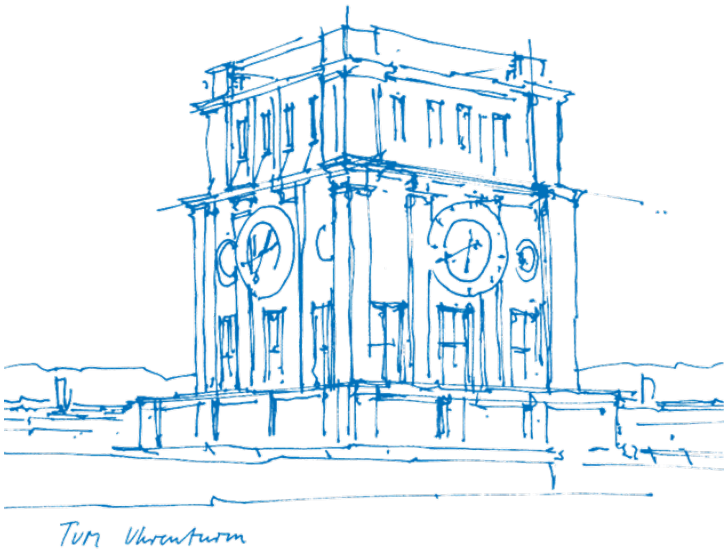
Computer Vision Group
Department of Informatics
Technical University of Munich

Integration of Deep Optical Flow in Visual-Inertial Odometry

Jingkun Feng



Computer Vision Group
Department of Informatics
Technical University of Munich

Integration of Deep Optical Flow in Visual-Inertial Odometry

Jingkun Feng

Semester Thesis

at the Department of Mechanical Engineering of the Technical University of Munich.

Examiner:
Prof. Dr. Daniel Cremers

Supervisor:
Mariia Gladkova

Submitted:
Freising, January 6, 2022

I hereby declare that this thesis is entirely the result of my own work except where otherwise indicated. I have only used the resources given in the list of references.

Freising, January 6, 2022 Jingkun Feng

Abstract

Optical flow is a useful technique in visual Simultaneous Localization and Mapping (SLAM). Handcrafted optical flow has been widely used in the front-end of visual SLAM to track feature points for a long period. Recently deep-learning-based optical flow estimation has been shown as a promising alternative to classic methods. This semester thesis aims to explore the possibility to improve the accuracy and robustness of visual-inertial odometry by integrating optical flow inferred using neural networks in the Basalt [1]. Basalt is a stereo visual SLAM system consists of visual-inertial odometry (VIO) and visual-inertial mapping. The VIO subsystem employs sparse optical flow based on KLT for feature tracking. We propose to replace the adapted pyramidal KLT tracker in Basalt with optical flow which is inferred using per-trained model. To evaluate the presented system we conduct evaluation on the KITTI odometry dataset [2] and the EuRoC MAV dataset [3]. The former one contains only images, while the latter one additionally provides cues from inertial measurement unit (IMU). We leverage two neural network frameworks and their pretrained models, i.e. LiteFlowNet [4] and RAFT [5] to infer the optical flow forwards and backwards for image sequences captured by stereo cameras. We implement a two-stage approach which computes the forward-backward-inconsistency of the flow vector as well as the epipolar constraint to remove outliers. The result of the integrated system shows improved accuracy in tracking and pose estimation in most of the evaluated sequences. However, the system is not real-time capable due to the long inference time of the neural networks.

Contents

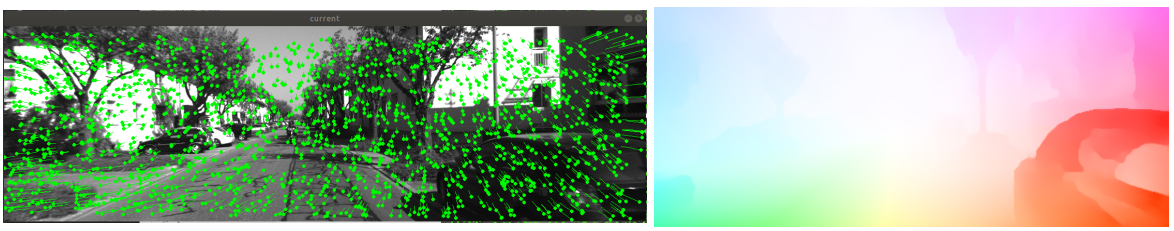
1 Introduction

Visual odometry (VO) and its extension, visual-inertial odometry (VIO) are popular research topics for motion tracking in computer vision and robotics domains. Recently neural network has become the method of choice in a variety of computer vision tasks. Deep-learning-based optical estimation methods and visual odometry systems have received significant attention from robotics and computer vision community. A considerable number of systems utilizing neural networks for both optical flow estimation and pose estimation have been proposed in the past five years. However, the visual odometry which combines deep optical flow and traditional pose estimation method is still out of the spotlight. The main idea of this thesis is inspired by DF-VO [6]. We aim to explore the possibility of replacing classic methods for optical flow estimation with deep-learning-based methods in visual odometry task by integrating optical flow inferred by neural networks (deep optical flow) for feature tracking in a visual-inertial odometry system.

VO is the process of estimating the ego-motion merely from visual data, i.e. images and VIO is its extension which additionally make use of IMU data. There are a large variety of methods for solving VO and these methods can be roughly divided into feature-based methods and direct methods [7]. Feature-based VO works by extracting a sparse set of feature points and tracking them in sequence of frames. Tracking of these feature points has been solved by calculating the descriptors which can uniquely represent a certain point and matching the descriptors between images for a long period. Researcher soon realized this solution is very inefficient, since the extraction of key points and the calculation of descriptors are very time consuming. As a transition from traditional feature-based methods to direct methods, the optical flow has taken the stage.

Optical flow is the apparent motion on an image plane caused by the relative motion between an observer and the scene [8]. It is manifested as the per-pixel motion between consecutive video frames as shown in Figure 1.1, telling us where in the adjacent image we can possibly find a particular pixel which appears in the current image. Therefore, it can be used in VO to tracked the motion of the keypoints, while the descriptors are discarded. This can speed up the VO, since the time taken by calculation optical flow itself is less than the computing and the matching of descriptors.

Traditionally the estimation of optical flow is treated as a hand-crafted optimization problem. To solve this problem it is assumed that the observed brightness of any object point is constant over time and nearby points in the image plane move in a similar manner [9] [10]. These traditional approaches, however, strongly rely on tight continuity of image frames and robust illumination condition, which is not always the case in practice.



(a) Sparse optical flow. **(b)** Dense optical flow.

Figure 1.1 Optical flow. (a) illustrates sparse optical flow which is computed by a pyramidal KLT tracker. (b) shows dense optical flow estimated by the LiteFlowNet and color coded with a method introduce by Baker et al. [11].

A breakthrough for these limitations appears with the emergence of many novel deep learning based methods for optical flow estimation. They show comparable or even superior performance than classical methods on the benchmarks such as KITTI [2] and MPI-Sintel [12]. In contrast to traditional methods for

optical flow estimation, deep learning can avoid formulating the estimation as an optimization problem and train a network to directly predict flow [5].

Although geometry based VO has dominated for a long period providing reliable accuracy, more and more researchers are turning their attention to solving the VO problem with neural networks [7]. End-to-end learning-based approaches have been shown being an option to perform visual odometry [13] [14] [15]. However, there are only a few works in the direction of combining deep optical flow and geometry based VO, which is the motivation of our work.

The main contribution of this thesis is exploring the possibility and advantage of using optical flow estimated by deep-learning-based methods for camera tracking in VO and VIO systems. For this purpose, we propose a deep optical flow integrated visual-inertial odometry system, which is built upon a state-of-the-art system, the Basalt VIO[1]. For flow estimation, deep-learning-based methods are employed and the estimates are fed into the system for feature tracking, outlier removal and camera pose estimation. We warp the extracted key points with the forward optical flow in subsequent frame. To remove incorrectly tracked points, we measure the forward-backward-inconsistency of optical flow vectors. A further stage for outlier filtering is completed by constraining the epipolar geometry of the correspondences in stereo image pairs under an absolute threshold. For pose estimation we retain the hierarchical approach in Basalt, which outperforms many other state-of-the-art methods with respect to trajectory accuracy. An overview of the original visual-inertial odometry subsystem of Basalt is provided in section 3.3.

The subsequent parts of this thesis is structured as following: In chapter 2, it is given an overview of the related work in optical flow estimation and visual odometry. In chapter 3, we introduce some required background knowledge, including the main concept of optical flow and visual-inertial odometry, as well as the involved building blocks of our project, i.e. Basalt VIO, LiteFlowNet and RAFT. The integration of deep optical flow and the approaches for outlier removal are presented in chapter 4. The evaluation of the proposed system with integrated deep optical flow will be shown in chapter 5. Afterwards, we present the result of the ablation studies and discuss the contribution of this project as well as interesting directions for future research in chapter 6. Finally, we summarize this thesis with a conclusion.

2 Related Work

2.1 Optical Flow

Traditionally, optical flow estimation has been considered as an energy minimization problem which normally consists of a data term and a regularization term. Classical methods for optical flow estimation can be split into two main groups, sparse and dense optical flow (s. Figure 1.1). The well known Lukas-Kanade optical flow [10] is a classic example of the former one, while Horn and Schunck [9] proposed a gradient approach for estimating a smooth flow field as a continuous optimization problem. In many works, the optimization problem is been solved in a coarse-to-fine framework [16] [17]. This class of methods struggle from the complex energy optimization problem and is not feasible for real-time applications [18].

Recent approaches for optical flow estimation tend to apply deep learning techniques, which have shown impressive results. FlowNet [19] and FlowNet2 [20] are pioneers in using CNN for optical flow estimation. They directly predict a dense flow field of the input image pairs, side-stepping the optimization problem. Similar to variational solutions, many approaches also employ a coarse-to-fine processing to predict large displacement [4] [21] [22]. Moreover, iterative refinement has emerged as popular approaches in many recent works [23] [24] to improve flow estimation results. In contrast, RAFT [5] conducts flow field prediction while maintaining high-resolution of the images. Approaches like variants of FlowNet [19] [20] [4], SPyNet [22], PWC-Net [21], RAFT [5] and more each outperforming one another on various benchmarks show competitive or even superior performance than traditional methods.

Computing optical flow with deep neural networks requires large amount of training data. However, they are particularly hard to obtain, because labeling optical flow in video footage requires sub-pixel accurate understand of exact motion of every points on images. To solve this issue, some synthetic dataset are generated by simulating massive realistic worlds with computer graphics techniques. As one example MPI-Sintel [12] [25] is an open source computer generated imaginary movie with optical flow labeling rendered. Another widely used example is FlyingChairs [19] and its successor [26], a dataset of many chairs flying on random backgrounds. Moreover, the dataset called FlyingThings3D [27] which consists of daily objects flying along randomized 3D trajectories attracts considerable attention of researchers. With wide-ranging attention on self-driving vehicles, KITTI Flow 2012/2015 [2] is the most popular dataset. However, it focuses on a particular task and hence has a very restricted set of motion patterns. Other than that, it contains merely 394 image pairs for training including 194 pairs in KITTI Flow 2012 and 200 pairs in KITTI Flow 2015, which makes the models only trained on KITTI have bad generalization.

2.2 Visual Odometry and Visual-Inertial-Odometry

Visual odometry as crucial components in visual SLAM, have been a popular topic in robotics community for a long period. It mainly solves the task of estimating camera pose by taking cues from the images. When additionally information from the inertial measurement unit (IMU) is used, it is referred to visual-inertial odometry (VIO).

