two problems when realizing an efficient edge-based stereo VO: high ambiguities on edge matching and redundancy of edge pixels. The key contributions of this chapter can be summarized as follows:

- **Efficient edge culling method:** An efficient edge culling method is proposed to suppress many cluttered edge responses; consequently, the required computational load is reduced while maintaining VO performance.

- **Robust edge labeling and matching methods:** Edge pixels are labeled by the image gradient directions, and edge matching speed and success rates are improved by the proposed multiple quadtrees and node caching schemes.

- **Edge and point-based hybrid robust VO:** This research proposes an extension of edge-based VO combining the conventional point features. In this extension, brightness changes between image frames are robustly incorporated into the photometric error minimization problem.

**Chapter 3: Scale-aware MVO using Vehicle Kinematics**

This chapter proposes the scale-aware MVO framework explicitly utilizing the vehicle kinematic constraints on the camera motions. For completeness of the formulation, a self-contained camera-vehicle extrinsic pose calibration method is additionally designed only using the monocular VO motions. Main contributions of this chapter are as follows;

- **Self-contained camera-vehicle extrinsic calibration:** A relative pose of the arbitrarily-attached camera to the vehicle can be estimated by proposing a self-contained camera-vehicle extrinsic pose calibration method only using monocular camera motions constrained by vehicle kinematics.

- **Absolute scale observer on turning motion:** By utilizing the local geometry of the constrained camera motions, a new scale observing method is designed for a monocular translation motion when the vehicle turns. In addition, theoretical analysis of the scale observer is addressed to determine stable states for observing the scale.

- **Absolute scale recovery of unobserved scales:** The unobserved scale of the straight region between turns can be estimated by an absolute scale using the metric scale on the turning motions.

# 2

# Robust Visual Odometry Systems using Image Edge and Point Features

This chapter addresses robust visual odometry (VO) systems using image edges and point features. At first, an efficient edge-based VO is proposed using multiple quadtrees created according to image gradient orientations. To characterize edges, they are classified into eight orientation groups according to their image gradient directions. Using the edge groups, eight quadtrees are constructed with overlapping areas belonging to adjacent quadtrees for robust and efficient matching. For further acceleration, previously visited tree nodes are stored and reused at the next iteration to warm-start. An edge culling method is designed to extract prominent edgelets and prune redundant edges. The camera motion is estimated by minimizing point-to-edge distances within a re-weighted iterative closest points (ICP) framework, and simultaneously, 3-D structures are recovered by static and temporal stereo settings.

To leverage edge and point characteristics, a robust hybrid VO is additionally proposed by utilizing both edge and point pixels. Among the cluttered raw edges, structural edgelets and corner points are efficiently classified by testing the homogeneity of image gradient orientations. At the matching step, the structural edgelets are matched by multiple quadtrees of the proposed edge-based method generated according to image gradient orientations. Then, 6-DoF camera motion is optimized by minimizing edge and point joint re-projection errors.

To analyze the effects of the proposed methods, extensive simulations are conducted in various settings. Then, the robust and accurate performance of the proposed method is evaluated on public datasets by additionally imposing drastic brightness changes. Quantitative results on public datasets confirm that the proposed approach has competitive performance with state-of-the-art stereo methods. In addition, the practical values of the proposed system are demonstrated in author-collected modern building scenes with curved edges only. The results show that the proposed method has promising advantages in real-world scenarios with few features and arbitrary brightness conditions.

## 2.1 Introduction

Recent advances in the accuracy and real-time performance of VO have been striking thanks to standard pipelines such as sparse point-based approaches [8, 10, 11], direct methods which find an optimal camera motion minimizing intensity residual between two images [9, 16, 17], and iterative closest points (ICP)-based algorithms [18, 19, 20, 21] that align a pair of large point sets, e.g., 3-D point clouds, to obtain relative camera motions.

Despite such maturity, robustness to featureless scenes and fluctuating illuminations is not yet sufficient. To alleviate this problem, irregular brightness changes have been considered as an affine model and compensated for semi-dense regions [22], and straight-line features are invited to maintain VO to keep track of motions even in low-textured scenes [23, 24].

As another attempt to improve robustness, edge-based VO systems have been introduced recently [25], [26]. The image edges can be stably detected by a traditional method

[27] even in monotonic surfaces often encountered in the man-made world. Moreover, a continuum of edge pixels gives more 3-D structural information of surroundings than sparse points, which fits interactive applications better.

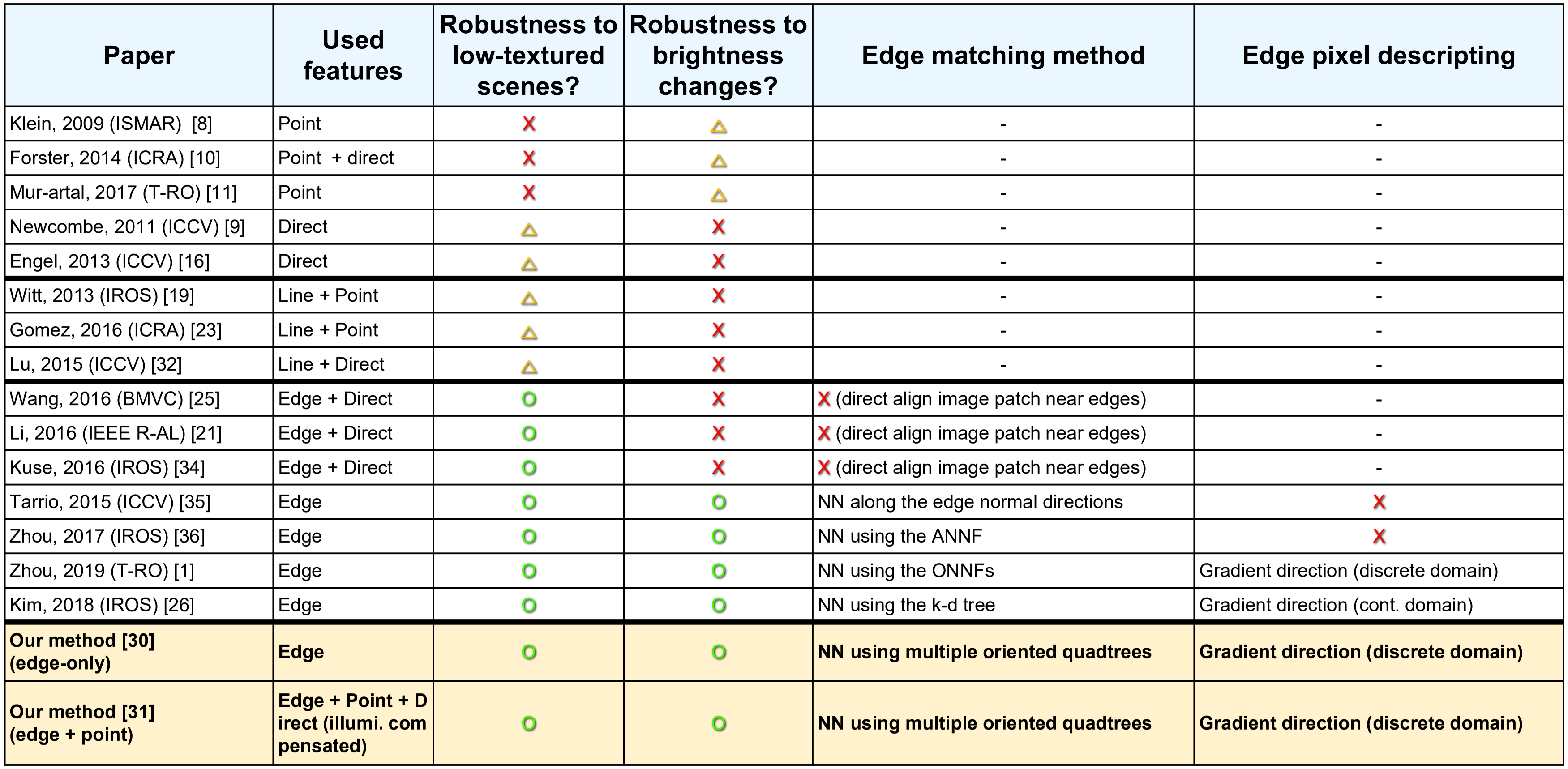
For utilizing image edges for VO, as reported in [1], several hurdles still remain. Especially 1) there are no apparent and efficient matching criteria for edges contrary to points and straight lines [28], [29], and 2) too many edge pixels cause a computational burden too heavy for real-time employment. In this chapter, the main objective is to deal with these two issues and efficiently incorporate free-formed edges into a robust stereo-visual odometry system. This algorithm has been addressed in three conference papers [26], [30], and [31].

### 2.1.1 Literature Review

Fig. 2.1 shows a comparison table of the literature review for the edge and point-based robust visual odometry systems.

**Points & direct intensity:** Early VO approaches have been mostly developed using point features, and point-based methods have shown high localization accuracy and robustness to large motions between frames with real-time operations [8, 10, 11]. Recently, VO methods utilizing intact brightness values for localization, so-called direct methods, are actively researched [9, 16]. Compared to the former, the latter is less susceptible to motion blurs and provides a denser representation, which is more attractive in practical aspects. Despite the successful research history, both methods still have limitations in real-world situations: point-based methods heavily rely on point features hardly existing in modern man-made scenes, and direct methods can be influenced by varying illuminations.

**Lines:** Straight lines are intermediate features between points and free curves, observed even in low-textured scenes. For enhancing robustness against those scenes, a stereo VO aligning multiple lines is proposed in [19], and [23] utilizes points with lines based on a monocular semi-direct approach [10]. In [32], a robust rgb-d direct VO combining points and lines is suggested. In those works, lines are not used as major features, but for additional



| Paper | Used features | Robustness to low-textured scenes? | Robustness to brightness changes? | Edge matching method | Edge pixel descripting |
|---|---|---|---|---|---|
| Klein, 2009 (ISMAR) [8] | Point | ✗ | △ | - | - |
| Forster, 2014 (ICRA) [10] | Point + direct | ✗ | △ | - | - |
| Mur-artal, 2017 (T-RO) [11] | Point | ✗ | △ | - | - |
| Newcombe, 2011 (ICCV) [9] | Direct | △ | ✗ | - | - |
| Engel, 2013 (ICCV) [16] | Direct | △ | ✗ | - | - |
| Witt, 2013 (IROS) [19] | Line + Point | △ | ✗ | - | - |
| Gomez, 2016 (ICRA) [23] | Line + Point | △ | ✗ | - | - |
| Lu, 2015 (ICCV) [32] | Line + Direct | △ | ✗ | - | - |
| Wang, 2016 (BMVC) [25] | Edge + Direct | ○ | ✗ | ✗ (direct align image patch near edges) | - |
| Li, 2016 (IEEE R-AL) [21] | Edge + Direct | ○ | ✗ | ✗ (direct align image patch near edges) | - |
| Kuse, 2016 (IROS) [34] | Edge + Direct | ○ | ✗ | ✗ (direct align image patch near edges) | - |
| Tarrio, 2015 (ICCV) [35] | Edge | ○ | ○ | NN along the edge normal directions | ✗ |
| Zhou, 2017 (IROS) [36] | Edge | ○ | ○ | NN using the ANNF | ✗ |
| Zhou, 2019 (T-RO) [1] | Edge | ○ | ○ | NN using the ONNFs | Gradient direction (discrete domain) |
| Kim, 2018 (IROS) [26] | Edge | ○ | ○ | NN using the k-d tree | Gradient direction (cont. domain) |
| **Our method [30] (edge-only)** | **Edge** | ○ | ○ | **NN using multiple oriented quadtrees** | **Gradient direction (discrete domain)** |
| **Our method [31] (edge + point)** | **Edge + Point + D irect (illumi. com pensated)** | ○ | ○ | **NN using multiple oriented quadtrees** | **Gradient direction (discrete domain)** |

Figure 2.1: **Literature review for the edge and point-based robust visual odometry systems**

constraints for point-based VO, because both end points of a line are not consistently extracted even by using state-of-the-art line descriptor [29].

**Edges:** Edges are generalized features, including points, lines, and arbitrary curves, and are easily observed in most scenes. Although some attempts to adopt edges into VO began in the early days [33], full-fledged edge-based VO systems have emerged only recently.

Edge-based VO methods can be largely divided into two types. The first one regards edges only as assistive profiles for photometric error minimization, not as major features. The work in [25] estimates rgb-d camera motions by minimizing both photometric and geometric errors of distance transform maps. Similarly, [21] suggests a rapid rgb-d VO by minimizing photometric errors around sparsely-sampled edge pixels. [34] develops a sub-gradient image aligning method using a distance transform around edges. They can enhance the robustness against low-textured scenes by imposing additional constraints by using edges. Nonetheless, they are still vulnerable to lighting changes due to the dependency on photometric properties.

The other type of edge-based VO methods utilizes intact edge pixels as the main features. In these methods, explicitly matching edge pixels is one of the most crucial parts. To mitigate difficulties in matching caused by the absence of proper descriptors for edges, several approaches are proposed in [35, 36, 1, 20, 26].

The matching methods of the latter type into two approaches can be further categorized: 1) *searching from geometry* and *2*) *searching from data structures*. Geometric approaches confine search regions by utilizing edge normal directions. In [35], searching is conducted along the normal direction of edge curves. [36] suggests an rgb-d VO using approximated neighbor fields (ANNFs) for fast edge matching, and the matching completeness is further improved via oriented edge neighbor fields (ONNFs) in [1].

The other approaches use data structures to find the most likely pair. This idea is originally from ICP algorithm [20] using a k-d tree structure. In [26], the robustness is improved by adopting a k-d tree considering image gradient vectors to compare edge similarities in cluttered regions. Note that these edge VO methods still exploit photometric

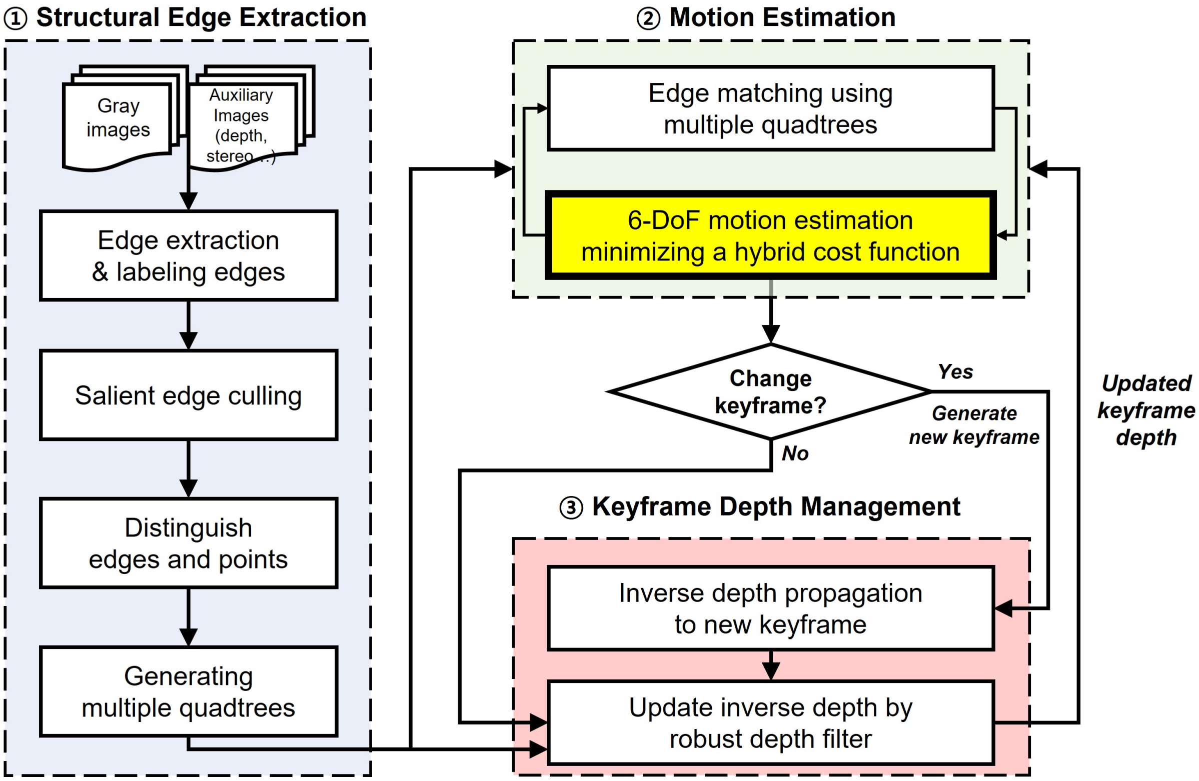

① Structural Edge Extraction
Gray images
Auxiliary Images (depth, stereo ...)
Edge extraction & labeling edges
Salient edge culling
Distinguish edges and points
Generating multiple quadtrees
② Motion Estimation
Edge matching using multiple quadtrees
6-DoF motion estimation minimizing a hybrid cost function
Change keyframe?
Yes
Generate new keyframe
No
Updated keyframe depth
③ Keyframe Depth Management
Inverse depth propagation to new keyframe
Update inverse depth by robust depth filter

Figure 2.2: **Flowchart of the proposed edge-point VO system**

information, and most methods rely on rgb-d sensors to obtain the 3-D information of edge regions.

The main goal is to further improve an edge-based VO with an explicit match process by proposing a new dedicated data structure and efficient edge pre-processing steps.

### 2.1.2 Contributions of the Chapter

In this chapter, in-depth consideration is given to solving two problems when realizing an efficient edge-based stereo VO: high ambiguities on edge matching and redundancy of edge pixels. The key contributions can be summarized as follows:

- An ICP-based efficient visual odometry system is designed by using the dedicated multiple quadtrees structure and the edge culling method.

- By using the proposed edge culling method, many cluttered edge responses are suppressed; consequently, the required computational load is reduced while maintaining VO performance.
- Edge matching speed and success rates are improved by the proposed multiple quadtrees and node caching schemes.
- An extension of edge-based VO is proposed by combining the conventional point features. In this extension, brightness changes between image frames are robustly incorporated into the photometric error minimization problem.
- Experimental results demonstrate that the proposed method has robust and competitive performance with state-of-the-art stereo VO on publicly available datasets and author-collected scenes.

### 2.1.3 Algorithm Overview

A flowchart of the proposed system is illustrated in Fig. 2.2. For every stereo stream, all the edge pixels are classified into eight orientation bins with mutually inclusive regions according to their image gradient directions. In Section. 2.3, to reduce redundant edge pixels, several thousands of raw edge pixels are additionally condensed into well-distributed structural edgelets by the proposed edge culling method. This research includes proposing an efficient multiple quadtrees structure composed of eight orientation bins, and storing the previously matched nodes for warm-starting in the next iteration, which are detailed in Section 2.4. Section 2.5 details the ICP-based camera motion estimation by minimizing stereo point-to-edge normal distances between current images and keyframe images, and the static and temporal stereo method for updating edge inverse depths. The extensive analysis of each core part and experimental results are following in Sections 2.7 and 4.1.

## 2.2 Preliminaries

### 2.2.1 Notation Rules of This Chapter

Bold letters are used for column vectors and matrices, and right superscripts $c$ and $k$ to denote variables represented in the current frame and keyframe, respectively. The secondary right superscripts $l$ and $r$ express the left and right frames of stereo cameras.

For example, let me denote the $i$-th pixel coordinate on a left key image as $\mathbf{p}_i^{k,l} \in \mathbb{R}^2$ with its inverse depth $\rho_i^{k,l} \in \mathbb{R}^+$ as suggested in [37]. The perspective relationship between $\mathbf{p}$ having the inverse depth $\rho$ and corresponding 3D point $\mathbf{X} \in \mathbb{R}^3$ can be represented as $\mathbf{p} = \pi\left(\mathbf{X}\right) : \mathbb{R}^3 \mapsto \mathbb{R}^2$, and its inverse mapping is $\mathbf{X} = \pi^{-1}\left(\mathbf{p}, \rho\right)$. Let me define an image gradient vector of $\mathbf{p}$ on the left key image as $\mathbf{g}^{k,l}\left(\mathbf{p}\right) : \mathbb{R}^2 \mapsto \mathbb{R}^2$. For simplicity, all image gradient vectors are assumed to be normalized.

### 2.2.2 3-D Geometry of Camera Motions and Edges

The 6-DoF camera motion from a current frame to a keyframe is parametrized by Lie algebra $\xi_{c,k}^l \in se\left(3\right)$ where corresponding rotational matrix on special orthogonal group $\mathbf{R}_{c,k}^l \in SO\left(3\right)$ and translation $\mathbf{t}_{c,k}^l \in \mathbb{R}^3$. The 3-D warping function transferring $\mathbf{p}^{k,l}$ to a corresponding pixel point $\mathbf{p}^{c,l}$ on the current frame is defined as,

$$\mathbf{p}^{c,l} = w\left(\mathbf{p}^{k,l}, \xi_{c,k}^l\right) = \pi\left(\mathbf{R}_{c,k}^l \cdot \pi^{-1}\left(\mathbf{p}^{k,l}, \rho^{k,l}\right) + \mathbf{t}_{c,k}^l\right). \tag{2.1}$$

A right camera motion $\xi_{c,k}^r$ can be denoted with an operator $\oplus$ on $se\left(3\right)$ and a fixed stereo pose $\xi_{l,r}^l \in se\left(3\right)$,

$$\xi_{c,k}^r = \xi_{c,k}^l \oplus \xi_{l,r}^l. \tag{2.2}$$

### 2.2.3 ICP-based Edge Alignment

The proposed method estimates camera motions by successively aligning matched pairs of edges within the ICP framework. To get it working, the most probable pixel correspondences among current and key edges should be established in advance. This matching process is called the nearest neighbor searching (NNS) [20]. For a query $\mathbf{q}$, the nearest pair in a pixel set $\mathcal{R} \subset \mathbb{R}^2$ can be founded by a NNS function $nn_{\mathcal{R}}(\mathbf{q})$,

$$nn_{\mathcal{R}}(\mathbf{q}) = \underset{\mathbf{p} \in \mathcal{R}}{\arg\min} \left\| \mathbf{q} - \mathbf{p} \right\|_2 \in \mathcal{R}, \tag{2.3}$$

where $\left\| \cdot \right\|_2$ is a 2-norm operator.

The well-distributed edges, however, are not always guaranteed, and many false edge responses hinder the correct matching. To characterize and find more informative edges, the author develops an edge culling method by making use of the fact that structural edges along object boundaries commonly possess long series of pixels with regular and high image gradients. A detailed explanation follows in the next section.

Using the correspondences, an ICP algorithm estimates an optimal camera motion $\xi^*_{c,k}$ by minimizing the sum of squared distances of the residual vector $\mathbf{d} \in \mathbb{R}^{N_p}$,

$$\xi^*_{c,k} = \underset{\xi \in se(3)}{\arg\min}\, \mathbf{d}^{\mathrm{T}} \mathbf{d}, \tag{2.4}$$

where $N_p$ is the number of matched pixel pairs. The $n$-th element of the residual vector, $d_n$, formulated between the $n$-th pixel $\mathbf{p}^k_n$ on the key image and its matched point in the pixel set $\mathcal{R}^c$ of the current image can be noted as 2-norm distance,

$$d_n = \left\| w\left(\mathbf{p}^k_n, \xi_{c,k}\right) - nn_{\mathcal{R}^c}\left(w\left(\mathbf{p}^k_n, \xi_{c,k}\right)\right) \right\|_2 . \tag{2.5}$$

As the proposed system uses both stereo images to track camera motions, this research proposes a new stereo cost function and additional methods to restrain outliers, which are

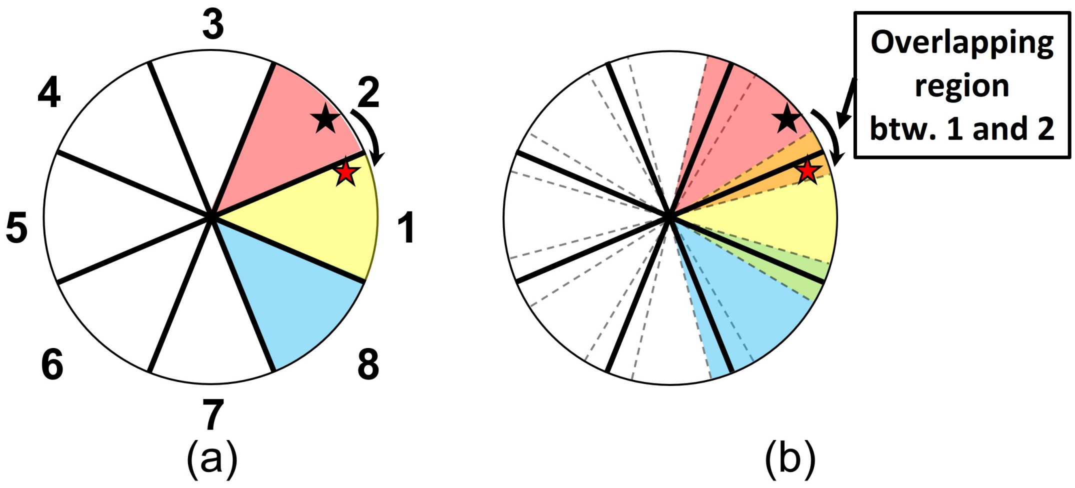


Figure 2.3: **Edge label bins with overlapping regions** (a) the original binning method proposed in [1] divides eight exclusive directional sets where pixels adjacent to boundaries could be wrongly matched, (b) the proposed method with overlapping regions gives flexibility to some extent for boundary pixels.

also discussed in the following sections.

## 2.3 Edge Extraction and Culling

This section addresses details on how edge pixels are distinguished and the salient structural edgelets are extracted out of raw pixels.

### 2.3.1 Edge Labeling using Overlapping Regions

When using edges, the major difficulty originates from the absence of dedicated descriptors for edges. To relieve this problem, [26] and [1] exploit image gradient directions to distinguish edges differently.

In [26], a similarity score between two pixels is quantified by a weighted sum of a pixel Euclidean distance and an inner product of normalized image gradients. Despite improved matching success rates, this method relies on a heuristically-defined weighting parameter between two terms, making it improper to be used in universal situations.

The second work [1] divides edges into eight bins according to image gradient directions, and *absolutely* labels each edge as one of eight mutually exclusive groups like Fig. 2.3(a). In this way, the search space can be effectively reduced, and simultaneously matching success rates be increased.

This research follows the main concept of the *absolute* labeling method but additionally augments overlapping regions between every neighboring bins where pixels can belong as duplicates. As can be seen in Fig. 2.3(a), only with small rotations on the image, the black point labeled as group 2 easily crosses the decision boundary between groups 1 and 2, and both black and red points become mutually exclusive. In this case, the original approach could fail to find the correct pair.

In contrast, by setting up the proposed overlapping regions, the red one is now labeled as a duplicate of groups 1 and 2, and can be included in searching candidates of the black one regardless of some extent of rotations as depicted in Fig. 2.3(b). The labeling result is used to extract salient edgelets and make multiple quadtrees in the following sections.

### 2.3.2 Finding Salient Edgelets out of Labeled Edges

A high signal-to-noise ratio from a number of pixels can be helpful for more accurate motion estimations [1]. However, many points could be redundant to get sufficient estimation performance, and spurious edges could hinder finding correct pairs rather than improve overall performance.

To address both problems, this research proposes an efficient edge culling method that filters false responses on cluttered edges and only sorts out prominent structural edgelets. As shown in Fig. 2.4(a), edges along object rims generally have regular image gradients, and edges in cluttered regions show many unconnected pixels. Given these observations, this research considers a long series of connected edge pixels with the same labeling group as structural edgelets, and, if not, as cluttered edges.

The structural edgelets can be determined by recursively connecting adjacent pixels in the same directions. A simple example is described in Fig. 2.5. The first recursion initiates

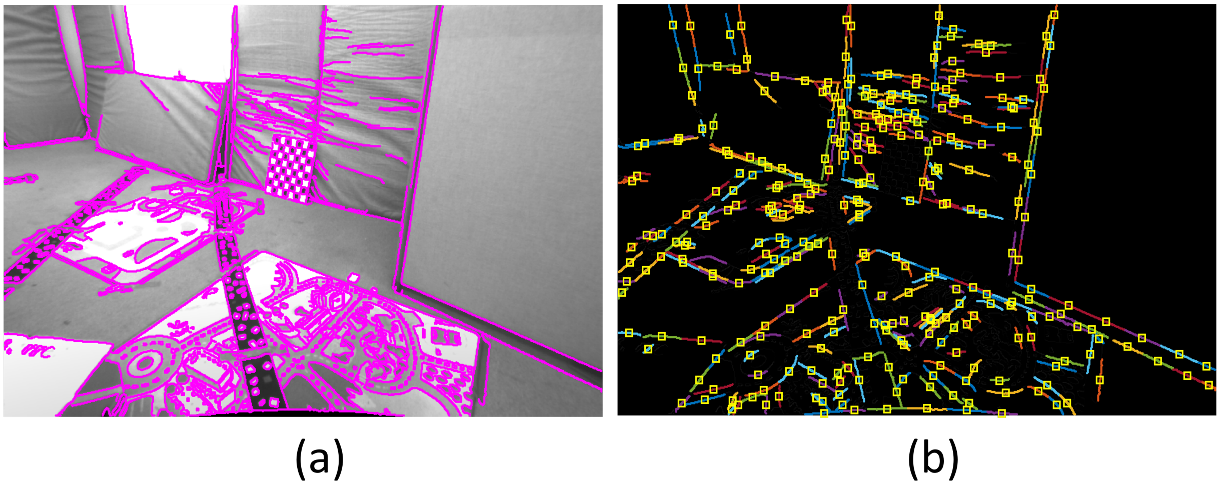


Figure 2.4: **Extracted salient edgelets and center points** (a) raw edges (24,672 pixels), (b) salient edgelets extracted by the proposed method in different colors. The yellow squares represent center points of edgelets. Total 574 center points are well distributed across the image.

at one of the color-coded starting query points. For the current query point, $3 \times 3$ adjacent pixels are stored into a list $L$ if the pixels have the same labels with the query point. If there is no more pixel to be connected, the list $L$ is regarded as a new structural edgelet. The algorithm restarts at non-visited new query points, and ceases when all the edge points in the image are visited.

After grouping, the length of each edgelet is checked by double threshold values, $l_{min}$ and $l_{max}$. As the step (c) in Fig. 2.5, fragmented edgelets under the minimum length $l_{min}$ are more likely to be spurious responses on cluttered regions. Thus, those edgelets are rejected.

For more compact query sets, this module additionally computes the centers of the edgelets and considers them as representative query points. It is found that several long edgelets can yield very sparse center points. To prevent this, the maximum length limit $l_{max}$ is set to get more uniform length edgelets. Then, the prominent edgelets are finally obtained along the object profiles, and the query points are evenly distributed over the entire image area as small yellow squares shown in Fig. 2.4(b). In the author's experience,

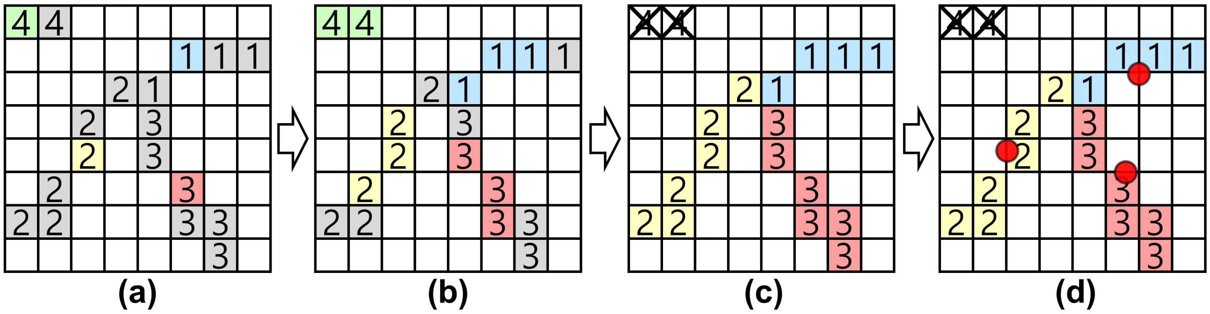


Figure 2.5: **Example of depth-first search for finding salient edgelets** Each color represents a different edgelet. Red points denote centers of respective edgelets.

it is found that the practical value for $l_{max}$ is about 30 in general images like Fig. 2.4.

All procedures can be efficiently performed by adapting the conventional depth-first search algorithm, and a pseudo code of the method is written in Algorithm 1.

## 2.4 Robust Edge Matching via Oriented Quadtrees

For estimating the camera motion between key and current frames, one of the most important and exhaustive parts is to match point pairs repetitively. To realize faster and more accurate ICP-based edge alignments, it is crucial to efficiently find the correct correspondences among the massive number of edge pixels. To this end, this research proposes an accelerated NNS strategy using oriented multiple quadtrees dedicated to the proposed edge-based VO.

### 2.4.1 Generating Multiple Oriented Quadtrees

To begin with, let me regard extracted edgelets of a current image as reference points for making quadtrees, and calculated center points of a key image as query points to be matched. Based on this assumption, multiple quadtrees are built in accordance with eight orientation labels by using a set of salient edgelets on the current image $\mathcal{R}^c \subset \mathbb{R}^2$. It is denoted that each set of edge points consisting of its relevant tree as $\mathcal{R}^c_i \subset \mathcal{R}^c$ where an

**Algorithm 1** Find Salient Edgelets and Centers

```
$E$; an image of labeled edge pixels.
$edgelets$; a list of detected edgelets.
$pts_c$; center pixels of selected edgelets.
for All pixel $q$ in $E$ do
    $L \leftarrow$ an empty list for a new edgelet;
    if $\mathbf{E}\left(q\right)$ is an edge then
        Create an empty $stack$ and push $q$ to $stack$;
        while $!isEmpty\left(stack\right)$ do
            $p \leftarrow frontAndPop\left(stack\right)$;
            for All $p_n$ neighbor of $p$ do
                if $size\left(L\right) < l_{max} \& E\left(p_n\right) = E\left(q\right)$ then
                    Add $p_n$ to $stack$ and $edgelet$;
                end if
            end for
        end while
        if $size\left(L\right) > l_{min}$ then
            Add $L$ to $edgelets$, and $mean\left(L\right)$ to $pts_c$;
        end if
    end if
end for
```

indicator $i$ denotes each directional bin. The difference between a normal quadtree and the proposed trees is illustrated in Fig. 2.6.

As depicted before in Fig. 2.3(b), the pixels near boundaries among neighboring bins are doubly inserted into two neighboring trees. Thus, both boundary query and its correct match can remain reachable to each other regardless of a certain degree of image rotation, which makes the matching process more robust to rotational motions.

Another advantage of multiple quadtrees is the decreased tree depth. As seen in Fig. 2.6, the normal case inevitably has a deeper depth than each of the multiple quadtrees because all points are in only one tree, which results in the increased number of travel nodes to reach the leaf nodes B and C. In contrast, the number of nodes in each of the multiple trees is lesser than in the normal case, which could imply a faster-searching speed.

### 2.4.2 Fast NNS Strategy storing Visited Nodes

At every iteration of the ICP-based motion estimation, it is necessary to warp key points onto the current image and find their correspondences within the current quadtrees.

For a simple explanation, Let me assume a situation to find a matching pair for a single key point $\mathbf{p}^k$ with an orientation labeling $i$. Given the motion $\xi_{c,k}$, let $\mathbf{p}^{k\prime} = w\left(\mathbf{p}^k, \xi_{c,k}\right)$ be a warped point of $\mathbf{p}^k$. Thanks to the orientation labeling, potential candidates can be directly narrowed down within $\mathcal{R}_i^c$, and the nearest pixel can be determined by the NNS function $nn_{\mathcal{R}_i^c}\left(\mathbf{p}^{k\prime}\right)$.

Due to gradual motion updates of the ICP-based approach, warped points move only a few pixels at each iteration. It implies that the previously matched node is much more likely to be re-matched at the very next iteration.

Inspired by this observation, the matching process can be further accelerated by caching the address of the matched nodes at the previous iteration, and warm-starting the search from the cached nodes. If the best match does not reside in the cached node, the procedure restarts to search from the root node, which merely occurs in practice.

When seeking a true match in node C depicted in Fig. 2.6(a), for the normal case, exhaustive re-entry to the root is required at every iteration, and many nodes are visited on the way from the root to the node C. In contrast, the proposed trees in Fig. 2.6(b) can compactly tighten the search space by the orientation label, and moreover, the warm-start from the previously cached node B reduces a large number of visited nodes to reach the node C compared to the normal case.

## 2.5 Motion Tracking and 3-D Reconstruction

### 2.5.1 Stereo Point-to-edge Distances Minimization

The left camera motion $\xi_{c,k}^l$ is estimated by minimizing the stereo point-to-edge normal distances generated by matched edge pairs of the current and keyframes. In Fig. 2.8, it is

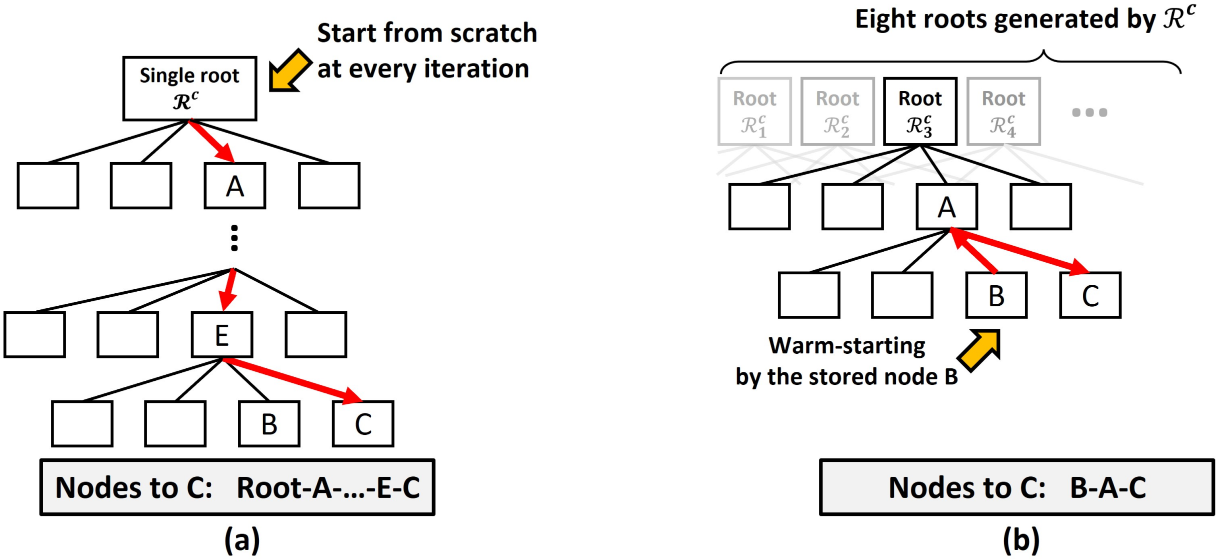


Figure 2.6: **Comparison between a normal single-rooted quadtree and the multiple cached quadtrees** (a) the original quadtree travels through all the nodes from a root to get C. (b) the proposed quadtrees cache the previously visited nodes and warmly start from the cached nodes, which considerably reduces nodes to be traveled.

assumed that the $n$-th query pixel $\mathbf{p}_n^{k,l}$ has an orientation label $i$, and its warped point by the camera motion $\xi_{c,k}^l$ is denoted by $\mathbf{p}_n^{k,l\prime} = w\left(\mathbf{p}_n^{k,l}, \xi_{c,k}^l\right)$. The warped pixel $\mathbf{p}_n^{k,l\prime}$ is matched to the current pixel coordinate $\tilde{\mathbf{p}}_n^{k,l\prime} := nn_{\mathcal{R}_i^{c,l}}\left(\mathbf{p}_n^{k,l\prime}\right)$ by the NNS function. The flow vector of two pixels is defined as,

$$\mathbf{F}_n^l := \mathbf{p}_n^{k,l\prime} - \tilde{\mathbf{p}}_n^{k,l\prime} \in \mathbb{R}^2. \tag{2.6}$$

A scalar value formed by $\mathbf{F}_n^l$ projected onto a unit gradient vector $\mathbf{g}^{c,l}\left(\tilde{\mathbf{p}}_n^{k,l\prime}\right)$ can be used as a signed residual $d_n^l \in \mathbb{R}$,

$$d_n^l = \mathbf{g}^{c,l}\left(\tilde{\mathbf{p}}_n^{k,l\prime}\right)^{\mathrm{T}} \cdot \mathbf{F}_n^l. \tag{2.7}$$

To make use of the absolute scale of the fixed stereo position, an additional residual term is defined induced by the current right image. Analogous to the left case, a warped point onto the right current image of the query point $\mathbf{p}_n^{k,l}$ is denoted by $\mathbf{p}_n^{k,l\prime\prime} = w\left(\mathbf{p}_n^{k,l}, \xi_{c,k}^l \oplus \xi_{l,r}^l\right)$, and its matched right current pixel is represented by $\tilde{\mathbf{p}}_n^{k,l\prime\prime} := nn_{\mathcal{R}_i^{c,r}}\left(\mathbf{p}_n^{k,l\prime\prime}\right)$.

A flow vector on the right current image is also defined as,

$$\mathbf{F}_n^r := \mathbf{p}_n^{k,l\prime\prime} - \tilde{\mathbf{p}}_n^{k,l\prime\prime}, \tag{2.8}$$

and the right residual term can be written as,

$$d_n^r = \mathbf{g}^{c,r}\left(\tilde{\mathbf{p}}_n^{k,l\prime\prime}\right)^{\mathrm{T}} \cdot \mathbf{F}_n^r. \tag{2.9}$$

By arranging total $N_p$ pairs of residuals into one column vector, the residual vector can be formulated as,

$$\mathbf{r} = \left[d_1^l, ..., d_{N_p}^l, d_1^r, ..., d_{N_p}^r\right]^{\mathrm{T}} \in \mathbb{R}^{2N_p}. \tag{2.10}$$

The optimal motion can be estimated by minimizing the weighted sum of squared residuals,

$$\xi_{c,k}^* = \underset{\xi_{c,k}^l \in se(3)}{\arg\min}\, \mathbf{r}^{\mathrm{T}}\mathbf{W}\mathbf{r} \tag{2.11}$$

where $\mathbf{W}$ is a weighting matrix. The motion update $\delta\xi \in se\,(3)$ can be calculated by the second-order Gauss-Newton method as,

$$\delta\xi = -\left(\mathbf{J}^{\mathrm{T}}\mathbf{W}\mathbf{J}\right)^{-1}\mathbf{J}^{\mathrm{T}}\mathbf{W}\mathbf{r}, \tag{2.12}$$

with the Jacobian matrix $\mathbf{J} = \frac{\partial \mathbf{r}}{\partial \xi} \in \mathbb{R}^{2N_p \times 6}$. The motion update is iteratively implemented until convergence,

$$\xi_{c,k}^l \leftarrow \xi_{c,k}^l \oplus \delta\xi. \tag{2.13}$$

Note that, as seen in Fig. 2.7, the unit vectors of the image gradient vectors on the edge pixels can be regarded as the 2-D normal vector locally perpendicular to the direction of the edge. Thus, the image gradient direction can be used as the unique 2-D normal vector on the edge pixels because the image gradient direction is invariant unless the image is not changed. Thanks to this property of the image gradient vector on the edges, the

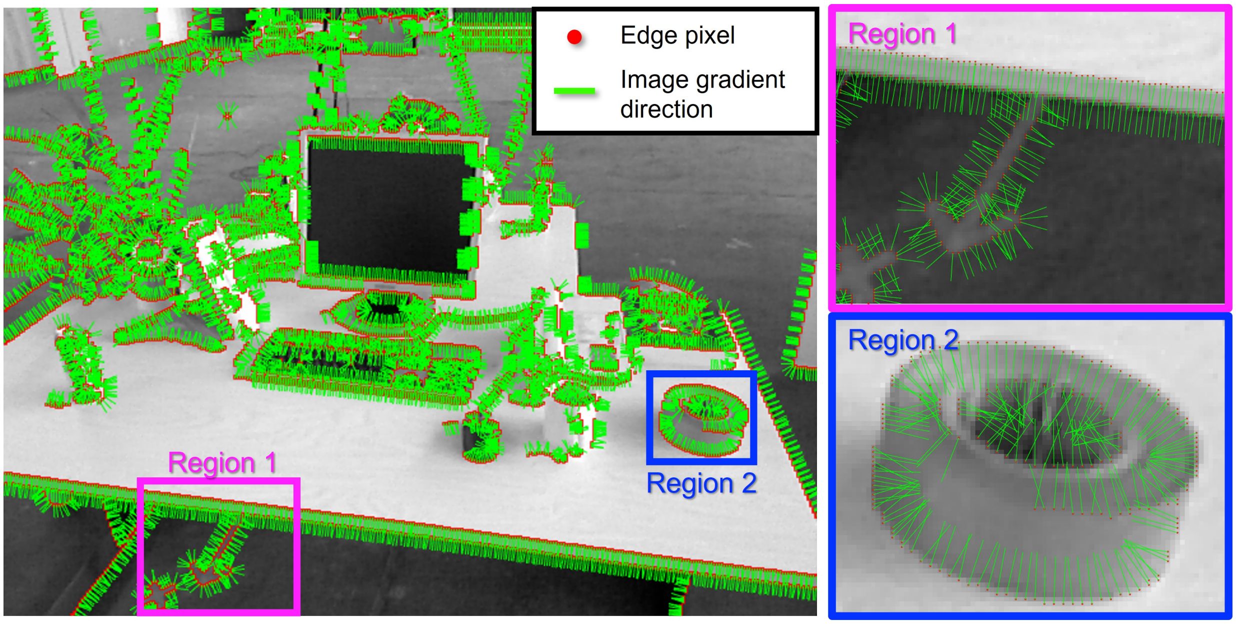


Figure 2.7: **Illustration of the image gradient vector on the image edge pixels**

signed distance in (2.7) can be formulated and give constraining effects on the optimization problem.

To suppress inevitably occurring wrong match pairs, a t-distribution weighting scheme [36] is employed for $\mathbf{W}$, and it is recalculated by the current residual distribution at every iteration. If the matching results are correct in both stereo images, two gradient vectors on the matched pixels should have a consistent direction. From this, additional outliers can be detected by testing whether their inner product is under a certain margin or not,

$$\mathbf{g}^{c,l}\left(\mathbf{p}_n^{k,l\prime}\right)^{\mathrm{T}} \cdot \mathbf{g}^{c,r}\left(\mathbf{p}_n^{k,l\prime\prime}\right) < \eta \tag{2.14}$$

where a threshold value $\eta$ is empirically set to 0.99 in this work, corresponding to about ten degrees angular difference.

If the camera motion is excessively large or the number of keypoints decreases under a certain level, the keyframe is replaced with the current frame, and all the points on the key frame with inverse depth information are updated and propagated to the new keyframe, which is discussed in the following.

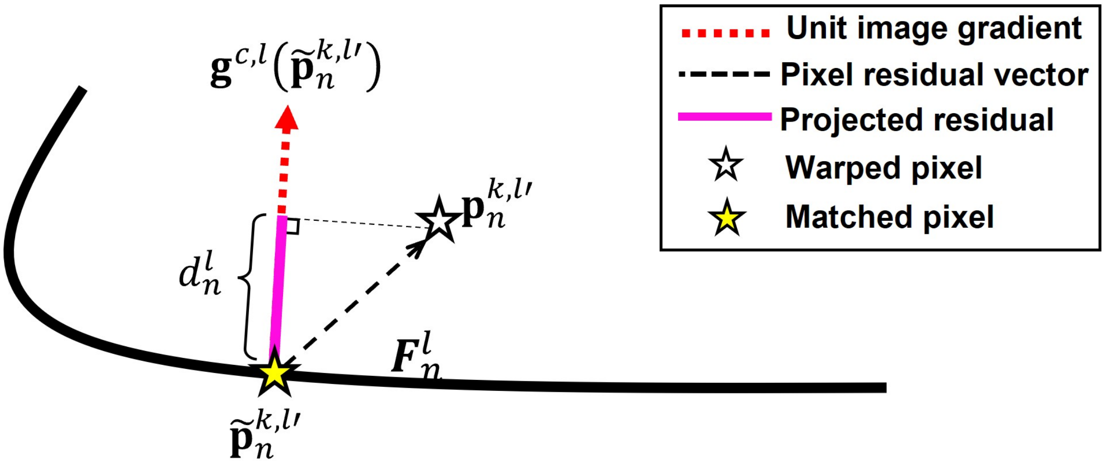

$\mathbf{g}^{c,l}(\tilde{\mathbf{p}}_n^{k,l\prime})$
$\mathbf{p}_n^{k,l\prime}$
$d_n^l$
$\boldsymbol{F}_n^l$
$\tilde{\mathbf{p}}_n^{k,l\prime}$
Unit image gradient
Pixel residual vector
Projected residual
Warped pixel
Matched pixel

Figure 2.8: **Illustration of the point-to-edge normal distance induced by a matched pair of points**

### 2.5.2 Edge Inverse Depth Reconstruction and Propagation

As reported in [17], a laterally-fixed stereo can yield reliable 3-D information only for vertical edges because horizontal features cannot be distinguished by the static stereo. For the sake of complete 3-D depth maps in all directions, this work follows the static and temporal stereo inverse depth estimation scheme [17]. The depth reconstruction procedure consists of four steps as illustrated in Fig. 2.9.

For probabilistic updates, it is assumed that an inverse depth observation $\rho$ follows the normal distribution with the standard deviation $\sigma$ around itself. The geometry error model is used to compute $\sigma$ proposed in [24]. Note that the disparity is searched by evaluating the normalized cross-correlation (NCC) with a $5 \times 9$ patch along an epipolar line, and the sub-pixel disparity is refined by using parabolic interpolation.

Using the static stereo of the keyframe, the initial inverse depth $\rho_k$ with $\sigma_k$ is first calculated. In this step, only vertical edges can be reliably reconstructed (step ① of Fig. 2.9).

Then, the previous observation $\rho_k$ is warped to the left current image for the temporal stereo update. In Fig. 2.9, by the static stereo configuration between current and keyframes,

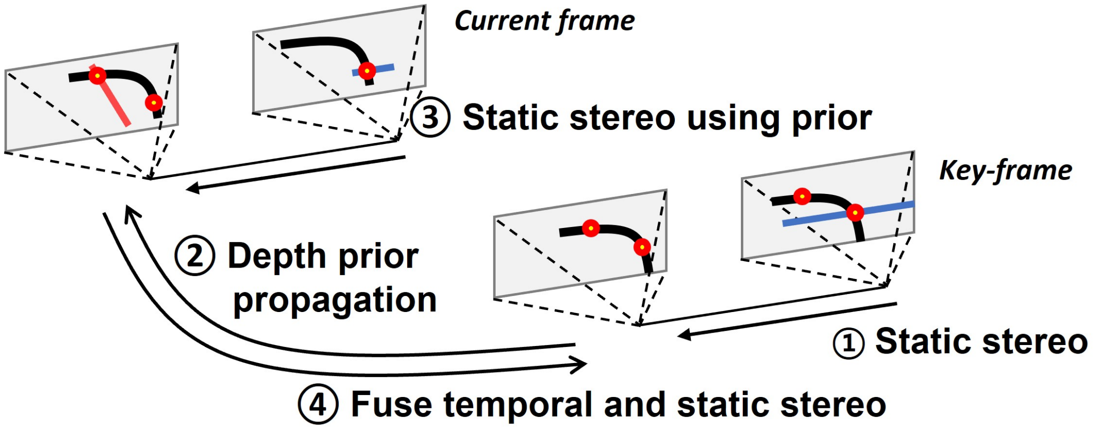

Current frame
③ Static stereo using prior
Key-frame
② Depth prior propagation
① Static stereo
④ Fuse temporal and static stereo

Figure 2.9: **Static and temporal stereo configurations**

horizontal edges can now be recovered, and $\rho_k$ can be updated if a red generalized epipolar line for each observation is almost perpendicular to the edge profile as depicted in Fig. 2.9 (steps ② and ③).

The inverse depth value $\rho_c$ with $\sigma_c$ detected by the temporal stereo is re-projected back, and updated with $\rho_k$ to find an optimal estimation $\rho_k^*$,

$$\rho_k^* = \frac{\sigma_k^2 \rho_c + \sigma_c^2 \rho_k}{\sigma_k^2 + \sigma_c^2},\ \sigma_k^{*2} = \frac{\sigma_k^2 \sigma_c^2}{\sigma_k^2 + \sigma_c^2}, \tag{2.15}$$

where $\sigma_k^*$ is the fused standard deviation (step ④ of Fig. 2.9).

The inverse depth values are consecutively updated by this procedure for every incoming image, and if the new keyframe is received, inverse depth values are propagated to the new keyframe like [17].

## 2.6 Leveraging Feature Modalities: VO System combining Edges and Points

Despite the robust performance of the edge-based VO, there still exists a problem when using edge-only VO. In this section, this problem is addressed, and the solution for this is

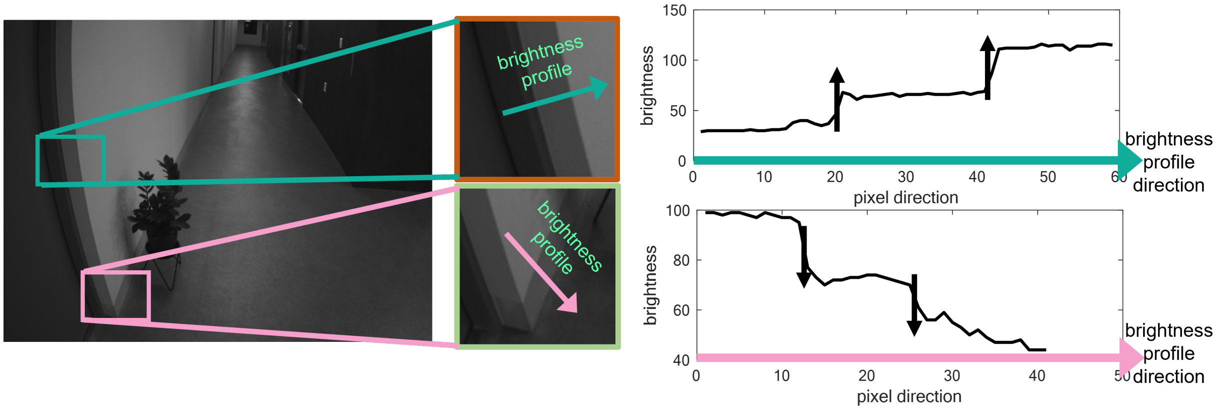


Figure 2.10: **Ambiguous edges in the modern indoor scenes**

also proposed: a hybrid visual odometry combining the edge and point features altogether.

Fig. 2.10 shows a potential limitation of the edge-based pixel matching methods. In modern indoor scenes, neighbor edges often have the same directions with similar monotonous color distributions. In this case, edges are difficult to be distinguished by the proposed gradient-based edge matching scheme. Exemplary scene and brightness profiles perpendicular to the edges are depicted in Fig. Along the green line, the same rising edges are detected, and they can be regarded as the same edges for gradient-based edge-matching methods. Analogously, the same problem occurs along the pink line. This problem mainly influences the edge-based VO methods using edge distance fields like [1] because the same type of nearby edges generates multiple ambiguous distance minima, which attracts the algorithm to wrong estimations.

Like this, only using edge-based VO is not always superior in all cases, especially in feature-abundant scenes as Fig. 2.11. Even in edge-dominant environments, few point features can still be available as Fig. 2.12. A comparison of characteristics of various image features is written in Fig. 2.13. As seen in the figure, combining edges and points could have complementary advantages.

Thus, this section presents an extended version of the edge-based visual odometry system combined with the conventional point-based approach. Among the cluttered raw edges,

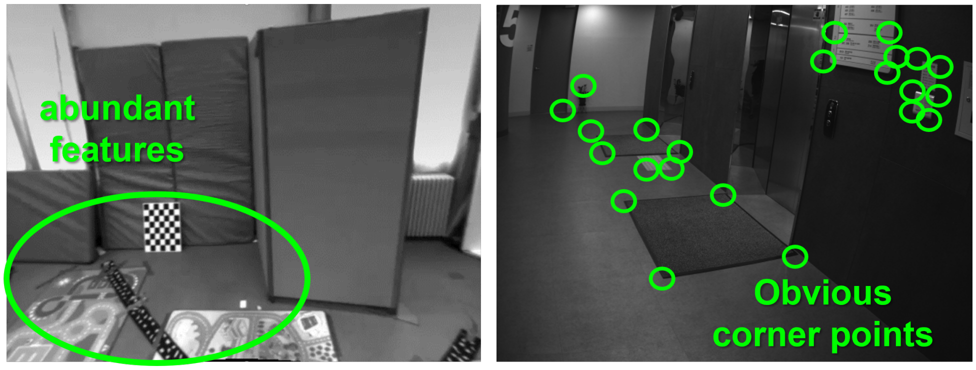


Figure 2.11: **Point abundant regions**

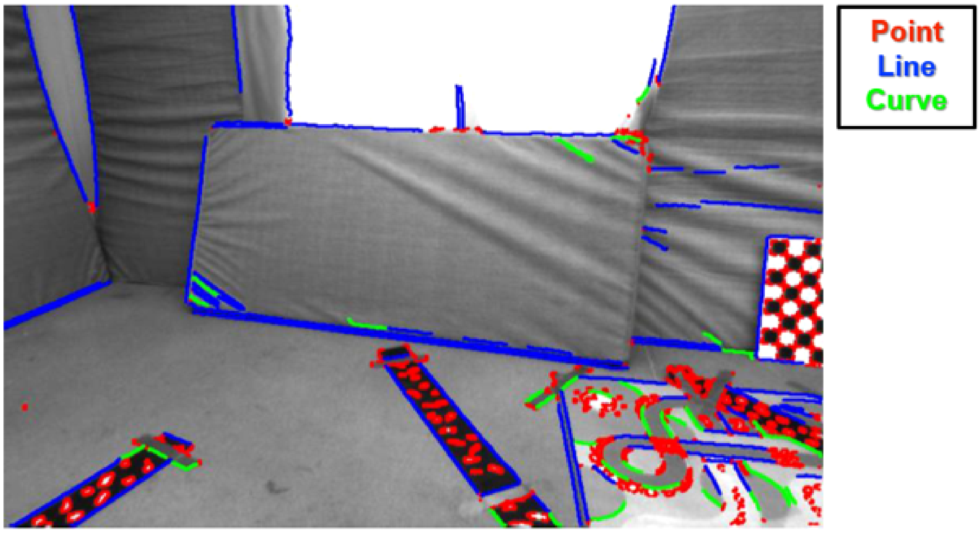
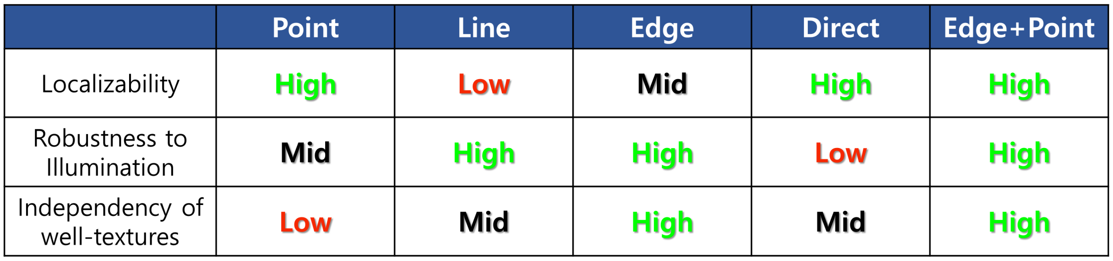


Figure 2.12: **Edges, points, and lines in edge-dominant scenes**

the prominent structural edgelets and corner points are efficiently classified by testing the homogeneity of image gradient orientations. At the matching step, the prominent structural edgelets are matched by multiple quadtrees generated according to image gradient orientations. Then, 6-DoF camera motions are optimized by minimizing edge and point joint re-projection errors. The robust and accurate performance of the proposed method is evaluated on public datasets by additionally imposing drastic brightness changes. The experimental results show that the proposed hybrid method has promising advantages in real-world scenarios with few features and arbitrary brightness conditions.



| | Point | Line | Edge | Direct | Edge+Point |
|---|---|---|---|---|---|
| Localizability | High | Low | Mid | High | High |
| Robustness to Illumination | Mid | High | High | Low | High |
| Independency of well-textures | Low | Mid | High | Mid | High |

Figure 2.13: **Comparison of characteristics of various image features**

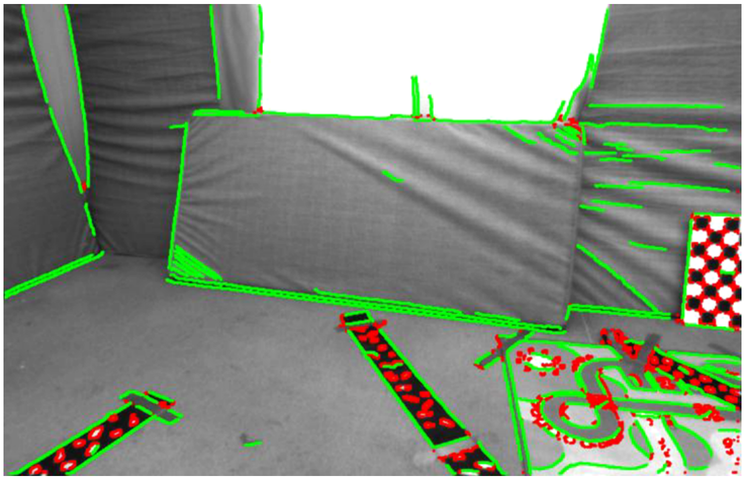


Figure 2.14: **Salient edges and points in the image**

### 2.6.1 Selective Point and Edge Extraction with Image Binning

The corners of the object boundary line are easy to distinguish due to the constant brightness change pattern, but corners without regularity in complex patterns are difficult to distinguish from adjacent pixels, degrading matching performance. Accordingly, only corner regions in which image slopes in the same direction continuously occur are used as structural corners. In complex patterns, indistinguishable edges occur, but apparent point features can be seen. The Shi-Tomasi score is calculated as a measure of the strength of the point feature for the corresponding corner, and the image is divided into 20×20 squares as seen in Fig. 2.15 to use the highest-scoring pixels in each square as a point feature. An example can be seen in Fig. 2.14. By doing this, the computational load can be maintained to a moderate level thanks to the consistent number of features in an image.

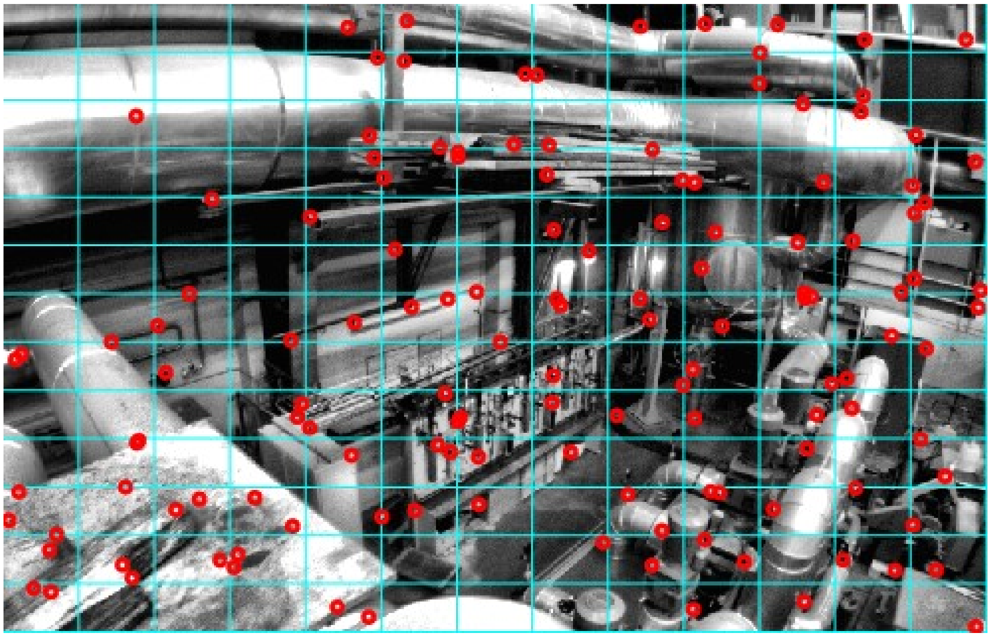


Figure 2.15: **Feature bucketing strategy to maintain the number of edge and point pixels**

### 2.6.2 Edge and Point-based Hybrid Camera Motion Estimation with Illumination Compensation

This subsection introduces a hybrid camera motion estimation combining edge and point features with illumination compensation. The global brightness variation between the reference image $I_k$ and the current image $I_c$ can be modeled in the affine form as

$$I_k = e^{\alpha} I_c + \beta, \tag{2.16}$$

where $\alpha, \beta \in \mathbb{R}$ are illumination changes of contrast and brightness between frame $k$ and $c$, respectively. Different from the edge-only formulation avoiding the brightness changes by using edges, the brightness change model is explicitly incorporated into the camera motion estimation problem as optimization parameters in the hybrid VO.

The hybrid motion estimation problem consists of three cost functions: a) point-to-edge distance, b) photometric error near edges, and c) photometric error near points. In

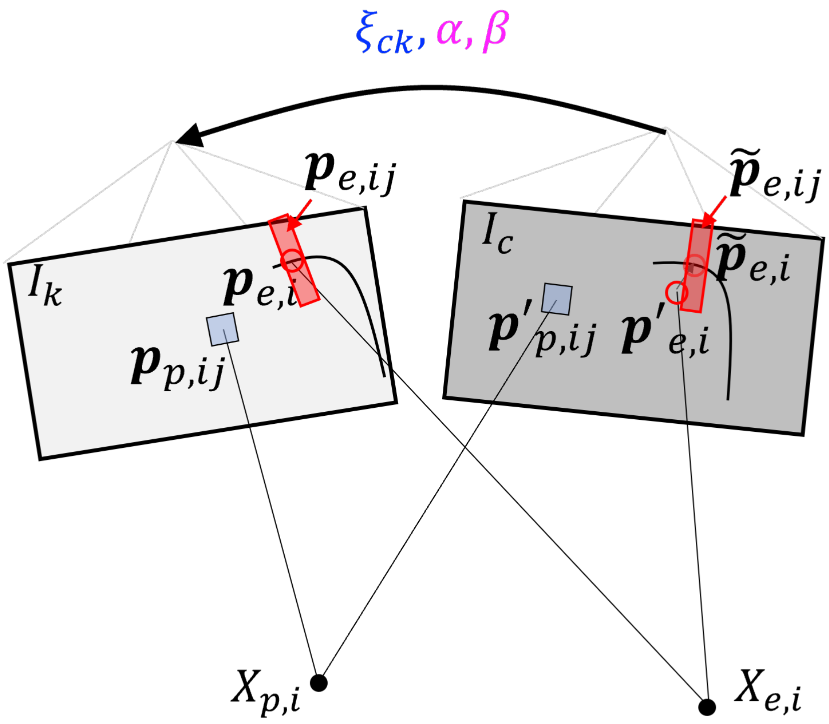


Figure 2.16: **Illustration of geometry of the consecutive cameras, pixels, and image patches**

the followings, each cost function will be detailed.

In this formulation, the edge pixel pairs previously matched in Section 2.4 are used.

First, let $\mathbf{p}'_{e,i} \in \mathbb{R}^2$ be a corresponding edge pixel of the $i$-th pixel $\mathbf{p}_{e,i} \in \mathbb{R}^2$ warped to the current image by $\xi_{ck}$ where $\xi_{ck} \in se(3)$ is the three-dimensional camera motion between the reference image and the current image. Additionally, let $\tilde{\mathbf{p}}_{e,i} \in \mathbb{R}^2$ be the matched pixel of the $\mathbf{p}'_{e,i}$ on the current image by multiple quadtrees. By defining the normalized image gradient vector on the pixel $\tilde{\mathbf{p}}_{e,i}$ as $g(\tilde{\mathbf{p}}_{e,i})$, the point-to-edge distance $r_{e,i} \in \mathbb{R}$ can be derived as,

$$r_{e,i} = g(\tilde{\mathbf{p}}_{e,i})^\top \cdot \left(\mathbf{p}'_{e,i} - \tilde{\mathbf{p}}_{e,i}\right). \tag{2.17}$$

As shown in Fig. 2.16, red patches denote the neighbor pixel regions of the edge pixels. Let $\mathbf{p}_{e,ij} \in \mathbb{R}^2$ be the $j$-th pixel of the patch near the $i$-th edge pixel $\mathbf{p}_{e,i}$ on the previous image $I_k$, and $\tilde{\mathbf{p}}_{e,ij} \in \mathbb{R}^2$ be the $j$-th pixel of the patch near the $i$-th matched edge pixel

$\tilde{\mathbf{p}}_{e,i}$ on the current image $I_c$. The brightness value can be obtained by defining the image brightness function $I_c[\mathbf{p}] : \mathbb{R}^2 \mapsto \mathbb{R}^+$. Then, the brightness difference of the two red patches between the current image and the reference image can be used as the additional residual terms. It is called the photometric error near the edge, and the residual term $r_{e,ij} \in \mathbb{R}$ of the $j$-th patch pixel of the $i$-th edge pixel can be written as,

$$r_{e,ij} = I_c\left[\tilde{\mathbf{p}}_{e,ij}\right] - e^{\alpha} I_k\left[\mathbf{p}_{e,ij}\right] - \beta. \tag{2.18}$$

In this hybrid formulation, the photometric error near the point feature is additionally considered. The $i$-th point feature on the previous image is defined as $\mathbf{p}_{p,i} \in \mathbb{R}^2$, and the $j$-th patch pixel near the $\mathbf{p}_{p,i}$ is written as $\mathbf{p}_{p,ij} \in \mathbb{R}^2$ which is denoted by the blue rectangular regions in Fig. 2.16. Analogous to the edge photometric error, the residual term $r_{p,ij} \in \mathbb{R}$ of the $j$-th patch pixel of the $i$-th corner point pixel can be written as,

$$r_{p,ij} = I_c\left[\mathbf{p}'_{p,ij}\right] - e^{\alpha} I_k\left[\mathbf{p}_{e,ij}\right] - \beta, \tag{2.19}$$

where $\mathbf{p}'_{p,ij} \in \mathbb{R}^2$ is the warped corner point pixel of $\mathbf{p}_{p,ij}$ onto the current image.

By aggregating $N_e$ edge pixels, $N_p$ corner point pixels, $n_p$ patch points near the corner points, and $n_e$ patch points near the edge pixels, the problem of estimating the camera motion $\xi_{ck}$ and the image brightness parameters $\alpha, \beta$ can be formulated as a form of a cost minimization as,

$$\underset{\boldsymbol{\xi_{ck}},\boldsymbol{\alpha},\boldsymbol{\beta}}{\text{argmin}}\ w_e \left( \sum_{i=1}^{N_e} r_{e,i}^2 \right) + w_p \left( \sum_{i=1}^{N_p} \sum_{j=1}^{n_p} r_{p,ij}^2 + \sum_{i=1}^{N_e} \sum_{j=1}^{n_e} r_{e,ij}^2 \right), \tag{2.20}$$

where $w_e$ and $w_p$ are the weighting parameters to normalize the influence of each term of the optimization problem. To solve this problem, the Levenberg-Marquardt nonlinear optimization method is employed.

Note that when the VO converges, the average value of the point-to-edge distance is under one pixel, and the average values of the photometric error terms are under ten. In this work, the weighting parameters are determined to make the overall cost under ten in average after convergence. Thus, the weight values $w_e$ and $w_p$ are computed by the below rules,

$$\begin{aligned} w_e &= 10/N_e \\ w_p &= 1/\left(N_e \cdot n_e + N_p \cdot n_p\right), \end{aligned} \tag{2.21}$$

where the two weight values are re-calculated whenever the new image incomes with varying $N_e$ and $N_p$.

Note that, even though the residual term in (2.19) is governed by all parameters, $\xi_{ck}, \alpha, \beta$, the residual term in (2.17) does not rely on $\alpha, \beta$ because the edge matching can be conducted regardless of the brightness changes. Thus, the author considers that (2.17) is a function of $\xi_{ck}$ only and (2.19) is a function of $\alpha, \beta$ only. From this, the convergence of the optimization problem in (2.20) can be improved, which is verified by the various experiment results in Chapter 5.

## 2.7 Performance Analysis

The author found that the overall VO performance is affected by three dominant factors: 1) usage of multiple quadtrees, 2) storing matched nodes, and 3) the edge culling method with the minimum length $l_{min}$ for edgelets. This section evaluates the benefits of each factor with extensive variations of parameter settings. All computations are conducted in C++ with `-O2` compiler flag on AMD Ryzen 5 3.6 GHz CPU.

### 2.7.1 Analysis 1: Normal Quadtree vs. Multiple Oriented Quadtrees

This analysis demonstrates the enhanced performance of the matching process by using the proposed multiple quadtrees. Two settings are considered: a normal quadtree and multiple quadtrees without storing the visited nodes. To separately evaluate each part, only labeled

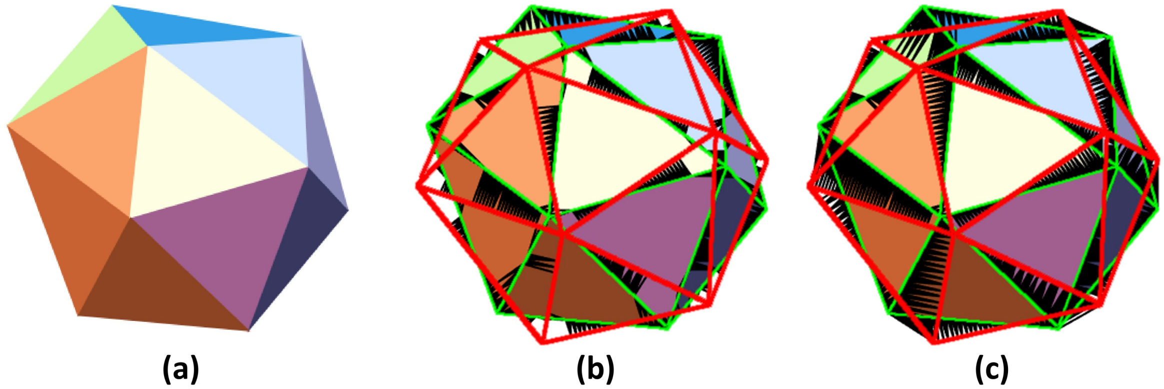


Figure 2.17: **Example of the matching results of the normal quadtree and the proposed multiple quadtrees** (a) a polyhedron model, (b) a matching result using the normal quadtree, (c) a matching result using the multiple oriented quadtrees. Key and current edges are in green and red, respectively. Black lines connect matched pairs.

raw edges without the edge culling method are used in this analysis.