According to the camera setup, VO can be categorized as monocular VO which makes use of a single camera, and stereo VO if two cameras in stereo setup are involved. Depending on how visual information is obtained and used, approaches of VO can be split into feature-based (indirect) methods and direct methods [28]. The feature-based methods extract and track feature points in image sequence, while direct method estimates ego-motion based on the photometric information. Feature-based VO is always popular and never lacks of contributions from researchers. VISO2 [29] and ORB-SLAM2 [30] are both successful examples which apply feature-based approaches. In works such as [31], optical flow is used to track the

motion of the detected feature points. Thanks to that, the computation of descriptors is discarded and the computational effort is reduced. Different from indirect methods, calculation of keypoints and descriptors can be completely discarded by direct methods, which results in lower computational effort and solves the problems caused by featureless scenes, e.g. a white wall [32]. Direct methods become more and more important with the emergence of projects such as LSD-SLAM [33], SVO [34], DSO [35] and etc.

Similar to optical flow estimation and many other computer vision tasks, a considerable number of deep learning based VO systems have been proposed in the last five years. In 2017, Wang et al. [36] introduce a recurrent network to learn VO from videos. Zhou et al. [37] propose the first self-supervised approach for jointly learning depth estimation and camera motion. Yang et al. [13] feed depth predictions into DSO [35]. Yang et al. [14] proposes a three-level framework for monocular VO that combines deep depth, pose and uncertainty estimation. DF-VO [6], in contrast to the above methods, effectively combines a CNN for depth estimation and another for optical estimation to create a simple VO system based on standard multi-view geometry, which is the essential inspiration of this thesis.

3 Preliminaries

In the first part of this section, we revisit some background knowledge required for understanding this research topic, including classic variational methods for feature tracking and optical flow estimation, and the main concept of visual odometry. In the second part, we briefly introduce the frameworks on which this project is built, including a visual-inertial SLAM framework, Basalt, and two deep learning frameworks for estimating optical flow, LiteFlowNet and RAFT.

In the following parts of this thesis, we denote vectors with bold lowercase (e.g. $\boldsymbol{v}$) and matrices as with bold capital letter (e.g. $\mathbf{R}$). The rigid body motion in 3D is denoted by $\mathbf{T} \in SE(3)$ consisting of rotation $\mathbf{R} \in SO(3)$ and translation $\mathbf{t} \in \mathbb{R}^3$. The notation $[\mathbf{R}, \mathbf{t}]$ is equivalent to $\mathbf{T}$.

3.1 Feature Tracking and Optical Flow

Optical flow is defined as the apparent motion of image brightness patterns in an image sequence [10]. It describes per-pixel motion in video sequences, as shown in Figure 3.1. In general, existing approaches can be divided into sparse optical flow and dense optical flow. Sparse optical flow refers to computing motion of only selected pixels, for example motion of feature points, while dense optical flow computes motion of all pixels. This section mainly introduces Lucas-Kanade optical flow [10], since it is not only simple but also very useful in SLAM. Compared to the Horn and Schunck [9] algorithm which takes high computational time in solving the problem iteratively, Lucus-Kanade algorithm is more efficient by implementing the concept of least-square method.

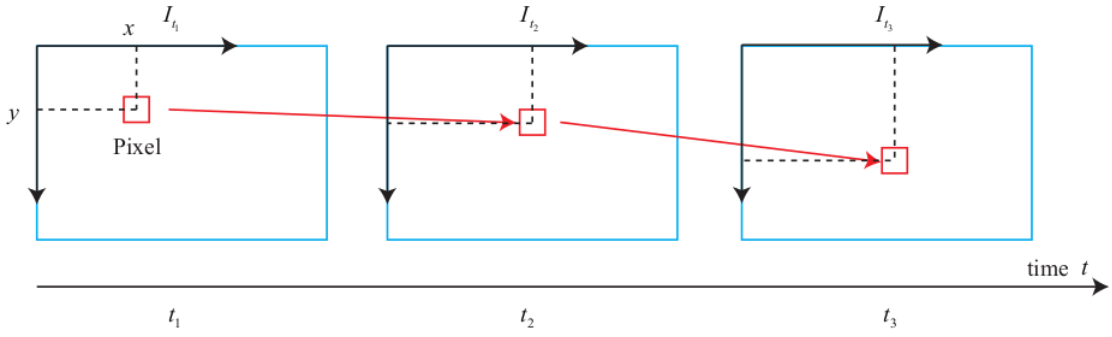


Figure 3.1 Optical flow for a single pixel. Constant intensity is assumed: $\mathbf{I}(x_1, y_1, t_1) = \mathbf{I}(x_2, y_2, t_2) = \mathbf{I}(x_3, y_3, t_3)$.

We assume the images change over time and intensity at pixel (x, y) at time t can be expressed as a function of pixel coordinates and time

$$\mathbf{I}(x, y, t)$$

When camera moves, we suppose the position of pixels will be different in the new frame. To estimate their new position, we bring in the basic but most important assumption of the optical flow method, i.e. the constant brightness assumption. Specifically, a point is presumed to maintain the same intensity value across all observations. Hence, we have

$$\mathbf{I}(x + dx, y + dy, t + dt) = \mathbf{I}(x, y, t) \tag{3.1}$$

for the pixel at (x, y) at time t moving to $(x + dx, y + dy, t + dt)$ at time $t + dt$. It is to note that this is a very strong assumption which can not be true at most of the time in practice. Once we accept this assumption, we can use the first-order Taylor expansion on the left-hand side of Equation 3.1

$$\mathbf{I}(x + dx, y + dy, t + dt) \approx \mathbf{I}(x, y, t) + \frac{\partial \boldsymbol{I}}{\partial x}dx + \frac{\partial \mathbf{I}}{\partial y}dy + \frac{\partial \mathbf{I}}{\partial t}dt \tag{3.2}$$

Because of the brightness assumption (Equation 3.1), we have

$$\frac{\partial \mathbf{I}}{\partial x}dx + \frac{\partial \mathbf{I}}{\partial y}dy = -\frac{\partial \mathbf{I}}{\partial t}dt \tag{3.3}$$

Dividing both sides by dt results in

$$\underbrace{\frac{\partial \mathbf{I}}{\partial x}}_{\mathbf{I}_x}\overbrace{\frac{dx}{dt}}^{u} + \underbrace{\frac{\partial \mathbf{I}}{\partial y}}_{\mathbf{I}_y}\overbrace{\frac{dy}{dt}}^{v} = -\underbrace{\frac{\partial \mathbf{I}}{\partial t}}_{\mathbf{I}_t} \tag{3.4}$$

Where u and v are the x and y components of the optical flow of the pixel (x, y) at time t. We can rewrite the above equation in vectors as

$$\nabla \mathbf{I} \cdot \begin{bmatrix} u \\ v \end{bmatrix} = -\mathbf{I}_t \tag{3.5}$$

There are two variables in this equation and we can not solve it by a single pixel . Thus, we introduce another constraint which assumes pixels inside a certain window (e.g. a window of size $w \times w$) have the same motion

$$\begin{bmatrix} \mathbf{I}_x & \mathbf{I}_y \end{bmatrix}_k \begin{bmatrix} u \\ v \end{bmatrix} = -\mathbf{I}_{t,k}, \qquad k = 1, ..., w^2 \tag{3.6}$$

Now we have a over-determined set of equations and the problem can be solved with least square

$$\begin{bmatrix} u \\ v \end{bmatrix}^* = -(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}, \quad \text{with} \mathbf{A} = \begin{bmatrix} \begin{bmatrix} \mathbf{I}_x & \mathbf{I}_y \end{bmatrix}_1 \\ \vdots \\ \begin{bmatrix} \mathbf{I}_x & \mathbf{I}_y \end{bmatrix}_k \end{bmatrix} \quad \text{and} \quad \mathbf{b} = \begin{bmatrix} \mathbf{I}_{t,1} \\ \vdots \\ \mathbf{I}_{t,k} \end{bmatrix} \tag{3.7}$$

In practice, we iterate this calculation several times, e.g. with Gauss-Newton method to obtain good results, since the gradient is only valid locally. With the estimated optical flow we can track motion of pixels from frame to frame, and inspired from that, the KLT feature tracker [38] is introduced. Although we use translation motion above to interpret optical flow, in general other type of displacement model such as affine transformation is reasonable. It should be noted that the accuracy of LK optical flow and many other traditional methods is, however, limited by problems including dynamic objects with high speed, occlusions, motion blur and homogeneous surfaces.

3.2 Visual Odometry

Visual odometry is the crucial component in visual SLAM with the main task of estimating the ego-pose of cameras or robots in consecutive video frames and reconstruct a local map. Traditional algorithms of visual odometry can be divided into indirect and direct methods. Indirect method based on feature tracking dominates for a long period and is a mature solution, because it is robust, and in-sensitive to illumination variance [7]. Features are small informative regions in images. Mostly corners are desired as it can side-step the aperture problem [39]. There exist many algorithms to extract feature points, e.g. Harris corner detection [40], FAST [41], SIFT [42], SURF [43], ORB [44] and etc. Feature points extracted commonly consist of key points and descriptors.

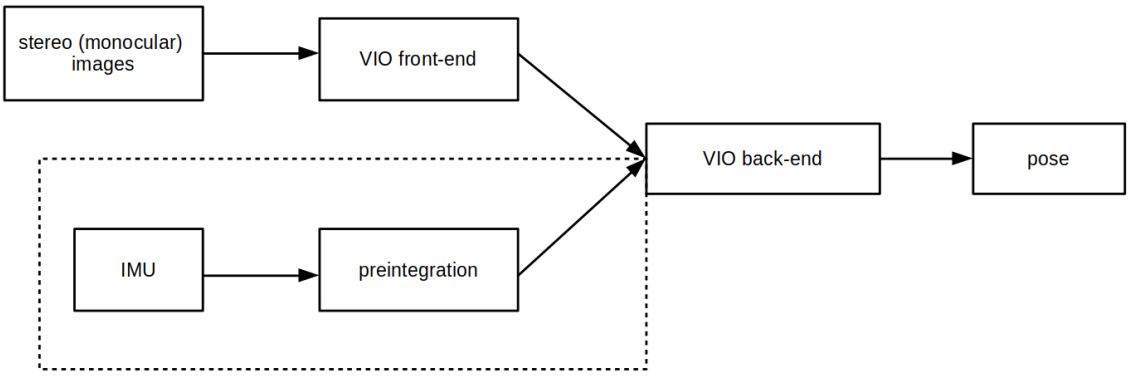


Figure 3.2 Basic framework of VO and VIO. Most of the existing approaches to VO are based on these stages: 1) Take images from monocular or stereo cameras; 2) Apply feature based method, direct method or combine them; 3) Check matched correspondences and remove outliers (For VIO, information from IMU is taken as additional input to the back-end); 4) Estimate camera motion and formulate the back-end problem into a filter or nonlinear optimization problem; 5) Add keypoints periodically to ensure homogeneous coverage across the image.

Feature matching and feature tracking play an important role in indirect visual odometry. It solves the data association problem in SLAM, i.e. it determines the association of landmarks which are currently seen and seen before [7]. Different from feature matching, to track features using optical flow requires only key points but not descriptors of features. Optical flow estimation is more efficient than the computation and matching of descriptors. In general, more accurately we match or track features from frame to frame, easier it becomes for the following pose estimation and optimization.

Another approach for VO is the direct method [35], which computes location of arbitrary pixels in the next frame by minimizing photometric error. In contrast to feature based method, direct method can calculate sparse to dense reconstruction. However, the computation of dense reconstruction is very time and computational power consuming. For the VO problem where quick estimation of camera pose is demanded sparse direct method is mostly the right choice.

3.3 Basalt VIO

Basalt VIO[1] is a state-of-the-art method which combines components such as patch tracking, landmark representation, fist-estimate Jacobians and marginalization scheme. It formulates the incremental motion tracking over time as fixed-lag smoothing problem. Since the integration proposed in thesis mainly makes changes to the feature tracking components of the Basalt, we will introduce this component in detail but the bundle adjustment framework in short. Therefore, we encourage the readers to refer to the work of Usenko et al. [1] for deeper understanding.