In the beginning, for an intuitive example, Fig. 2.17 shows matching results on the polyhedron image with the 15 degrees of the camera roll. The result of using the proposed quadtrees shows more robustness to rotations and qualitatively desirable matching tendencies.

For quantitative evaluation, the monotonous and cluttered images are warped as seen in Fig. 2.18 along u, v, and camera roll axes and compare the iterations and elapsed times required for the matching sequences of the ICP algorithm to converge. Errors are applied with 15-pixel deviations along the u- and v- axes, and 10 degrees rotations with respect to the roll axis.

According to the results on the monotonous image described in Figs. 2.19(a-c), the number of iterations slightly decreases by using the multiple quadtrees. In contrast, the improvements by the multiple quadtrees on the cluttered image are more prominent in regions with large displacement as shown in Figs. 2.20(a-c). Note that the time consumption of the multiple quadtrees decreases by about 30 % on average compared to the normal one. The author noticed that the number of calculations for the NNS function is considerably

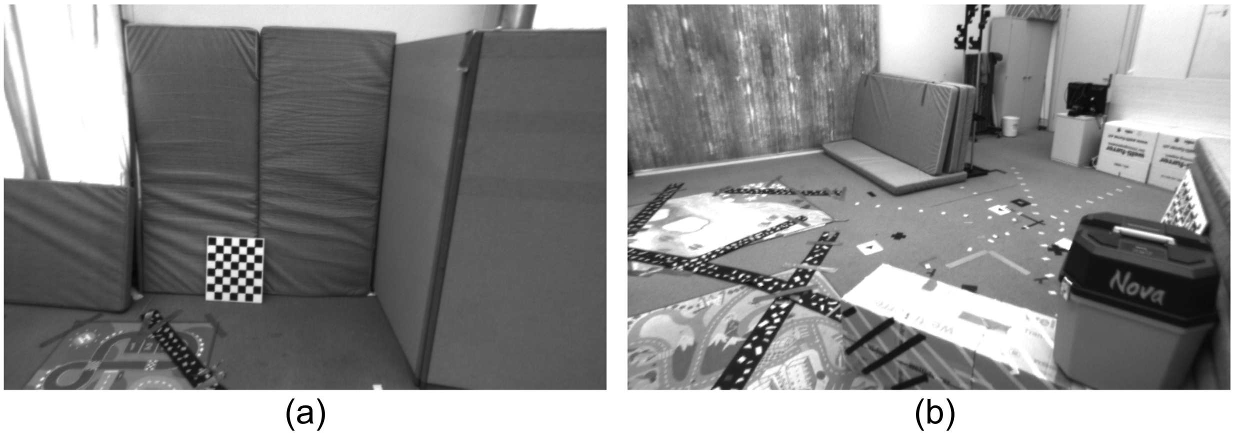


Figure 2.18: **Selected two images on the EuRoC V1_01 dataset for quantitative evaluations** (a) a simple image and (b) a cluttered image.

reduced because a query point is compared to a fraction of reference points only within its related tree out of eight trees.

For the rolling motions larger than ±5 degrees, iterations are saturated as in Fig. 2.20(c), which means that the ICP algorithm falls into local minima. In the author's experiences, some very cluttered regions in Fig. 2.18(b) yield a lot of wrong matches, and the estimation fails in this case. It will be addressed in Section 2.7.3.

### 2.7.2 Analysis 2: Effect of storing the Previously Matched Nodes

This subsection demonstrates further improvements in the matching speed by storing the previously matched nodes. In this analysis, the same simulation settings in the previous section are used, and only node storage functionality is switched on.

As can be seen in Figs. 2.19 and 2.20, the number of iterations remains almost the same with the pure multiple quadtrees because the result of starting from the stored nodes is theoretically identical to starting from the root node. Thanks to the warm-start from the stored nodes, the number of nodes traversing considerably decreases as depicted in Fig. 2.6, and consequently, the time consumption is further reduced up to 70 % compared to the normal quadtree in most cases. When the ICP algorithm fails, large discrepancies in

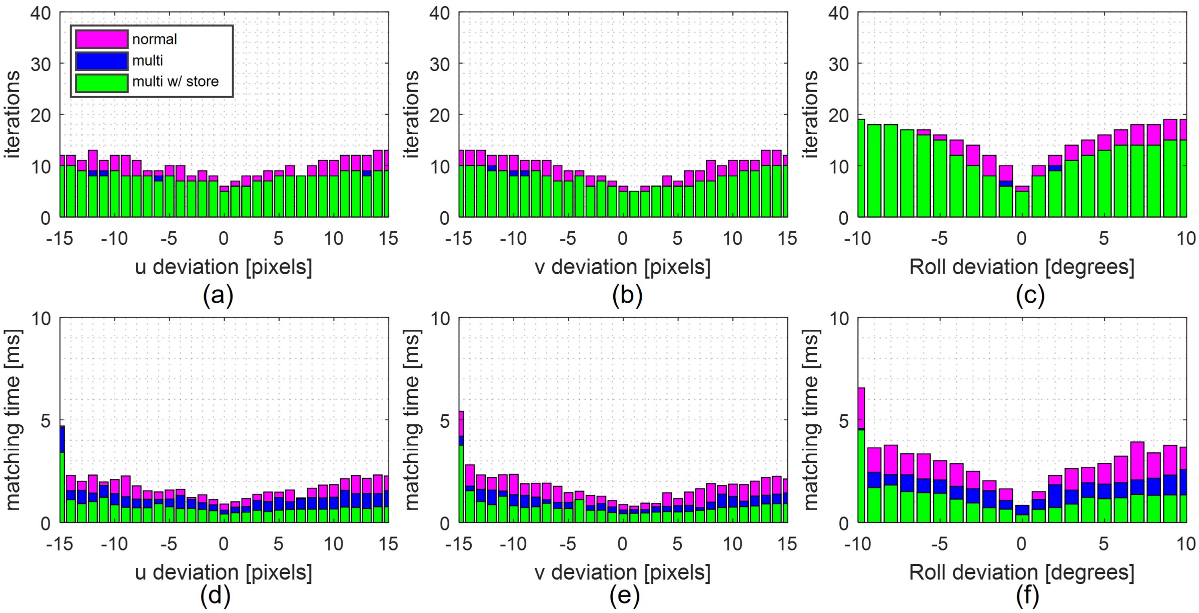


Figure 2.19: **ICP iterations and time consumption to align the simple image in Fig. 2.18(a).** (a-c) ICP iterations versus camera motions, (d-f) time consumption versus camera motions. Overall 9,031 query points are used.

iterations occur because the wrongly matched previous nodes can lead the current ICP iteration toward another wrong direction.

### 2.7.3 Analysis 3: Effect of the Edge Culling Method

This subsection evaluates the effect of the edge culling method on the efficiency and robustness of the ICP process by changing $l_{min}$. In this analysis, $l_{max}$ is fixed to 30, and $l_{min} = \{5, 15, 25\}$ are considered. Compared to the previous simulation setting, the only difference is replacing raw edges with the culled edges.

The edge culling results with the three $l_{min}$ values are reported in Fig. 2.22. The cluttered responses on the floor and false responses on the wall are effectively suppressed by the increasing $l_{min}$ while the prominent edgelets remain.

Fig. 2.21 presents the comparison results. The results of $l_{min} = 5$ case are similar to the result of the setting without the edge culling method because many complex responses

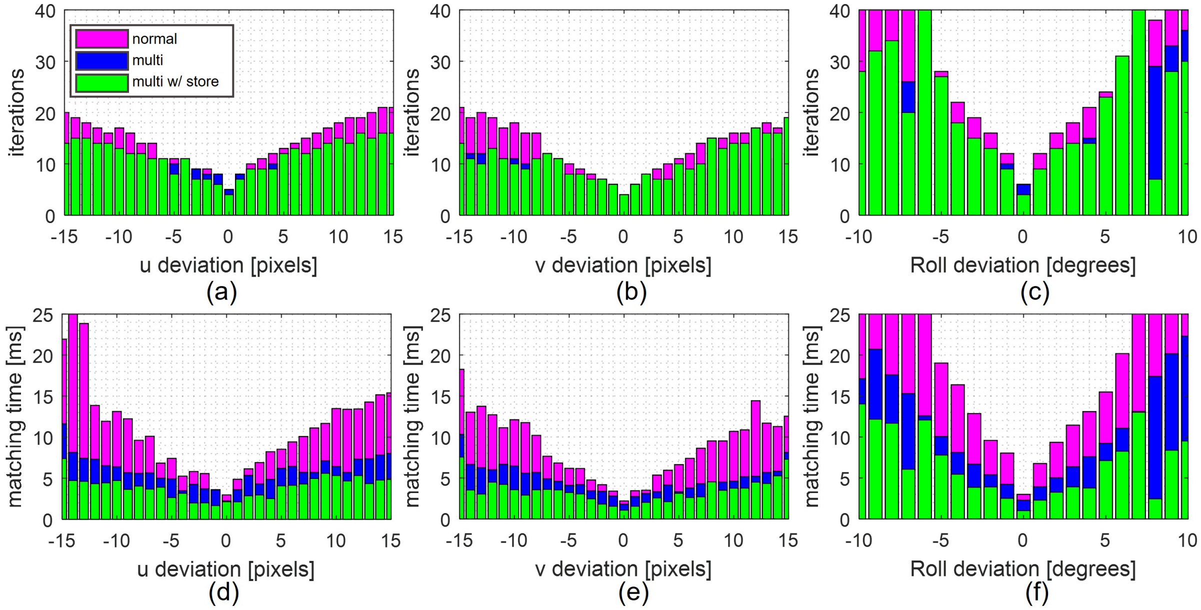


Figure 2.20: **ICP iterations and time consumption to align the cluttered image in Fig. 2.18(b).** (a-c) ICP iterations versus camera motions, (d-f) time consumption versus camera motions. Overall 33,602 query points are used.

still exist in the left image like Fig. 2.22. For higher values of $l_{min}$, the number of iterations and the calculation time significantly decrease despite large motions thanks to the reduced cluttered edges as depicted in the right two images of Fig. 2.22. From these results, It can be concluded that the edge-culling method considerably enhances both the robustness and efficiency of the proposed edge-based ICP motion estimation.

Note that, to show the overall performance of the proposed method, the author evaluates the overall performance of the proposed method using EuRoC stereo datasets [38] and author-collected datasets gathered in low-textured indoor office situations. Results can be found in Chapter 5.

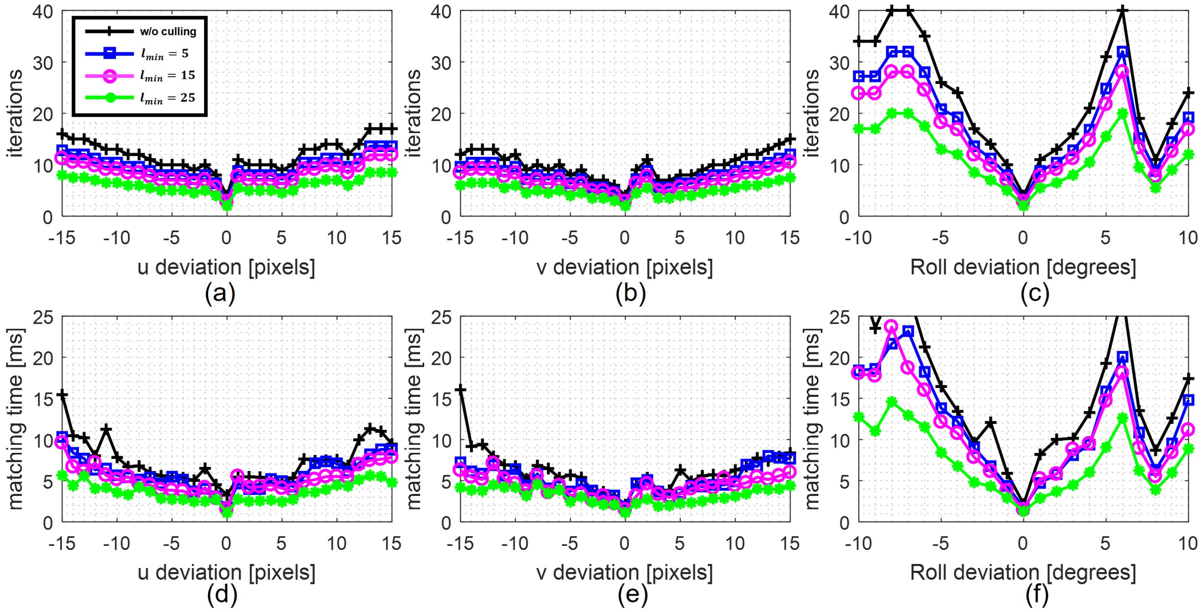


Figure 2.21: **ICP iterations and time consumption by using the edge culling method to align the cluttered image in Fig. 2.18(b).** (a-c) ICP iterations versus camera motions, (d-f) time consumption versus camera motions.

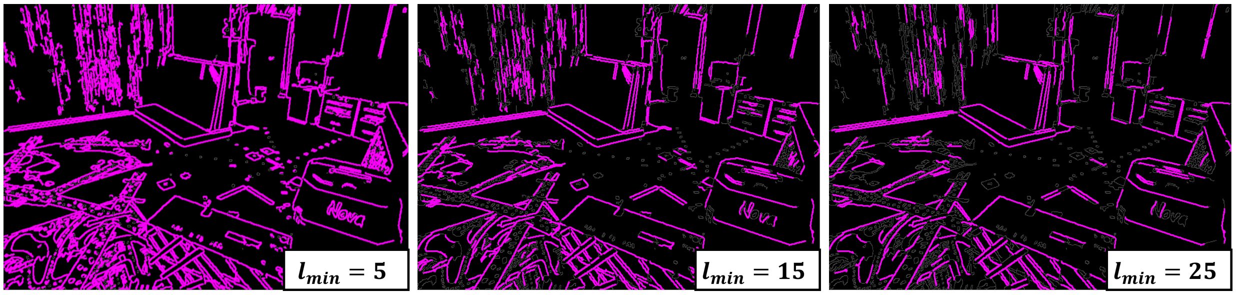


Figure 2.22: **Results of the edge culling method with** $l_{min} = \{5, 15, 25\}$

# 3

# Scale-aware Monocular Visual Odometry using Vehicle Kinematic Constraint

This chapter proposes a new approach to scale-aware monocular visual odometry (VO) and extrinsic calibration using constraints on camera motion by vehicle kinematics. The main idea is to utilize the Ackermann steering model to observe the absolute metric scale in turning motion.

To describe the motion of the camera attached to the vehicle, the unknown camera-vehicle relative pose is firstly estimated by the proposed extrinsic calibration method. To stably observe scale, turn regions are detected, and an observer is designed to estimate the absolute scale as a function of the camera rotation and direction of translational motion during turning. Using the observed scale, an absolute scale recovery is proposed to estimate the unknown scale between turns. Because the proposed scale observer becomes singular near zero rotation, sensitivity analysis is conducted on the scale observer, and appropriate conditions for stable scale estimation are investigated. For quantitative evaluation of the extrinsic calibration and the absolute scale recovery, synthetic driving datasets are randomly generated with various noise conditions, and the performance of each module is statistically evaluated by Monte Carlo simulations on the synthetic datasets.

To evaluate the overall performance, the author implements the proposed method and state-of-the-art monocular and stereo VO methods in the public outdoor driving KITTI dataset, and the proposed method shows competitive scale recovery performance with no external sensor and no assumption on surroundings such as planar ground landmarks. To show promising applicability, the author collects real-world driving datasets in two multi-floor underground parking lots, and demonstrates the accurate absolute scale recovery performance of the proposed method in indoor driving situations.

## 3.1 Introduction

Navigation is one of the fundamental capabilities of an autonomous mobile vehicle. For navigation, ego-motion estimation using cameras called visual odometry (VO) has received attention due to its compact setting and rich environment expression from an image. Thus, the VO has been actively studied with various configurations: a single camera [11, 39], multiple cameras [40, 41], VO with an inertial measurement unit (IMU) [42, 43], and combining vehicle dynamics [44, 45, 46].

Especially, VO using a single camera, monocular VO (MVO), is an attractive solution for automobile navigation due to its minimum setting. Furthermore, because one or more cameras can be easily found in most vehicles as forms of driving assistant systems and user-mounted dashboard cameras, the MVO implementation targeted for mobile vehicles is highly valuable.

However, due to the monocular projective nature, absolute metric information disappears from an image, and MVO can only yield up-to-scale translation motion, making the MVO-only setup more challenging without additional metric measurements. It is called the scale ambiguity problem [15]. Although the scale ambiguity often means scale drifts over time, it is specifically used only to denote absolute scale vanishing in this chapter.

A common approach to recover the scale in the MVO is integrating additional sensors providing metric information such as inertial measurements from IMU [42],[47], low-resolution time-of-flight range sensors [48], and a single and multiple distance meters settings [49]. Although the utilization of additional sensors can improve performance, the need for the sensors and precise extrinsic calibration among them might not be affordable for arbitrary settings.

For the ground vehicle settings, a popular approach is to utilize the consistent height of the camera rigidly attached to the vehicle and planar ground observations with plenty of image features [50, 5, 51, 52, 53, 54, 55, 56]. They show successful performance when planar features are available abundantly. Although they target ground vehicle applications, the vehicle kinematics is not fully exploited but implicitly considered as a planar and level traverse of the camera. Furthermore, in most research, the relative pose between the camera and the vehicle is commonly assumed to be an identity, which is not always true in vehicular settings.

This chapter introduces a scale-aware monocular VO system utilizing a vehicle kinematic motion model. Different from the previous scale-aware MVO works [50, 5, 51, 52, 53, 54, 55, 56], the vehicle kinematics is explicitly used to model the monocular camera motions. To exactly obtain the fixed relative pose of the camera and the vehicle, a self-contained extrinsic calibration method is developed to estimate the relative pose between the camera and the vehicle. Then, the author designs a scale observer that estimates the absolute translation scale from the geometric constraint of the frame-to-frame camera turning motion. To propagate the observed absolute scale on turning regions, a method is proposed to recover the unobserved scale between turns.

In the following, related works are reviewed, and the main contributions of this chapter are listed compared to the related works. The references are listed and analyzed in Fig. 3.1.

### 3.1.1 Literature Review

The author surveys monocular VO methods with scale awareness, and categorizes them into three types according to methods to obtain absolute scale information: 1) additional sensors, 2) environmental properties, and 3) a learning-based approach.

**Additional sensors:** Additional sensors are frequently used to observe the metric scale of the motion estimation. For VO, multiple cameras [40], [41], [42] with known relative pose and baselines are widely used to triangulate landmarks in 3-D space and estimate the metric camera translation motion. For more compact settings, monocular visual-inertial odometry (V-IO) is proposed [43], [47]. By double-integrating acceleration measurements, the metric translation change of the IMU is incorporated into the MVO motion optimization problem.

Other works utilize different sensor modalities that provide metric information to the MVO framework. In [48], [49], multiple 1-D point laser sensors are used to obtain the metric distance of the center pixel of the camera and to recover the trajectory scale.

As mentioned before, two hurdles exist to implement this type of method; the calibration among various sensors with different modalities is nontrivial, and some sensor combinations might not be readily available.

**Environmental properties:** Most monocular scale-aware VO methods [50, 5, 51, 52, 56, 53, 54, 55] utilize two environmental conditions: planar ground observations and constant camera height. An early work [50] extracts point features on the road from a fixed quadrilateral image region to estimate the planar homography transform between frames. By decomposing the homography matrix, they compute the camera height from the plane and adjust the scale of camera motion using consistent camera height assumption.

The strategy using the planar homography is still popular in recent studies, and several variations are proposed to extract planar information accurately and stably; [5, 51] combine sparse features and direct illumination on the ground plane to estimate the homography matrix, and [52] and [56] geometrically model the plane regions by the Delaunay triangulation with point feature nodes and stably prune out outliers. In [54] and [55], robust

| Paper | Type | Minimal camera setting? | Independence on point distributions? | Metric scale recovery? | OK with unknown camera-vehicle pose? | Fully utilizing vehicle kinematics? |
|---|---|---|---|---|---|---|
| Wang, 2017 (ICCV) [40] | add. sensors | X, 2 cams. | O | O | - (no vehicular VO) | X |
| Won, 2020 (ICRA) [41] | add. sensors | X, 4 cams. | O | O | - (no vehicular VO) | X |
| Qin, 2018 (T-RO) [43] | add. sensors | X, 1 cam. + IMU | O | O | - (no vehicular VO) | X |
| Chiodini, 2020 (T-IM) [48] | add. sensors | X, 1 cam. + 1 laser dist. | O | O | - (no vehicular VO) | X |
| Olmez, 2021 (DSP) [49] | add. sensors | X, 1 cam. + 4 laser dist. | O | O | - (no vehicular VO) | X |
| Wang, 2017 [59] | learning | △, 1 cam. train w/ stereo | O | X (up-to-scale) | X (assume known) | X |
| Tateno, 2017 [60] | learning | △, 1 cam. train w/ stereo | O | X (up-to-scale) | X (assume known) | X |
| Yang, 2018 [61] | learning | △, 1 cam. train w/ stereo | O | X (up-to-scale) | X (assume known) | X |
| Liu, 2019 [62] | learning | △, 1 cam. train w/ stereo | O | X (up-to-scale) | X (assume known) | X |
| Campos, 2022 [64] | learning | △, 1 cam. train w/ stereo | O | X (up-to-scale) | X (assume known) | X |
| Chen, 2019 [65] | learning | △, 1 cam. + IMU | O | O | X (assume known) | X |
| Han, 2019 [66] | learning | △, 1 cam. + 3-D LiDAR | O | O | X (assume known) | X |
| Kitt, 2011 [50] | env. properties | O, 1 cam. | X (planar fts.) | O | X (assume known) | △ (level motion) |
| Fanani, 2017 (IV) [51] | env. properties | O, 1 cam. | X (planar fts.) | O | X (assume known) | △ (level motion) |
| Wang, 2018 (ICRA) [52] | env. properties | O, 1 cam. | X (planar fts.) | O | X (assume known) | △ (level motion) |
| Zhang, 2021 (T-M) [56] | env. properties | O, 1 cam. | X (planar fts.) | O | X (assume known) | △ (level motion) |
| Tian, 2021 (ICRA) [52] | env. properties | O, 1 cam. | X (planar fts.) | O | X (assume known) | △ (level motion) |
| Zhou, 2020 (T-ITS) [54] | env. properties | O, 1 cam. | X (planar fts.) | O | X (assume known) | △ (level motion) |
| Fan, 2020 (ACCESS) [53] | env. properties | O, 1 cam. | X (planar fts.) | O | X (assume known) | △ (level motion) |
| **Ours** | **env. properties** | **O, 1 cam.** | O | O | O | **O (kinematically modeled cam. Motion)** |

Figure 3.1: **Literature review for the scale-aware monocular visual odometry algorithms.**

plane fitting is proposed. In recent work [53], road regions are segmented pixel-wise by deep learning to robustly find planar features.

These methods show stable and accurate scale-maintaining performance in planar feature-abundant environments; however, they may become infeasible in some regions with no texture on ground planes. Furthermore, most research assumes a known attachment pose of a vehicle-mounted camera or assume zero-pitch camera pose.

**Learning-based approaches:** In recent years, deep learning has undoubtedly achieved considerable advances in computer vision, and many deep applications are derived from several influential works such as [57], [58]. Following the trend, a number of MVO attempts using deep learning are also introduced [59, 60, 61, 62, 63, 64]. In [59], an end-to-end MVO network is proposed by directly training conventional VO results using deep recurrent convolutional neural network (RCNN), and other methods [60, 61, 62, 63, 65, 66, 67] utilize deep depth prediction in training steps. By using depth, these methods can provide consistently scaled translation motion over sequences; however, still yield up-to-scale estimation only due to the monocular nature. To fill the metric gap, deep monocular V-IO (MV-IO) methods [65, 66, 67] are emerging recently.

It is noted that most existing learning-based methods require more data than monocular images, such as stereo images [61] or 3-D LiDAR points [62], in the training step of MVO or inference step of MV-IO. Even more, machine learning methods still suffer from a generalization gap between training and test sets, and their performance might degrade in unseen conditions.

According to the author's survey, it is found that there are few approaches operating independently of the additional sensors and assumptions on surrounding environments and landmark distributions. Especially, the MVO methods with the scale recovery for vehicles mainly focus on indirectly using vehicle characteristics, such as planar ground features and consistent height of the camera. In several cases, the camera-vehicle relative pose is also assumed to be known.

This chapter proposes the scale-aware MVO framework that explicitly utilizes the vehicle kinematic constraints on the camera motion. A self-contained camera-vehicle extrinsic pose calibration method is also introduced for the completeness of the formulation.

### 3.1.2 Contributions of the Chapter

Compared to the related works, major contributions of the chapter are listed as follows;

- An arbitrarily attached camera pose to the vehicle can be estimated by proposing a self-contained camera-vehicle extrinsic pose calibration method using camera motion constrained by vehicle kinematics.
- By utilizing local geometry of the constrained camera motion, a new scale observation method can be formulated when the vehicle turns. The scale observer is also theoretically analyzed to determine stable states for observing the scale.
- Unobserved scale between turning regions can be estimated by the absolute scale using the metric scale on the turning motion.

Note that, different from the scale-aware MVO using the planar ground features [50, 5, 51, 52, 53, 54, 55, 56], the proposed method has an additional advantage of no need for assumption on the uncontrollable external environment such as ground feature distributions.

### 3.1.3 Algorithm Overview

The proposed algorithm in this chapter is illustrated in Fig. 3.2, and the rest of the chapter is structured as follows: Section 3.2 describes preliminaries including notation rules used in this chapter, vehicle motion model, visual processing and data structures required for the proposed method. In Section 3.3, a camera-vehicle extrinsic pose calibration method is proposed to estimate arbitrarily installed monocular camera pose with respect to the vehicle. In Section 3.4, an absolute scale observer is designed by using the kinematically constrained camera motion model in turning motion, and the absolute scale recovery method between turning regions is proposed in Section 3.4.3. Section 4.2 presents an in-depth analysis of the

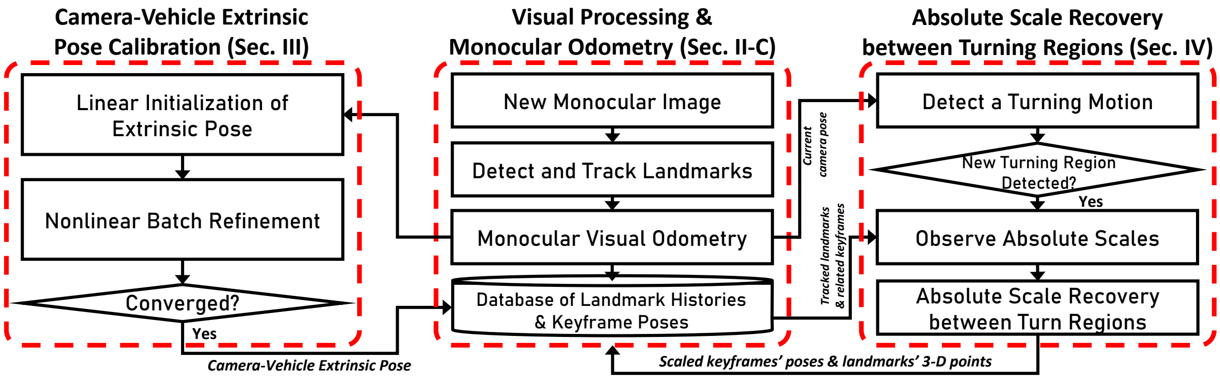

Camera-Vehicle Extrinsic Pose Calibration (Sec. III)
Linear Initialization of Extrinsic Pose
Nonlinear Batch Refinement
Converged?
Yes
Camera-Vehicle Extrinsic Pose
Visual Processing & Monocular Odometry (Sec. II-C)
New Monocular Image
Detect and Track Landmarks
Monocular Visual Odometry
Database of Landmark Histories & Keyframe Poses
Current camera pose
Tracked landmarks & related keyframes
Absolute Scale Recovery between Turning Regions (Sec. IV)
Detect a Turning Motion
New Turning Region Detected?
Yes
Observe Absolute Scales
Absolute Scale Recovery between Turn Regions
Scaled keyframes' poses & landmarks' 3-D points

Figure 3.2: **Block diagram of the proposed scale-aware monocular visual odometry and extrinsic calibration system**

proposed modules and demonstrates the comparable performance of the proposed method on publicly available datasets. The final part of Section 4.2 highlights the effectiveness of the proposed method, especially in indoor driving circumstances by experiments on author-collected indoor driving datasets. The author-collected datasets and related parameters are publicly shared as rosbag files at `https://chkim.net/scalemvo`.

## 3.2 Preliminaries

Before detailed description, notations and the 3-D geometry between a monocular camera and landmarks are defined. Then, the derivation of the monocular camera motion model constrained by vehicle kinematics is proposed. The front-end visual processing and data structures for the proposed system will be explained at the end of this section.

### 3.2.1 Notation Rules of This Chapter

Throughout this chapter, the author expresses column vectors with bold lowercase letters, and matrices are in bold capital letters. The exception is for using $\mathbf{X}$ to denote a 3-D point. Let $\mathbf{X}_i \in \mathbb{R}^3$ be the $i$-th 3-D point represented in the world frame $\{W\}$, and $\mathbf{X}_{ij} \in \mathbb{R}^3$ be the expression of $\mathbf{X}_i$ in the $j$-th camera frame $\{C_j\}$. The 3-D rotation matrix and translation vector from $\{C_a\}$ to $\{C_b\}$ are described as $\mathbf{R}_{C_b}^{C_a} \in \mathrm{SO}(3)$ and $\mathbf{t}_{C_b}^{C_a} \in \mathbb{R}^3$, respectively, and the corresponding rigid body transform is defined as $\mathbf{T}_{C_b}^{C_a} := \left[\mathbf{R}_{C_b}^{C_a}, \mathbf{t}_{C_b}^{C_a}; \mathbf{0}_3^\top, 1\right] \in \mathrm{SE}\,(3)$ where $\mathbf{0}_3$ is a 3-D zero vector. The projection relationship of $\mathbf{X}_i$ to the corresponding 2-D pixel $\mathbf{p}_{ij} \in \mathbb{R}^2$ on the pixel plane of $j$-th camera frame can be computed by $\pi_j\left(\mathbf{X}_i\right) \in \mathbb{R}^2$ by defining a function $\pi_j\left(\cdot\right) : \mathbb{R}^3 \mapsto \mathbb{R}^2$ projecting a 3-D point expressed in $\{W\}$ onto the pixel plane of $\{C_j\}$. For simplicity, abbreviated notations $\mathrm{c}\left(\cdot\right)$, $\mathrm{s}\left(\cdot\right)$, $\mathrm{t}\left(\cdot\right)$ are used throughout this chapter to denote cosine, sine, and tangent functions, respectively.

### 3.2.2 Camera Motion constrained by Vehicle Kinematics

As depicted in Fig. 3.3, the chassis part of a four-wheeled automotive vehicle is designed for all wheels to experience concentric circular motions. This kinematics, called the Ackermann steering geometry [68], enforces locally planar and circular motion.

As shown in Fig. 3.3, it is considered that the vehicle frame $\{V\}$ is on the rear axle of the vehicle, and the z- and x-axes of $\{V\}$ head forward and right of the vehicle, respectively. Using this, the vehicle motion from $\{V_{j-1}\}$ to $\{V_j\}$ can be represented as

$$\mathbf{R}_{V_j}^{V_{j-1}} = \begin{bmatrix} \mathrm{c}\psi_j & 0 & \mathrm{s}\psi_j \\ 0 & 1 & 0 \\ -\mathrm{s}\psi_j & 0 & \mathrm{c}\psi_j \end{bmatrix}, \mathbf{t}_{V_j}^{V_{j-1}} \begin{bmatrix} \rho_j\,\mathrm{s}\gamma_j \\ 0 \\ \rho_j\,\mathrm{c}\gamma_j \end{bmatrix}. \tag{3.1}$$

where $\psi_j, \rho_j \in \mathbb{R}$ are turning angle and distance between centers of $\{V_{j-1}\}$ to $\{V_j\}$, respectively, and $\gamma_j := \psi_j/2$.

The motion of the camera rigidly attached to the vehicle can be modeled by vehicle kinematics. It is considered that the original camera frame $\{C\}$ is at the distant $L \in \mathbb{R}$ from the origin of $\{V\}$ along the z-axis of $\{V\}$, and the camera pose is $\mathbf{Q} \in \mathrm{SO}(3)$. The author additionally defines an auxiliary camera frame $\{A\}$ sharing the origin of $\{C\}$ but having the same pose with $\{V\}$, i.e., $\mathbf{R}_A^V = \mathbf{I}_3 \in \mathrm{SO}(3)$. The translation vector between $\{V\}$ and $\{A\}$ is $\mathbf{t}_A^V = [0, 0, L]^\top$ where $\mathbf{I}_3$ is a 3-D identity matrix.

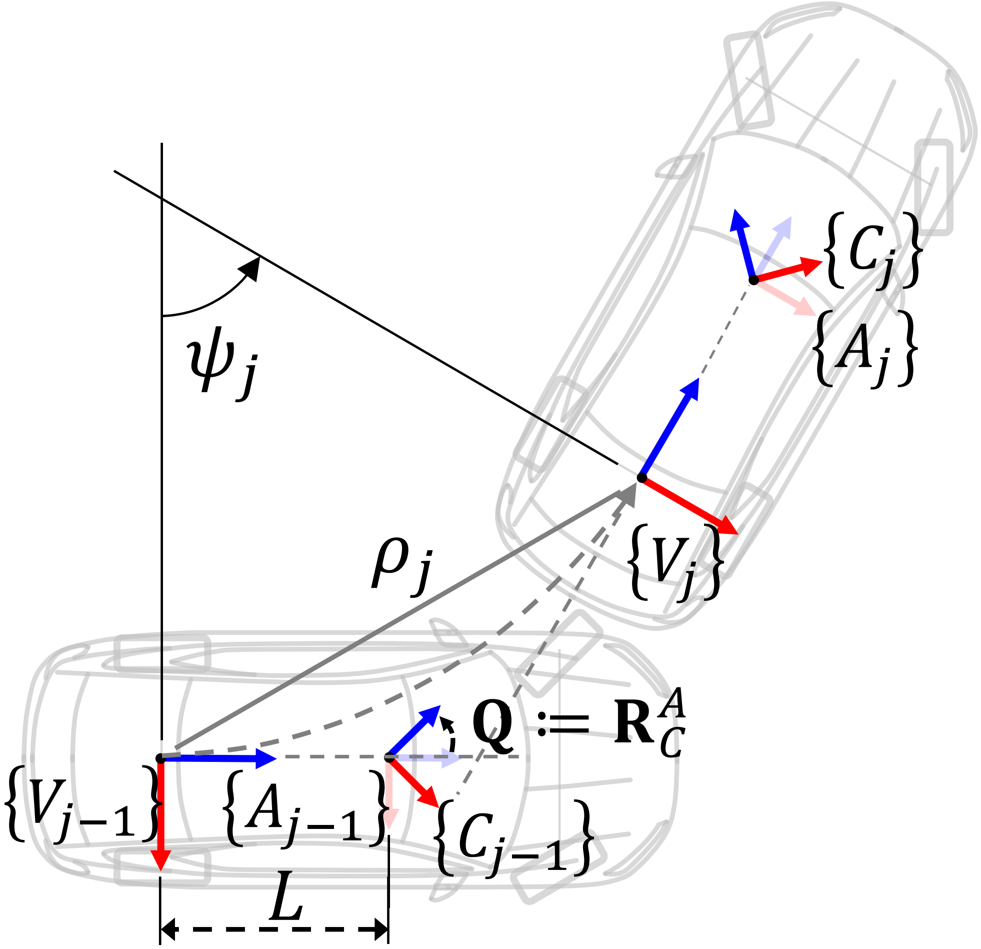


Figure 3.3: **Illustration of the vehicle kinematics** This figure shows the Ackermann steering geometry of the vehicle between $\{V_0\}$ and $\{V_1\}$. Red and blue arrows denote x- and z-axes of each frame. By the right-hand rule, the y-axis directs to the paper. The shaded frames are auxiliary camera frames.

The relative motion between $\{A_{j\text{-}1}\}$ and $\{A_j\}$ $\mathbf{T}_{A_j}^{A_{j\text{-}1}} \in \mathrm{SE}(3)$ can be represented as

$$\mathbf{T}_{A_j}^{A_{j\text{-}1}} = \mathbf{T}_V^A \mathbf{T}_{V_j}^{V_{j\text{-}1}} \mathbf{T}_A^V, \tag{3.2}$$

where rotation and translation parts of $\mathbf{T}_{A_j}^{A_{j\text{-}1}}$ are written as

$$\mathbf{R}_{A_j}^{A_{j\text{-}1}} = \begin{bmatrix} \mathrm{c}\psi_j & 0 & \mathrm{s}\psi_j \\ 0 & 1 & 0 \\ -\mathrm{s}\psi_j & 0 & \mathrm{c}\psi_j \end{bmatrix}, \ \mathbf{t}_{A_j}^{A_{j\text{-}1}} = \begin{bmatrix} \rho_j \mathrm{s}\gamma_j + L\mathrm{s}\psi_j \\ 0 \\ \rho_j \mathrm{c}\gamma_j - L + L\mathrm{c}\psi_j \end{bmatrix}. \tag{3.3}$$

Finally, the constrained camera motion $\mathbf{T}_{C_j}^{C_{j\text{-}1}} \in \mathrm{SE}(3)$ can be written as

$$\mathbf{T}_{C_j}^{C_{j\text{-}1}} = \mathbf{T}_A^C \mathbf{T}_{A_j}^{A_{j\text{-}1}} \mathbf{T}_C^A = \left[\mathbf{Q}^\top \mathbf{R}_{A_j}^{A_{j\text{-}1}} \mathbf{Q}, \mathbf{Q}^\top \mathbf{t}_{A_j}^{A_{j\text{-}1}};\ \mathbf{0}_3^\top, 1\right] \tag{3.4}$$

where $\mathbf{T}_C^A = \left[\mathbf{Q}, \mathbf{0}_3; \mathbf{0}_3{}^\top, 1\right] \in \mathrm{SE}(3)$.