Basalt tracks a sparse set of FAST key points in the 2D image plane between adjacent frames using patch-based optical flow based on KLT [10]. The tracking is accurate and robust thanks to the inverse-compositional approach accompanied with a patch dissimilarity norm. The estimation of an optical flow here is based on the pattern of brightness. Instead of zero-normalized cross-correlation, it applies locally-scaled sum of squared differences (LSSD) [45] to obtain illumination-invariant optical flow. According to [1], the main concept of the LSSD KLT can be formulated as

$$\underset{\mathbf{T}\in SE(2)}{\operatorname{argmin}} \sum_{\mathbf{x}_i\in\Omega} \left(r_i(\boldsymbol{\xi})\right)^2 \tag{3.8}$$

where $\mathbf{T} \in SE(2)$ is the desired transformation between two matching patches in adjacent images, Ω defines a patch, and $\overline{I}$ is the average intensity of all pixels in the patch. The residual r_i of an increment $\boldsymbol{\xi}$ is defined as

$$r_i(\boldsymbol{\xi}) = \frac{I_{t+1}(\mathbf{T}\mathbf{x}_i)}{\overline{I_{t+1}}} - \frac{I_t(\mathbf{x}_i)}{\overline{I_t}} \qquad \forall \mathbf{x}_i \in \Omega \tag{3.9}$$

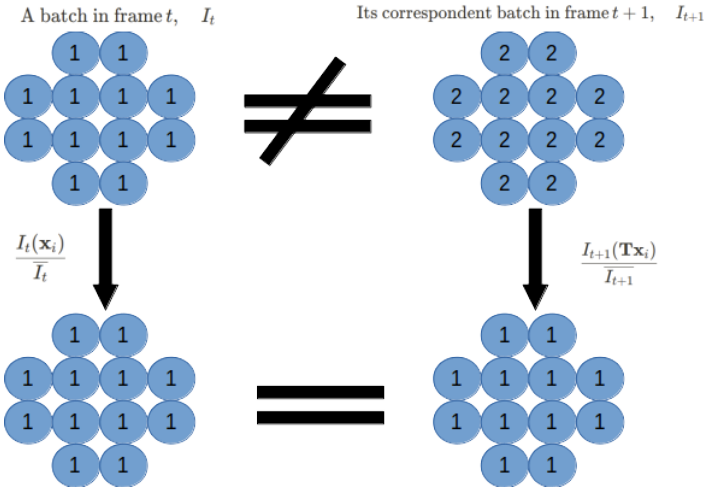


Figure 3.3 LSSD KLT applied on a single patch. Note that the first term in Equation 3.9 can only ensure the optical flow being invariant to the change of photometric scale due to various exposure time but not to the photometric drift.

Afterwards, we can compute the desired transformation, i.e. the optimal increment $\boldsymbol{\xi}^*$ by solving Equation 3.8 with the Gauss-Newton method. To calculate $\delta\boldsymbol{\xi}$ for each iteration, we firstly calculate the Jacobian

$$\boldsymbol{J}_i(\boldsymbol{\xi}) = \frac{\partial r_i(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} \tag{3.10}$$

$$= \frac{\partial}{\partial \boldsymbol{\xi}} \frac{I_{t+1}(\mathbf{T}\mathbf{x}_i)}{\overline{I_{t+1}}(\boldsymbol{\xi})} \tag{3.11}$$

$$= \frac{\partial}{\partial \boldsymbol{\xi}} \frac{I_{t+1}(\exp(\boldsymbol{\xi}^\wedge)\mathbf{x}_i)}{\overline{I_{t+1}}(\boldsymbol{\xi})} \tag{3.12}$$

$$= \frac{\partial}{\partial \mathbf{x}'} \frac{I_{t+1}(\mathbf{x}')}{\overline{I_{t+1}}(\boldsymbol{\xi})} \frac{\partial \mathbf{x}'}{\partial \boldsymbol{\xi}} |_{\mathbf{x}'=\exp(\boldsymbol{\xi}^\wedge)\mathbf{x}_i} \tag{3.13}$$

The updated $\boldsymbol{\xi}_{new}$ after each iteration can be obtained by

$$\boldsymbol{\xi}_{new} = \boldsymbol{\xi}_{old} - \left(\sum_i \boldsymbol{J}_i(\boldsymbol{\xi}_{old}) \left(\boldsymbol{J}_i(\boldsymbol{\xi}_{old})\right)^T \right)^{-1} \left(\sum_i \left(\boldsymbol{J}_i(\boldsymbol{\xi}_{old})\right)^T r_{i,\Omega}(\boldsymbol{\xi}_{old}) \right) \tag{3.14}$$

To speed up the above algorithm, the Jacobian around the point $\mathbf{x}_i$ on I_t is calculated and retained for each iteration. This is only possible, when the baseline of the corresponding patches in two images is small enough. Therefore an initial position is required for the iterative optimization. Basalt simply treats the position of the keypoint in the last frame as the initial position, while we propose to use the position resulted by applying deep optical flow on it as the prior.

Basalt tackles the large displacement problem by adopting a coarse-to-fine framework. Besides, patches with large forward-backward-inconsistency which is defined in section 4.2 are considered as outliers and discarded. In terms of implementation, Basalt is unique in parallelizing in constructing image pyramids and tracking keypoints by applying thread building blocks [46].

At the back-end stage, based on the obtained information from the front-end an error that consists of point reprojection and inertial measurement unit (IMU) propagation terms is minimized in a bundle-adjustment framework [1].

3.4 LiteFlowNet

Following the step of FlowNet [19] and FlowNet2 [26] which pioneer in leveraging convolutional neural networks for optical flow prediction, LiteFlowNet [4] proposed a lightweight alternative network. It is 30

times smaller in the model size and 1.36 times faster than FlowNet2 while maintaining reliable performance on the challenging Sintel [12] and KITTI benchmarks [2]. It utilizes a lightweight cascaded network to estimate flow at each pyramid more effectively. The efficiency is also increased by pyramidal feature extraction and embraces feature warping. Meanwhile, the estimation accuracy is improved through early correction and a flow regularization layer which alleviates the outlier problems [4].

For our experiment, we employ a pretrained model of LiteFlowNet which is fine-tuned on Sintel. According to [4], this model has the best test accuracy in compare with models using other training schema. It achieves the accuracy of 5.38 average end-point error (AEE) on Sintel final testing set, and 1.6 AEE on KITTI flow 2012, which outperforms approaches such as LDOF [47], SPyNet [22], PCA-Layers [48] and etc [4].

3.5 RAFT

Recurrent All-Pairs Field Transforms (RAFT) [5] is an end-to-end deep learning based model for optical flow estimation which is motivated by traditional optimization-based methods. This method is unique in applying a large number of lightweight, recurrent update operators. It extracts per-pixel features, builds multi-scale correlation volumes for all pairs of pixels and updates a flow field through a recurrent unit iteratively. With strong cross-dataset generalization, RAFT achieves state-of-the-art accuracy in various datasets and shows also high efficiency regarding to inference time [5].

We test the proposed system by integrating optical flow estimated using two various pretrained models of RAFT. The model RAFT-s combines KITTI, HD1K [49], and Sintel date when fine-tuning on Sintel. It ranked the first place on both Sintel final with AEE of 2.86 and KITTI with Fl-all [1] of 5.1% at the time of publication [5]. We additionally use the model RAFT-k which is fine-tuned on KITTI for the evaluation on KITTI.

[1] Fl-all refers to the percentage of optical flow outliers over all ground-truth pixels.

4 Integration of Deep optical Flow in Basalt VIO

In this section, we introduce our solution to integrating deep optical flow in Basalt VIO for feature tracking and the approaches we applied to remove inconsistent optical flow vectors as well as incorrectly tracked points.

4.1 Feature Tracking

The first step of out algorithm is to extract a sparse set of keypoints. Same as the original system, we use FAST corner detector [41]. To ensure a homogeneous distribution of the detected points in a frame, we divide the image plane into cells with a user-defined grid size and choose the FAST corners with the strongest response in each cell. The optical flow is estimated by applying either pretrained models of LiteFlowNet [4] or RAFT [5]. We formulate the frame-to-frame feature tracking problem as warping keypoints $\boldsymbol{x_{t,i}}$ with the estimated forward optical flow $\boldsymbol{F_t^{t+1}}$

$$\boldsymbol{x_{t+1,i}} = \boldsymbol{F_t^{t+1}}(\boldsymbol{x_{t,i}}) + \boldsymbol{x_{t,i}} = \begin{bmatrix} u_i \\ v_i \end{bmatrix}_t^{t+1} + \begin{bmatrix} x_i \\ y_i \end{bmatrix}_t \tag{4.1}$$

$\boldsymbol{x_{t+1,i}}$ is the coordinates of keypoint $\boldsymbol{x_{t,i}}$ in the $(t+1)$-th frame. It is to notice that $\boldsymbol{x_{t+1,i}}$ is likely locate outside of the image plane in case of the optical flow is invalid. Therefore we need to validate it against the image resolution. Meanwhile, either $\boldsymbol{x}$ or $\boldsymbol{x} + \boldsymbol{F_t^{t+1}}(\boldsymbol{x})$ may locate in between pixels, hence interpolation or approximation is required in order to obtain the correspondent optical flow vector $\boldsymbol{F_t}$. While some similar works use its nearest neighbor or average over its surrounding four corners to approximate the result flow [50] [6], our system applies bi-linear interpolation to resample its flow vector. To make advantage of the patched-based optical flow estimation in the original Basalt VIO system, we adopt the iterative optimization algorithm introduced in section 3.3 to refine the warping after removing outliers. Instead of the original position of the keypoint in the current frame $\boldsymbol{x_{t,i}}$, its new position $\boldsymbol{x_{t+1,i}}$ resulted by Equation 4.1 is now considered as the initialization for the iterative optimization formulated in Equation 3.8 and Equation 3.9.

Compared to the pyramidal KLT employed in the original system for temporal feature tracking, the new system side-steps the limitation of a coarse-to-fine framework [16] by replacing it with integrating deep optical flow. Besides, since the predicted flow field of the pretrained models can be very noisy on static frames or when motion is extremely small [51], the nonlinear optimization introduced in section 3.3 is adopted in our integration to refine the warping.

To track keypoints in stereo image pairs, we retain the pyramidal KLT tracker of the original system. In general, optical flow does not pertain to stereo matching because in this case Δt in equation Equation 3.4 is zero and hence the equation becomes invalid.