The above derivation is analogous to the vehicular MVO research [44] incorporating the vehicle kinematics to make the 1-point MVO. However, [44] used two major assumptions: zero displacement $L = 0$ and ideal camera pose $\mathbf{Q} = \mathbf{I}_3$, which might be invalid in general camera settings. In fact, the author of [44] reported that the two assumptions are valid only when the steering angle is sufficiently small. In the large steering motion, $L$ is no longer negligible because of increasing terms multiplied by $L$ in $\mathbf{t}_{A_j}^{A_{j\text{-}1}}$.

In this chapter, to deal with the general camera installation, the camera-vehicle extrinsic pose calibration method is proposed in Section 3.3. In addition, the nonzero $L$ is considered to realize the scale-aware MVO, which will be detailed in Section 3.4.

### 3.2.3 Visual Processing Front-end and Data Structures

As in other feature-based VO algorithms, the proposed method in this chapter utilizes associations between visual landmark correspondences and camera frames. Requirements for the proposed scale awareness module are listed.

Each visual landmark should store:

- 2-D pixel tracking history over images
- Address of frames where the landmark was seen
- 3-D point of the landmark represented in $\{W\}$

Each image frame should include:

- Address of landmarks observed in the frame
- 6-DoF camera motion from $\{W\}$

For the visual landmark, the author uses the ORB image feature [28]. To evenly distribute landmarks throughout the image, the image is divided into $n_s$ cells, and $n_f$ features are selected for each cell with the highest FAST scores.

Existing landmarks are tracked for every new image, and store all tracking histories. The proposed method selects keyframes among the image frames to reduce the problem size and to obtain a sufficiently large turning motion between frames. To obtain the camera motion, the proposed MVO module is implemented by following the motion tracking scheme of the successful MVO method, ORB-SLAM2 [11].

## 3.3 Camera-Vehicle Extrinsic Pose Calibration

The exact extrinsic pose $\mathbf{Q}$ of vehicle-installed cameras, such as driving assistance cameras and custom dashcams, are not generally available. In this section, the author introduces the two-step calibration method to estimate the camera-vehicle extrinsic pose by only using the motion of the camera installed in the vehicle.

### 3.3.1 Problem Formulation

The kinematic constraint of the vehicle is generated by a chassis part. In normal driving conditions, it can be considered that the vehicle body part experiences the same rigid body motion with the chassis part. In this case, the motion of the camera attached to the body part can also be expressed by the constrained motion model in (3.4).

Based on the above description, desired conditions of the calibration problem for the $j$-th frame are written as

$$\hat{\mathbf{R}}_j = \mathbf{Q}^\top \mathbf{R}_j \mathbf{Q}, \quad \hat{\mathbf{t}}_j = \mathbf{Q}^\top \mathbf{t}_j, \tag{3.5}$$

where simplified notations are defined as

$$\hat{\mathbf{R}}_j := \mathbf{R}_{C_j}^{C_{j-1}}, \;\; \mathbf{R}_j := \mathbf{R}_{A_j}^{A_{j-1}} \in \mathrm{SO}(3) \tag{3.6}$$

and

$$\hat{\mathbf{t}}_j := \mathbf{t}_{C_j}^{C_{j-1}}, \mathbf{t}_j := \mathbf{t}_{A_j}^{A_{j-1}} \in \mathbb{R}^3, \tag{3.7}$$

respectively. Note that unconstrained camera motion $\hat{\mathbf{R}}_j$ and $\hat{\mathbf{t}}_j$ can be computed by the MVO algorithm.

As (3.3) and (3.4), right-hand sides of two equations in (3.5) are functions of $\mathbf{q}_s$, $\rho_j$, and $\psi_j$ where $\mathbf{q}_s \in \mathbb{R}^4$ is a unit quaternion of $\mathbf{Q}$. Let me define a parameter vector with unknowns as

$$= \left[ {\mathbf{q}_s}^\top, \rho_1, \cdots, \rho_N, \psi_1, \cdots, \psi_N \right]^\top \in \mathbb{R}^{2N+4}. \tag{3.8}$$

By aggregating $N$ poses, an optimization problem with respect to  can be formulated as

$$\operatorname{argmin} \sum_{j=1}^{N} \|\hat{\mathbf{R}}_j - \mathbf{Q}^\top \mathbf{R}_j \mathbf{Q}\|_F^2 + \|\hat{\mathbf{t}}_j - \mathbf{Q}^\top \mathbf{t}_j\|_F^2, \tag{3.9}$$

where $\|\cdot\|_F$ is the Frobenius norm.

Note that the problem in (3.9) is a large-scale nonlinear batch optimization problem. Without proper initial parameter values, it could fall into the wrong minima, or diverge. To prevent this, the author first calculates the initial guess of each part of  separately by linear algebraic approaches.

### 3.3.2 Linear Initialization of $\psi_j$ and $\mathbf{Q}$

First, it is explained how to extract initial guesses of turning angles $\psi_j$ from the unconstrained camera rotations $\hat{\mathbf{R}}_j$ regardless of the unknown $\mathbf{Q}$. Then, using the initial $\psi_j$, a linear algebraic approach is proposed to calculate the initial value of $\mathbf{q}_s$.

#### Extracting $\psi_j$ from the unconstrained rotation $\hat{\mathbf{R}}_j$

In the rotation part of (3.5), $\hat{\mathbf{R}}_j$ and $\mathbf{R}_j$ are *similar* matrices by $\mathbf{Q}$. By the property of similar matrices, the two matrices should have the same eigenvalues regardless of a choice of $\mathbf{Q} \in \mathrm{SO}(3)$.

From the definition of $\mathbf{R}_j$ in (3.3), three eigenvalues of $\mathbf{R}_j$ are one and $\cos\psi_j \pm i \sin\psi_j$ where $i$ is the unit imaginary number. They are also eigenvalues of $\hat{\mathbf{R}}_j$ due to the eigenvalue invariance of similar matrices. Because the sum of eigenvalues is equal to the trace of the matrix, an equation related to $\psi_j$ can be derived as

$$\operatorname{trace} \hat{\mathbf{R}}_j = 2\cos\psi_j + 1. \tag{3.10}$$

From (3.10), only the magnitude $|\psi_j|$ can be obtained. To determine its sign, $\mathbf{t}_j$ is employed. Due to the Ackermann geometry, $\hat{\mathbf{t}}_j$ is locally constrained to the x-z plane of $\{A\}$.

When the steering motion is larger than the roll and pitch motion, it can be considered that a vector rotation $\hat{\mathbf{t}}'_j := \hat{\mathbf{R}}_j\hat{\mathbf{t}}_j \in \mathbb{R}^3$ is mainly governed by the steering motion. Then, from the directional difference between $\hat{\mathbf{t}}'_j$ and $\hat{\mathbf{t}}_j$, the direction of rotation of the vehicle can be computed.

In sum, the initial guess of $\psi_j$ can be calculated as a closed form with a 3-D cross product operator × as

$$\psi_j = \operatorname{sign}\left(\hat{\mathbf{t}}_j \times \hat{\mathbf{t}}'_j\right) \cdot \left|\arccos\left(\frac{\operatorname{trace}\hat{\mathbf{R}}_j - 1}{2}\right)\right|. \tag{3.11}$$

### Linear solution of S in a quaternion representation

Using the initial guesses $\psi_j$, $\mathbf{Q}$ can be estimated by solving the least squares problem in quaternion space. Let $\hat{\mathbf{q}}_j$, $\mathbf{q}_j \in \mathbb{R}^4$ be unit quaternions of $\hat{\mathbf{R}}_j$ and $\mathbf{R}_j$, respectively. This research work follows the Hamilton quaternion convention with right-handed algebra. the A pure quaternion of the vector $\mathbf{v} \in \mathbb{R}^3$ is defined with zero at the first element as $\breve{\mathbf{v}} := \begin{bmatrix} 0, & \mathbf{v}^\top \end{bmatrix}^\top \in \mathbb{R}^4$. Then, (3.5) can be rewritten as

$$\begin{aligned} \mathbf{Q}\hat{\mathbf{R}}_j = \mathbf{R}_j\mathbf{Q} &\rightarrow \mathbf{q}_s \otimes \hat{\mathbf{q}}_j = \mathbf{q}_j \otimes \mathbf{q}_s \\ \mathbf{Q}\hat{\mathbf{t}}_j = \mathbf{t}_j &\rightarrow \mathbf{q}_s \otimes \breve{\hat{\mathbf{t}}}_j \otimes \mathbf{q}_s^* = \breve{\mathbf{t}}_j, \end{aligned} \tag{3.12}$$

where ⊗ means the quaternion product operator, and $\mathbf{q}^* \in \mathbb{R}^4$ denotes conjugate of a quaternion $\mathbf{q}$. Because the MVO can only provide up-to-scale translation motion, unit vectors $\hat{\mathbf{u}}_j$ and $\mathbf{u}_j$ are used corresponding to $\hat{\mathbf{t}}_j$ and $\mathbf{t}_j$, respectively.

The quaternion equations in (3.12) can be transformed into matrix forms,

$$\begin{aligned} \Omega_r\left(\hat{\mathbf{q}}_i\right)\mathbf{q}_s &= \Omega_l\left(\mathbf{q}_j\right)\mathbf{q}_s \\ \Omega_r\left(\breve{\hat{\mathbf{u}}}_i\right)\mathbf{q}_s &= \Omega_l\left(\breve{\mathbf{u}}_i\right)\mathbf{q}_s, \end{aligned} \tag{3.13}$$

where matrix forms of left and right quaternion products $\Omega_l(\mathbf{q})$, $\Omega_r(\mathbf{q}) : \mathbb{R}^4 \mapsto \mathbb{R}^{4\times4}$ are denoted as

$$\Omega_l(\mathbf{q}) = \begin{bmatrix} q_0 & -\mathbf{n}^\top \\ \mathbf{n} & q_0\mathbf{I}_3 - [\mathbf{n}]_\times \end{bmatrix}, \Omega_r(\mathbf{q}) = \begin{bmatrix} q_0 & -\mathbf{n}^\top \\ \mathbf{n} & q_0\mathbf{I}_3 + [\mathbf{n}]_\times \end{bmatrix} \tag{3.14}$$

where $\mathbf{q} := \left[q_0, \mathbf{n}^\top\right]^\top$ with a scalar $q_0$ and $\mathbf{n} \in \mathbb{R}^3$, and $[\mathbf{n}]_\times \in \mathbb{R}^{3\times3}$ is a matrix satisfying $[\mathbf{n}]_\times\mathbf{v} = \mathbf{n} \times \mathbf{v}$ with $\mathbf{v} \in \mathbb{R}^3$.

Note that, in this initialization step, $L = 0$ is temporally assumed to neglect unknown values $\rho_j$ of $\mathbf{t}_j$. Then, the simplified form of $\mathbf{u}_j$ becomes a function of only $\psi_j$

$$\mathbf{u}_j = \frac{\mathbf{t}_j}{\|\mathbf{t}_j\|_2} = \begin{bmatrix} \rho_j \mathrm{s}\gamma_j + L\mathrm{s}\psi_j \\ 0 \\ \rho_j \mathrm{c}\gamma_j - L + L\mathrm{c}\psi_j \end{bmatrix} / \|\mathbf{t}_j\|_2 \approx \begin{bmatrix} \mathrm{s}\psi_j \\ 0 \\ \mathrm{c}\psi_j \end{bmatrix}. \tag{3.15}$$

$\rho_j$ values will be re-considered in a refinement step in the following subsection.

By concatenating $N$ equations, a least squares problem to estimate $\mathbf{q}_s$ with a matrix $\mathbf{M} \in \mathbb{R}^{8N \times 4}$ can be formulated as

$$\mathbf{M}\mathbf{q}_s = \begin{bmatrix} \Omega_r\left(\hat{\mathbf{q}}_1\right) - \Omega_l\left(\mathbf{q}_1\right) \\ \Omega_r\left(\breve{\hat{\mathbf{u}}}_1\right) - \Omega_l\left(\breve{\mathbf{u}}_1\right) \\ \vdots \\ \Omega_r\left(\hat{\mathbf{q}}_N\right) - \Omega_l\left(\mathbf{q}_N\right) \\ \Omega_r\left(\breve{\hat{\mathbf{u}}}_N\right) - \Omega_l\left(\breve{\mathbf{u}}_N\right) \end{bmatrix} \mathbf{q}_s = \mathbf{0}_{8N}. \tag{3.16}$$

A solution $\mathbf{q}_s$ can be computed by the right nullspace of $\mathbf{M}$. Using the singular value decomposition to $\mathbf{M}$, the solution can be obtained as the right singular vector corresponding to the smallest singular value.

To distinguish the smallest one out of four singular values, the ratio of the two smallest singular values is checked. If the second one is more than twice the smallest one, the solution can be considered reliable. In Fig. 3.4, it is found that the second one becomes sufficiently larger than the smallest one at the first turning motion. Then, the estimated $\mathbf{q}_s$ converges to true values with the dashed line.

### 3.3.3 Full Refinement of the Initial Guesses

In the previous linear step, $L = 0$ is assumed for simple derivations. In this step, the nonzero $L$ is re-considered to incorporate effects of $\mathrm{s}\psi_j$ and $\mathrm{c}\psi_j - 1$ terms of $\mathbf{t}_j$. Without

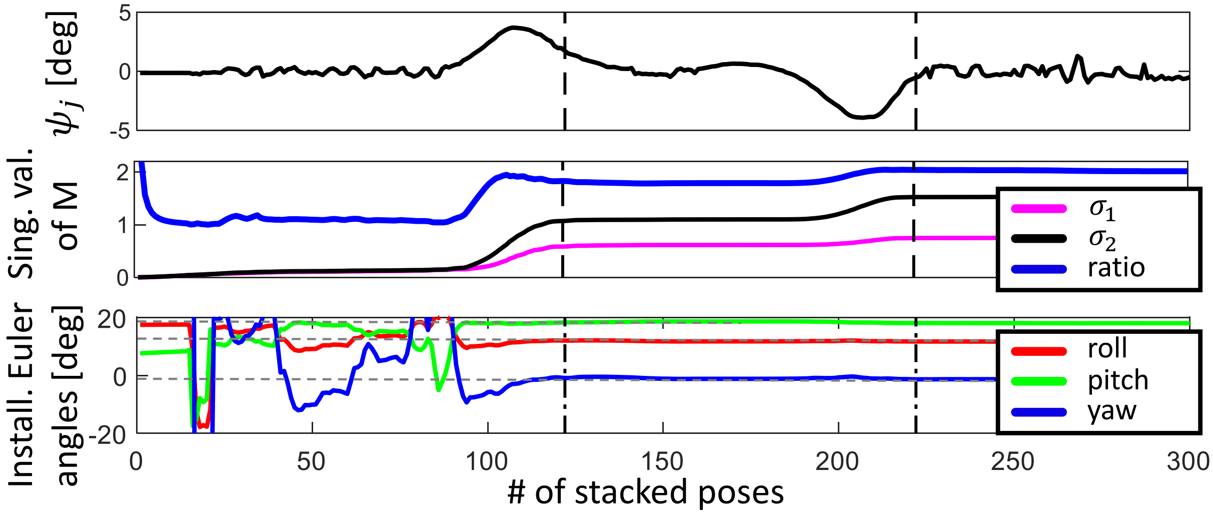


Figure 3.4: **Singular value history of the linear initialization of $\mathbf{q}_s$** In the first graph, turning angles are obtained by the ground truth poses to the first 500 frames of 00 dataset [2]. $\sigma_1$ and $\sigma_2$ are the two smallest singular values of $\mathbf{M}$, respectively, and their histories are drawn according to the number of stacked poses in the middle graph. The third graph exhibits histories of the estimated Euler angles of $\mathbf{Q}$. Two vertical lines denote ends of each turning motion, and horizontal dashed lines are true values of Euler angles of $\mathbf{Q}$. The author intentionally rotates the camera pose with Euler angles of $\{12, 19, -1\}$ degrees to simulate an arbitrary pose $\mathbf{Q}$.

loss of generality, $L = 1$ is used. Because one cannot obtain the scale of $\mathbf{t}_j$ from the MVO, the author uses the normalized translation vector $\mathbf{t}_j / \|\mathbf{t}_j\|_2$ in the full refinement problem. Then, the modified problem of (3.9) is written as

$$\operatorname{argmin} \sum_{j=1}^{N} w_H \begin{pmatrix} \|\hat{\mathbf{R}}_j - \mathbf{Q}^\top \mathbf{R}_j \mathbf{Q}\|_F^2 \\ + \| \left(\mathbf{Q}\hat{\mathbf{u}}_j\right)^\top \mathbf{t}_j / \|\mathbf{t}_j\|_2 - 1\|_2^2 \end{pmatrix} \tag{3.17}$$
$$\text{subject to } \|\mathbf{q}_s\|_2 = 1.$$

As seen in (3.17), the original translation term in (3.9) is modified into the difference of unit directions of two translation representations to delete the unknown magnitude of the estimated monocular translation motion.

The real-world vehicle motion could slightly deviate from the planar motion model. To suppress the bad effects of the off-planar motions on the optimization process, the Huber norm $w_H(r)$ is invited with the threshold value $r_{th} \in \mathbb{R}^+$ as

$$w_H(r) = \begin{cases} r_{th}/|r| & \text{if} \quad |r| > r_{th} \\ 1 & \quad\quad \text{otherwise} \end{cases}, \tag{3.18}$$

where $r_{th}$ is set to 60 % value of the residuals, and recalculated for each optimization step.

The above nonlinear optimization problem can be efficiently solved by a nonlinear programming solver, CasADi [69]. Note that the resulting scale estimations $\rho_j$ are proportional to the $L$. If the exact metric $L$ can be known in advance, the absolute value of $\rho_j$ can be estimated. Reversely, the metric $L$ can be recovered using metric motion measurements from additional sensors, such as wheel odometer and global positioning system (GPS). Anyway, any choice of real positive $L$ does not affect estimating $\mathbf{Q}$.

## 3.4 Absolute Scale Recovery between Turning Regions

In this section, the author introduces a method to observe the absolute metric scale of the MVO motion, and a strategy to detect turning frame regions that can provide absolute scale observations stably. Then, an absolute scale recovery (ASR) method is proposed to scale the translation motion and 3-D points between turns by using the observed absolute scale of the turn regions.

### 3.4.1 Observing Absolute Scale via Kinematic Geometry

This subsection details how to observe the absolute scale $s_{\alpha,j}$ of the monocular camera translation motion $\mathbf{t}_j$ at the vehicle turning. When the vehicle turns to an angle of $\psi_j$, one can draw two triangles by joining the origins of vehicle body frames and camera frames as depicted in Fig. 3.5(a). For the red isosceles triangle $\triangle ACA'$, lengths of $\overline{AC}$ and $\overline{A'C}$ can be calculated as

$$\overline{AC} = \overline{A'C} = \frac{\rho_j}{2\,\mathrm{c}\gamma_j}. \tag{3.19}$$

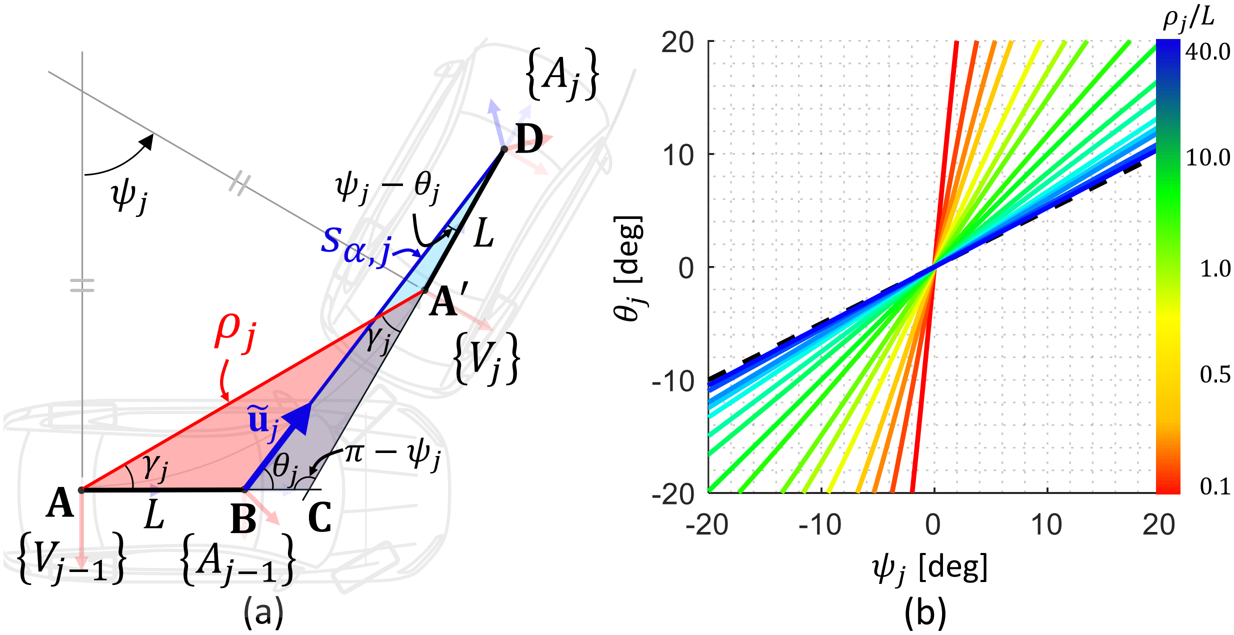


Figure 3.5: **Two triangles formed by the turn of the vehicle and relationship between $\psi$ and $\theta$** (a) $\psi_j$ is the turn angle of the vehicle, and $\theta_j$ is the subtended angle between the z-axis of $\{A_{j\text{-}1}\}$ and the rotated unit translation vector $\tilde{\mathbf{u}}_j$. $\rho_j$ is the distance between centers of $\{V_{j\text{-}1}\}$ and $\{V_j\}$, and $s_{\alpha,j}$ is the scale of the monocular translation motion which is the objective to estimate. (b) The relationship between $\psi$ and $\theta$ is plotted with respect to the value of $\rho_j/L$.

By using $\overline{AC}$ and $\overline{A'C}$, each side of the blue triangle $\triangle BCD$ can be calculated with $\overline{AB} = \overline{A'D} = L$ as

$$\overline{BC} = \frac{\rho_j}{2\,\mathrm{c}\gamma_j} - L,\ \overline{CD} = \frac{\rho_j}{2\,\mathrm{c}\gamma_j} + L. \tag{3.20}$$

As seen in Fig. 3.5(a), the objective $s_{\alpha,j}$ is a side of the blue triangle. If angles $\psi_j$, $\theta_j$, and $\gamma_j$ are known, one can calculate $s_{\alpha,j}$ by utilizing the sine rule on the blue triangle. Because the initial value of $\psi_j$ can be known by (3.11) and $\gamma_j = \psi_j/2$, the unknown value of the angle $\theta_j$ can be computed, which is the subtended angle between the z-axis of $\{A_{j\text{-}1}\}$ and a unit vector $\tilde{\mathbf{u}}_j := \mathbf{Q}\hat{\mathbf{u}}_j \in \mathbb{R}^3$ rotated to the auxiliary frame. By defining $\mathbf{k}_V \in \mathbb{R}^3$ as the unit vector of the z-axis of $\{V\}$, $\theta_j$ is calculated as

$$\theta_j = \arctan\left\{(\mathbf{k}_V \times \tilde{\mathbf{u}}_j)/\left(\mathbf{k}_V{}^{\top}\tilde{\mathbf{u}}_j\right)\right\}. \tag{3.21}$$

Applying the sine rule on the blue triangle, an equality is finally obtained as,

$$\frac{\frac{\rho_j}{2\,\mathrm{c}\gamma_j} - L}{\mathrm{s}(\psi_j - \theta_j)} = \frac{\frac{\rho_j}{2\,\mathrm{c}\gamma_j} + L}{\mathrm{s}\theta_j} = \frac{s_{\alpha,j}}{\mathrm{s}\psi_j}. \tag{3.22}$$

From the first equality in (3.22), the temporal distance $\rho_j$ of the vehicle is expressed as a function of $\psi_j$ and $\theta_j$ up to $L$,

$$\frac{\rho_j}{L} = 2\,\mathrm{c}\gamma_j\,\frac{\mathrm{s}\theta_j + \mathrm{s}(\psi_j - \theta_j)}{\mathrm{s}\theta_j - \mathrm{s}(\psi_j - \theta_j)}. \tag{3.23}$$

Then, the scale observer can be derived by substituting (3.23) to the second equality of (3.22) as

$$\frac{s_{\alpha,j}}{L} = \frac{2\,\mathrm{s}\psi_j}{\mathrm{s}\theta_j - \mathrm{s}(\psi_j - \theta_j)}. \tag{3.24}$$

In Fig. 3.5(b), the graph of $\psi_j$ and $\theta_j$ with respect to various $\rho_j/L$ settings is depicted. $\theta_j$ can be derived from (3.23) as

$$\theta_j = \arctan\left(\frac{\rho_j \mathrm{t}\gamma_j + 2L\,\mathrm{s}\gamma_j}{\rho_j - 2L\,\mathrm{s}\gamma_j\,\mathrm{t}\gamma_j}\right), \tag{3.25}$$

where $\theta_j$ is a function of $\psi_j$ and $\rho_j$, which allows us to treat $s_{\alpha,j}$ as a function of $\psi_j$ and $\theta_j$.

As seen in the figure, $\theta_j$ is approximately proportional to $\psi_j$ for all $\rho_j/L$. It is found that the ratio $\theta_j/\psi_j$ asymptotically converges to 0.5 when $\rho_j$ goes to infinite, which guarantees a nonzero positive denominator of (3.24) by assuming $|\psi_j| = |2\gamma_j| < \pi$. Because general vehicles cannot steer over 90 degrees in a short period like the camera image acquisition interval, the turning angle assumption is valid in most cases.

The scale observer in (3.24) becomes singular when $\psi_j$ goes to zero. To discuss this problem, the section 3.5.1 will investigate the relationship among $\psi_j$, $\theta_j$, and $s_{\alpha,j}$, and analyze which condition is desirable to stably observe the scale by a sensitivity analysis on the scale observer.

### 3.4.2 Detecting Turning Regions

As denoted in the previous subsection, the scale observer (3.24) becomes singular for small turning angle $\psi_j$. To obtain the reliable observations, keyframes with sufficiently large turning motions are detected.

Let $\mathcal{F}$ be an index set of all keyframes between turning regions. $\mathcal{F}$ consists of two subsets, $\mathcal{F}_t$ and $\mathcal{F}_u$, which are index sets of keyframes on turning and non-turning regions, respectively. The author additionally separates $\mathcal{F}_t$ into two index sets of previous and current turning regions, $\mathcal{F}_{tp}$ and $\mathcal{F}_{tc}$, respectively. Each index set is depicted as a shaded region with a dashed boundary in Fig. 3.6.

Once $|\psi_j|$ becomes larger than a threshold angle $\psi_{th}$, the $j$-th keyframe is regarded as a turning candidate frame, and it is counted how many candidates follow sequentially. If the number of the candidates exceeds a threshold count value $n_{th}$, a new turning region $\mathcal{F}_{tc}$ can be found from the $j$-th keyframe to a keyframe whose next keyframe is no longer the candidate frame. If not, all the candidates from the $j$-th keyframe are passed to the non-turning region index set $\mathcal{F}_u$. This procedure is written in Algorithm 2.

**Algorithm 2** Detecting a New Turning Region

```
$i_{op}$: an operating indicator. Default is True.
$n$: a counter value. Default is zero.
for each incoming frames, current $j$ -th frame do
    Do monocular VO
    $|\psi_j| \leftarrow$ a steering angle calculated by (3.11)
    if $|\psi_j| \geq \psi_{th}$ then
        if $!i_{op}$ then
            $i_{op} \leftarrow$ True
        end if
        $\mathcal{F}_{tc} \leftarrow \mathcal{F}_{tc} \cup j$; $++n$;
    else
        $i_{op} \leftarrow$ False; $n \leftarrow 0$;
        if $n \geq n_{th}$ then
            $\mathcal{F}_{tp} \leftarrow \mathcal{F}_{tc}$; $\mathcal{F}_{tc} \leftarrow \varnothing$;
        else
            $\mathcal{F}_{u} \leftarrow \mathcal{F}_{u} \cup \mathcal{F}_{tc}$; $\mathcal{F}_{tc} \leftarrow \varnothing$;
        end if
    end if
end for
```

### 3.4.3 Recovering Unknown Scale by Nonlinear Programming with Equality Constraints

In this subsection, the author introduces the ASR module. Using the observed scale values on the turning keyframes $\mathcal{F}_t$, unknown scale values of the monocular translation motion and 3-D landmark points between the turning regions $\mathcal{F}_u$ are recovered.

Fig. 3.6 illustrates a factor graph of the $i$-th landmark and its related keyframes. The landmark is associated with the keyframes by 2-D pixel tracks $\mathbf{p}_{ij} \in \mathbb{R}^2$. The pixel reprojection error $\mathbf{r}_{ij}$ induced by $\mathbf{X}_i$ and $\{C_j\}$ is written as

$$\mathbf{r}_{ij} := \pi_j \left(X_i\right) - \mathbf{p}_{ij} \in \mathbb{R}^2. \tag{3.26}$$

By aggregating all error vectors generated by $N$ keyframes and $M$ landmarks, the

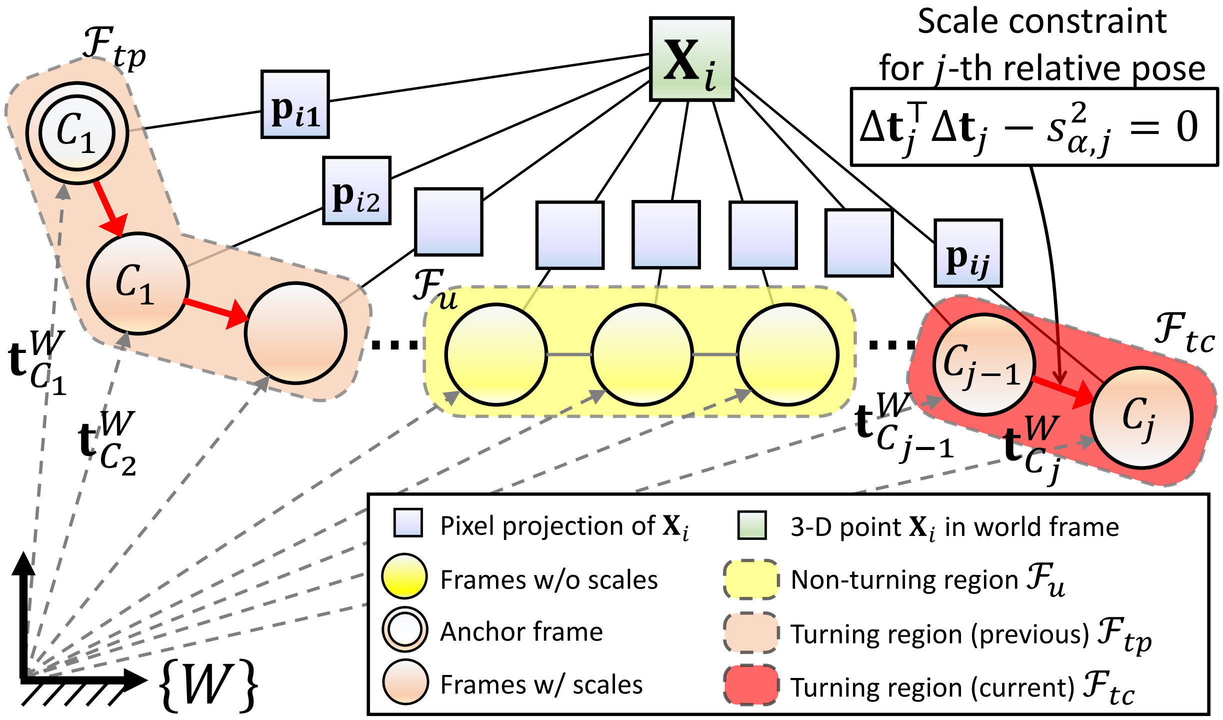


Figure 3.6: **Factor graph of a landmark and related keyframes for the absolute scale recovery** An $i$-th landmark is connected to turning and non-turning keyframes by 2-D pixel tracks. In this illustration, 1's index rule is used. The red arrow means a constrained relative translation motion in the turning regions. All translation motions of the keyframes are represented with respect to the world frame.

residual vector $\mathbf{r}$ is defined as

$$\mathbf{r} := \left[ o_{11}\mathbf{r}_{11}{}^{\top}, \cdots, o_{NM}\mathbf{r}_{NM}{}^{\top} \right]^{\top} \in \mathbb{R}^{2MN}, \tag{3.27}$$

where an indicator $o_{ij}$ becomes true if the $i$-th point is seen in the $j$-th keyframe, otherwise false.

Let $\zeta$ be the parameter vector to be scaled as

$$\zeta := \left[ \mathbf{t}_2^{W\top}, \cdots, \mathbf{t}_N^{W\top}, \mathbf{X}_1{}^{\top}, \cdots, \mathbf{X}_M{}^{\top} \right]^{\top} \in \mathbb{R}^{P}, \tag{3.28}$$

where $P := 3\left(N-1\right)+3M$ and the first keyframe $\{C_1\}$ is fixed to avoid the gauge freedom. Then, one can formulate a reprojection error minimization problem with respect to $\zeta$ as

$$\underset{\zeta}{\operatorname{argmin}}\ \mathbf{r}\left(\zeta\right)^{\top}\mathbf{r}\left(\zeta\right), \tag{3.29}$$

where $\mathbf{r}\left(\zeta\right) \in \mathbb{R}^{2MN}$ is the residual vector as a function of $\zeta$.

The optimization problem in (3.29) is a popular nonlinear programming problem in computer vision, called bundle adjustment (BA) [15]. Unlike the original BA, the observed scale values are additionally incorporated into the problem to recover the unobserved scale between turns.

Conceptually, the scale of the turning keyframes can be propagated to the associated non-turning keyframes through the 2-D pixel tracks. Like the red arrows depicted in Fig. 3.6, the observed scale on $\mathcal{F}_t$ can be used to constrain the relative translation motion $\Delta\mathbf{t}_j := \mathbf{t}_{C_j}^{W} - \mathbf{t}_{C_{j-1}}^{W} \in \mathbb{R}^3$. If the cardinality of $\mathcal{F}_t$ is $K$ and the $k$-th element of $\mathcal{F}_t$ is $\mathcal{F}_t\left(k\right)$, the $k$-th scale constraint can be written as an equality constraint

$$g_k\left(\zeta, \mathbf{s}_\alpha\right) = \Delta\mathbf{t}_{\mathcal{F}_t(k)}{}^{\top}\Delta\mathbf{t}_{\mathcal{F}_t(k)} - s_{\alpha,\mathcal{F}_t(k)}^2 = 0, \tag{3.30}$$

where $\mathbf{s}_\alpha := \left[ s_{\alpha,\mathcal{F}_t(1)}, \cdots, s_{\alpha,\mathcal{F}_t(K)} \right]^{\top} \in \mathbb{R}^K$.

Defining the Lagrangian $L(\zeta, \boldsymbol{\lambda}) := \mathbf{r}(\zeta)^{\top}\mathbf{r}(\zeta) + \boldsymbol{\lambda}^{\top}\mathbf{g}(\zeta, \mathbf{s}_{\alpha}) \in \mathbb{R}$ with the Lagrange multiplier vector $\boldsymbol{\lambda} \in \mathbb{R}^{K}$, a minimization problem is finally set up as

$$\underset{\zeta, \boldsymbol{\lambda}}{\operatorname{argmin}}\ L(\zeta, \boldsymbol{\lambda}) \text{ subject to } \mathbf{g}(\zeta, \mathbf{s}_{\alpha}) = \mathbf{0}_{K}, \tag{3.31}$$

where an equality constraint vector $\mathbf{g}$ is defined as

$$\mathbf{g}(\zeta, \mathbf{s}_{\alpha}) := \left[g_{1}(\zeta, \mathbf{s}_{\alpha}), \cdots, g_{K}(\zeta, \mathbf{s}_{\alpha})\right]^{\top} : \mathbb{R}^{P} \mapsto \mathbb{R}^{K}. \tag{3.32}$$

The above nonlinear programming with equality constraints can be solved by sequential quadratic programming (SQP) [70]. Whenever a new turning region is detected, the ASR module is operated, and this procedure is repeated for overall image sequences.

## 3.5 Performance Analysis

This section analyzes the proposed three modules: the scale observer, the camera-vehicle extrinsic calibration, and the ASR module.

### 3.5.1 Noise Sensitivity Analysis of the Scale Observer

First, the author performs an in-depth analysis of the scale observer. Noise in $\psi_j$ and $\theta_j$ estimation is inevitable due to imperfect camera motion estimation and off-planar vehicle vibration. In Section 3.4.1, (3.24) is ill-defined near $\psi_j = \theta_j = 0$, which implies high noise-sensitivity around the zero. To address this problem, the author analyzes the noise sensitivity of the scale observer $s_{\alpha,j}$ with respect to $\psi_j$ and $\theta_j$ with various $\rho_j/L$ settings. Without loss of generality, the normalized scale $s'_{\alpha,j} := s_{\alpha,j}/L$ is considered during this analysis.