4.2 Outlier Removal

We adopt a two-fold approach to remove outliers in order to achieve robust and accurate tracking. As the first step, we check the consistency of the estimated optical flow of registered keypoints forwards and backwards. Therefore as input to the system, we requires one pair of optical flow per camera in a stereo setup. Each pair consists of a forward and a backward inference, i.e. from the current frame t to the target frame $(t+1)$ and vice versa. Keypoint coordinates in the target frame $\boldsymbol{x_{t+1,i}}$ are computed with

Equation 4.1. To retrieve the keypoint's location $\hat{\boldsymbol{x}}_{t,i}$ in the current frame t, we apply the backward optical flow on its coordinates in the target frame $t+1$

$$\hat{\boldsymbol{x}}_{\boldsymbol{t},\boldsymbol{i}} = \begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix}_t = \boldsymbol{F}^{\boldsymbol{t}}_{\boldsymbol{t+1}}(\boldsymbol{x}_{\boldsymbol{t+1},\boldsymbol{i}}) + \boldsymbol{x}_{\boldsymbol{t+1},\boldsymbol{i}} \tag{4.2}$$

$$= \begin{bmatrix} u \\ v \end{bmatrix}^t_{t+1} + \begin{bmatrix} x \\ y \end{bmatrix}_{t+1} \tag{4.3}$$

In this equation $\boldsymbol{F}^{\boldsymbol{t}}_{\boldsymbol{t+1}}(\boldsymbol{x}_{\boldsymbol{t+1},\boldsymbol{i}})$ is the backward optical flow at $\boldsymbol{x}_{t+1,i}$. As $\hat{\boldsymbol{x}}_{t,i}$ normally does not return to the initial location, we define its squared Euclidean distance to $\boldsymbol{x}_{t,i}$ as forward-backward-inconsistency. It can be computed and simplified by combining Equation 4.1 and Equation 4.3

$$\mathbf{C}_i = ||\hat{\boldsymbol{x}}_{t,i} - \boldsymbol{x}_{t,i}||^2 \tag{4.4}$$

$$= \left|\left| \begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix}_t - \begin{bmatrix} x \\ y \end{bmatrix}_t \right|\right|^2 \tag{4.5}$$

$$= \left|\left| \left(\begin{bmatrix} u \\ v \end{bmatrix}^t_{t+1} + \begin{bmatrix} x \\ y \end{bmatrix}_{t+1} \right) - \begin{bmatrix} x \\ y \end{bmatrix}_t \right|\right|^2 \tag{4.6}$$

$$= \left|\left| \begin{bmatrix} u \\ v \end{bmatrix}^t_{t+1} + \begin{bmatrix} u \\ v \end{bmatrix}^{t+1}_t \right|\right|^2 \tag{4.7}$$

$$= ||\boldsymbol{F}^t_{t+1}(\boldsymbol{x}_{\boldsymbol{t+1},\boldsymbol{i}}) + \boldsymbol{F}^{t+1}_t(\boldsymbol{x}_{t,i})||^2 \tag{4.8}$$

It is proved that better consistency of optical flow vectors prone to higher tracking accuracy [6]. In another words, points with higher inconsistency, i.e. with larger $\mathbf{C}_i$, have higher possibility to be outliers. Based on that, we set a feasible inconsistency threshold to filter outliers.

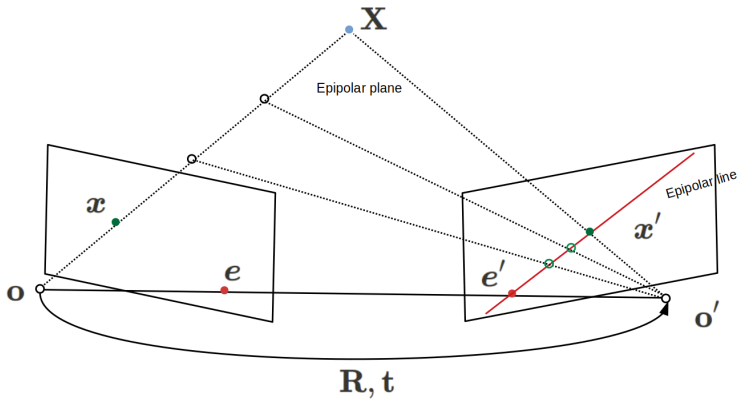


Figure 4.1 Epipolar constraint.

The second stage for outlier removal verifies the epipolar constraint of correspondences in stereo image pairs. For this purpose, we require the essential matrix $\mathbf{E}$ for calibrated cameras or the fundamental matrix $\mathbf{F}$ for uncalibrated. Fundamental Matrix $\mathbf{F}$ has a more general description in terms of projective geometry and it is related to the essential matrix $\mathbf{E}$ by camera intrinsic matrix $\mathbf{K}$ such that [8]

$$\mathbf{F} = \mathbf{K}'\mathbf{E}\mathbf{K}^{-1} \tag{4.9}$$

Suppose we have obtained a pair of correspondences with homogeneous image coordinates, $\boldsymbol{x}$ and $\boldsymbol{x}'$. The epipolar constraint for a stereo image pair can be described as

$$\boldsymbol{x}'^T\mathbf{K}'\mathbf{E}\mathbf{K}^{-1}\boldsymbol{x} = 0 \tag{4.10}$$

In practice, calibration information is provided with the image sequences in most of public datasets, which is our case as well. Therefore, the transformation $[\mathbf{R}, \mathbf{t}]$ between the stereo cameras is normally known and the essential matrix is computed as

$$\mathbf{E} = [\mathbf{t}]_{\times}\mathbf{R} \tag{4.11}$$

where $[\mathbf{t}]_{\times}$ is the skew-symmetric matrix of translation $\mathbf{t}$. Since keypoints are tracked separately for the stereo frames, we perform epipolar constraint only if a keypoint can be observed both in the right and in the left frame. The matched features are then reprojected to normalized 3D coordinates and their epipolar errors are compared against a threshold. If the result exceeds the threshold, we consider them as outliers and remove them from observations in the right frame. Observations in the left frame are retained to avoid the tracking with deep optical flow being interrupted.

5 Evaluation

To evaluate the deep optical flow integrated system, we conduct evaluation on the KITTI Odometry and the EuRoC dataset and compare it to the original system. In this section, we concisely introduce the dataset and criteria we use for evaluation. After that, we present the results of the proposed system and comparison in details.

5.1 Dataset

The presented system is evaluated on two datasets, i.e. KITTI Odometry [2] and EuRoC MAV [3]. In our experiment, We employ pretrained model of LiteFlowNet and RAFT to estimate optical flow. The evaluation criteria are absolute trajectory error and relative pose error [52]. Additionally, for KITTI Odometry we also evaluate the performance regarding to average translational and rotational error which are based on the metric utilized in KITTI benchmark [2].

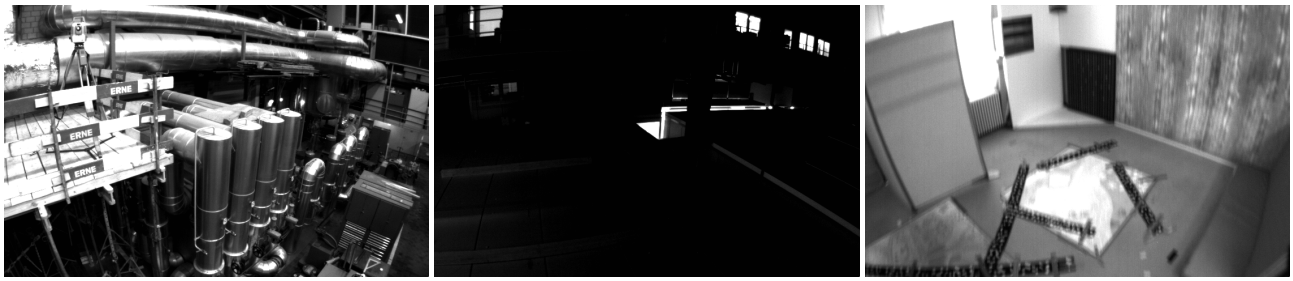
5.1.1 KITTI Odometry

KITTI Odometry dataset contains 22 stereo sequences of various driving scenarios, but only 11 of them are accompanied with publicly available ground-truth [2]. Meanwhile, merely 8 of these 11 sequences are involved in our experiments. Due to storage limitation, the other three sequences which are long will be excluded.

The dataset provides both gray-scale and RGB images. The ground-truth obtained with GPS/OXTS.

(a) Highway scenario in sequence 01. **(b)** Country road in sequence 03. **(c)** Urban scenario in sequence 07.

Figure 5.1 Representative images of the KITTI Odometry dataset.



(a) Scene with good texture in MH_02. **(b)** Dark scene in MH_05. **(c)** Blur scene in V1_03.

Figure 5.2 Representative images of the EuRoC MAV dataset.

5.1.2 EuRoC MAV

EuRoC MAV [3] is a dataset collected on-board a Micro Aerial Vehicle (MAV), containing stereo gray-scale images, synchronized IMU measurements, and accurate motion ground-truth. There are 11 long

sequences provided with different difficulties. The first batch of datasets consists of 5 sequences recorded in an industrial machine hall with some dark scenes, while the second batch with 6 sequences are captured in a room with good illumination. These sequences are of different complexity to process in terms flight dynamics, lighting conditions and motion blur. The characteristics of all sequences are summarized in Table 5.1 and some representative images of the dataset are shown in Figure 5.2. Ground-truth position measurements of the dataset are provided by a laser tracker and estimates of the 6D pose of the MAV are provided at the IMU sampling rate.

Name	Characteristic
MH_01_easy	Good texture, bright scene
MH_02_easy	Good texture, bright scene
MH_03_medium	Fast motion, bright scene
MH_04_difficult	Fast motion, dark scene
MH_05_difficult	Fast motion, dark scene
V1_01_easy	Slow motion, bright scene
V1_02_medium	Fast motion, bright scene
V1_03_difficult	Fast motion, motion blur
V2_01_easy	Slow motion, bright scene
V2_02_medium	Fast motion, bright scene
V2_03_difficult	Fast motion, motion blur

Table 5.1 Characteristics of all sequences of the EuRoC MAV dataset.

5.2 Evaluation Criteria

Methods for evaluating the accuracy of SLAM systems have been investigated over a long period. For now a number of metrics are used in research community. Detailed definition of these metrics can be found in works such as [52], [28] and [53]. Below we briefly review metrics we applied for evaluation, i.e. the absolute trajectory error (ATE), the relative pose error (RPE), as well as the average translational and rotational error which is specifically considered in KITTI odometry benchmark [2].

In the following, we use $\mathbf{Q} \in SE(3)$ and $\mathbf{P} \in SE(3)$ to denote the estimated and the true camera poses respectively. Besides, the translation of a rigid body transformation matrix in $SE(3)$ is defined with an operation $trans(\cdot)$, and the rotation part with $rot(\cdot)$.

5.2.1 Absolute Trajectory Error

Absolute Trajectory error [52] is an intuitive metric to evaluate global consistency of the estimated trajectory by measuring the root-mean-square error between predicted camera poses and ground-truth. Because the estimated and the ground-truth trajectory can be in arbitrary coordinate frames, we first need to align them by using for example the Umeyama alignment method [54], which brings out a transformation matrix $\mathbf{S}$. Then we define the absolute trajectory error matrix at time step i as

$$\mathbf{E}_i := \mathbf{Q}_i^{-1}\mathbf{S}\mathbf{P}_i \tag{5.1}$$

For this metric, we compute the root mean squared error (RMSE) over all time indices (n) of the translational part

$$\text{ATE} := \sqrt{\frac{1}{n}\sum_{i=1}^{n} ||trans(\mathbf{E}_i)||^2} \tag{5.2}$$

5.2.2 Relative Pose Error

To evaluate the local accuracy of the estimated trajectory, relative pose error (RPE) is in particular useful [52]. It compares the reconstructed relative transformation between nearby poses to the ground truth transformation. Assume $\mathbf{Q_i}$ is the estimated pose of the current frame and $\mathbf{Q_{i+1}}$ the next frame, given the respective ground-truth poses $\mathbf{P_i}$ and $\mathbf{P_{i+1}}$, the RPE matrix can be formulated as

$$\mathbf{F}_i := (\mathbf{Q}_i^{-1}\mathbf{Q}_{i+1})^{-1}(\mathbf{P}_i^{-1}\mathbf{P}_{i+1}) \tag{5.3}$$

$\mathbf{F}_i$ consists of a translation and a rotational error. For a clear distinction in evaluation, we normally compute the them separately. Similar to absolute trajectory error, we calculate the RMSE of the translational component $trans(\mathbf{F}_i)$ over all possible time intervals

$$\text{RPE}_{trans} := \sqrt{\frac{1}{m}\sum_{i=1}^{m}||trans(\mathbf{F}_i)||^2} \quad \text{for} \quad i = 1, ..., n \tag{5.4}$$

where $m = n - 1$. Note that although this equation looks similar to Equation 5.2, they are essentially not the same because $\mathbf{E}_i$ and $\mathbf{F}_i$ are defined differently. For the rotational part of $\mathbf{F}_i$ we use mean error approach [53]:

$$\text{RPE}_{rot} := \frac{1}{m}\sum_{i=1}^{m}\angle\mathbf{F}_i \quad \text{for} \quad i = 1, ..., n \tag{5.5}$$

where $\angle\mathbf{F}_i$ is defined as:

$$\angle\mathbf{F}_i := \arccos\left(\frac{tr\left(rot(\mathbf{F}_i)\right) - 1}{2}\right) \tag{5.6}$$

5.2.3 Average Translational and Rotational Error

To evaluate the system performance specific on KITTI odometry, we adopt the metric specifically utilized in KITTI Odometry benchmark which measures errors as functions of the trajectory length. According to the definition in [2], the translation and rotation errors are computed separately as

$$E_{trans}(\mathcal{F}) := \frac{1}{|\mathcal{F}|}\sum_{(i,j)\in\mathcal{F}}||\mathbf{G}_{i,j}|| \tag{5.7}$$

$$E_{rot}(\mathcal{F}) := \frac{1}{|\mathcal{F}|}\sum_{(i,j)\in\mathcal{F}}\angle[\mathbf{G}_{i,j}] \tag{5.8}$$

where the matrix $\mathbf{G}_{i,j}$ is defined as

$$\mathbf{G}_{i,j} := (\mathbf{Q}_i^{-1}\mathbf{Q}_j)^{-1}(\mathbf{P}_i^{-1}\mathbf{P}_j) \quad \text{with} \quad i, j \in \mathcal{F} \tag{5.9}$$

$\mathcal{F}$ is a set of frames (i, j), $|\mathcal{F}|$ is the respective trajectory length and $\angle[\cdot]$ is the rotation angle computed as Equation 5.6.

Because the ground-truth of KITTI odometry is obtained by GPS/OXTS, the ground-truth error is large for very small sub-sequences and bias the evaluation results. Hence, we take the average errors for all possible sub-sequences of length (100, 200, ..., 800) meters into account to better indicate the true performance of the odometry. The final evaluation results are the average translational errors $t_{err}(\%)$ and rotational errors $r_{err}(^\circ/100\text{m})$ computed as

$$t_{err} := \frac{1}{\sum_l |\mathcal{F}_l|}\sum_l E_{trans}(\mathcal{F}_l)\cdot|\mathcal{F}_l| \tag{5.10}$$

$$t_{rot} := 100 \times \frac{1}{\sum_l |\mathcal{F}_l|}\sum_l E_{rot}(\mathcal{F}_l)\cdot|\mathcal{F}_l| \tag{5.11}$$

where $\mathcal{F}_l$ are all sets of frames in the l-th possible sub-sequences of length (100, 200, ..., 800) meters.

5.3 Results

In this part we present the evaluation results of the proposed extended Basalt VIO which integrates deep optical flow on KITTI Odometry dataset and on EuRoC MAV dataset. We compute ATE, RPE and average translational and rotational error to measure our system's accuracy on KITTI odometry , while on EuRoC we consider merely ATE and RPE. We use different system parameters when evaluating on different datasets, which will be specified in the respective subsections.

5.3.1 Evaluation on KITTI Odometry

There is no IMU data provided by KITTI Odometry dataset. Thus, the components related to IMU data in our system will not contribute to the pose estimation in this evaluation. To extract keypoints, we divide images into cells with size of 30 pixels. As mentioned in section 4.1, the system initializes one keypoints for each cell and resamples new points for cells in which the keypoint is loss. Here we use a 4-level pyramidal KLT tracker for stereo matching. It should be noted that resolution of images in KITTI is different from sequence to sequence. In, for example, images from sequence 05 with resolution of 370×1226, about 200-300 keypoints can be tracked per frame. In this evaluation, we additionally apply the mean track length (MTL) which measures the average number of frames in that a point kept tracked to represent the robustness of feature tracking.

On KITTI Odometry dataset, we evaluate not only the original system but also new systems integrated with three pretrained inference models, i.e. LFN-ft of LiteFlowNet [4], RAFT-k and RAFT-s of RAFT [5]. They are all first trained on FlyingChair and FlyingThings, but they are finetuned with different strategies. LFN-ft is finetuned with a mixture of KITTI Flow 12 and KITTI Flow 15 on Sintel [4]. RAFT-k is finetuned using only KITTI data on KITTI, while RAFT-s combines data from KITTI, HD1K [49], and Sintel when finetuning on Sintel. The comparison between these three models is detailed in chapter 6, while we merely focus on the difference between systems with integrated deep optical flow and the original Basalt VIO for now.

The quantitative evaluation results is presented in Table 5.2. When comparing with the original Basalt VIO which applies KLT tracker, the new systems with deep optical flow shows distinct advantages with larger mean track length and the lower errors in almost all of the evaluated sequences except in sequence 07. In this sequence, our system fails at a single frame, where the ego-vehicle stops at a crossing and a large truck crosses in front of it as shown in Figure 5.4. One possible way to alleviate this issue is to filter the points sampled on a dynamic objects.

5.3.2 Evaluation on EuRoC MAV

For evaluation on EuRoC sequences, not only images but also IMU data are available. As suggested in the original paper of Basalt [1], we divide images of EuRoC into a grid of cells with size of 50 pixels $\times 50$ pixels. Since we have smaller displacement in this case, we adjust the level of image pyramid to 3. The other parameters remain the same as evaluation on KITTI except that IMU data is considered for the pose estimation here. In this experiment, we only compare the original system with the system integrated with RAFT-s which has the best performance on the previous evaluation.

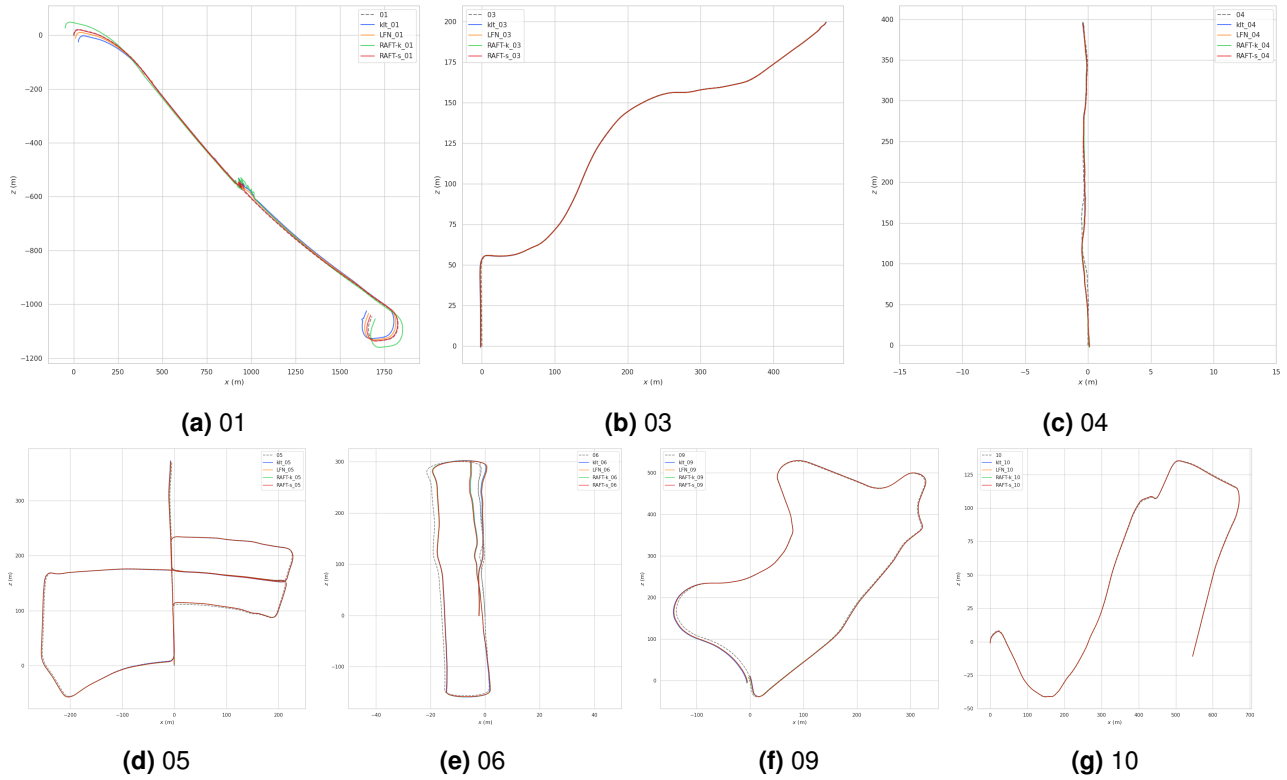
The evaluation results on EuRoC MAV dataset are quantitatively reported in Table 5.3 and illustrated in Figure 5.3. Considering RMS ATE, the proposed system shows better results in eight out of ten sequences. The original system performs slightly better on the sequence MH_02 and V1_01, while the propose system can achieve about 30% lower ATE on the sequence V1_02, V1_03, V2_01 and V2_02. In terms of relative pose error, the proposed system has overall better performance. Based on these results, it is clear that the integration of deep optical flow can significantly help improving the accuracy of a VIO system.

Method	Metric	01	03	04	05	06	07	09	10	Avg. excl. 01
KLT	MTL	2.0893	3.3427	2.3315	2.7874	2.2958	3.0446	2.4493	2.5116	2.6804
	t_{err}	4.5239	0.9962	1.1921	0.7646	1.0605	0.8625	1.0590	0.5892	0.9320
	r_{err}	0.1713	0.2293	**0.1922**	0.2276	**0.2313**	0.4851	0.1937	0.2652	0.2606
	ATE	30.7334	1.3648	1.2690	2.7245	2.5591	1.5547	4.3127	0.9834	2.1098
	RPE_{tran}	0.6737	0.0143	0.0267	0.0136	0.0183	0.0113	0.0213	0.0139	0.0171
	RPE_{rot}	0.0469	0.0328	0.0237	0.0309	0.0239	0.0281	0.0332	0.0383	0.0301
LFN-ft	MTL	2.4084	5.9444	3.0587	**3.4437**	3.2274	5.3516	3.6193	4.0942	4.1056
	t_{err}	3.7765	0.9007	1.0723	0.7019	0.9497	X	**0.9591**	**0.5604**	0.8573
	r_{err}	0.1828	**0.2140**	0.2340	0.2253	0.2460	X	**0.1735**	**0.2330**	**0.2210**
	ATE	18.2617	1.0455	1.1437	2.3292	2.5230	X	**3.6707**	**0.8712**	1.9306
	RPE_{tran}	0.4930	**0.0132**	0.0256	0.0120	0.0149	X	0.0194	**0.0131**	0.0164
	RPE_{rot}	0.0465	**0.0325**	**0.0226**	0.0304	0.0229	X	**0.0321**	0.0379	0.0297
RAFT-k	MTL	**2.0825**	5.6359	2.6124	3.1286	2.9939	5.1324	3.3664	3.8669	3.8195
	t_{err}	5.5563	**0.8853**	1.5338	0.7138	0.9508	X	0.9703	0.6128	0.9445
	r_{err}	0.1834	0.2269	0.2852	**0.2220**	0.2371	X	0.1879	0.2550	0.2357
	ATE	37.5889	**1.0174**	1.6687	2.3022	**2.3997**	X	3.7039	0.9243	2.0027
	RPE_{tran}	0.6743	0.0133	0.0335	0.0123	0.0162	X	0.0200	0.0132	0.0181
	RPE_{rot}	0.0410	0.0326	0.0237	0.0304	0.0231	X	0.0324	0.0383	0.0301
RAFT-s	MTL	2.4889	**5.9630**	**3.2364**	3.4286	**3.3288**	**5.4351**	**3.6955**	**4.19409**	**4.1831**
	t_{err}	**1.7562**	0.9033	**0.9665**	**0.6996**	**0.9144**	X	0.9602	0.6122	**0.8427**
	r_{err}	**0.1258**	0.2144	0.2342	0.2262	0.2432	X	0.1819	0.2459	0.2243
	ATE	**5.1679**	1.0309	**1.0170**	**2.2426**	2.4629	X	3.7208	0.9023	**1.8961**
	RPE_{tran}	**0.2653**	0.0133	**0.0240**	**0.0118**	**0.0146**	X	**0.0188**	**0.0131**	**0.0159**
	RPE_{rot}	**0.0324**	**0.0325**	**0.0226**	**0.0303**	**0.0228**	X	0.0322	**0.0380**	**0.0297**

Table 5.2 Quantitative result on KITTI Odometry (Seq. 01, 03-07, 09, 10). Units of the metrics are: MTL [frame], t_{err} [%], r_{err} [$^\circ/100$m], ATE [m], RPE_{tran} [m], RPE_{rot} [$^\circ$]. The best results are stressed in bold. Some sequences are excluded due to limited computational power and thus extremely long inference time. Results on sequence 01 is ignored when calculating the average, because it commonly recognized as very difficult to solve in research community and results from different models have high bias. KLT refers to the original Basalt VIO [1] which applies KLT tracker, while the others are systems integrated deep optical flow. LFN-ft, RAFT-k and RAFT-s refer to different pretrained models for deep optical estimation. LFN-ft is a model of LiteFlowNet [4] which is finetuned on Sintel dataset [12]. RAFT-k and RAFT-s are models of RAFT [5]. The former on is finetuned on KITTI Flow 15, while the later is finetuned on Sintel. The systems leveraging deep optical flow outperform the original system on the majority of the sequences in terms of ATE. The system integrated with RAFT-s has the best average results w.r.t four of the considered metrics.