$s'_j$ is differentiated to two parameters $\psi_j$, $\theta_j$, and resulting sensitivity equations are as belows:

$$\frac{\partial s'_{\alpha,j}}{\partial \psi_j} = \frac{-\mathrm{s}\theta_j}{\mathrm{c}(\psi_j - 2\theta_j) - 1}, \quad \frac{\partial s'_{\alpha,j}}{\partial \theta_j} = \frac{2\left(\mathrm{s}(\psi_j - \theta_j) + \mathrm{s}\theta_j\right)}{\mathrm{c}(\psi_j - 2\theta_j) - 1}. \tag{3.33}$$

Using the above two derivatives, the author draws multiple graphs by changing $\rho_j/L$ in Figs. 3.7(a)–(b). According to the graphs, the scale observer becomes less sensitive to noise in the angle estimation of $\psi_j$ and $\theta_j$ when the turning angle $\psi_j$ is large. Both sensitivities show similar tendency because $\theta_j$ is governed by $\psi_j$ as Fig. 3.5(b).

When increasing the relative vehicle speed $\rho_j/L$, both noise sensitivities also increase as seen in Fig. 3.7(a). From these tendencies, one can conclude that more stable scale observations can be obtained in apparently large turning motions at low driving speeds. As seen in Fig. 3.7(b), the scale observer is slightly more sensitive to error in $\psi_j$ than $\theta_j$. In other words, the accuracy of the turning angle estimation is more crucial for accurate scale observation than the translation vector estimation. Fortunately, it is found that MVO yields sufficiently accurate turning angle $\psi_j$ in average error less than 0.1 degrees in the

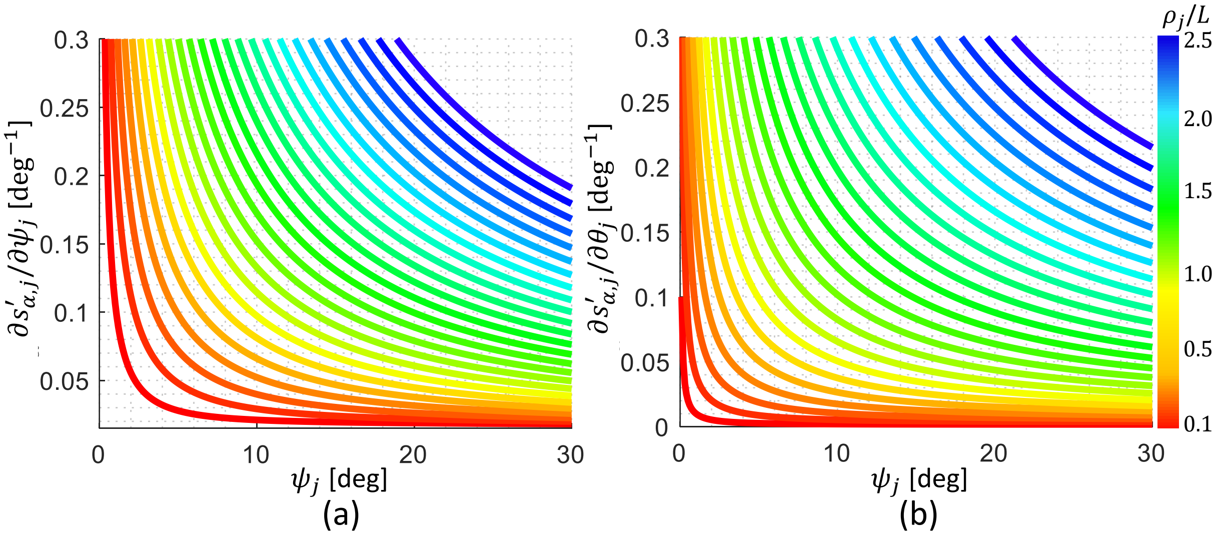


Figure 3.7: **Noise sensitivities of the scale observer** (a) Noise sensitivity with respect to $\psi_j$, (b) noise sensitivity with respect to $\theta_j$. The curves are color-coded according to $\rho_j/L$.

KITTI datasets [2]. For $\psi_j = 5$ degrees and $\rho_j = 0.4$ m with $L = 1$ m, the 0.1 degree error corresponds to about 0.02 m scale error which is only 1/20 of the scale observation error.

As mentioned before, both noise sensitivities are governed by $\rho_j/L$. Without changing the metric distance $\rho_j$ between $\{V_{j\text{-}1}\}$ and $\{V_j\}$, the term $\rho_j/L$ can be decreased by increasing $L$. In general, the camera on the vehicle is mounted around the windshield to look forward, and such a setup can guarantee sufficiently large $L > 1$ m, which implies that the proposed method is suitable for general automobile environments.

In Fig. 3.8(a), the author plots the history of $\psi_j$ and $\rho_j/L$ estimated by the proposed MVO from the author-collected parking-lot datasets which will be detailed in Section 4.2.2. As seen in the graph, during turns, $\rho_j/L$ is mostly in the range $[0.2, 0.4]$, and the turning angle is over 3 degrees in average. Note that $\rho_j/L \in [0.2, 0.4]$ corresponds to the vehicle speed 20–30 km/h (12–19 mi/h) with $L = 1$ m for 10 Hz image acquisition frequency. Based on these motion characteristics of the parking-lot datasets, the noise tolerance of the scale observer is evaluated. two situations are considered: $\Delta\psi_j = \Delta\theta_j = 0.05$ degrees and $\Delta\psi_j = \Delta\theta_j = 0.1$ degrees where $\Delta\psi_j, \Delta\theta_j \in \mathbb{R}$ denote absolute values of the estimation

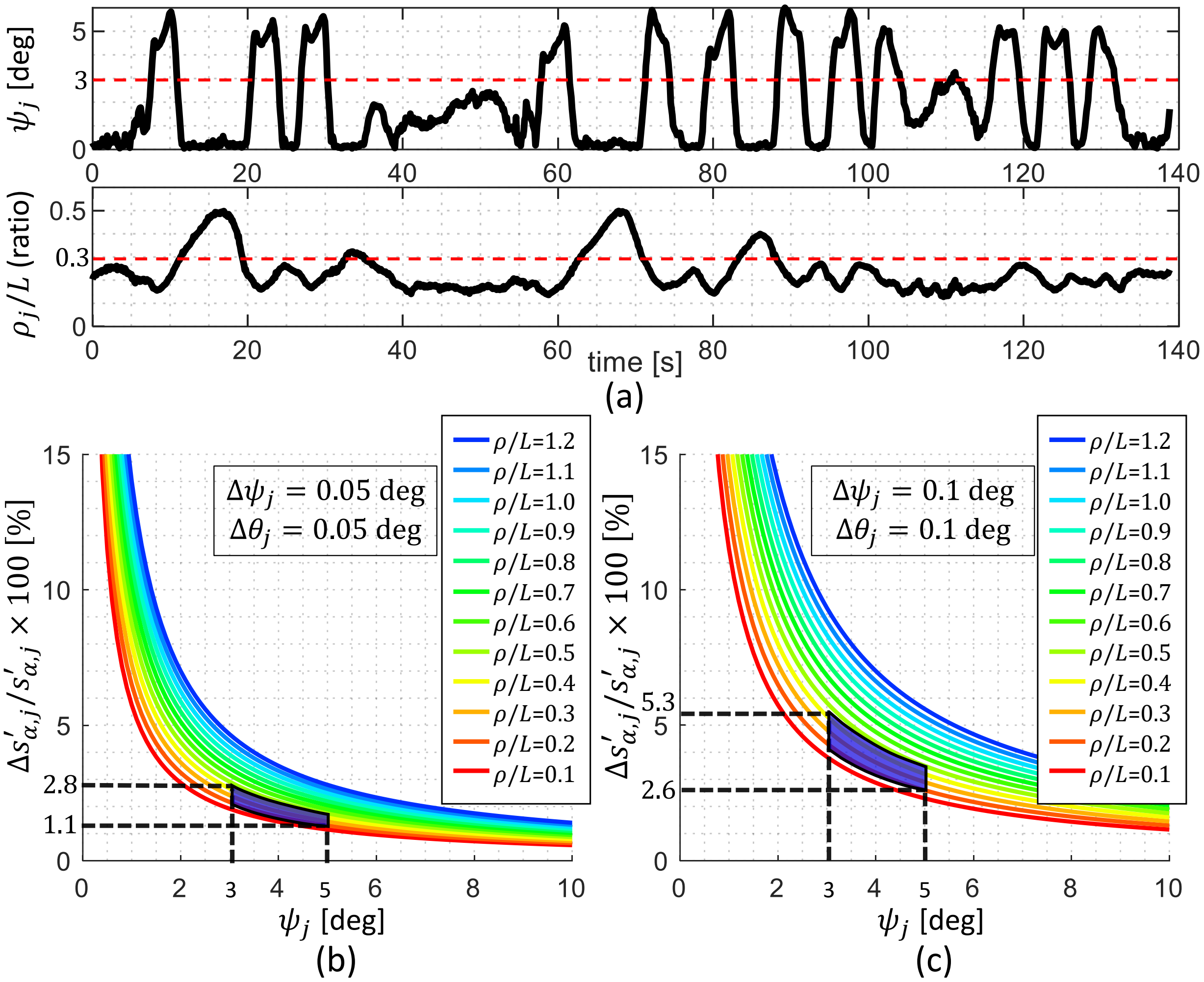


Figure 3.8: **Turn angle $\psi_j$, relative distance $\rho_j/L$, and the error over the scale on the author-collected parking lots driving datasets** (a) This graph shows the time history of $\psi_j$ and $\rho_j/L$ on `bldg_39` of the author-collected datasets to be detailed in Section 4.2.2. The red dashed line means the average value during turns. (b) The scale estimation error ratio when the angle estimation error is $\Delta\psi_j = \Delta\theta_j = 0.05$ deg. (c) The scale estimation error ratio when the angle estimation error is $\Delta\psi_j = \Delta\theta_j = 0.1$ deg. The dark blue region corresponds to $\psi_j \in [3,5]$ degrees and $\rho_j/L \in [0.2, 0.4]$.

error on $\psi_j$ and $\theta_j$, respectively. The error values are determined based on the average 0.1 degrees rotation error in the KITTI datasets mentioned before. The error of the scale estimation, $\Delta s'_{\alpha,j} \in \mathbb{R}$, are calculated with respect to $\Delta\psi_j$ and $\Delta\theta_j$ as

$$\Delta s'_{\alpha,j}(\Delta\psi_j, \Delta\theta_j) \approx \left|\frac{\partial s'_{\alpha,j}}{\partial \psi_j}\right| \Delta\psi_j + \left|\frac{\partial s'_{\alpha,j}}{\partial \theta_j}\right| \Delta\theta_j. \tag{3.34}$$

The author computes the percentage of the error over the normalized scale, $\Delta s'_{\alpha,j}/s'_{\alpha,j} \times 100$ [%]. In Figs. 3.8(b)–(c), the dark blue region is the region of interest of this chapter $\psi_j \in [3, 5]$ degrees and $\rho_j/L \in [0.2, 0.4]$. As seen in Figs. 3.8(b)–(c), the error percentage in the real-world situation such as parking lots can be quite small, about 2.5 % in average and 5.5 % in the worst case.

From this, the proposed method will be effective for common indoor driving situations. In Section 4.2.2, the author will verify the effectiveness of the proposed method on the author-collected driving datasets obtained in multi-floor underground parking lots.

### 3.5.2 Analysis via Implementations on Synthetic Data

The performance of the camera-vehicle extrinsic calibration and the ASR module are extensively evaluated through Monte-Carlo simulation on a synthetic driving dataset. The shape of the synthetic dataset is depicted in Fig. 3.9. This dataset has a 1.2 km trajectory with several 90-degree turning motions, and about $4,000$ points scattered along the trajectory.

The data association of 2-D pixel tracks and keyframes is established by projecting the 3-D points to each camera frame with a field of view limit of 100 m. For each Monte-Carlo simulation, the author changes the distribution of the 3-D points and their 2-D pixel projection error. For realistic simulation, several camera rotation pose error settings are considered with different noise levels.

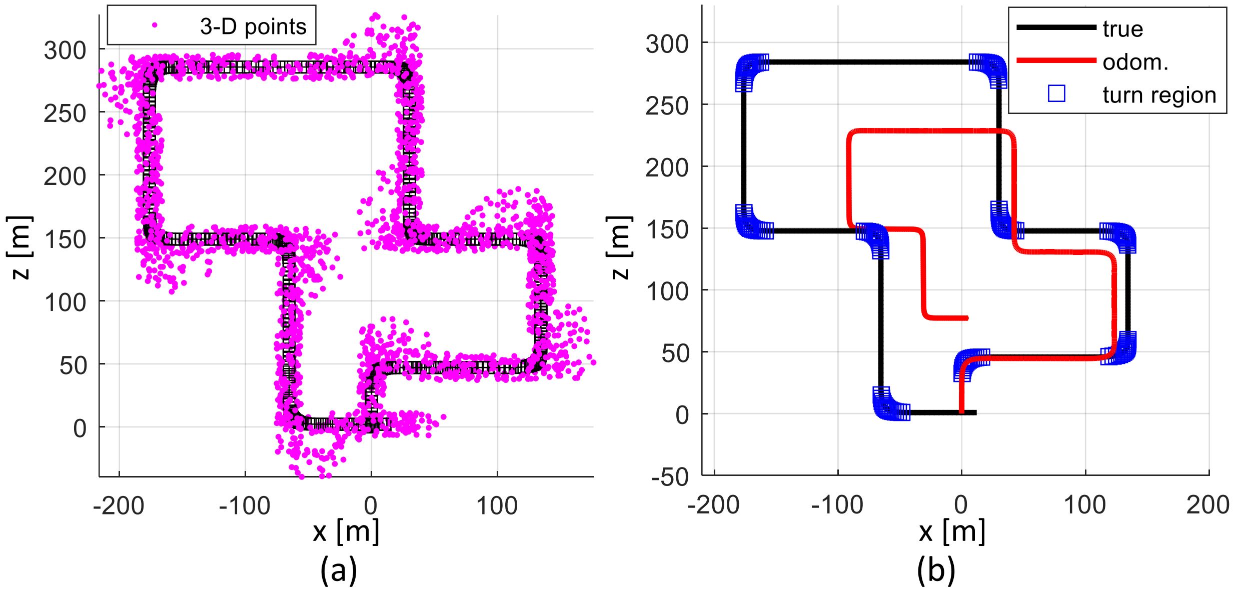


Figure 3.9: **Trajectory and 3-D points of the synthetic dataset, and the turning region detection results** (a) The trajectory is 1.2 km long with 3, 500 data points and nine turning spots. (b) The author simulates the noisy and drifted MVO estimation by augmenting rotation and translation error to the true in black trajectory. The blue boxes are the detected turning regions by the proposed method.

### Camera-vehicle extrinsic calibration results

The camera intrinsic parameter is set the same as the sensor suite of the first data sequence `00` of the KITTI odometry datasets. An artificial monocular camera is considered with $L = 1.0$ m displacement from the rear axle, and the camera installation pose $\mathbf{Q}$ is set by $\{5,\ 15,\ -10\}$ degrees z-y-x Euler angles.

The author evaluates the accuracy of the proposed camera-vehicle pose calibration method by changing noise in the camera rotation motion estimation with 0.05, 0.2, 0.5, and 1.0-degree random noise for each frame. For each noise level, total 100 simulations are repeated for meaningful statistics.

The simulation results are plotted in Figs. 3.10(a)–(b). Two settings of the calibration method are considered: (a) linear initialization only and (b) full refinement. As seen in Fig. 3.10(a), under linear initialization, the estimation accuracy rapidly degrades when the

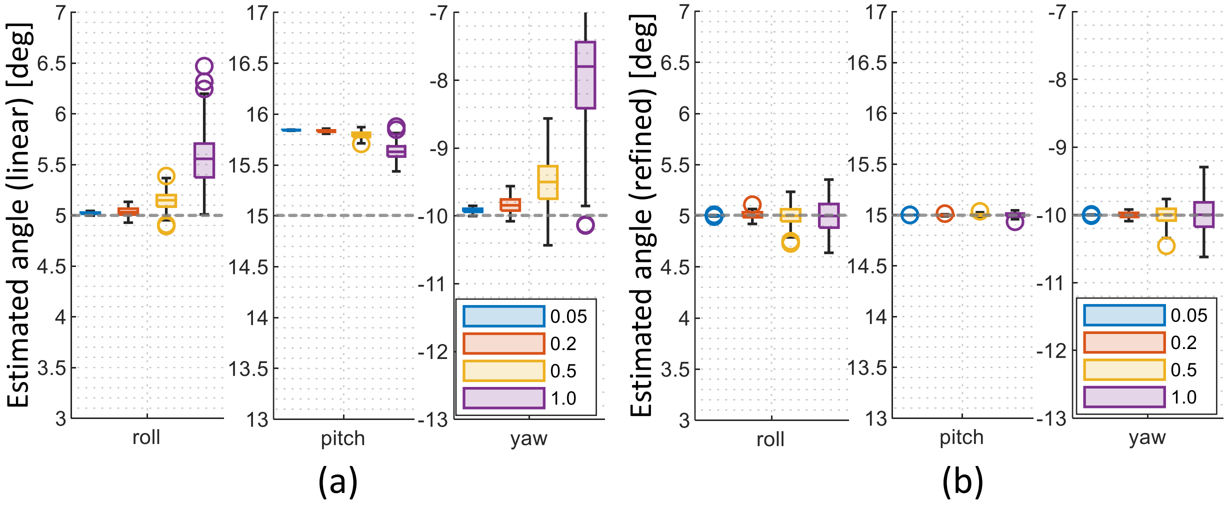


Figure 3.10: **Results of the camera-vehicle extrinsic calibration on the synthetic dataset** (a) results of the linear method only, (b) results after the full refinement. The boxplots are colored according to each noise level. The horizontal lines in each boxplot denote mean values. The gray dashed lines denote each truth value, and each box means 1-sigma region. The black vertical lines mean ranges of the resulting values.

noise level increases. Especially, the pitch angle estimation corresponding to the rotation around the y-axis of $\{A\}$ shows large offset errors for all noise conditions. It is found that the offset error is caused by the deviated direction vector $\mathbf{u}_j$ in (3.15) by assuming $L = 0$. If one compensates the true $L$ and $\rho_j$ values in the linear initialization step, no offset error occurs on the pitch angle.

Contrary to the linear-only setting, the full refinement module yields unbiased estimation regardless of the noise level because $\rho_j$ is explicitly optimized with the non-zero $L$ in the refinement step. Furthermore, thanks to the noise suppression effect of the Huber norm, the standard deviation of the estimated Euler angles is decreased to 0.5 degrees for the 1-degree noise condition. As mentioned in Fig. 3.4, only one turning motion is sufficient to excite the extrinsic calibration module. Considering all of these, the proposed calibration method can stably estimate the accurate camera-vehicle extrinsic pose with the noisy data from a monocular camera with one turning region only.

### Absolute scale recovery results

The performance of the ASR module is evaluated in the synthetic dataset. For evaluations, several pixel tracking error conditions are considered: zero-mean random error with a standard deviation of $\{0.5, 1.0, 2.0\}$ pixels. For the rotation motion error, the random error is fixed with a standard deviation of 0.5 degrees. The turning angle threshold $\psi_{th}$ is set to 2.5 degrees.

In Fig. 3.9(b), the simulated odometry trajectory is in red. The author intentionally augments translation drifts to the camera motion to imitate the monocular scale drift. The scale of the simulated trajectory is successively decreased by 0.1 % per frame, which corresponds to the total 33 % scale decreasing at the end.

The detected turning frames are marked with blue squares on the black true trajectory in Fig. 3.9. The scale value observed by (3.24) is plotted in the second row of Fig. 3.11(a). In the figure, the scale of the raw MVO gradually decreases due to the motion drifts. In contrast, for the apparent turning motion in the yellow-shaded regions, the observed scale by the proposed method accurately follows the true value in the black dashed line. As expected, the scale observations during the small turning motion take arbitrary values due to the singularity at the small angle as denoted in (3.24).

By utilizing the observed scale on the turns, the ASR module is conducted to adjust the drifted raw trajectory. The resulting scale history is depicted in the last row of Fig. 3.11(a), and overall trajectories are shown in Fig. 3.11(b). In the figures, the words low, mid, and high denote the track noise conditions of 0.5, 1.0, and 2.0 pixels, respectively. For all the noise conditions, the unobserved scale values of the non-turning regions are successfully recovered, and then, the shapes of the recovered trajectories follow the ground truth well.

Table 3.1 shows the quantitative results for each noise condition. The root-mean-square error (RMSE) values are calculated for four variables: steering angle $\psi_j$, translation direction angle $\theta_j$, absolute error of scale estimation $s_{\alpha,j}$, and scale error ratio $r_j$ calculated

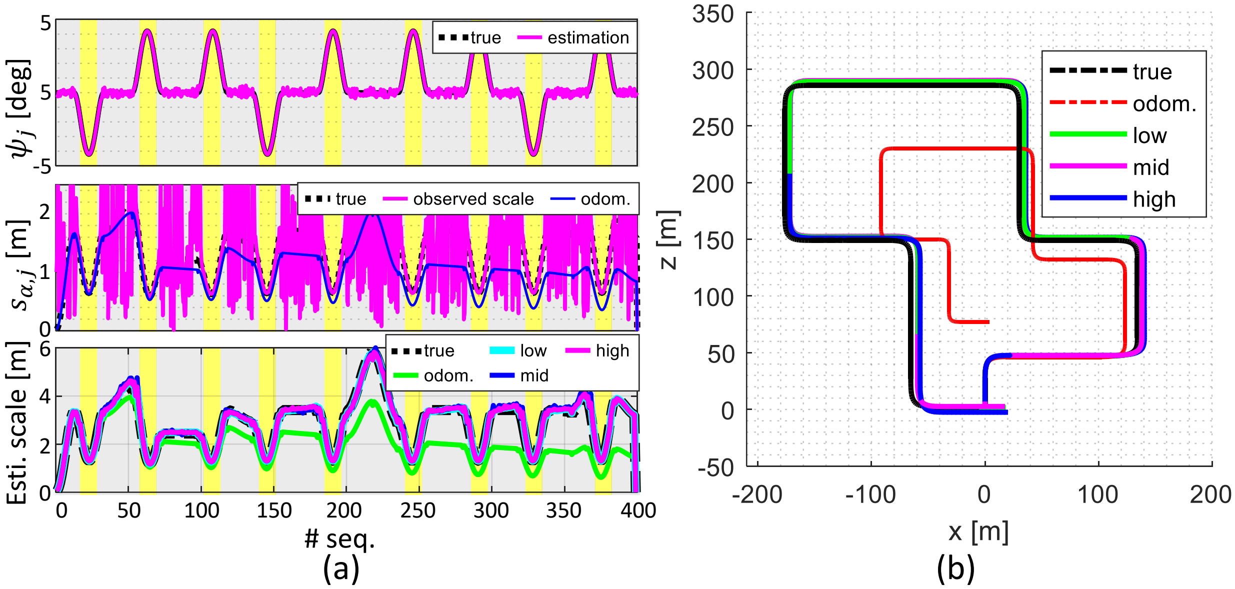


Figure 3.11: **Results of the absolute scale recovery on the synthetic dataset** (a) The first graph is about frame-to-frame turning angle, and the second one shows the scale observation history. The last graph depicts the estimated scale history after the ASR. Yellow and gray shaded regions express the turning and non-turning regions, respectively. (b) Trajectories of the raw monocular odometry and the ASR module with the various camera motion noise settings.

by

$$r_j = \left| s_{\alpha,j} - s_{\alpha,j,true} \right| / s_{\alpha,j,true} \times 100\,[\%]\,, \tag{3.35}$$

where $s_{\alpha,j,true} \in \mathbb{R}$ is the true value for the estimated scale $s_{\alpha,j}$. To separately evaluate the performance of turning and non-turning regions, the two metrics related to the scale estimation are computed for the turning regions only and the entire sequences, respectively.

Table 3.1: **RMSE comparisons of angles and scale estimations on the synthetic dataset**

| Noise<br>[px] | $\psi_j$<br>[deg] | $\theta_j$<br>[deg] | $s_{\alpha,j}$<br>(turn)<br>[m] | $r_j$<br>(turn)<br>[%] | $s_{\alpha,j}$<br>(all)<br>[m] | $r_j$<br>(all)<br>[%] |
|---|---|---|---|---|---|---|
| odom. | 0.500 | - | - | - | 1.098 | 34.79 |
| 0.5 | 0.110 | 0.045 | 0.066 | 1.01 | 0.100 | 3.38 |
| 1.0 | 0.214 | 0.058 | 0.067 | 1.14 | 0.107 | 3.56 |
| 2.0 | 0.238 | 0.062 | 0.067 | 1.52 | 0.145 | 4.57 |

The raw MVO trajectory shows severe scale drifts. But for turning regions, the absolute scale RMSE error shows 0.067 m and the scale error ratio is under 2 % for all noise conditions. In terms of the entire sequence, the absolute scale RMSE error is about 0.1 m, and the scale error ratio increases to 5 %, which results from the recovered scale from the long straight regions making the weak pixel-to-frame connectivity.

From the results, it can be concluded that the proposed method is much more effective in driving conditions with frequent turns and short straight corridors. Those environments can often be seen in actual driving situations such as parking lots. To demonstrate the applicability of the proposed method to the mentioned situations, the author acquires real-world driving image datasets in multi-floor underground parking lots and applies the proposed method, which will be detailed by the experimental results in Chapter 5.

# 4

# Experimental Results on Real-world Scenarios

## 4.1 Implementations on Indoor Office Datasets

The author evaluates the overall performance of the proposed edge-based VO using EuRoC stereo datasets [38]. To verify the practical usability, additional demonstrations of the proposed method are conducted on author-collected datasets gathered in low-textured indoor situations. The proposed method is compared with two state-of-the-art stereo VO algorithms: stereo ORB-SLAM2 [11], and stereo DSO [40]. To compare in terms of pure VO, the SLAM functionality of the ORB-SLAM is switched off. For the quantitative comparison of VO performance, the relative pose error (RPE) is used, which is proposed in [71]. The author publicly shares the experiment datasets used in the research at `https://chkim.net/iros2020`.

### 4.1.1 EuRoC MAV Datasets

To show the robustness to the illumination changing, the previously proposed illumination changing model used in [26] is additionally applied, and the modified datasets are distin-

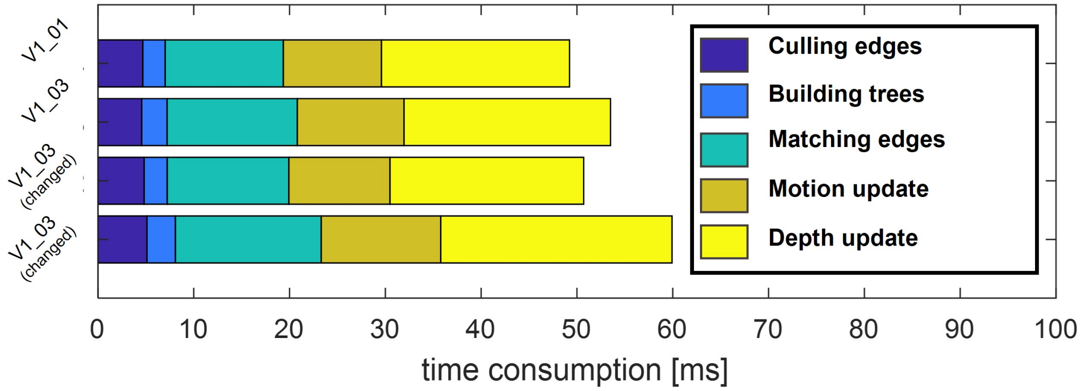


Figure 4.1: **Calculation times of four implementations on EuRoC datasets.**

Table 4.1: **Performance comparison on EuRoC datasets** The bold letter denotes the best performance for each dataset.

| | Relative pose errors (RPE) [m/s] | | |
|---|---|---|---|
| Dataset | Proposed | ORB-stereo | stereo DSO |
| V1_01 | 0.038 | 0.031 | **0.024** |
| V1_03 | **0.046** | 0.052 | failed |
| V1_01(change) | **0.031** | 0.055 | failed |
| V1_03(change) | **0.055** | 0.068 | failed |

guished by a suffix of *change*. V1_01 contains moderated motions and illuminations with abundant textures, and V1_03 has severe blurs from aggressive motions and illumination changes induced by the auto exposure control. The comparison results are shown in Table. 4.1. According to the results, the proposed edge VO shows comparative performance with the others in the normal V1_01. In V1_03, due to the high illumination, DSO fails to operate when auto exposure control excessively intervenes. ORB-VO continues to track motions for all datasets; however, the performances on datasets with changed illuminations are significantly degraded. Nevertheless, the proposed VO shows robust and accurate performance throughout the datasets. The average calculation times per stereo frame are about 50–60 ms for EuRoC datasets as seen in Fig. 4.1.

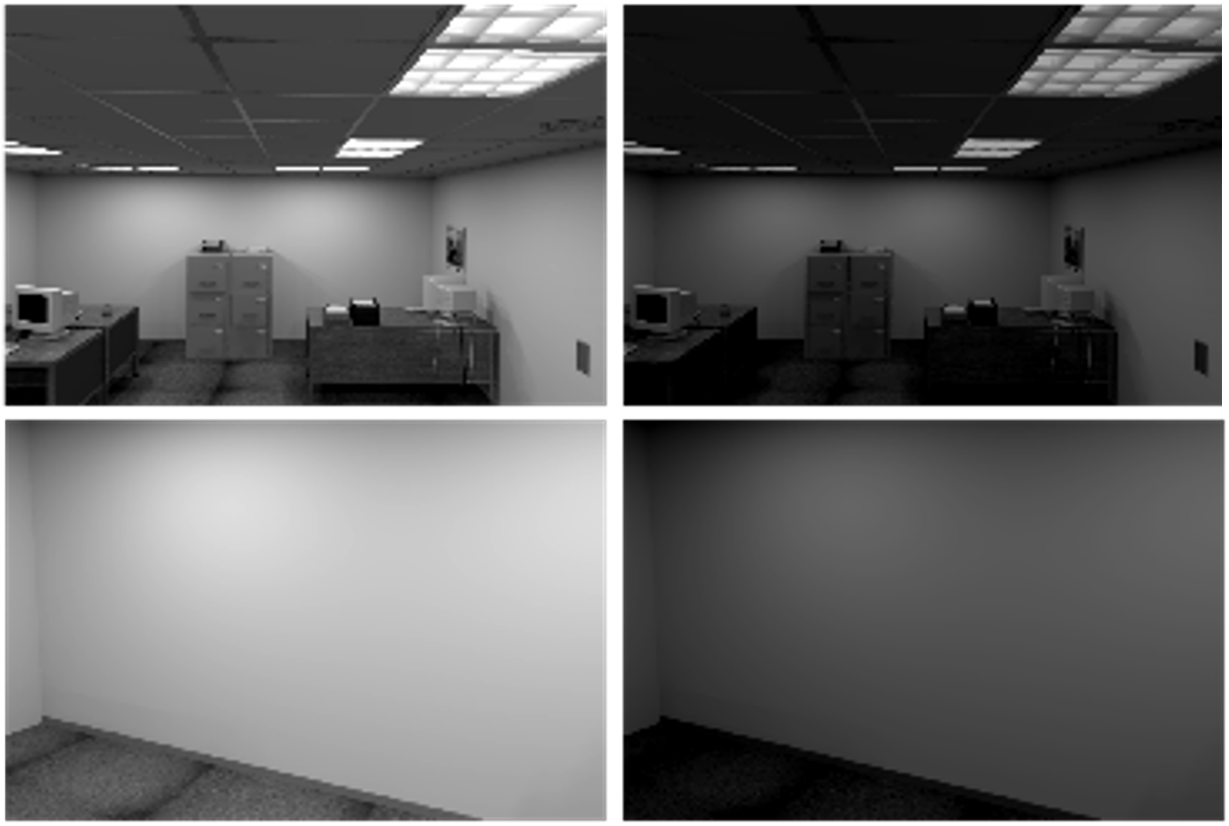


Figure 4.2: **Representative scenes of the ICL-NUIM datasets (left figures), and simulated brightness changes (right figures)[3].**

### 4.1.2 ICL-NUIM Datasets

For performance evaluation of the edge and point combined hybrid method, the ICL-NUIM dataset [3] is used. This dataset simulates the office environment with fewer dot features as shown in Fig. 4.2, and is analyzed in addition to the case of applying a rapidly changing light of ±20 % level in a one-second cycle (ilu.) to verify the robustness to brightness changes. The mathematical model of the affine brightness change is written as

$$\begin{aligned} \alpha &= 1 + 0.2\sin(0.2k) \\ \beta &= 5\sin(0.2k) - 5, \end{aligned} \tag{4.1}$$

Table 4.2: **Performance comparison on ICL-NUIM office datasets** Bold letters denote the best performance for each dataset. The asteriod mark means jumped (diverged) cases, and x mark means a failed case.

| | Absolute Traj. Error (ATE) [m] | | | | Relative Pose Error (RPE) [cm/s] | | | |
|---|---|---|---|---|---|---|---|---|
| Dataset | P(w/o) | P(w/) | E | E+P | P(w/o) | P(w/) | E | E+P |
| office_00 | *0.127 | *0.150 | 0.062 | **0.026** | *0.78 | *0.79 | 0.73 | **0.67** |
| office_01 | *0.099 | *0.116 | x | **0.056** | *0.52 | *0.58 | x | **0.44** |
| office_02 | 0.073 | x | *0.639 | **0.066** | 0.49 | x | *2.41 | **0.46** |
| office_03 | **0.025** | 0.026 | 0.029 | 0.027 | **0.39** | 0.26 | 0.48 | 0.39 |
| office_00(ilu.) | x | *0.150 | 0.056 | **0.028** | x | *0.79 | 0.73 | **0.66** |
| office_01(ilu.) | x | *0.116 | x | **0.058** | x | *0.58 | x | **0.44** |
| office_02(ilu.) | x | x | *0.282 | **0.062** | x | x | *1.44 | **0.49** |
| office_03(ilu.) | x | 0.026 | 0.029 | **0.023** | x | **0.39** | 0.49 | **0.39** |

where $k \in \mathbb{N}^+$ is the number of sequences of each dataset.

The algorithm performance is analyzed for the case where only point features are used (P), only corners were used (E), and both features are used (E+P). In particular, when only point features were used, the performance according to the presence or absence of brightness change compensation was additionally analyzed. For quantitative performance comparison, absolute trajectory error (ATE), the sum of squares of the difference between the true camera posture value and the estimate for each image, was used as an indicator. In addition, the odometry performance is compared by comparing the relative pose error (RPE). The resulting trajectories are depicted in Figs. 4.3,4.4,4.5,4.6.

As the result of Table 4.2, office_01 and office_02 data, which lack a lot of patterns in Figs. 4.4 and 4.5, used only point features, a jump occurred in posture estimation, and if the brightness changes, posture estimation failed. When only the corners were used, a similar result could be derived even in the brightness change, but it failed in office_01 where the corners were not evenly distributed, and the average accuracy was measured to be low. On the other hand, it was confirmed that the proposed algorithm achieves stable and high accuracy in all cases.

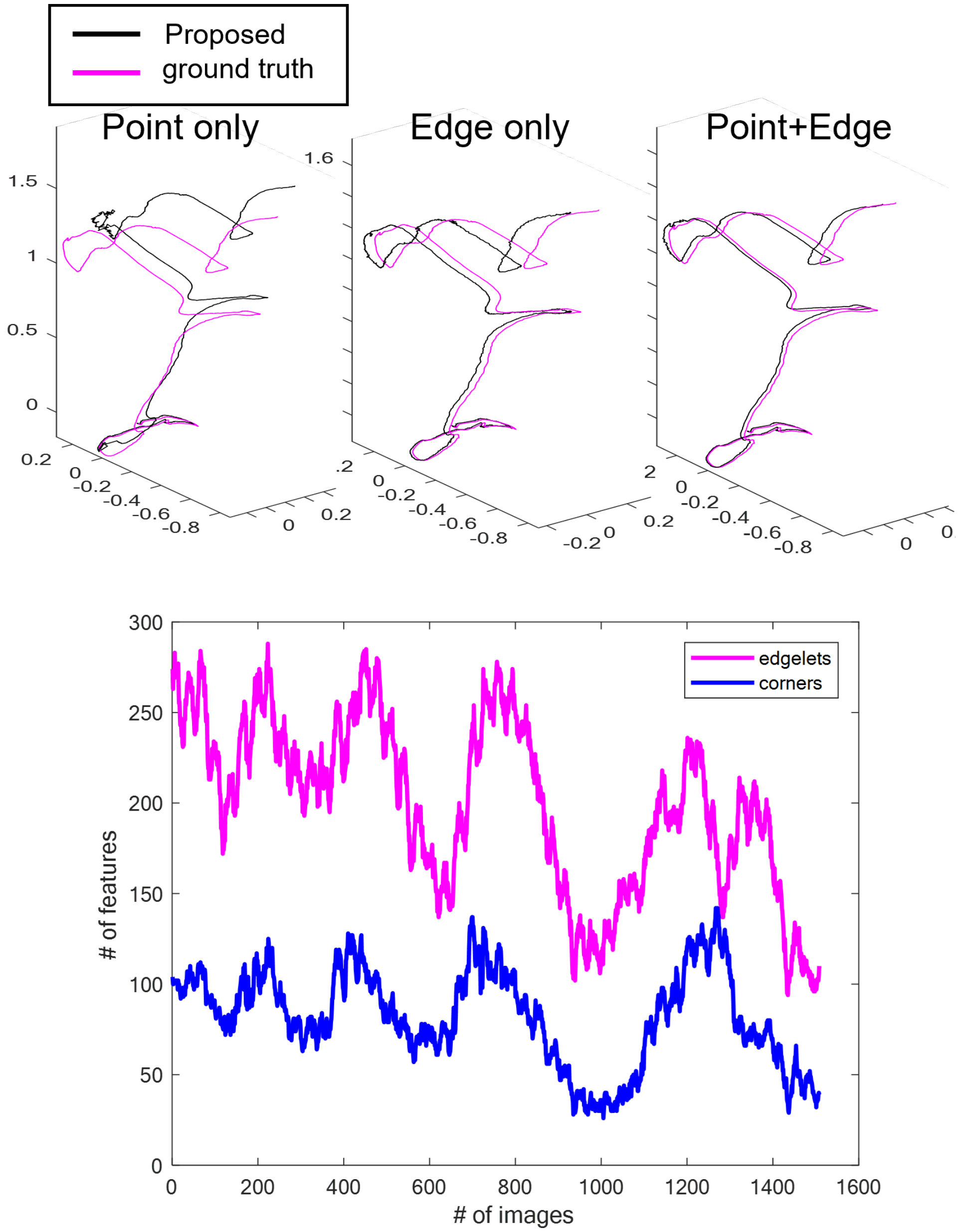


Figure 4.3: **Results of the edge and point-based VO with various settings on `office_00` of ICL-NUIM dataset.**

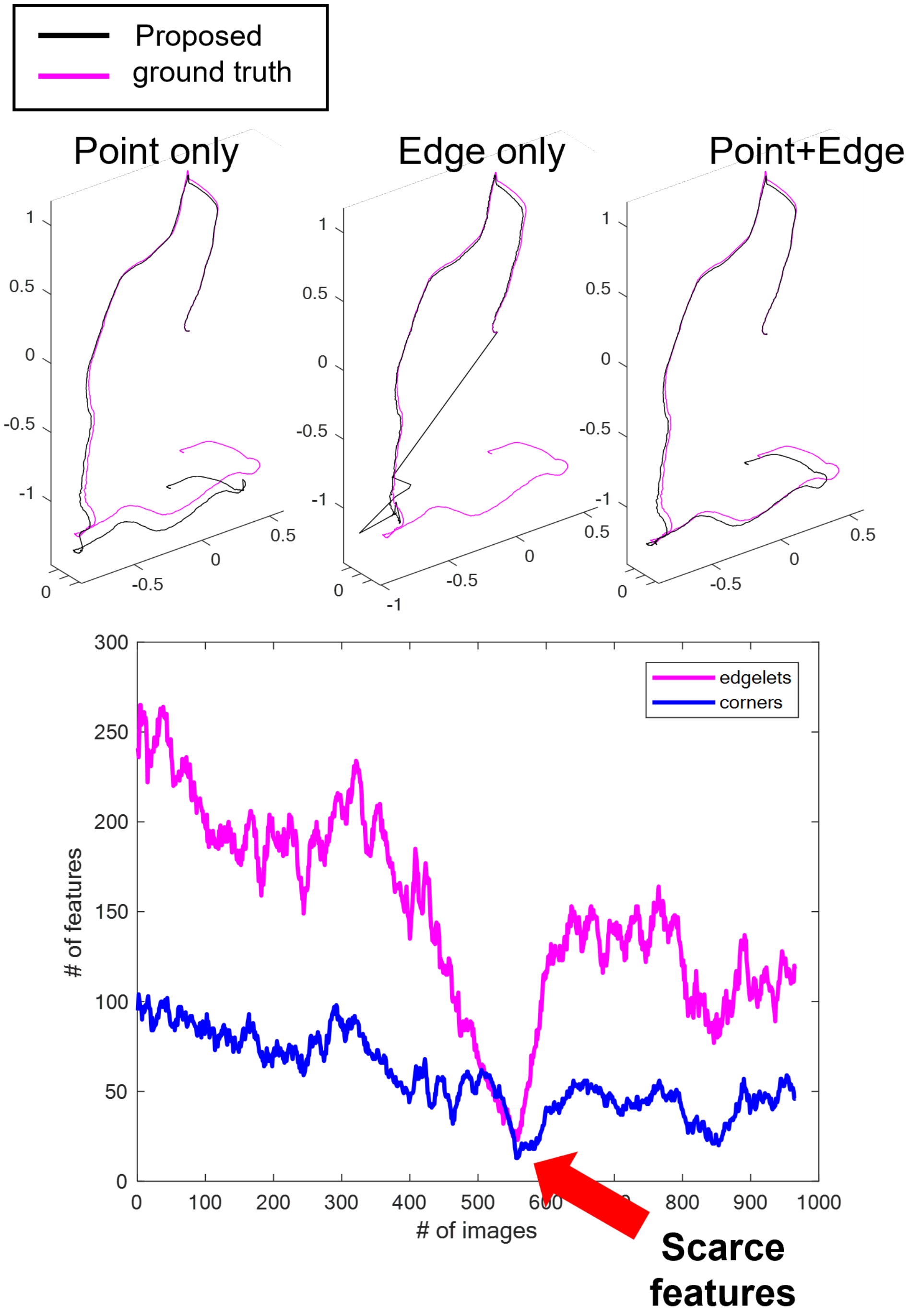


Figure 4.4: **Results of the edge and point-based VO with various settings on** `office_01` **of ICL-NUIM dataset.**

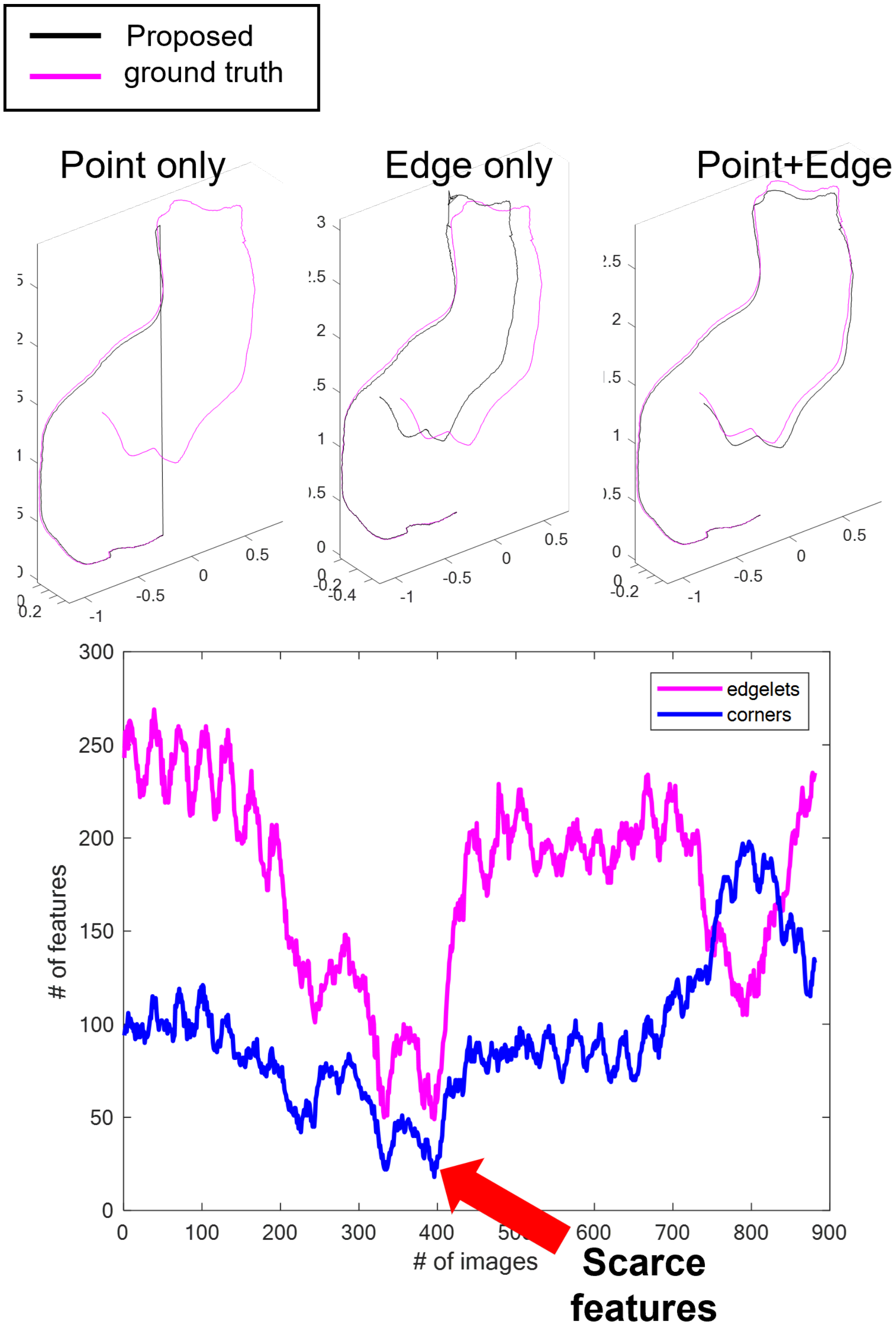


Figure 4.5: **Results of the edge and point-based VO with various settings on `office_02` of ICL-NUIM dataset.**

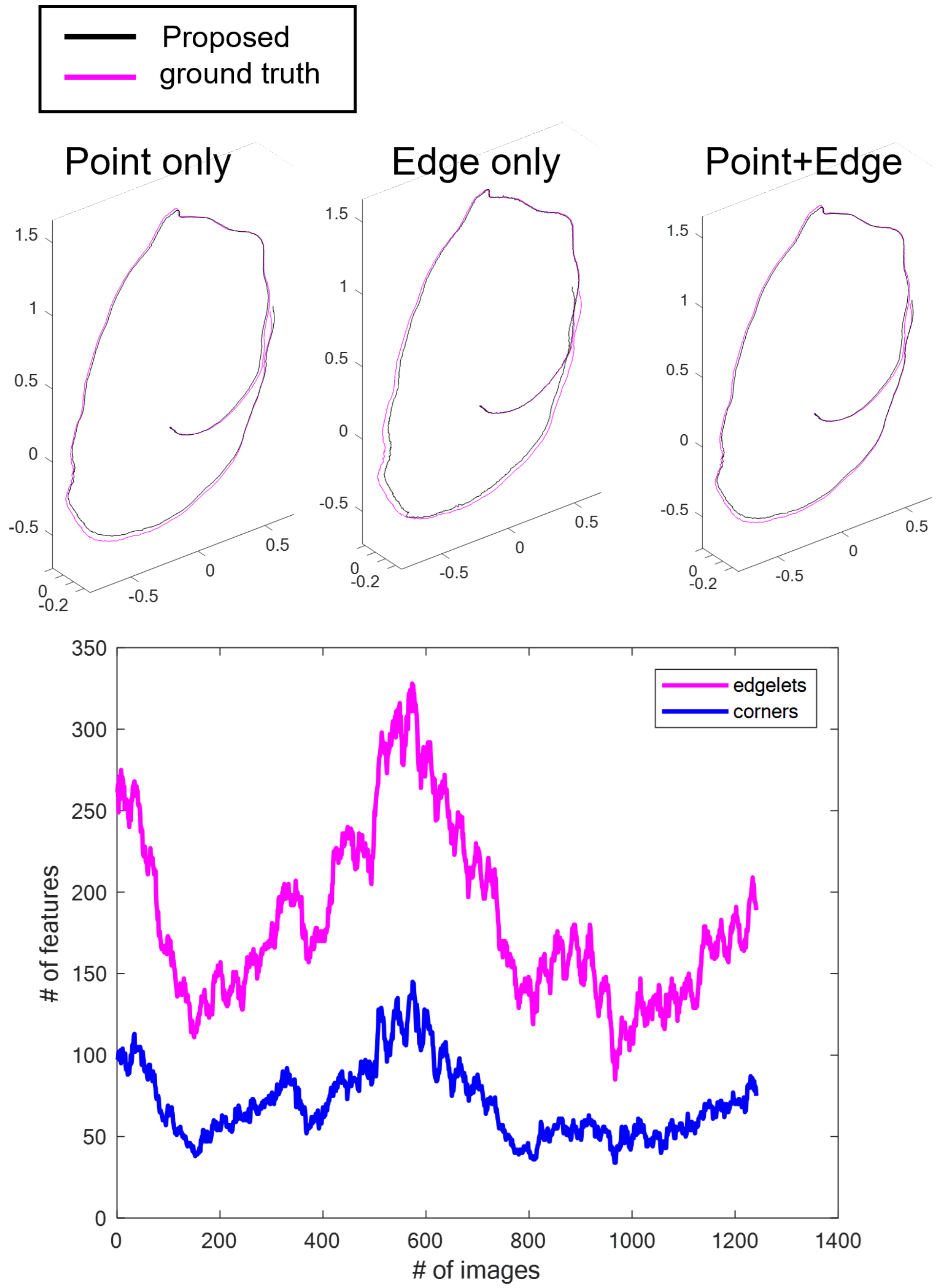


Figure 4.6: **Results of the edge and point-based VO with various settings on `office_03` of ICL-NUIM dataset.**

### 4.1.3 SNU Modern Building Indoor Datasets

To verify the real-world applicability of the proposed method, the author collected challenging man-made scenes that include few free-formed edges only. Because there is no ground truth trajectory, the camera is carefully moved to maintain the same height throughout the loop, and starting and end positions are set identically to check whether the results are consistent. As depicted in Fig. 4.7, the rectangular skeleton of the corridor is accurately recovered. Moreover, the proposed method maintains the stable height estimate and exactly returns back to the starting point blue circle without any help of re-localization ability like SLAM. Further results can be seen in Figs. 4.8, 4.9, and 4.10.

For the edge and point-based VO, the author additionally collects indoor datasets. As seen in Fig. 4.11, the shapes of the furniture with curve features are well reconstructed. In Fig. 4.12, the detailed inter-floor stair structures are also reconstructed well.

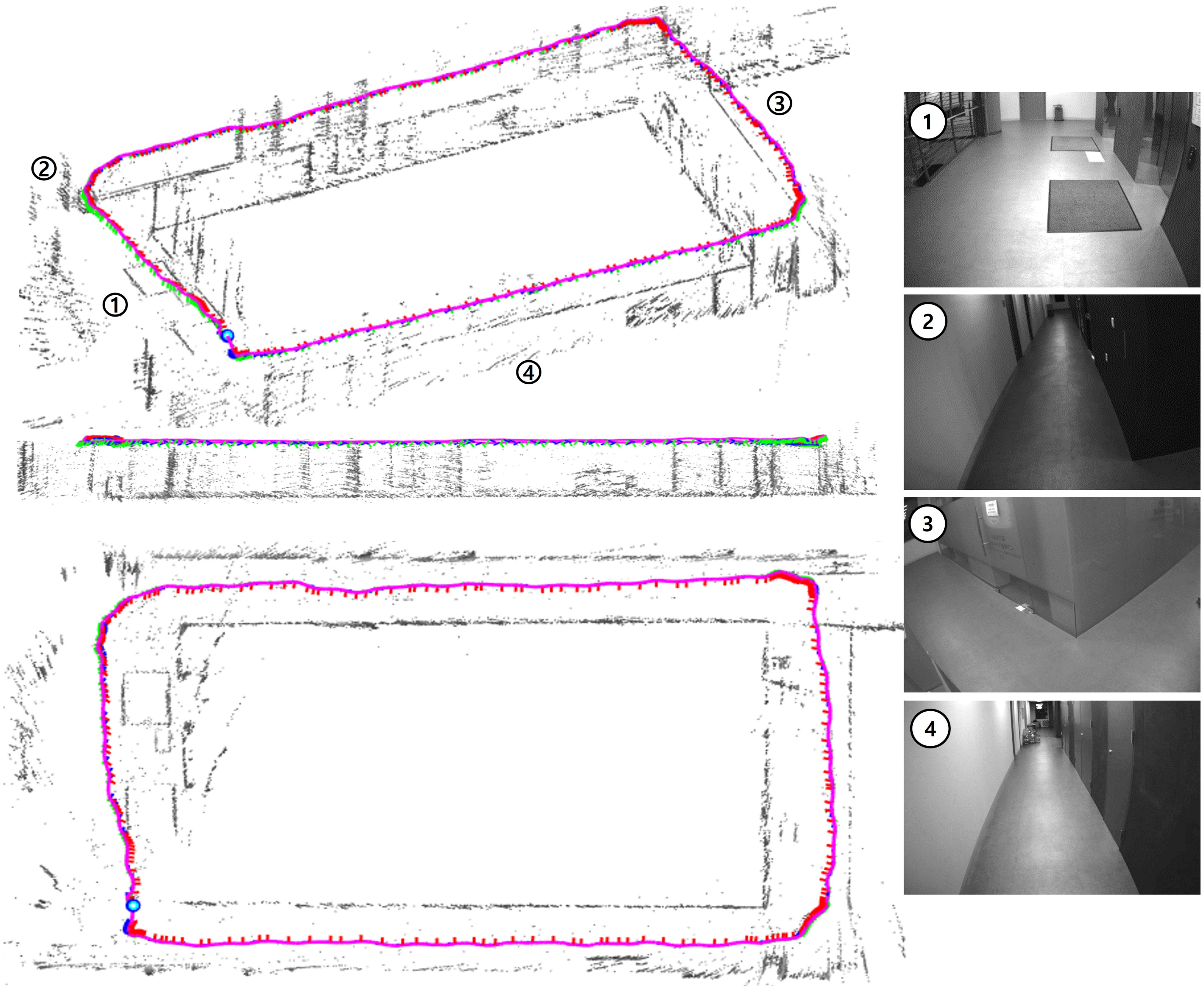


Figure 4.7: **VO and 3-D reconstruction results on the author-collected dataset.** The estimated trajectory is in magenta, ranging 7 m×12 m.

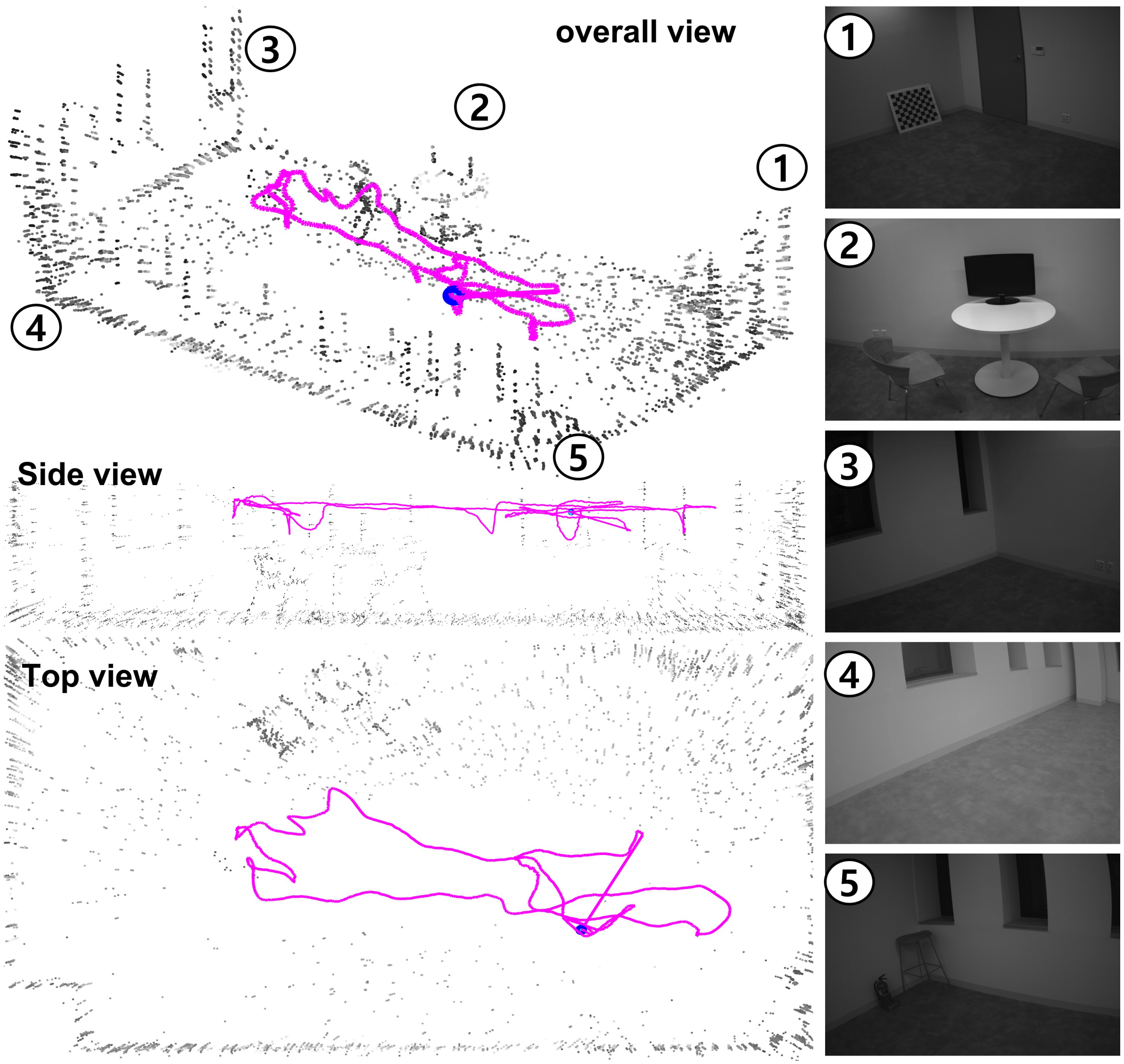


Figure 4.8: **VO and 3-D reconstruction results on the author-collected dataset.** The estimated trajectory is in magenta, ranging 7 m×12 m.

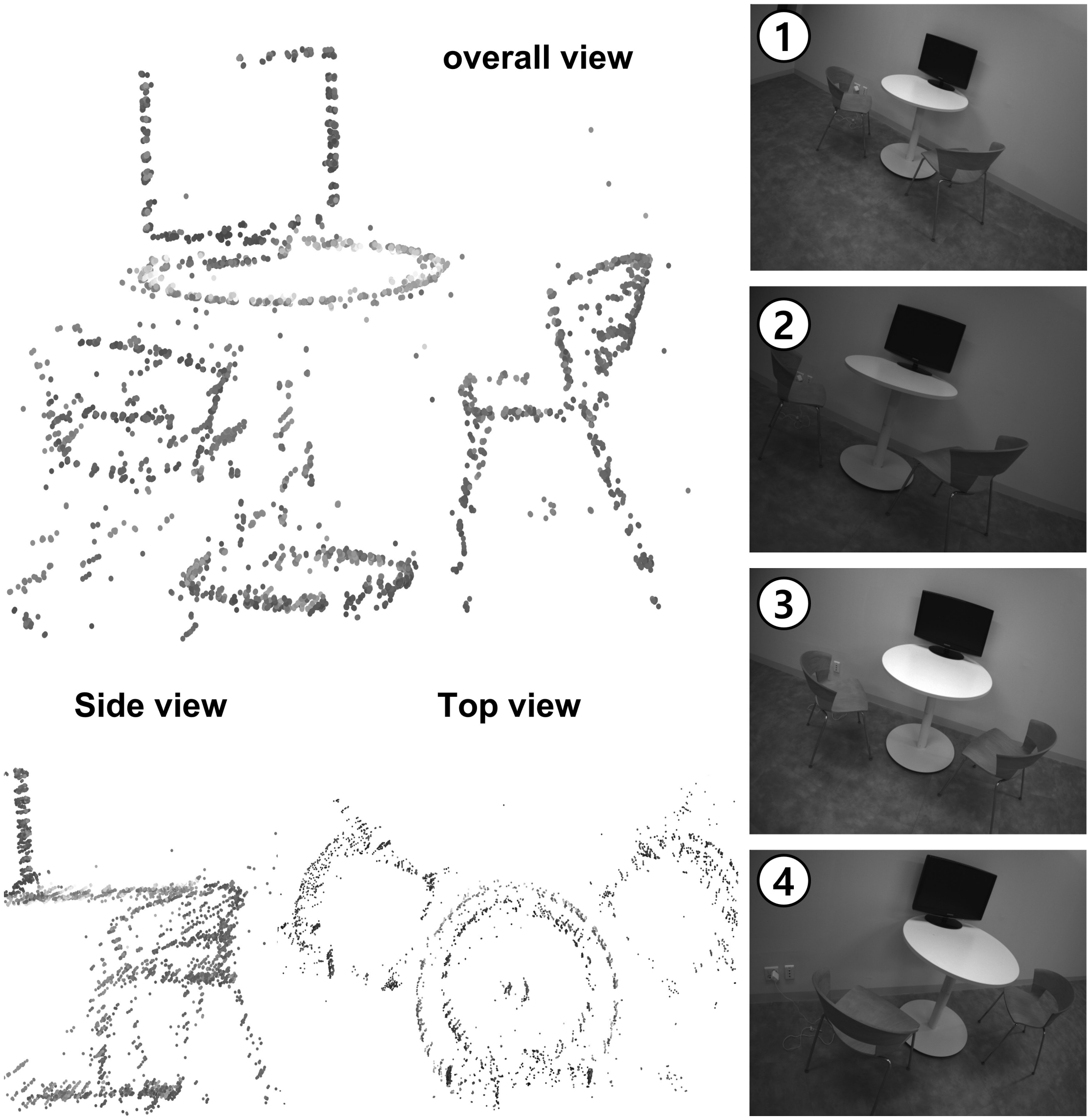


Figure 4.9: **VO and 3-D reconstruction results on the author-collected dataset.** The estimated trajectory is in magenta, ranging 7 m×12 m.

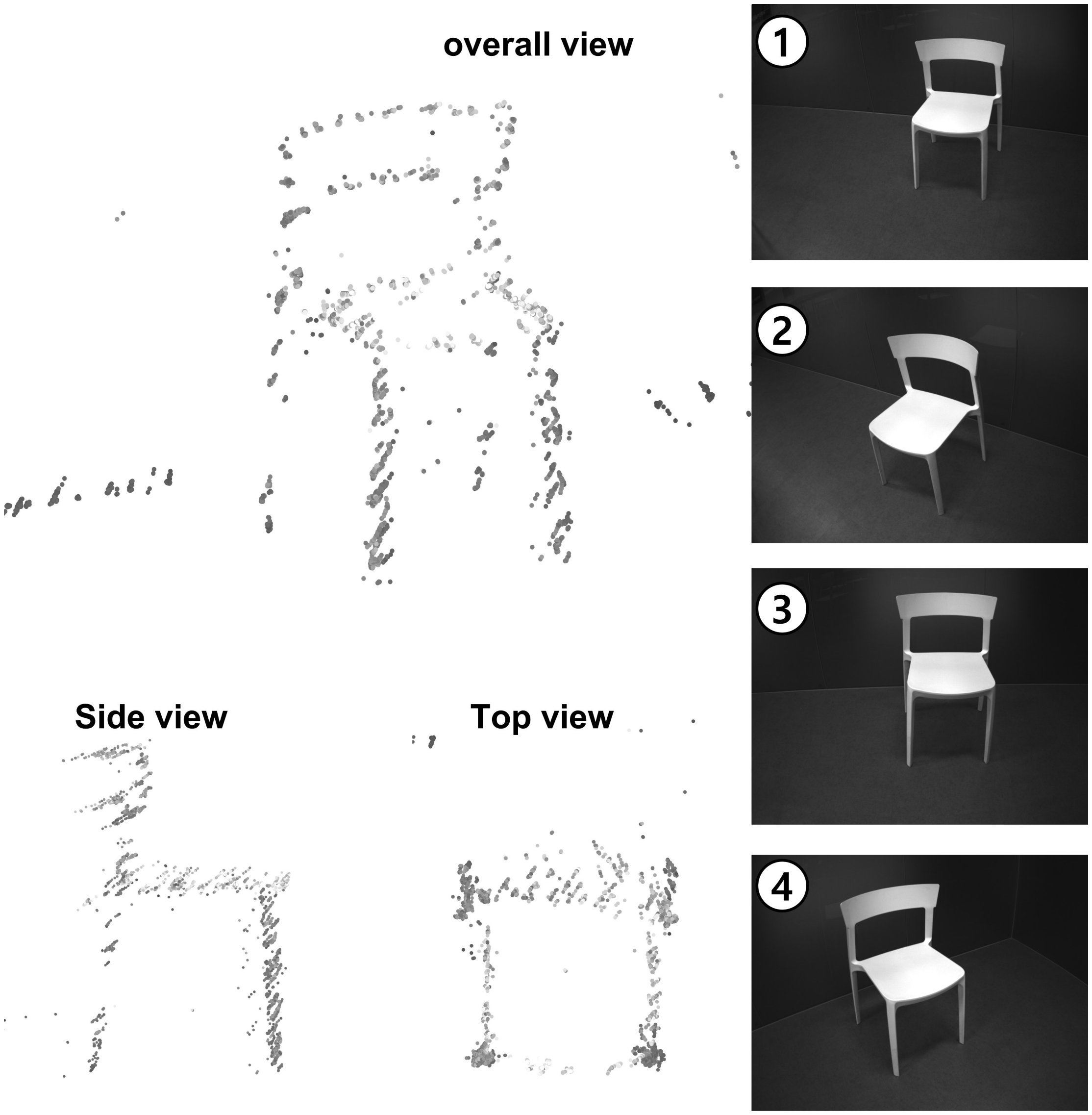


Figure 4.10: **VO and 3-D reconstruction results on the author-collected dataset.** The estimated trajectory is in magenta, ranging 7 m×12 m.

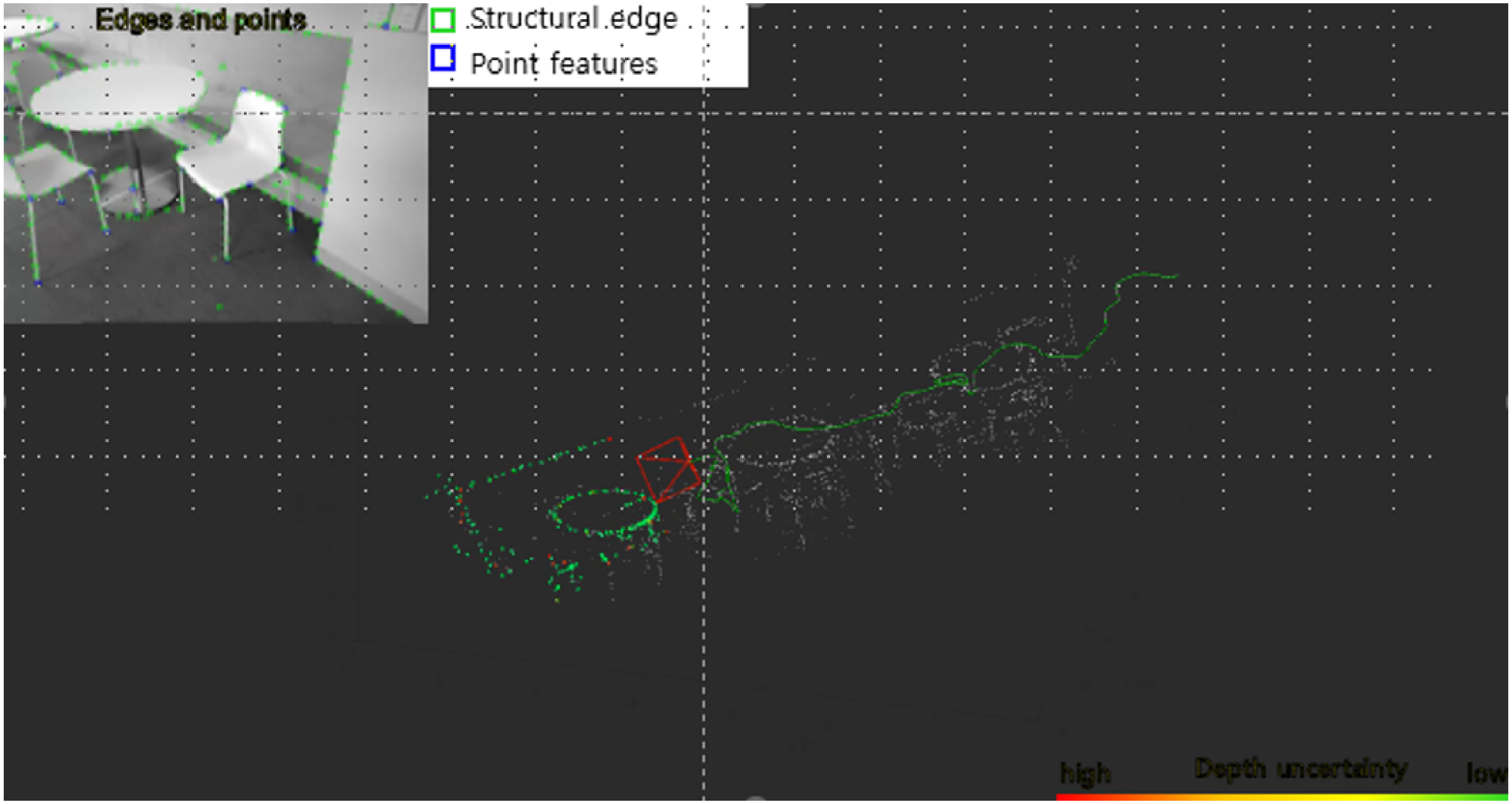


Figure 4.11: **Edge and point-based VO and 3-D reconstruction results on the author-collected dataset - library**

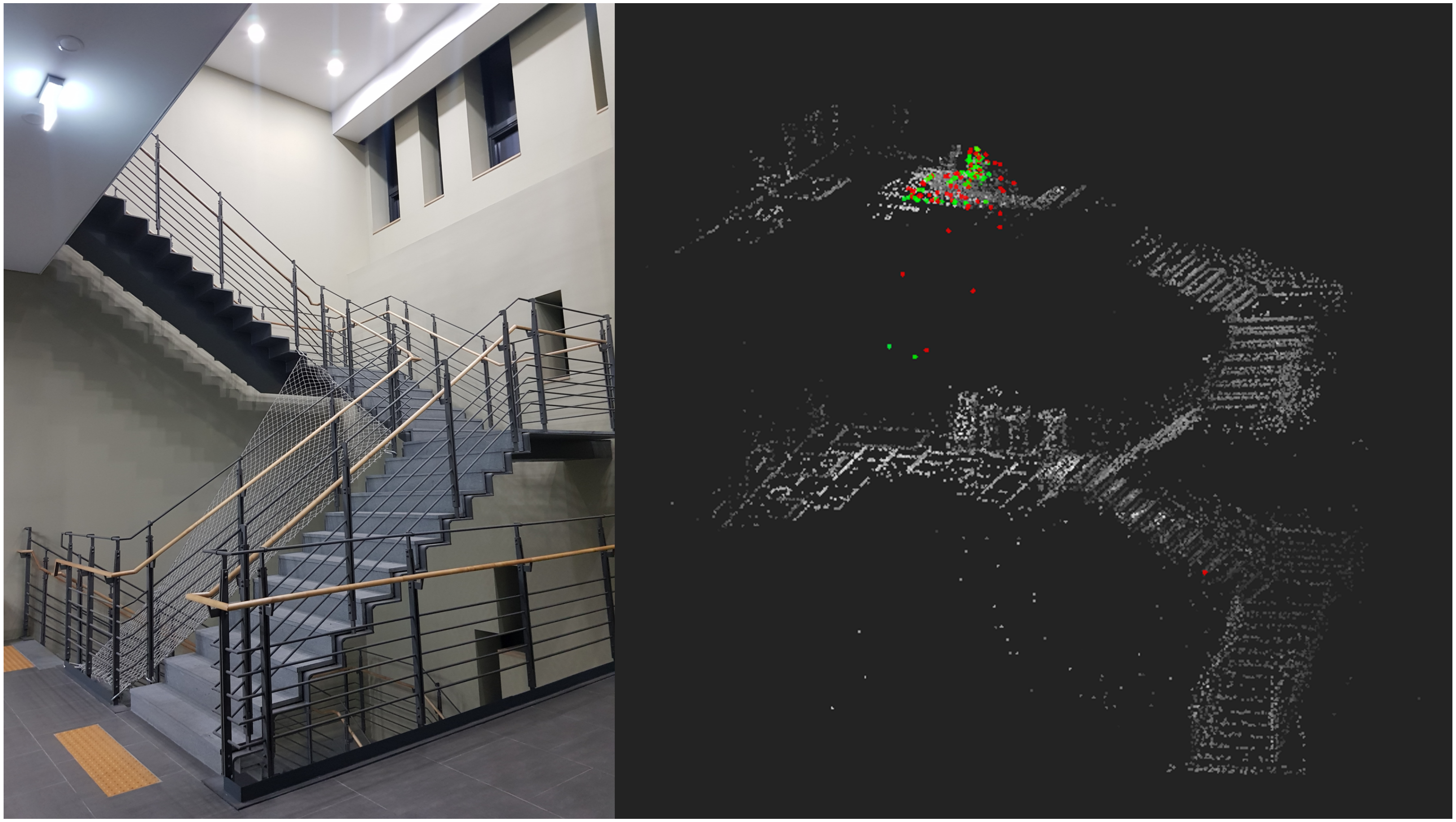


Figure 4.12: **Edge and point-based VO and 3-D reconstruction results on the author-collected dataset - stair**

## 4.2 Implementations on Indoor Driving Datasets

Then, the overall MVO performance of the proposed method is evaluated on publicly available outdoor driving datasets, i.e., KITTI odometry datasets [2]. In the end, the promising real-world applicability of the proposed method is demonstrated on author-collected indoor driving datasets acquired in two different underground parking lots with multiple floors.