Method	Metric	MH_01	MH_02	MH_03	MH_04	MH_05	V1_01	V1_02	V1_03	V2_01	V2_02	Avg.
KLT	ATE	0.09081	**0.05387**	0.08488	0.10852	0.12732	**0.04284**	0.05636	0.07201	0.05636	0.06414	0.07650
	RPE_{tran}	0.00138	0.00180	0.00374	0.00509	0.00370	0.00229	0.00295	0.00508	0.00118	0.00988	0.00371
	RPE_{rot}	0.00040	0.00043	0.00055	0.00069	0.00054	0.00068	0.00086	0.00107	0.00067	0.00098	0.00069
RAFT-s	ATE	**0.08618**	0.05395	**0.07096**	**0.10008**	**0.10767**	0.04322	**0.04114**	**0.04876**	**0.03777**	**0.03974**	**0.06295**
	RPE_{tran}	**0.00136**	**0.00138**	**0.00353**	**0.00485**	**0.00357**	**0.00228**	**0.00265**	**0.00348**	**0.00110**	**0.00298**	**0.00272**
	RPE_{rot}	**0.00038**	**0.00041**	**0.00054**	**0.00066**	**0.00051**	**0.00067**	**0.00082**	**0.00104**	**0.00065**	**0.00091**	**0.00066**

Table 5.3 ATE and RPE w.r.t. translation and rotation of the estimated trajectory on the EuRoC dataset. Units of the applied metrics are: ATE [m], RPE_{tran} [m], RPE_{rot} [$^\circ$]. KLT denotes the original Basalt VIO [1], while RAFT-s refers to the proposed system integrated with a model of RAFT [5] which is finetuned on Sintel dataset [12]. The better results among the two systems are shown in bold. The sequence V2_03 is not evaluated because a number of frames for one camera in it are missing.



(a) 01 **(b)** 03 **(c)** 04

(d) 05 **(e)** 06 **(f)** 09 **(g)** 10

Figure 5.3 Qualitative results on KITTI Odometry (Seq. 01, 03, 04, 05, 06, 09, 10).

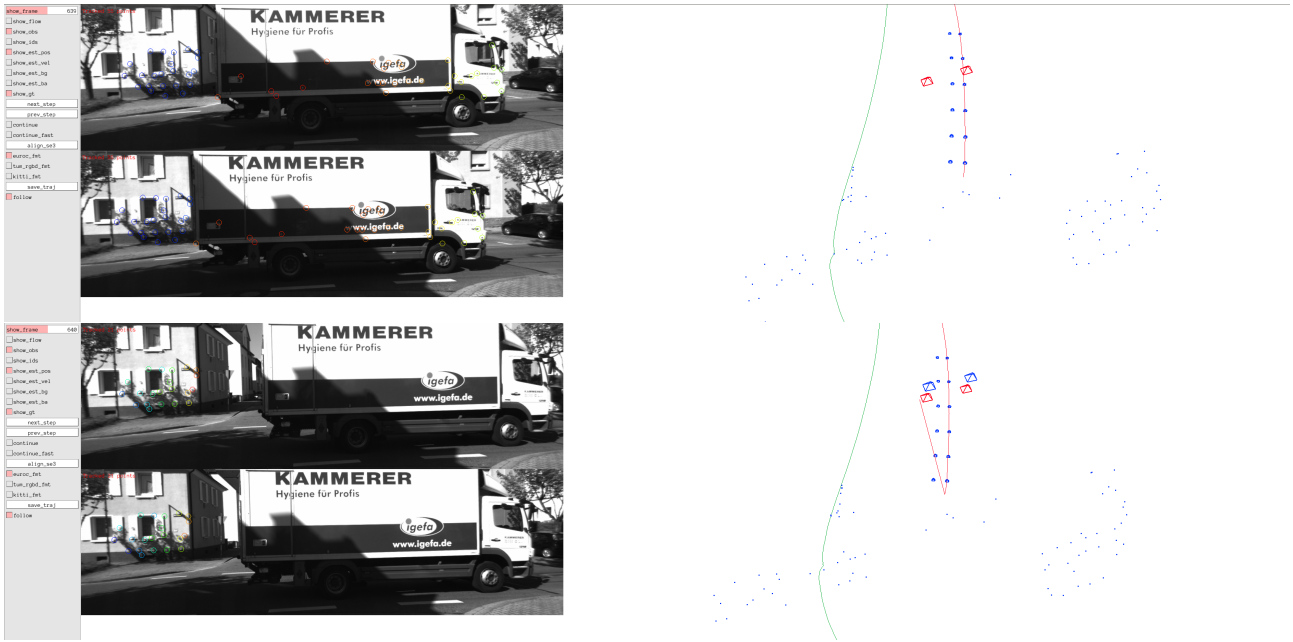
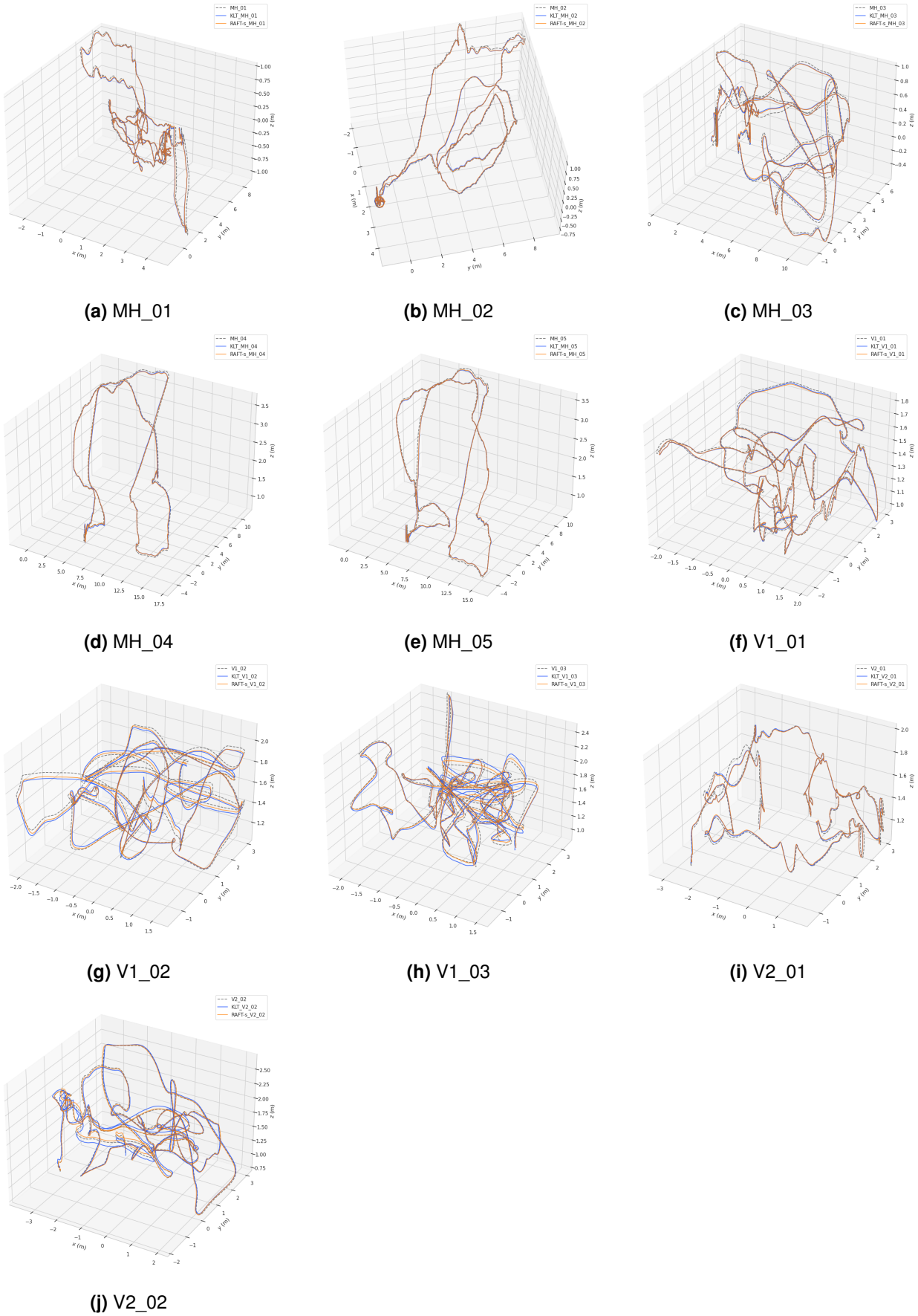


Figure 5.4 Scenario excerpted from sequence 07 of KITTI where the proposed system fails to estimate reasonable pose. A truck drives in front when the ego-vehicle stops at the intersection. About half of tracked points locate on the truck which meanwhile occludes around half of the view.



(a) MH_01 **(b)** MH_02 **(c)** MH_03

(d) MH_04 **(e)** MH_05 **(f)** V1_01

(g) V1_02 **(h)** V1_03 **(i)** V2_01

(j) V2_02

Figure 5.5 Qualitative results on EuRoC MAV (MH_01, MH_02, MH_03, MH_04, MH_05, V1_01, V1_02, V1_03, V2_01, V2_02).

5.3.3 Timing

According to Usenko et al. [1], the average time of the original Basalt VIO per frame on the EuRoC dataset is 7.83 ms[1], while our experiment[2] measures an average time of 7.4 ms per frame on EuRoC and 13.08 ms per frame on KITTI. The timing of optical flow inference stage using a Quadro P3200 GPU is provided in Table 5.4. Although the original system is around 4 times faster than the real-time playback, the proposed new system is not real-time capable at all due to high computational effort required for optical flow inference.

		KITTI	EuRoC
	Frame rate	0.10 s	0.03 s
Inference time	RAFT-s/k	0.55 s	0.40 s
	LFN-ft	0.10 s	0.065 s

Table 5.4 Timing per frame. Recording time indicates the frame rate of the datasets. KITTI captures images with 10 frames per second, while EuRoC 30 frames per second. Images of KITTI have larger resolution than images of EuRoC. The inferring time for optical flow of an image pairs are shown as statistic outside the bracket. The application of forward-backward-inconsistency check requires four passes of image pairs, i.e. two passes for each camera. Hence, the total time for the optical flow inference stage of our system is quadruple of the time reported here.

[1] The original implementation is tested on an Intel E5-1620 CPU with 4 cores and 8 virtual cores [1]

[2] We tested the original implementation on an Intel i7-8850H CPU with 6 cores and 12 virtual cores.