First, the overall performance of the proposed scale MVO is exhibited by using the publicly available outdoor driving image datasets, KITTI odometry datasets [2]. To highlight the practical value of the proposed MVO, the author additionally collects the real-world indoor driving sequences, called SNU underground parking lots datasets. The proposed method is quantitatively evaluated by comparing it with the popular visual navigation stack, ORB-SLAM [11], in monocular and stereo modes. For abbreviations, let them be called ORB-mono and ORB-stereo, respectively. In this implementation, the source code of the latest publication ORB-SLAM3 [72] is used. To compare in the manner of VO, the loop-closure and re-localization modules of the ORB-SLAM are deactivated.

### 4.2.1 KITTI Odometry Datasets

Sequences of the KITTI datasets are composed of the time-synchronized 10 Hz stereo images with the accurate ground-truth pose post-processed by the OXTS RT 3003 (GPS/IMU) inertial navigation solution. The stereo images are stereo-rectified and have 1240 × 376 pixels resolution. The monocular images obtained by the left grayscale monocular camera are used to operate the proposed method and the ORB-mono.

According to the sensor setup of the KITTI datasets, $L = 0.93$ m and $\mathbf{Q} = \mathbf{I}_3$ are considered, and $\psi_{th} = 2^\circ$ is set by considering that the average frame-to-frame rotation angle of the dataset is about 3 degrees.

The scale consistency performance of the proposed method is evaluated in 11 sequences, `00`–`10`. The sequence `01` is not used, for which most feature-based VO methods fail [5, 54, 55]. Table 4.3 shows quantitative results of the proposed method, ORB-mono, and stereo modes. As the performance metric, the scale error ratio RMSE (3.35) is computed for each sequence. To compensate the unknown initial scale of the monocular methods, the scale value of the initial ten frames from the true trajectory is provided.

As seen in Table 4.3, the scale error ratio RMSE of the ORB-mono increases over 50 % for several sequences. This scale drift problem has been reported in the original ORB-SLAM paper [11]. The ORB-stereo shows stable performance thanks to the metric length of the stereo baseline.

The proposed method shows competitive performance to the ORB-stereo in several sequences marked with $*$ in Table 4.3; however, in the other sequences, performance degrades similarly to the ORB-mono. To analyze this, the author additionally calculates statistics of each sequence in the table: the number of turns, minimum, maximum, and average distance between adjacent turning regions. Contrary to sequences with $*$ mark, non-marked sequences have very few distant turns, or no turn at all. Those sequences mainly have long straight paths between adjacent turning regions, and the vehicle changes driving directions very slowly with a huge radius of curvature, which is not the target environment of this chapter.

In Table 4.4, the author additionally computes the translation error suggested in [2] of the proposed method, and compares the performance of the proposed method to the ORB-mono and stereo, and the state-of-the-art plane-based scale-aware MVO works [5, 54, 55]. Because no open-source implementation is available for these works, the reported results in [5, 54, 55] are referred to. The method [55] shows the best and most stable performance thanks to the robust ground point extraction and aggregation strategies proposed in [55]. Similar to the results in Table 4.3, the proposed method shows comparable performance on the $*$-marked sequences with an average translation error of about 3.5 %. The method [54] fails to track motion in several sequences because it uses a fixed image region to obtain the ground features, and [5] reports divergence in 07 due to the occlusion of the fixed ground region by a dynamic object.

The representative trajectories for successful sequences are depicted in Fig. 4.14. In

Table 4.3: **Quantitative comparison on the KITTI odometry datasets - scale estimation error ratio**

| | Scale estimation error ratio RMSE [%] | | | Sequence statistics | | | |
|---|---|---|---|---|---|---|---|
| No. | ORB-mono | ORB-stereo | Proposed | # of turns | min. dist. [m] | max. dist. [m] | avg. dist. [m] |
| *00 | 45.7 | 1.7 | 8.2 | 28 | 17.4 | 447.9 | 125.5 |
| 02 | 43.0 | 1.2 | 13.7 | 17 | 27.5 | 648.9 | 230.1 |
| 03 | 9.9 | 1.8 | 9.9 | 0 | - | - | - |
| 04 | 63.4 | 0.9 | 63.4 | 0 | - | - | - |
| *05 | 116.0 | 1.9 | 5.8 | 9 | 53.0 | 450.8 | 182.6 |
| 06 | 28.8 | 1.1 | 28.8 | 2 | - | 443.9 | - |
| *07 | 71.8 | 3.0 | 6.9 | 6 | 67.9 | 146.8 | 98.3 |
| *08 | 85.8 | 2.1 | 10.5 | 18 | 4.8 | 386.1 | 160.1 |
| 09 | 31.2 | 1.4 | 17.8 | 4 | 20.7 | 579.9 | 307.7 |
| 10 | 7.8 | 1.6 | 7.7 | 2 | - | 674.3 | - |

addition, the overall turning region detection results are depicted in Fig. 4.13. The red markers denote the detected turning regions by the proposed turning region detection method. Except for several long straight regions, the proposed method yields the absolute metric trajectory that overlaps with the ground truth line. The average of the scale error ratio RMSE on 00, 05, 07, 08 is about 8 % corresponding to 1/10 of the naive ORB-mono case.

### 4.2.2 SNU Underground Parking Lots Datasets

To demonstrate the promising applicability of the proposed method for indoor driving, the author collects his own driving datasets, called SNU underground parking lots datasets. Different from the KITTI outdoor datasets, due to the absence of the external ground truth measurement such as the GPS/INS solution, the author additionally records the 3-D LiDAR pointcloud, and executes the LiDAR odometry and mapping (LOAM) algorithm [73] on the author-collected datasets to obtain the accurate metric trajectory. For the

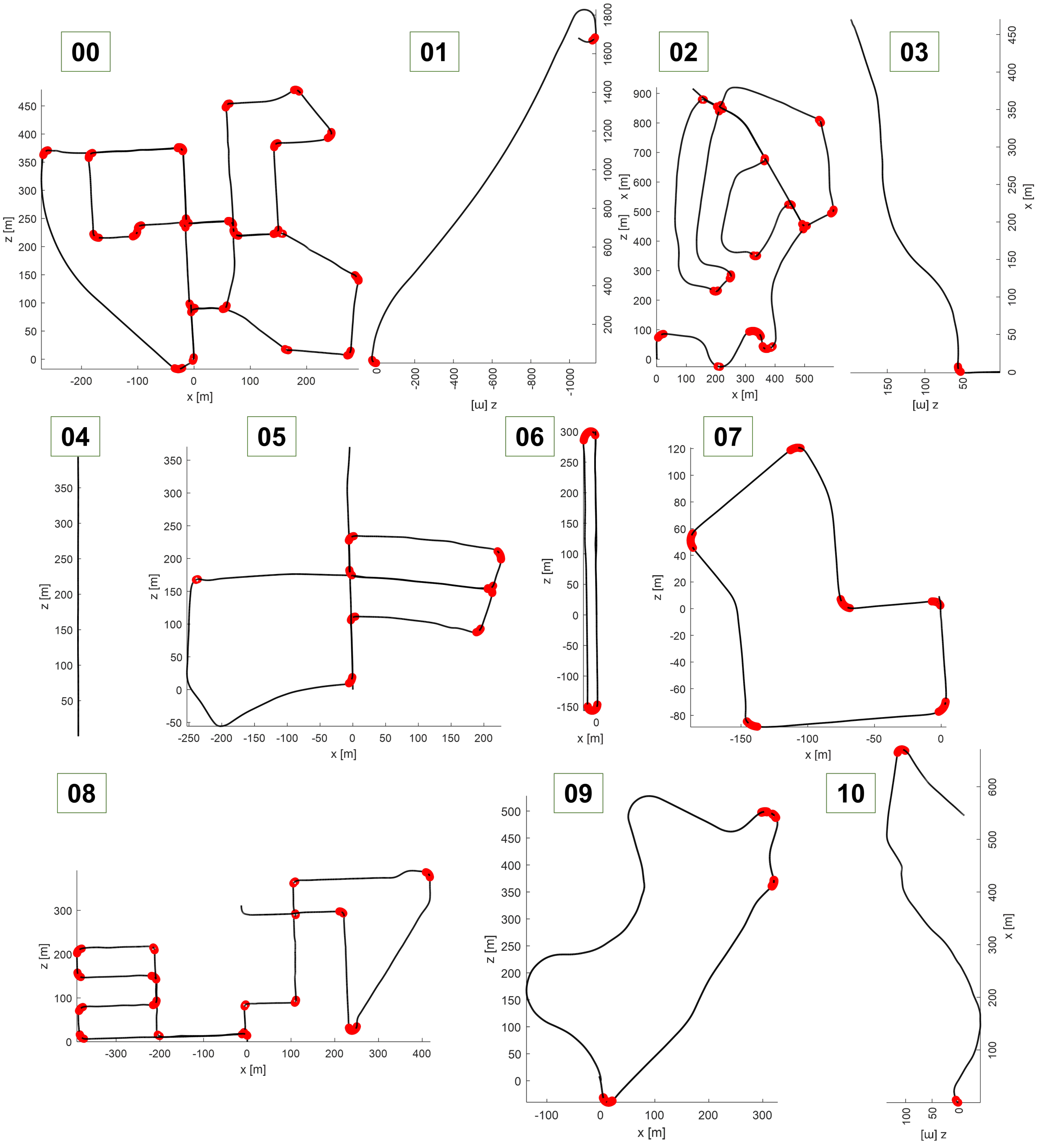


Figure 4.13: **Turning region detection results on KITTI odometry datasets** The red markers denote the detected turning regions by the proposed turning region detection method.

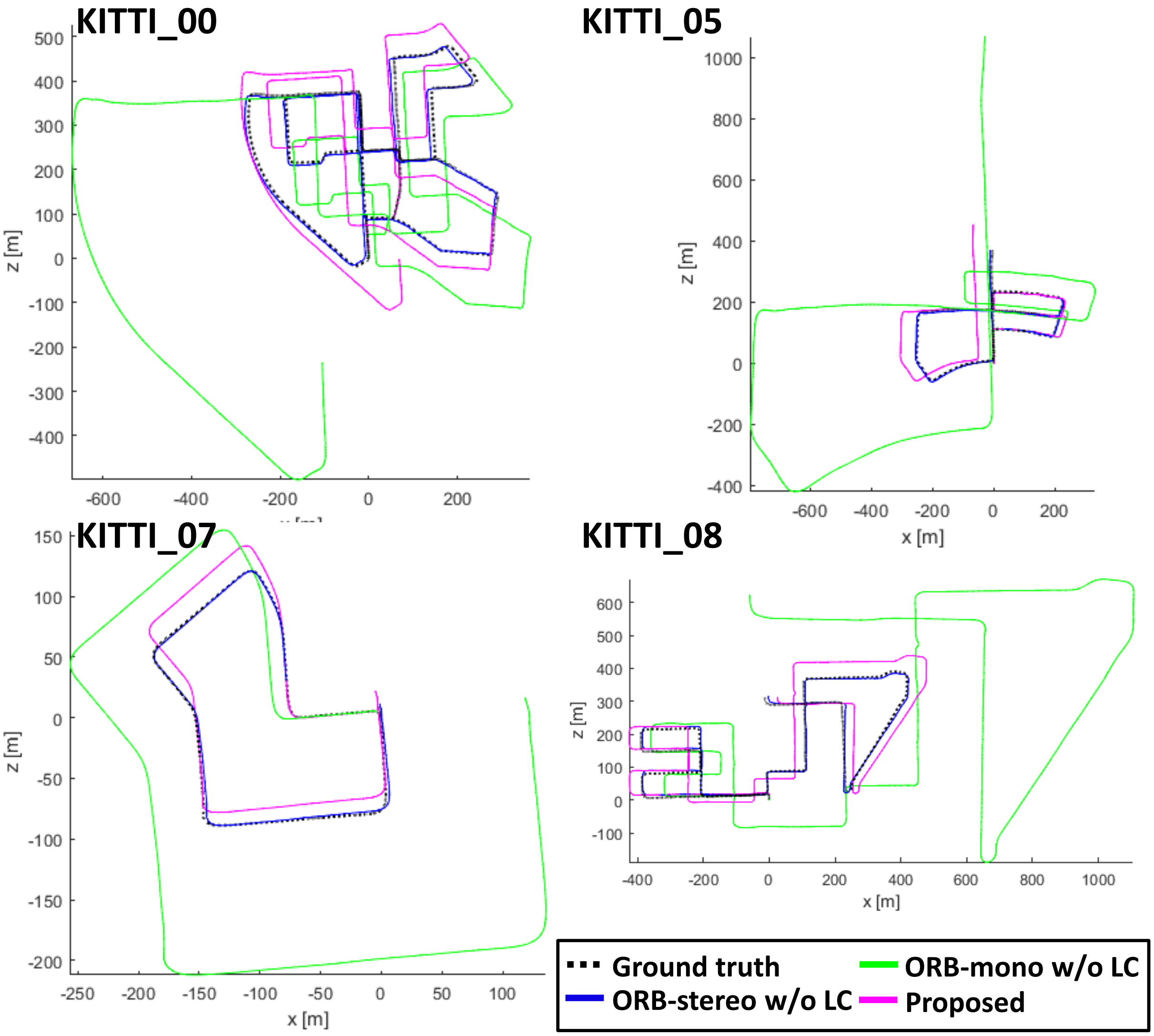


Figure 4.14: **Representative trajectories of the proposed method on the KITTI odometry datasets** Trajectories are the results on 00, 05, 07, and 08. The black dashed line denotes the ground truth trajectory, and the green and blue trajectories are of the ORB-mono and stereo settings, respectively. The results of the proposed method are depicted in magenta.

Table 4.4: **Quantitative comparison on the KITTI odometry datasets - translation error** The boldface means the best performance except for the stereo VO, ORB-stereo. Dash means failure cases.

| | Translation error [%] | | | | | |
|---|---|---|---|---|---|---|
| No. | ORB-mono | ORB-stereo | Song *et al.* [5] | Zhou *et al.* [54] | Tian *et al.* [55] | Proposed |
| *00 | 20.8 | 0.70 | 2.04 | 2.17 | **1.41** | 3.29 |
| 02 | 9.52 | 0.76 | **1.50** | - | 2.18 | 9.52 |
| 03 | 11.58 | 0.71 | 3.37 | - | **1.79** | 11.58 |
| 04 | 15.47 | 0.48 | 2.19 | 2.70 | **1.91** | 15.47 |
| *05 | 18.63 | 0.40 | **1.43** | - | 1.61 | 3.05 |
| 06 | 18.98 | 0.51 | 2.09 | - | **2.03** | 18.98 |
| *07 | 13.82 | 0.50 | - | - | **1.77** | 3.36 |
| *08 | 22.06 | 1.05 | 2.37 | - | **1.51** | 3.11 |
| 09 | 12.76 | 0.87 | **1.76** | - | 1.77 | 12.76 |
| 10 | 4.86 | 0.60 | 2.12 | 2.09 | **1.25** | 4.86 |

Table 4.5: **Hardware specifications of the author-collected dataset**

| Hardware | Qty. | Specifications |
|---|---|---|
| Vehicle | 1 | Hyundai Elantra CN7 2021<br>length: 4.68 m, width: 1.82 m<br>height: 1.41 m, wheel base: 2.72 m |
| Camera | 3 | Matrixvision mvBlueCOUGAR-X104iG<br>1032 × 772 pixels gray image at 10 Hz<br>Global shutter and hardware triggered<br>GiGE interface |
| 3-D LiDAR | 1 | Velodyne VLP-32C<br>32-channel 360 deg. laser scans at 10 Hz<br>20 deg. vertical field of view |
| IMU | 1 | Lord Microstrain 3DM-GX3-25 AHRS<br>3-axis acc., 3-axis gyro. at 250 Hz |
| Micro-controller | 1 | Arduino MKR Zero with the Ethernet Shield |

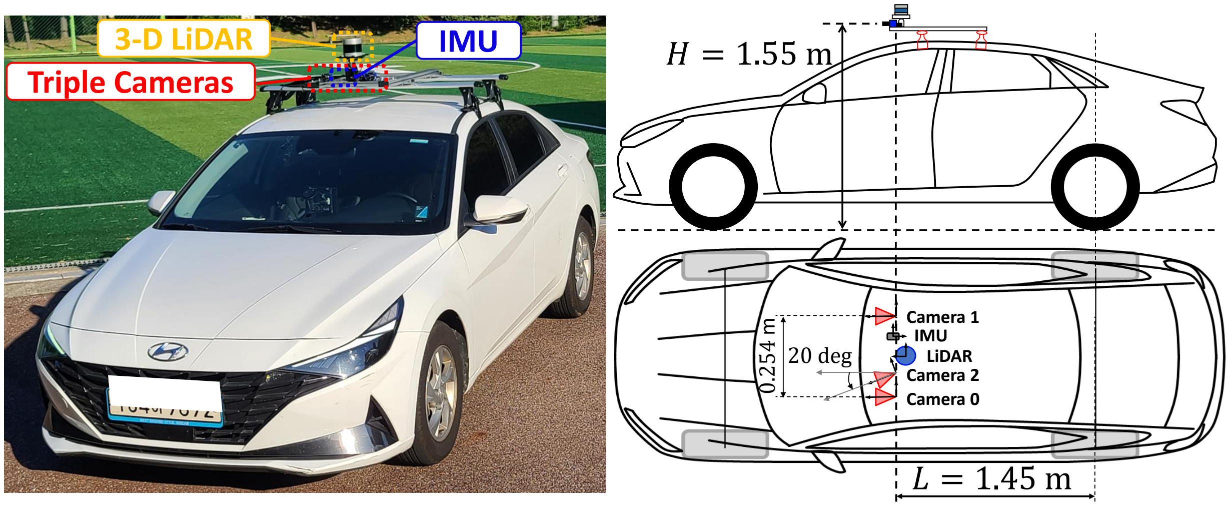


Figure 4.15: **Experimental setting for the author-collected dataset - overview**

LOAM trajectories, the author utilizes not the raw odometry result but the trajectory after the mapping procedure for high accuracy.

The hardware setting of the automobile and sensor suites is shown in Fig. 4.15, and sensor specifications are written in Table 4.5. Three global shutter grayscale cameras, a 32-channel 3-D LiDAR, and a 6-axis IMU are equipped on the roof of the vehicle. All cameras are triggered to capture time-synchronized 10 Hz images by the digital signal from the Arduino MKR Zero microcontroller. All the sensors and the microcontroller communicate to the Linux laptop computer by the ethernet interface.

The camera setting of the author-collected dataset has $L = 1.45$ m, and the height of the cameras is $H = 1.55$ m from the ground. Two main cameras numbered 0 and 1 face front, and an auxiliary camera with the number 2 is rotated left by 20 degrees. The extrinsic parameters of cameras, 3-D LiDAR, and IMU are calibrated by using the LiDAR and camera extrinsic calibration module in [74].

The author drives the automobile in two multi-floor underground parking lots on campus: `bldg_39` and `bldg_220`. Overviews and dimensions of both environments are illustrated in Figs. 4.18 and 4.19. `bldg_39` has two floors with identical shapes; `bldg_220` has three floors across the two different buildings.Especially in `bldg_220`, spiral inter-floor

Table 4.6: **Results of the camera-vehicle extrinsic calibration on the author-collected dataset**

| Camera-vehicle pose in z-y-x Euler angles [deg] | | | | | | |
|---|---|---|---|---|---|---|
| | Camera 0 | | | Camera 2 | | |
| | Roll | Pitch | Yaw | Roll | Pitch | Yaw |
| Truth | 0.00 | 0.00 | 0.00 | 0.00 | -20.00 | 0.00 |
| Linear-only | -0.15 | 3.97 | 0.18 | 0.26 | -23.55 | -0.15 |
| Refinement | 0.07 | 0.13 | 0.08 | 0.10 | -20.12 | -0.05 |

transitions are concentric, which could be a reference point for qualitative evaluations.

Representative scenes for each dataset are shown in Fig. 4.17 and each alphabetic label corresponds to locations with the same label in Figs. 4.23 and 4.24. Different from the KITTI datasets, there are only a few spurious image features on the ground generated by the specular reflection, which might not be suitable for the plane-based scale-aware MVO methods [50, 5, 51, 52, 56, 53, 54, 55].

First, the camera-vehicle extrinsic poses are estimated by the proposed extrinsic calibration method for the author's experimental settings. For the calibration, cameras 0 and 2 are used, which are depicted in the layout of Fig. 4.15. The author uses the pose trajectory of each camera obtained from the MVO between the first two turns of `bldg_39`. In Table 4.6, the linear-only method yields inaccurate results as reported in the analysis on the synthetic dataset. On the contrary, the full refinement shows very accurate performance with an average error smaller than 0.2 degrees. The resulting camera-vehicle extrinsic poses are used during the experiments.

Implementation results are shown in Figs. 4.23 and 4.24. As there is no ground-truth trajectory for the author-collected datasets, resulting trajectories are overlaid onto the real-scale floorplan drawing, and compare the proposed MVO with two absolute-scale navigation methods, i.e., the ORB-stereo and LOAM. The LOAM successfully operates on the `bldg_39` dataset; however, it fails to estimate forward motion at the spiral inter-floor passages of `bldg_220` because there are very few structural 3-D features along the driving direction as seen at the label Ⓖ of Fig. 4.17. Nevertheless, the trajectories on each floor are stably estimated, and they can be used as references of comparison.

While the ORB-stereo operates accurately for all sequences, the scale of the monocular version severely drifts. The author thinks that the severe drift of the ORB-mono is induced by many turns in small-scale environments, which makes the connectivity of the landmark tracks much weaker due to the frequent and large changes of the viewpoints. Contrarily, such driving environments are suitable for the proposed MVO, and consequently, the proposed method shows accurate metric-scale trajectories comparable to the ORB-stereo and LOAM.

In addition, an state-of-the-art open source lidar-inertial odometry and mapping (LIO-SAM) [4] is also implemented for comparisons. Different from the LOAM, the LIO-SAM makes a lidar point-feature map and re-localizes its position and pose by the loop closure module when a loop is detected. Furthermore, it utilizes a high-frequency IMU data to deal with the blinded epoch between lidar pointcloud scans. The author runs the LIO-SAM in the `bldg_39` and `bldg_220` datasets, and results are depicted in Figs. 4.20–4.21. In Fig. 4.20, resulting trajectories and maps with the loop closure functionality are highly distorted for both datasets. Because, in the underground parking lots, the appearances and 3-D structures of each floor are almost identical, the loop closure module wrongly merges the maps and yields wrong motion estimations. As seen in Fig. 4.20(a), the wrongly-detected loop connectivities are illustrated as the yellow lines, and wrongly merged trajectories after loop closing are shown in Fig. 4.20(b).

In Fig. 4.21, without the loop closure, the LIO-SAM shows similar performance to the original LOAM implementation because the odometry part of the LIO-SAM is based on the LOAM. Thus, the author only uses the LOAM trajectory for the performance comparisons instead of the LIO-SAM trajectory.

In Fig. 4.22, the fixed image region is additionally illustrated by the blue quadrilateral where the ground landmarks are likely to emerge. As seen in Fig. 4.22(a)–(b), off-planar features are included in the fixed region, and no planar landmark is detected in this region, which is not a favorable circumstance to the methods depending on the ground features.

Note that the proposed MVO method can recover the scale even in the non-flat ground of the inter-floor passages at the labels Ⓒ and Ⓖ of Fig. 4.17. This is because the proposed method does not depend on any specific feature distribution such as the flat ground features right in front of the camera assumed in the aforementioned plane-based methods.

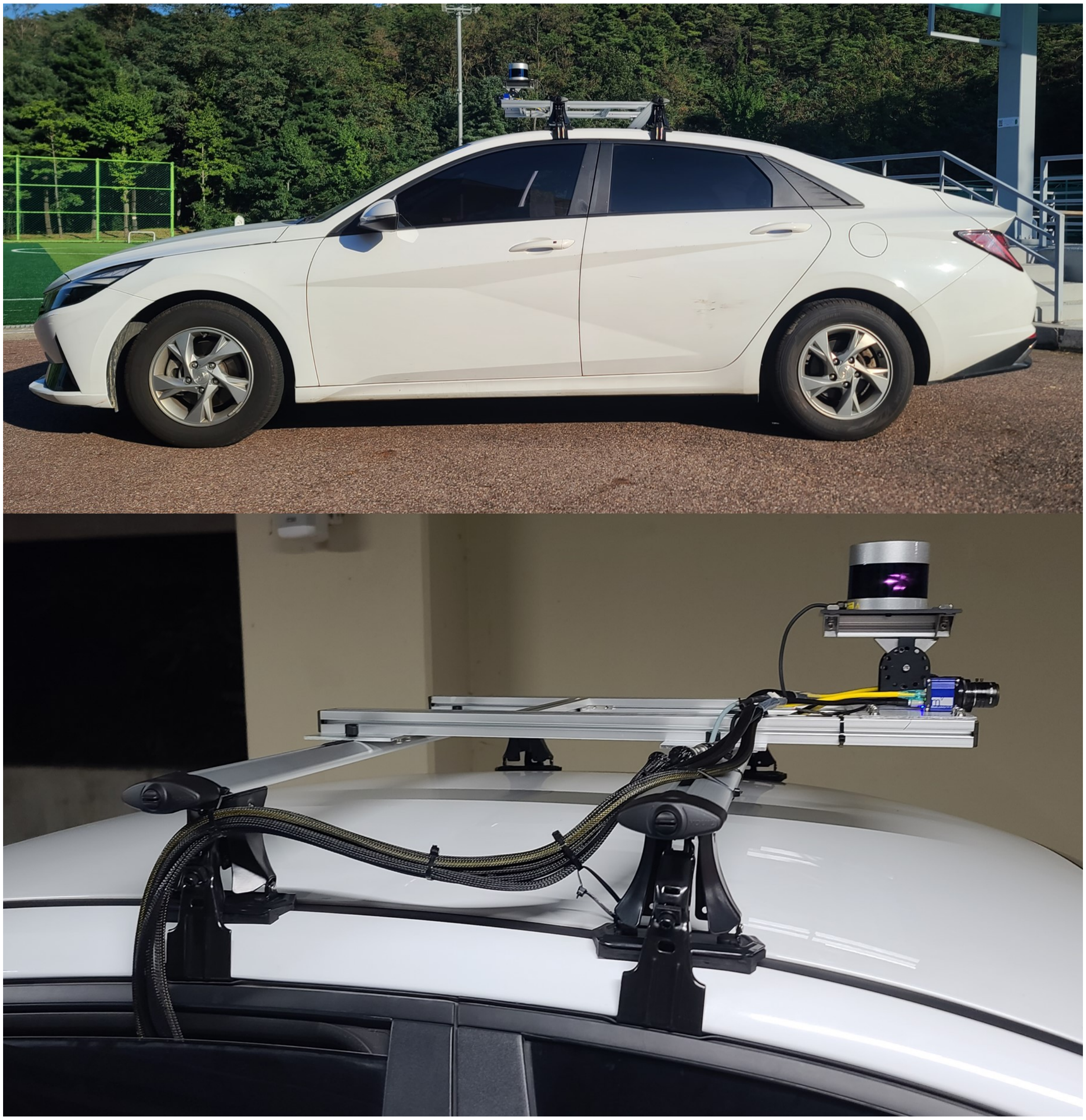


Figure 4.16: **Experimental setting for author-collected dataset - sideview**

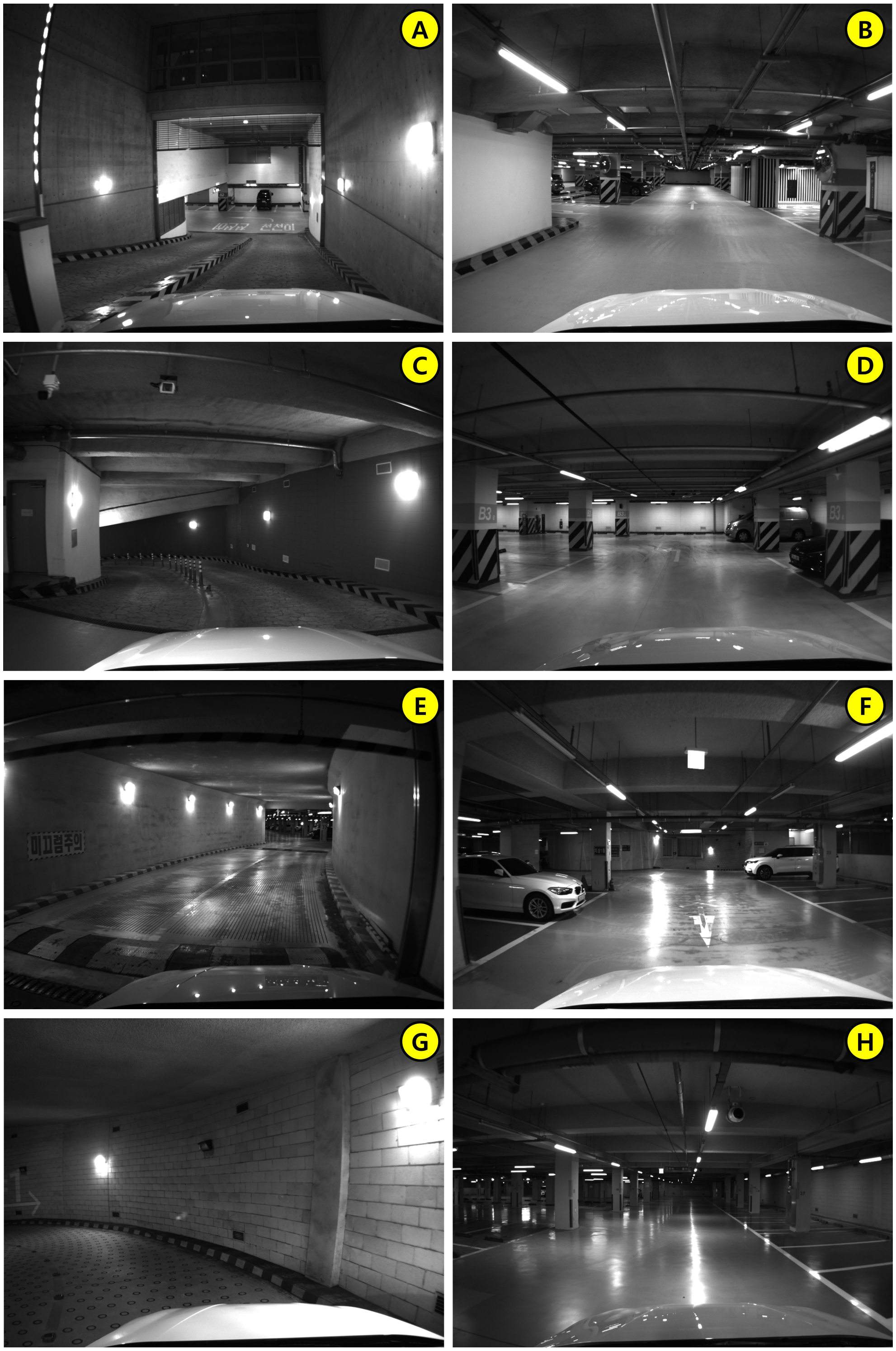


Figure 4.17: **Representative images of the author-collected SNU underground parking lots datasets.** Circled alphabets correspond to the locations marked the same symbol in Figs. 4.18 and 4.19.

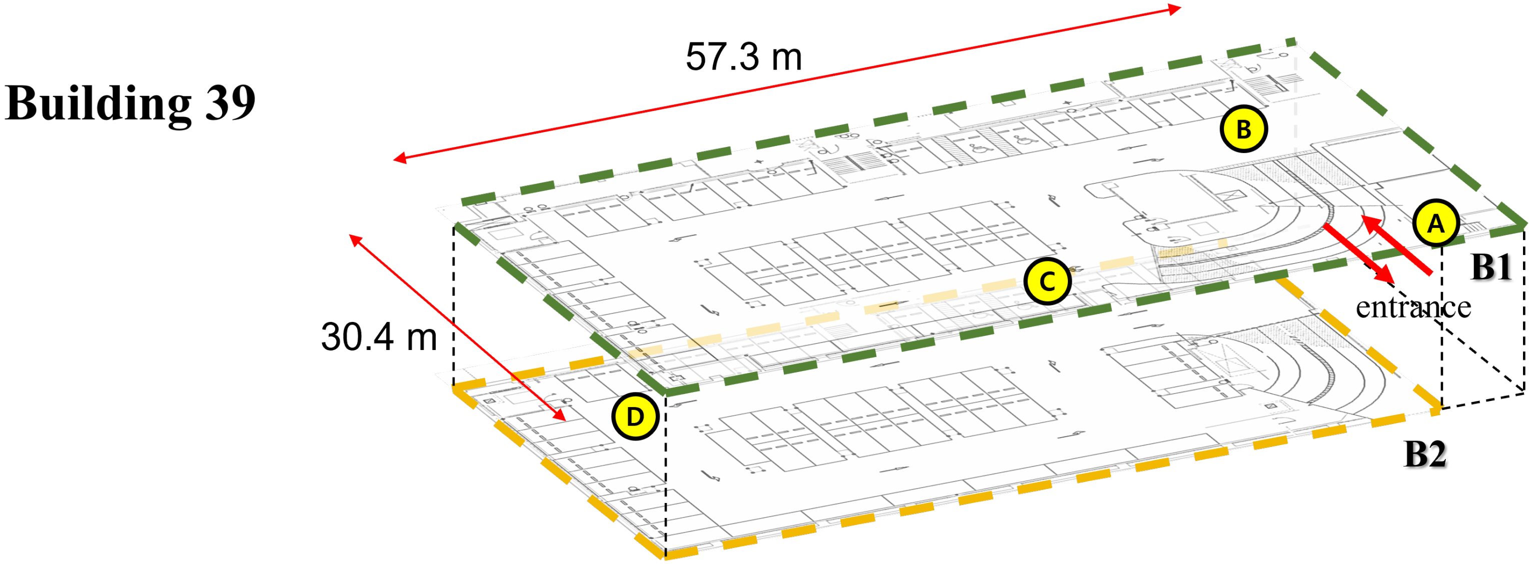


Figure 4.18: **Overviews of the author-collected datasets** `blgd_39` has two floors with the same shapes.

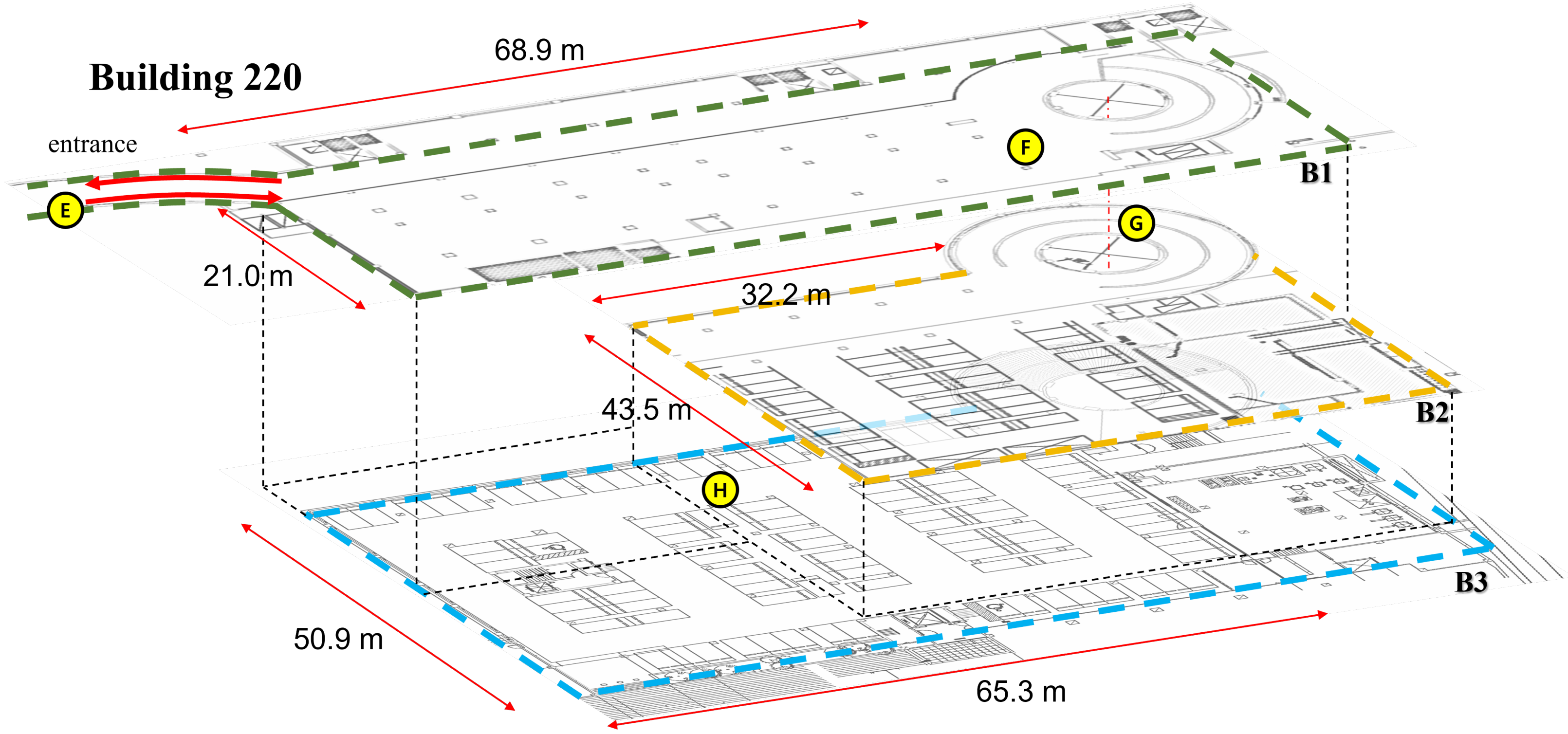


Figure 4.19: **Overviews of the author-collected datasets** `blgd_220` has three floors acros two buildings' underground spaces.

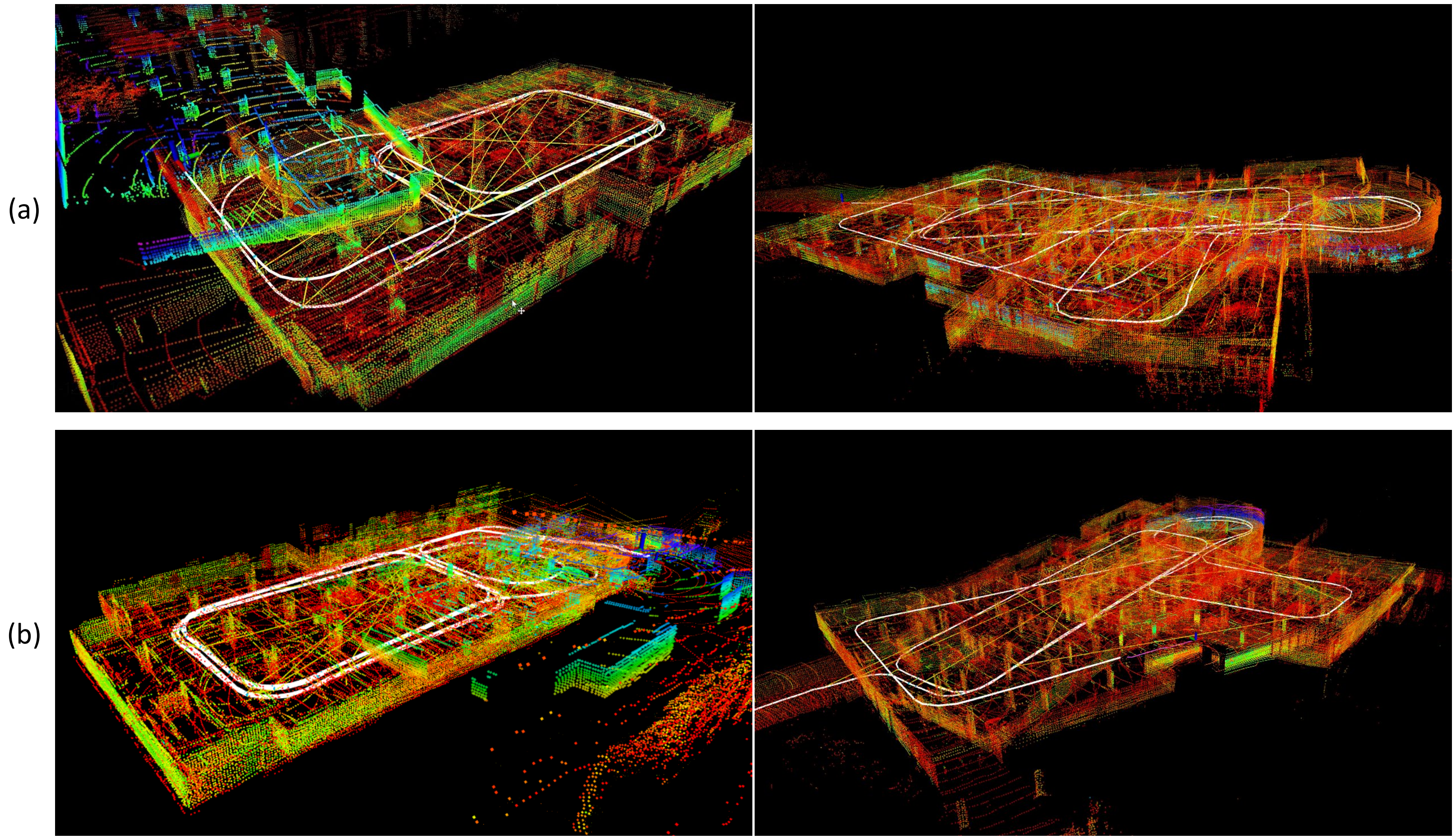


Figure 4.20: **LIO-SAM [4] results on the `bldg_39` and `bldg_220` datasets with the loop closure** The white line means the trajectory of the LiDAR, and the scattered points are generated by the LIO-SAM as the lidar point-feature map.

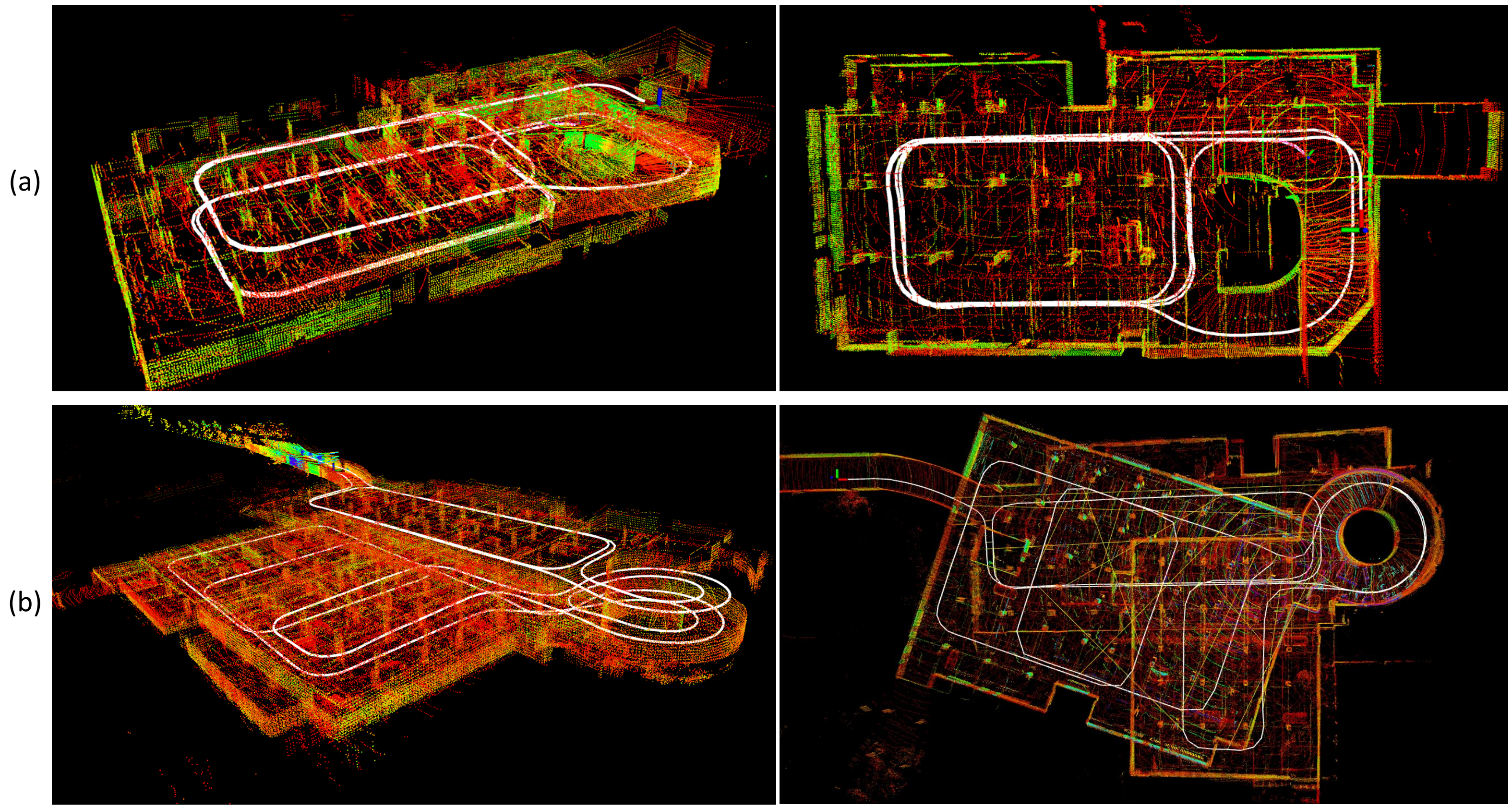


Figure 4.21: **LIO-SAM [4] results on the `bldg_39` and `bldg_220` datasets without the loop closure** The white line means the trajectory of the LiDAR, and the scattered points are generated by the LIO-SAM as the lidar point-feature map.

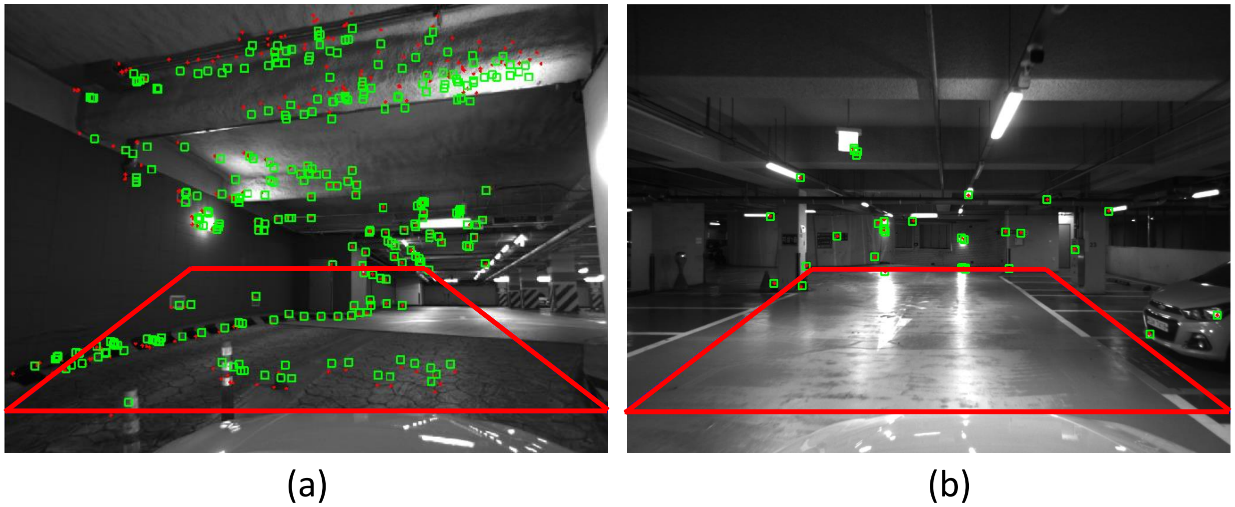


Figure 4.22: **Limitations of using the fixed image region for ground landmarks** The red quadrilateral region of interest (ROI) is commonly used as the ground plane region [5]. (a) In the non-flat passage, points on the pillar and slide are in the ROI. (b) No point is observed from the ground. The proposed MVO method does not require the assumption on the feature distribution such as ground points.

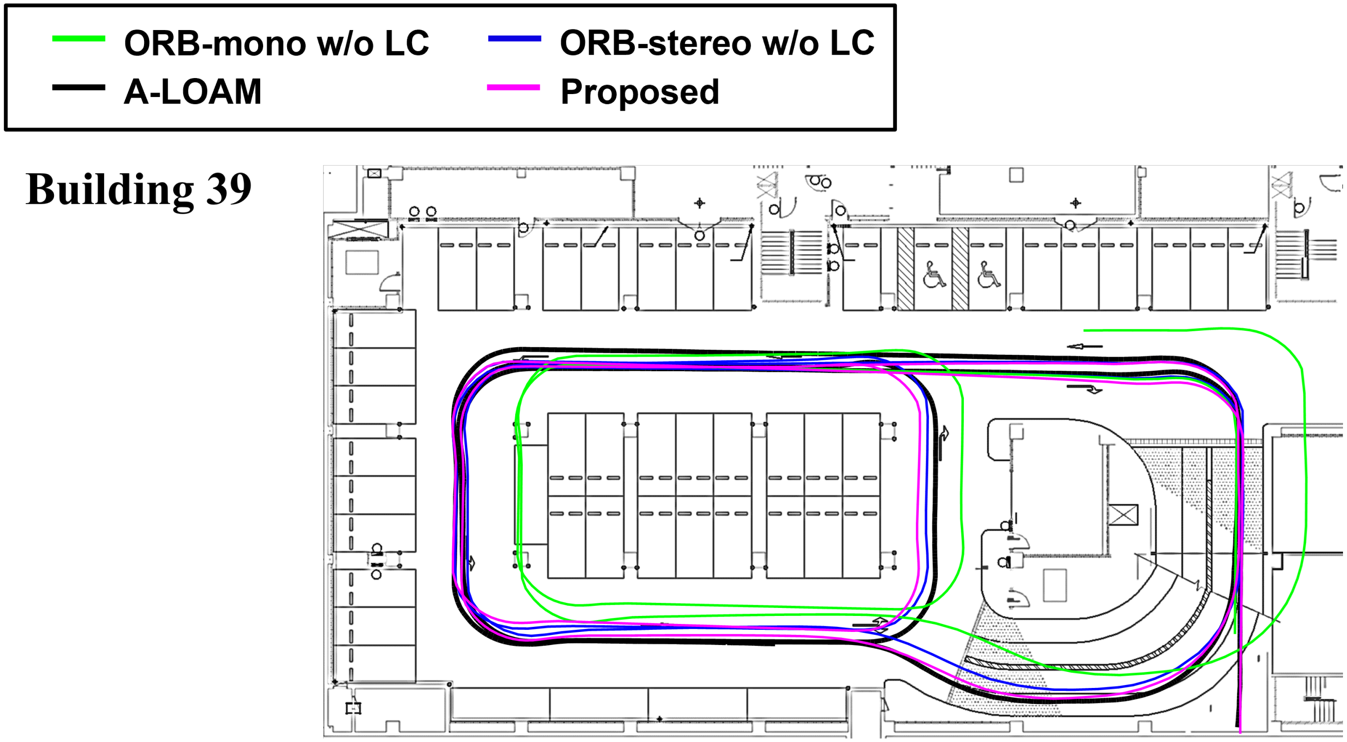


Figure 4.23: **Overall trajectories on `bldg_220` of the proposed method, ORB-SLAM with monocular and stereo settings, and the 3-D LiDAR odometry.**

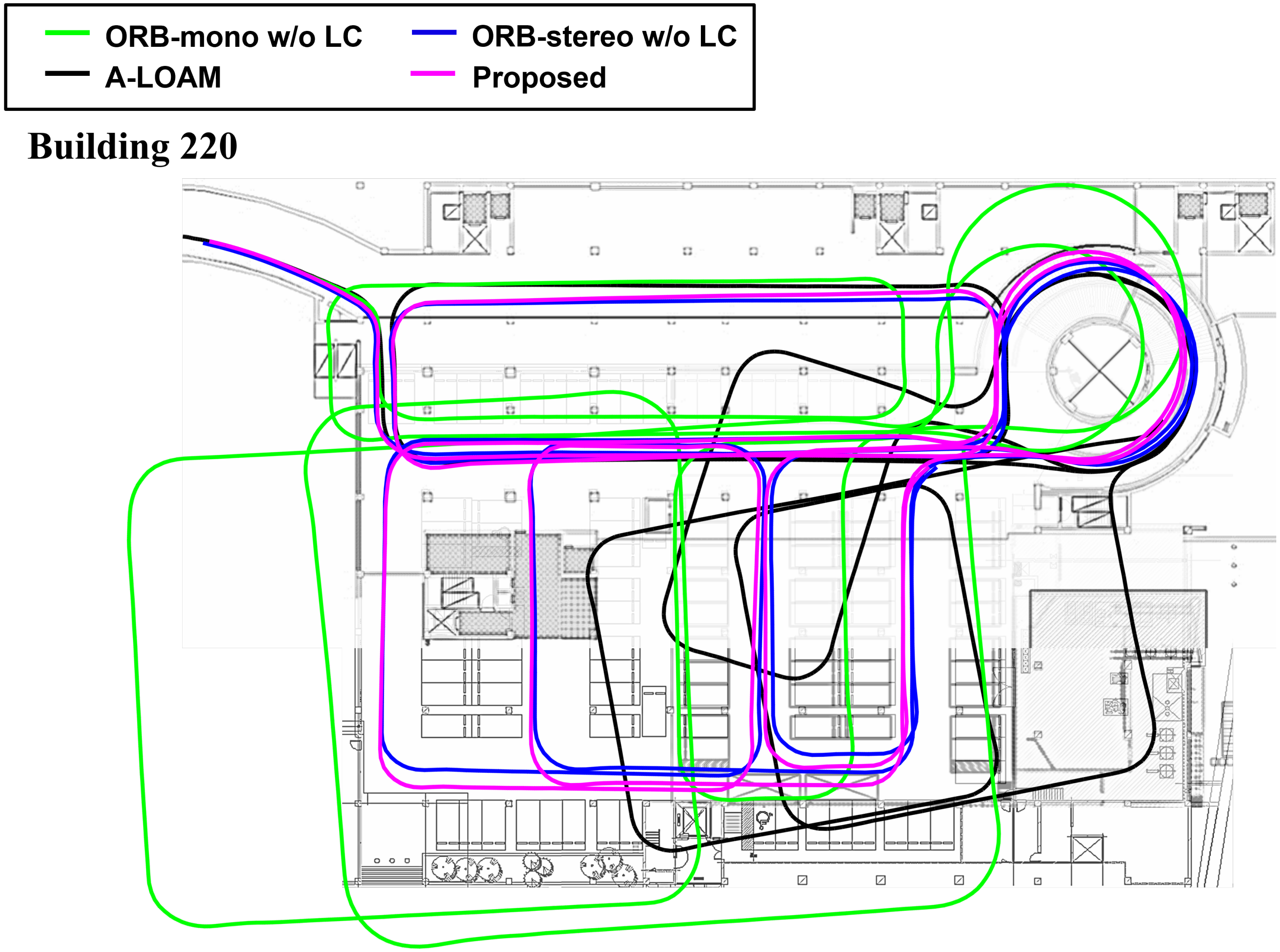


Figure 4.24: **Overall trajectories on `bldg_220` of the proposed method, ORB-SLAM with monocular and stereo settings, and the 3-D LiDAR odometry.**

# 5 Conclusion

This dissertation presented the robust visual odometry systems using image edges and points against low-textured and brightness changing conditions encountered in the indoor scenes, and the scale-aware monocular visual odometry backend using the vehicle kinematic motion model for the indoor driving applications.

In Chapter 2, the edge and point-based VO systems were proposed. Extracted edges were classified into eight overlapped subsets with respect to their image gradient directions. To mitigate high ambiguity on matching edges, they were segmented into eight orientation bins with shared regions between neighboring bins. In addition, the matching procedure was accelerated by multiple-quadtree structures memorizing visited nodes. To effectively reduce the large number of edge pixels observed in an image, hundreds of well-distributed prominent edgelets are only extracted from the tens of thousands of cluttered edge pixels. Camera motions were estimated by minimizing the point-to-edge distances, and the proposed method simultaneously updated edge inverse depths by the static and temporal stereo. For the hybrid setting using both edge and point features, the modified motion estimation problem was proposed by minimizing the point-to-edge distances, photometric

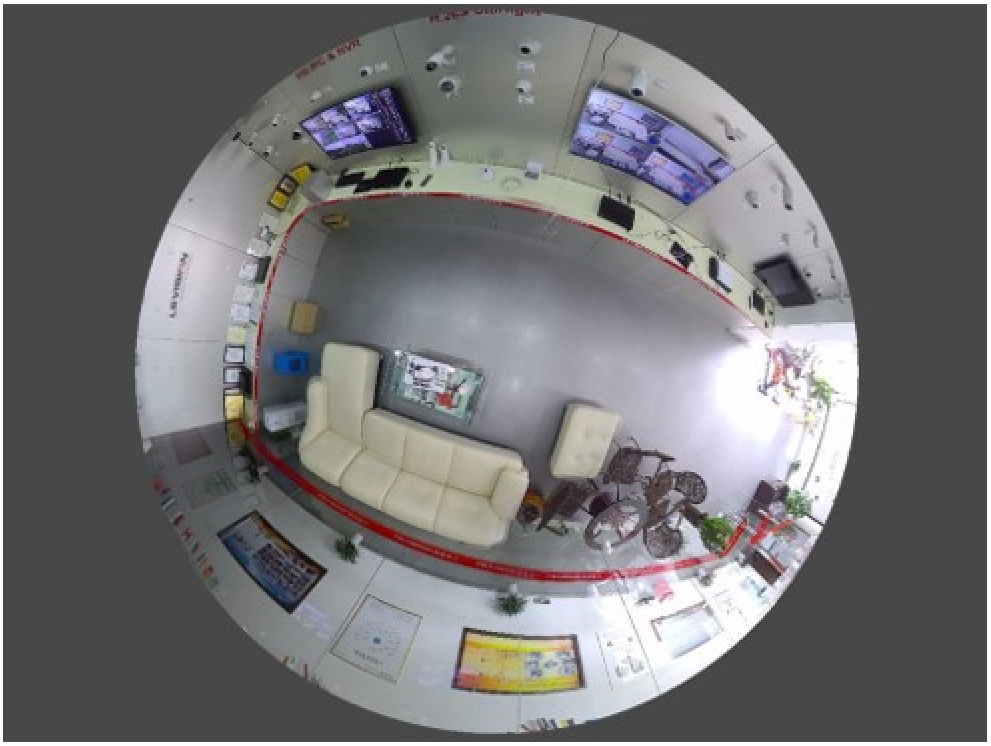


Figure 5.1: **Omnidirectional imagery in indoor environment** The 360 degrees view can be obtained by using the omnidirectional camera.

errors around the edge, and points with the affine brightness compensation. Then, the robust and accurate performance of the proposed method was demonstrated by using various datasets, including author-collected datasets with almost no texture and severe brightness changes.

In Chapter 3, the scale-aware monocular visual odometry system is proposed utilizing the vehicle kinematic constraint. The main idea of the proposed method was to utilize the vehicle kinematic motion model to observe the absolute metric scale in turning motions. To describe camera motions fixed to the vehicle, the camera-vehicle extrinsic pose was first estimated by the proposed extrinsic calibration method. To stably observe scale, the method was presented to detect turning frame regions, and the scale observer was formulated as the function of the camera rotation and the translation direction angles. By in-depth analysis on each proposed module and extensive experiments on driving datasets, it was shown that the proposed method could recover the absolute scale of camera translation motions with no external sensor and assumption on surrounding circumstances, such as planar ground landmarks.

The author suggests potential extensions of the proposed method; as reported in Section 4.2.1, the scale could not be propagated for long straight motions between turns. In this case, other scale-aware methods like a plane-based method [54] could be more effective. Therefore, combining the other methods and the proposed method will be promising work. Also, the author suggests using an omnidirectional camera as [41] because landmarks can be tracked over 360 degrees turning motions as seen in Fig. 5.1, which gives stronger connectivities among landmarks and keyframes. Furthermore, there might be no need to update new keyframes frequently compared to a pinhole camera set, and a larger rotation angle is guaranteed, enabling more stable scale observations.

# References

[1] Y. Zhou, H. Li, and L. Kneip, “Canny-vo: Visual odometry with rgb-d cameras based on geometric 3-d–2-d edge alignment,” *IEEE Transactions on Robotics*, vol. 35, no. 1, pp. 184–199, 2019.

[2] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the kitti vision benchmark suite,” in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 3354–3361.

[3] A. Handa, T. Whelan, J. McDonald, and A. Davison, “A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM,” in *IEEE Intl. Conf. on Robotics and Automation, ICRA*, Hong Kong, China, May 2014.

[4] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, “Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2020, pp. 5135–5142.

[5] S. Song, M. Chandraker, and C. C. Guest, “High accuracy monocular sfm and scale correction for autonomous driving,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 4, pp. 730–743, Apr. 2016.

[6] D. Nister, O. Naroditsky, and J. Bergen, “Visual odometry,” in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 1, 2004, pp. I–I.

[7] 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.*

Cham: Springer International Publishing, 2017, pp. 235–252. [Online]. Available: https://doi.org/10.1007/978-3-319-29363-9_14

[8] G. Klein and D. Murray, "Parallel tracking and mapping on a camera phone," in *2009 8th IEEE International Symposium on Mixed and Augmented Reality*. IEEE, 2009, pp. 83–86.

[9] 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.

[10] C. Forster, M. Pizzoli, and D. Scaramuzza, "Svo: Fast semi-direct monocular visual odometry," in *2014 IEEE international conference on robotics and automation* (*ICRA*). IEEE, 2014, pp. 15–22.

[11] 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*, vol. 33, no. 5, pp. 1255–1262, 2017.

[12] A. Geiger, J. Ziegler, and C. Stiller, "Stereoscan: Dense 3d reconstruction in real-time," in *2011 IEEE Intelligent Vehicles Symposium* (*IV*), 2011, pp. 963–968.

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

[14] C. Kerl, J. Sturm, and D. Cremers, "Robust odometry estimation for rgb-d cameras," in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 3748–3754.

[15] D. Scaramuzza and F. Fraundorfer, "Visual odometry [tutorial]," *IEEE robotics & automation magazine*, vol. 18, no. 4, pp. 80–92, 2011.

[16] J. Engel, J. Sturm, and D. Cremers, "Semi-dense visual odometry for a monocular camera," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 1449–1456.

[17] J. Engel, J. Stückler, and D. Cremers, "Large-scale direct slam with stereo cameras," in *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2015, pp. 1935–1942.

[18] 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 *2011 10th IEEE international symposium on mixed and augmented reality*. Ieee, 2011, pp. 127–136.

[19] J. Witt and U. Weltin, "Robust stereo visual odometry using iterative closest multiple lines," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 4164–4171.

[20] I. Dryanovski, R. G. Valenti, and J. Xiao, "Fast visual odometry and mapping from rgb-d data," in *2013 IEEE international conference on robotics and automation*. IEEE, 2013, pp. 2305–2310.

[21] S. Li and D. Lee, "Fast visual odometry using intensity-assisted iterative closest point," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 992–999, 2016.

[22] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 834–849.

[23] R. Gomez-Ojeda, J. Briales, and J. Gonzalez-Jimenez, "Pl-svo: Semi-direct monocular visual odometry by combining points and line segments," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 4211–4216.

[24] S. Yang and S. Scherer, "Direct monocular odometry using points and lines," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 3871–3877.

[25] X. Wang, W. Dong, M. Zhou, R. Li, and H. Zha, "Edge enhanced direct visual odometry." in *BMVC*, 2016.

[26] C. Kim, P. Kim, S. Lee, and H. J. Kim, "Edge-based robust rgb-d visual odometry using 2-d edge divergence minimization," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1–9.

[27] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.

[28] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International conference on computer vision*. Ieee, 2011, pp. 2564–2571.

[29] L. Zhang and R. Koch, "An efficient and robust line segment matching approach based on lbd descriptor and pairwise geometric consistency," *Journal of Visual Communication and Image Representation*, vol. 24, no. 7, pp. 794–805, 2013.

[30] C. Kim, J. Kim, and H. J. Kim, "Edge-based visual odometry with stereo cameras using multiple oriented quadtrees," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 5917–5924.

[31] C. Kim, S. Lee, and H. J. Kim, "Hybrid rgb-d visual odometry combining edges and point features," in *2020 the 35th ICROS annual conference (ICROS 2020)*, 2020.

[32] Y. Lu and D. Song, "Robust rgb-d odometry using point and line features," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3934–3942.

[33] M. Tomono, “Robust 3d slam with a stereo camera based on an edge-point icp algorithm,” in *2009 IEEE international conference on robotics and automation*. IEEE, 2009, pp. 4306–4311.

[34] M. Kuse and S. Shen, “Robust camera motion estimation using direct edge alignment and sub-gradient method,” in *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2016, pp. 573–579.

[35] J. J. Tarrio and S. Pedre, “Realtime edge-based visual odometry for a monocular camera,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[36] Y. Zhou, L. Kneip, and H. Li, “Semi-dense visual odometry for rgb-d cameras using approximate nearest neighbour fields,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 6261–6268.

[37] 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.

[38] 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.

[39] J. Engel, V. Koltun, and D. Cremers, “Direct sparse odometry,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 3, pp. 611–625, 2018.

[40] R. Wang, M. Schworer, and D. Cremers, “Stereo dso: Large-scale direct sparse visual odometry with stereo cameras,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[41] C. Won, H. Seok, Z. Cui, M. Pollefeys, and J. Lim, “Omnislam: Omnidirectional localization and dense mapping for wide-baseline multi-camera systems,” in *IEEE International Conference on Robotics and Automation*, May 2020, pp. 559–566.

[42] S. Heo, J. Cha, and C. G. Park, “Ekf-based visual inertial navigation using sliding window nonlinear optimization,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 7, pp. 2470–2479, July 2019.

[43] T. Qin, P. Li, and S. Shen, “Vins-mono: A robust and versatile monocular visual-inertial state estimator,” *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[44] D. Scaramuzza, “1-point-ransac structure from motion for vehicle-mounted cameras by exploiting non-holonomic constraints,” *International journal of computer vision*, vol. 95, no. 1, pp. 74–85, 2011.

[45] R. Kang, L. Xiong, M. Xu, J. Zhao, and P. Zhang, “Vins-vehicle: A tightly-coupled vehicle dynamics extension to visual-inertial state estimator,” in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 3593–3600.

[46] J. H. Jung, J. Cha, J. Y. Chung, T. I. Kim, M. H. Seo, S. Y. Park, J. Y. Yeo, and C. G. Park, “Monocular visual-inertial-wheel odometry using low-grade imu in urban areas,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 2, pp. 925–938, Feb. 2022.

[47] G. Nützi, S. Weiss, D. Scaramuzza, and R. Siegwart, “Fusion of imu and vision for absolute scale estimation in monocular slam,” *Journal of intelligent & robotic systems*, vol. 61, no. 1, pp. 287–299, 2011.

[48] S. Chiodini, R. Giubilato, M. Pertile, and S. Debei, “Retrieving scale on monocular visual odometry using low-resolution range sensors,” *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 8, pp. 5875–5889, 2020.

[49] B. Ölmez and T. E. Tuncer, “Metric scale and angle estimation in monocular visual odometry with multiple distance sensors,” *Digital Signal Processing*, vol. 117, p. 103148, 2021.

[50] B. M. Kitt, J. Rehder, A. D. Chambers, M. Schonbein, H. Lategahn, and S. Singh, "Monocular visual odometry using a planar road model to solve scale ambiguity," 2011.

[51] N. Fanani, A. Stürck, M. Barnada, and R. Mester, "Multimodal scale estimation for monocular visual odometry," in *2017 IEEE intelligent vehicles symposium (IV)*. IEEE, 2017, pp. 1714–1721.

[52] X. Wang, H. Zhang, X. Yin, M. Du, and Q. Chen, "Monocular visual odometry scale recovery using geometrical constraint," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 988–995.

[53] M. Fan, S.-W. Kim, S.-T. Kim, J.-Y. Sun, and S.-J. Ko, "Simple but effective scale estimation for monocular visual odometry in road driving scenarios," *IEEE Access*, vol. 8, pp. 175 891–175 903, 2020.

[54] D. Zhou, Y. Dai, and H. Li, "Ground-plane-based absolute scale estimation for monocular visual odometry," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 2, pp. 791–802, 2020.

[55] R. Tian, Y. Zhang, D. Zhu, S. Liang, S. Coleman, and D. Kerr, "Accurate and robust scale recovery for monocular visual odometry based on plane geometry," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 5296–5302.

[56] H. Zhang, X. Wang, X. Yin, M. Du, C. Liu, and Q. Chen, "Geometry-constrained scale estimation for monocular visual odometry," *IEEE Transactions on Multimedia*, 2021.

[57] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[58] C. Godard, O. Mac Aodha, and G. J. Brostow, “Unsupervised monocular depth estimation with left-right consistency,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 270–279.

[59] S. Wang, R. Clark, H. Wen, and N. Trigoni, “Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks,” in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 2043–2050.

[60] K. Tateno, F. Tombari, I. Laina, and N. Navab, “Cnn-slam: Real-time dense monocular slam with learned depth prediction,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 6243–6252.

[61] N. Yang, R. Wang, J. Stuckler, and D. Cremers, “Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 817–833.

[62] Q. Liu, R. Li, H. Hu, and D. Gu, “Using unsupervised deep learning technique for monocular visual odometry,” *Ieee Access*, vol. 7, pp. 18 076–18 088, 2019.

[63] S. Jia, X. Pei, X. Jing, and D. Yao, “Self-supervised 3d reconstruction and ego-motion estimation via on-board monocular video,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 7557–7569, July 2022.

[64] C. Campos and J. D. Tardós, “Scale-aware direct monocular odometry,” 2022.

[65] C. Chen, S. Rosa, Y. Miao, C. X. Lu, W. Wu, A. Markham, and N. Trigoni, “Selective sensor fusion for neural visual-inertial odometry,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 542–10 551.

[66] L. Han, Y. Lin, G. Du, and S. Lian, “Deepvio: Self-supervised deep learning of monocular visual inertial odometry using 3d geometric constraints,” in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 6906–6913.

[67] M. Abolfazli Esfahani, H. Wang, K. Wu, and S. Yuan, "Aboldeepio: A novel deep inertial odometry network for autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 5, pp. 1941–1950, May 2020.

[68] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to autonomous mobile robots*. MIT press, 2011.

[69] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "Casadi: a software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.

[70] S. Wright, J. Nocedal *et al.*, "Numerical optimization," *Springer Science*, vol. 35, no. 67-68, p. 7, 1999.

[71] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2012, pp. 573–580.

[72] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[73] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time." in *Robotics: Science and Systems*, vol. 2, no. 9. Berkeley, CA, 2014, pp. 1–9.

[74] J. Kim, C. Kim, Y. Han, and H. J. Kim, "Automated extrinsic calibration for 3d lidars with range offset correction using an arbitrary planar board," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 5082–5088.

# 국문초록

항법 정보는 자율 자동차, 드론 등 자율 로봇의 위치를 알기위한 필수적인 요소이다. 특히, 소형의 카메라만을 이용해 항법을 수행하는 영상 기반 항법은 GPS 등 외부 항법정보를 이용할 수 없는 실내 환경에서의 대안으로 각광받고 있다. 본 논문은 실내 환경에서 마주할 수 있는 무늬 부족 환경 및 밝기 변화 환경에서도 안정적으로 동작 할 수 있는 강인한 영상 항법을 제안한다. 그리고 실내 영상 항법의 실용적인 활용처로써, 차량에서 볼 수 있는 최소한의 센서 세팅인 단안 카메라만을 이용한 실내 주행 항법 상황을 고려하였다. 단안 항법의 스케일 모호성 문제를 해결하기위해 차량의 기구학을 이용한 절대 스케일 복원 방법을 제안하였고, 차량 관점의 항법이 가능하도록 카메라-차량 간 외부 자세 보정법을 제안한다.

첫 번째로 무늬 부족 및 밝기 변화 환경에서도 구동 가능한 이미지 모서리와 점 특징 기반의 강인한 영상 항법을 제안한다. 이미지 모서리를 이미지 기울기 방향에 따라 분류하여 다중 쿼드트리 구조로 효과적으로 매칭하였으며, 모서리와 점 특징의 재사영 오차, 광도 오차를 최소화하며 이미지 간 밝기 변화를 동시에 보상하는 하이브리드 영상 항법을 구현하였다. 다양한 시뮬레이션 분석을 통해 제안한 각 모듈의 성능을 평가하였으며, 무늬가 적은 데이터 셋에 예측 불가능한 빛 변화를 인가한 뒤, 최신 성능 알고리즘과의 비교를 통해 전체적인 항법 성능을 검증하였다. 또한, 일부 곡선 요소 외에는 두드러진 무늬가 없는 실제 실내 사무실 및 복도 환경에 대한 데이터셋을 취득하였고, 제안하는 영상 항법이 실제 환경에서도 안정적으로 동작 할 수 있음을 검증하였다.

다음으로, 차량 기구학을 제약된 카메라 움직임을 이용하여 단안 항법 스케일을 인식하는 방법을 다룬다. 우선, 차량에 부착된 카메라의 움직임을 기술하기 위해 카메라-차량 간 외부 자세를 추정하는 방법을 제안한다. 회전 영역에서 단안 항법의 절대 스케일을 관측하는 스케일 관측기를 설계하고, 스케일을 안정적으로 관측하기 위해 회전 영역을 감지하는 방법을 제안한다. 그리고 관측된 스케일을 사용하여, 회전 영역 사이의 관측되지 않은 전체 스케일을 추정하는 절대 스케일 복구 방법을 제안한다. 성능 검증을 위해서, 몬테카를로

시뮬레이션을 수행하여 각 모듈의 성능을 통계적으로 평가하였고, 공개된 차량 주행 데이터셋을 이용해 제안하는 방법론과 최신 성능 알고리즘과의 성능 비교 평가를 수행하였다. 추가로, 실내 차량 주행환경에 대해 제안하는 방법론의 유망한 적용 가능성을 보여주기 위해, 두 개의 다층 지하 주차장에서 실제 차량 주행 영상 데이터를 수집하였고, 해당 데이터에 대한 시연을 통해 실제 상황에서의 정확한 절대 스케일 복원 성능을 검증하였다.

**주요어:** 강인한 영상 항법, 단안 스케일 복원, 외부자세 보정, 이미지 모서리와 특징점, 차량 기구학

**학 번:** 2018-31816