6 Discussion

In this section, we discuss the proposed approach in terms of the methods we applied and possible work directions for future improvement. We conduct an ablation study to investigate possible factors that influences on the performance of the proposed system. For this purpose, we use a system with the following settings as baseline and study how performance may vary with different settings:

- Inference model: RAFT-s
- Images used for inference: gray-scale images
- Interpolation: bilinear interpolation
- Refinement: with refinement
- Keypoint extraction: FAST corner detection

It should also be noted that due to high difficulty of sequence 01 and failure of the proposed system on sequence 07, we merely conduct ablation studies on six sequences out of the 11 sequences with ground-truth of the KITTI Odometry dataset, i.e. the sequence 03, 04, 05, 06, 09 and 10. The EuRoC MAV dataset is not considered when studying the influence of inferring deep optical flow using RGB images on the accuracy of the proposed system, since this dataset contains only gray-scale images.

6.1 Inference Model

The accuracy of a visual odometry is highly related to the quality of the feature tracking process. Since our system leverages optical flow based feature tracking, the accuracy of the model we used to estimate deep optical flow remains critical. We compare performance of systems integrated with three different models. One of these models called LFN-ft is from LiteFlowNet [4] and the other two, i.e. RAFT-s and RAFT-k are from RAFT [5]. In addition to the different frameworks used, the training scheme used by these models also differ. LFN-ft is trained and finetuned The evaluation result of the corresponding systems is presented in Table 5.2.

By comparing the evaluation results, we find out that the models, i.e. RAFT-s and LFN-ft which are finetuned with mixed data from different datasets can result in more accurate estimation, while RAFT-k shows nearly no advantage of being finetuned merely on KITTI. In addition to the self-imposed problems of the KITTI dataset mentioned in chapter 2, an other possible reason for that is models finetuned with a combination of various datasets normally have better generalization. Besides, it can be seen that RAFT-s outperforms LFN-ft on the majority of sequences. This is likely due to the application of the recurrent operators in RAFT [5].

6.2 Images for Inference - Gray-Scale v.s. RGB

The existing supervised and self-supervised learning methods for optical flow estimation are mainly trained on color images, because RGB images contain richer information than gray images. Moreover, nearly all existing datasets so far consist only of RGB images, except the HD1K dataset. We initially estimate the deep optical flow using gray-scale images because the original system is designed for gray-scale images. However, since the deep optical flow are obtained using an extra inference model, we attempt to infer them on color images expecting better results for the visual odometry task.

The system integrated with deep optical flow which is inferred using color images achieves lower average error regarding to average translational and rotational error as well as ATE, while it achieve the same result in terms of translational and rotational relative pose errors as the baseline (s. Table 6.1). It can be concluded based on these results that inferring optical flow on RGB images is able to improve the tracking accuracy of the proposed system.

Method	Metric	03	04	05	06	09	10	Avg.
Gray-scale	t_{err}	0.9033	**0.9665**	0.6996	**0.9144**	**0.9602**	0.6122	0.8427
	r_{err}	**0.2144**	0.2342	0.2262	0.2432	**0.1819**	0.2459	0.2243
	ATE	1.0309	**1.0170**	2.2426	2.4629	**3.7208**	0.9023	1.8961
	RPE$_{tran}$	**0.0133**	**0.0240**	0.0118	**0.0146**	0.0188	0.0131	0.0159
	RPE$_{rot}$	0.0325	**0.0226**	0.0303	0.0228	0.0322	0.0380	0.0297
RGB	t_{err}	**0.8827**	1.0082	**0.6946**	0.9170	0.9644	**0.5622**	**0.8382**
	r_{err}	0.2249	**0.2282**	**0.2234**	**0.2420**	0.1849	**0.2278**	**0.2219**
	ATE	**1.0049**	1.0668	**2.1745**	**2.3706**	3.7324	**0.8626**	**1.8686**
	RPE$_{tran}$	0.0134	0.0241	0.0118	0.0147	0.0188	**0.0130**	0.0159
	RPE$_{rot}$	**0.0324**	0.0228	0.0303	0.0228	0.0322	0.0380	0.0297

Table 6.1 Evaluation results of ablation study about the image format used for inference on KITTI Odometry (Seq. 03, 04, 05, 06, 09, 10). The best results are shown in bold.

6.3 Interpolation - Nearest Neighbor v.s. Bilinear Interpolation

As mentioned in section 4.1, the extracted keypoints do not necessarily locate in a regular grid. To obtain its corresponding flow vector, interpolation methods are necessary. Nearest neighbor and bilinear interpolation are popular solutions in works facing similar problems. We compare the evaluation results of two systems (s. Table 6.2). One applies the optical flow of the nearest neighbor and the other uses bilinear interpolation to interpolate the flow vector, when a keypoint is located between pixels.

Regarding to the evaluation on KITTI Odometry dataset shown in Table 6.2, the system using bilinear interpolation achieves lower absolute trajectory errors on four out of six sequences. In terms of relative pose errors, the difference between them is tiny. On EuRoC dataset, using bilinear interpolation helps the system achieve lower ATE on half of the sequences but the system applying nearest neighbor has a lower average ATE. Based on these results, there is no obvious distinction between the system using nearest neighbor and the one using bilinear interpolation can be found.

6.4 With or Without Refinement

The application of nonlinear optimization for refinement described in section 3.3 is inspired by the KLT tracker of the original VIO of Basalt, but optical flow itself can be sufficient for tracing pixels in consecutive images. Thus we conduct this ablation study, in order to analyze whether the accuracy of our system can be benefited from the refinement.

The results shown in Table 6.4 and Table 6.5 confirm that using nonlinear optimization can significantly help obtain a more accurate tracking locally and thus more accurate local pose estimation. The baseline outperforms the system integrated with deep optical flow which is not refined on the EuRoC dataset and on the majority of the KITTI dataset. However, although using refined optical flow achieves better local accuracy on sequence 03, 04 and 06, the absolute trajectory error of not refined is lower on sequence 04, and much smaller on 03 and 06. For sequence 03 and 06, the trajectories estimated by the system extended with refined optical flow are drifted more as shown in Figure 6.1.

Method	Metric	03	04	05	06	09	10	Avg.
NN	t_{err}	0.9055	**0.9662**	0.7017	0.9149	0.9625	**0.6027**	0.8423
	r_{err}	**0.2093**	**0.2272**	0.2278	0.2442	0.1826	**0.2411**	**0.2220**
	ATE	1.0327	**1.0123**	2.2675	2.4847	3.7410	**0.8877**	1.9043
	RPE$_{tran}$	0.0133	0.0242	0.0118	0.0147	0.0188	**0.0130**	0.0160
	RPE$_{rot}$	**0.0324**	0.0227	0.0303	0.0228	**0.0321**	0.0380	0.0297
BI	t_{err}	**0.9033**	0.9665	**0.6996**	**0.9144**	**0.9602**	0.6122	**0.8427**
	r_{err}	0.2144	0.2342	**0.2262**	**0.2432**	**0.1819**	0.2459	0.2243
	ATE	**1.0309**	1.0170	**2.2426**	**2.4629**	**3.7208**	0.9023	**1.8961**
	RPE$_{tran}$	0.0133	**0.0240**	0.0118	**0.0146**	0.0188	0.0131	**0.0159**
	RPE$_{rot}$	0.0325	**0.0226**	0.0303	0.0228	0.0322	0.0380	0.0297

Table 6.2 Evaluation results of ablation study of interpolation methods on KITTI Odometry (Seq. 03, 04, 05, 06, 09, 10). NN refers to the nearest neighbor and BI to bilinear interpolation. The best results are shown in bold.

Method	Metric	MH_01	MH_02	MH_03	MH_04	MH_05	V1_01	V1_02	V1_03	V2_01	V2_02	Avg.
NN	ATE	**0.0855**	**0.0516**	0.0717	0.1023	**0.0934**	0.0432	0.0425	0.0493	0.0403	0.0397	**0.06194**
	RPE$_{tran}$	0.0014	0.0014	0.0035	0.0048	**0.0035**	0.0023	0.0027	0.0035	0.0011	**0.0029**	**0.00270**
	RPE$_{rot}$	0.0004	0.0004	0.0005	0.0007	0.0005	0.0007	0.0008	0.0010	0.0007	0.0009	0.00066
BI	ATE	0.0862	0.0540	**0.0710**	**0.1001**	0.1077	0.0432	**0.0411**	**0.0488**	**0.0378**	0.0397	0.06295
	RPE$_{tran}$	0.0014	0.0014	0.0035	0.0048	0.0036	0.0023	0.0026	0.0035	0.0011	0.0030	0.00272
	RPE$_{rot}$	0.0004	0.0004	0.0005	0.0007	0.0005	0.0007	0.0008	0.0010	0.0007	0.0009	0.00066

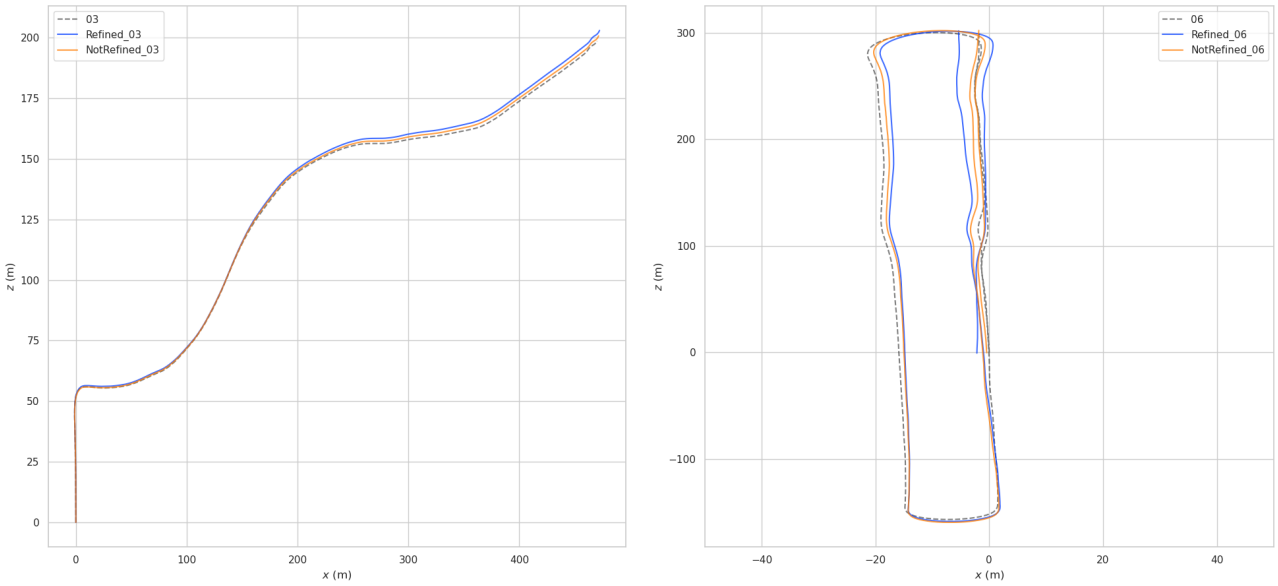
Table 6.3 Evaluation results of ablation study of interpolation methods on EuRoC MAV. NN refers to the nearest neighbor and BI to bilinear interpolation. The best results are shown in bold.

Method	Metric	03	04	05	06	09	10	Avg.
Refined	t_{err}	0.9033	**0.9665**	**0.6996**	**0.9144**	**0.9602**	**0.6122**	**0.8427**
	r_{err}	**0.2144**	**0.2342**	**0.2262**	**0.2432**	**0.1819**	**0.2459**	**0.2243**
	ATE	1.0309	1.0170	**2.2426**	2.4629	**3.7208**	**0.9023**	**1.8961**
	RPE$_{tran}$	**0.0133**	**0.0240**	**0.0118**	**0.0146**	**0.0188**	**0.0131**	**0.0159**
	RPE$_{rot}$	**0.0325**	**0.0226**	**0.0303**	**0.0228**	**0.0322**	**0.0380**	**0.0297**
Not refined	t_{err}	**0.6714**	1.0386	0.8153	1.0155	1.0376	0.6348	0.8689
	r_{err}	0.2595	0.4916	0.2730	0.3138	0.2466	0.3638	0.3247
	ATE	**0.6747**	**0.9074**	3.3607	**2.0447**	4.4613	1.1232	2.0953
	RPE$_{tran}$	0.0147	0.0385	0.0154	0.0225	0.0242	0.0165	0.0220
	RPE$_{rot}$	0.0333	0.0287	0.0327	0.0279	0.0352	0.0410	0.0331

Table 6.4 Evaluation results on KITTI Odometry (Seq. 03, 04, 05, 06, 09, 10) of the ablation study about refining the integrated deep optical flow using nonlinear optimization. The best results are written in bold.

Method	Metric	MH_01	MH_02	MH_03	MH_04	MH_05	V1_01	V1_02	V1_03	V2_01	V2_02	Avg.
Refined	ATE	**0.0862**	**0.0540**	**0.0710**	**0.1001**	**0.1077**	**0.0432**	**0.0411**	**0.0488**	**0.0378**	**0.0397**	**0.06295**
	RPE$_{tran}$	**0.0014**	**0.0014**	**0.0035**	**0.0048**	**0.0036**	**0.0023**	**0.0026**	**0.0035**	**0.0011**	**0.0030**	**0.00272**
	RPE$_{rot}$	**0.0004**	**0.0004**	**0.0005**	**0.0007**	**0.0005**	**0.0007**	**0.0008**	**0.0010**	**0.0007**	**0.0009**	**0.00066**
Not refined	ATE	0.2238	0.1707	0.1429	0.4098	0.3747	0.0543	0.0549	0.0508	0.0397	0.0530	0.15746
	RPE$_{tran}$	0.0022	0.0027	0.0044	0.0082	0.0063	0.0024	0.0030	0.0035	0.0014	0.0026	0.00368
	RPE$_{rot}$	0.0005	0.0005	0.0006	0.0008	0.0006	0.0007	0.0009	0.0011	0.0007	0.0009	0.00074

Table 6.5 Evaluation results on EuRoC MAV dataset of the ablation study about refining the integrated deep optical flow using nonlinear optimization. The best results are stressed in bold.



(a) Aligned trajectories of sequence 03

(b) Aligned trajectories of sequence 06

Figure 6.1 Aligned trajectories of sequence 03 and 06 estimated by systems integrated with and without refined deep optical flow.

Method	Metric	03	04	05	06	09	10	Avg.
FAST	t_{err}	0.9033	**0.9665**	**0.6996**	**0.9144**	**0.9602**	0.6122	**0.8427**
	r_{err}	0.2144	0.2342	**0.2262**	0.2432	**0.1819**	0.2459	0.2243
	ATE	**1.0309**	**1.0170**	**2.2426**	**2.4629**	**3.7208**	0.9023	**1.8961**
	RPE$_{tran}$	**0.0133**	**0.0240**	**0.0118**	**0.0146**	**0.0188**	**0.0131**	**0.0159**
	RPE$_{rot}$	0.0325	**0.0226**	**0.0303**	**0.0228**	**0.0322**	**0.0380**	**0.0297**
FBC	t_{err}	0.9136	1.2626	0.7521	0.9721	1.0056	0.5783	0.9141
	r_{err}	**0.1980**	**0.1456**	0.2305	0.2506	0.2125	0.2256	**0.2104**
	ATE	1.1586	1.3480	2.3011	2.5033	3.8490	0.8810	2.0068
	RPE$_{tran}$	0.0140	0.0300	0.0130	0.0168	0.0212	0.0138	0.0181
	RPE$_{rot}$	0.0325	0.0234	0.0308	0.0233	0.0329	0.0383	0.0302
Combine	t_{err}	**0.9003**	1.2315	0.7436	0.9292	0.9835	**0.5309**	0.8865
	r_{err}	0.2188	0.2346	0.2330	**0.2238**	0.2011	**0.2209**	0.2220
	ATE	1.0528	1.3271	2.5262	**2.1958**	4.0777	**0.8449**	2.0041
	RPE$_{tran}$	0.0135	0.0279	0.0124	0.0155	0.0196	0.0133	0.0171
	RPE$_{rot}$	**0.0324**	0.0231	0.0304	0.0232	0.0326	0.0382	0.0300

Table 6.6 Evaluation results of ablation study about keypoint extraction methods on KITTI Odometry (Seq. 03, 04, 06, 09). The best results are shown in bold. FAST denotes the baseline which extracts FAST corners. FBC is the system employing forward-backward-inconsistency for keypoint extraction. Combine refers to a system which combines the above two methods.

6.5 Keypoint Extraction - FAST v.s. Forward-Backward Consistent Points

In Basalt's original VIO system, FAST corner detection is applied to extract a set of keypoints to initialize the KLT tracker. But we have a different starting point because we already have the optical flow in hand before extracting keypoints. Therefore, the forward-backward inconsistency of optical flow introduced in section 4.2 can be applied to extract points and exclude some outliers in an early stage. This keypoint selection scheme is inspired by DF-VO [6]. For simplicity we will name the points extracted with this approach forward-backward-consistent keypoints (FBC). To prove the idea, we construct two new systems based on the baseline. One of them extracts FBC points. The other extracts FAST keypoints and applies forward-backward-inconsistency to remove outliers such that if the keypoint with strongest response in a cell does not have forward-backward-consistent optical flow, the point with second strongest response will be tested. The selection is performed iteratively, until either a point in this cell is suitable or all extracted FAST corners are checked.

The evaluation results on KITTI Odometry and EuRoC MAV are presented in Table 6.6 and Table 6.7 respectively. Nonetheless, the comparison of the results in these two tables shows that this idea can not bring us a distinct improvement. According to the statistic, the system applying FAST corner detection still dominates on the majority of the evaluated KITTI Odemetry sequences, while the system with a combined keypoint extraction scheme obtains the lowest ATE on over half of the EuRoC sequences as well as the smallest average ATE. The system which extracts FBC points has the most mediocre performance. We presume that the reason could lie in the fact that the points extracted with FBC are not reliable for pose estimation and scene reconstruction using bundle adjusment. However, it may be beneficial to combine FAST corner detection and forward-backward-inconsistency validation at the keypoint extraction stage, since we can remove outliers earlier in this way.

Method	Metric	MH_01	MH_02	MH_03	MH_04	MH_05	V1_01	V1_02	V1_03	V2_01	V2_02	Avg.
FAST	ATE	0.0862	0.0540	0.0710	0.1001	0.1077	**0.0432**	**0.0411**	**0.0488**	**0.0378**	0.0397	0.06295
	RPE_{tran}	**0.0014**	**0.0014**	**0.0035**	**0.0048**	0.0036	0.0023	**0.0026**	**0.0035**	**0.0011**	**0.0030**	**0.00272**
	RPE_{rot}	**0.0004**	**0.0004**	**0.0005**	**0.0007**	**0.0005**	**0.0007**	**0.0008**	**0.0010**	**0.0007**	**0.0009**	**0.00066**
FBC	ATE	0.0997	0.0750	0.0893	0.1757	0.1157	0.0432	0.0474	0.0447	0.0396	0.0319	0.07623
	RPE_{tran}	0.0015	0.0015	0.0039	0.0058	0.0041	0.0023	0.0028	0.0044	0.0012	0.0047	0.00323
	RPE_{rot}	0.0004	0.0004	0.0006	0.0007	0.0006	0.0007	0.0009	0.0012	0.0007	0.0010	0.00071
Combine	ATE	**0.0783**	**0.0536**	**0.0552**	**0.0992**	**0.1066**	0.0439	0.0472	0.0627	0.0437	**0.0342**	**0.06244**
	RPE_{tran}	0.0013	**0.0014**	**0.0035**	0.0055	**0.0034**	0.0023	0.0027	0.0036	**0.0011**	0.0041	0.00288
	RPE_{rot}	**0.0004**	**0.0004**	**0.0005**	**0.0007**	**0.0005**	**0.0007**	**0.0008**	**0.0010**	**0.0007**	**0.0009**	**0.00066**

Table 6.7 Evaluation results of ablation study about keypoint extraction methods on EuRoC MAV. The best results are shown in bold. FAST denotes the baseline which extracts FAST corners. FBC is the system employing forward-backward-inconsistency for keypoint extraction. Combine refers to a system which combines the above two methods.

6.6 Pyramidal Level of KLT for Stereo Matching

Different from the original Basalt VIO which utilizes pyramidal KLT tracker for both temporal tracking and stereo matching, our system applies deep optical flow for the first and retain the KLT tracker for the second. This feature can be exploited to solve the failure caused by dynamic objects, e.g. sequence 07 of KITTI. Instead of using a 4-level pyramidal KLT tracker in the evaluation above (s. Table 5.2), we reduce the level to 3. In this way, we can reject some correspondences on fast-moving objects in scene when performing stereo matching. We conduct evaluation of this setting on sequences 01 and 07 from KITTI odometry because these two sequences contain scenarios where dynamic objects are critical (s. Figure 5.4 and Figure 6.3a). The corresponding trajectories estimated with the new setting is illustrated in Figure 6.2 and Figure 6.3b respectively. With a lower level coarse-to-fine KLT, keypoints on fast-moving objects are rejected when matching correspondences in stereo image pairs. Thanks to that, the proposed system can

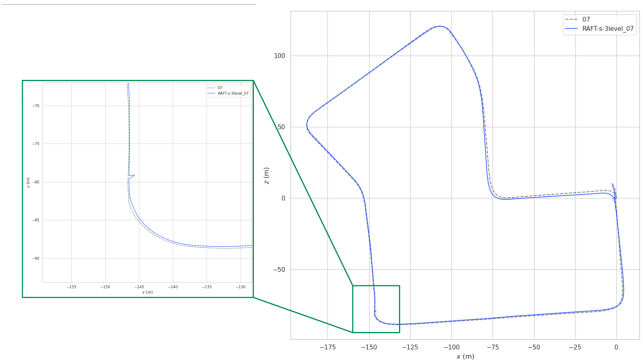
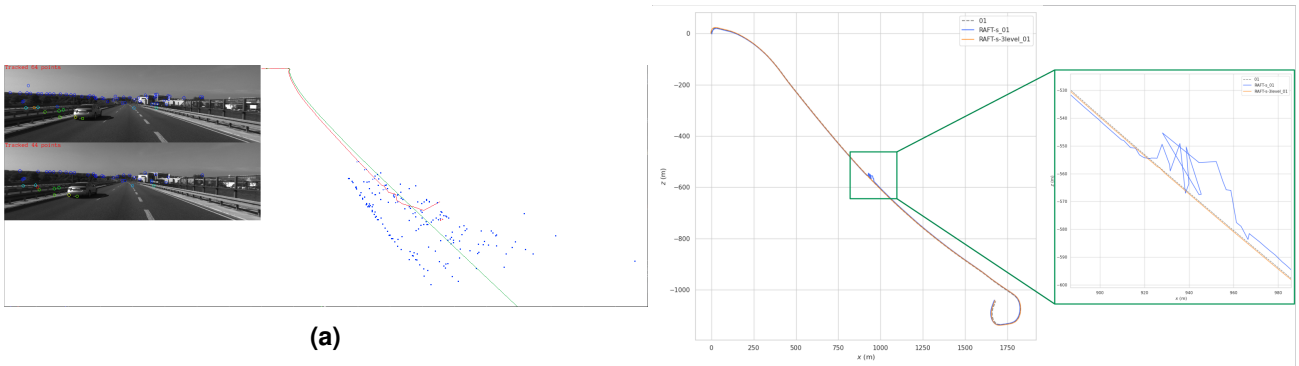


Figure 6.2 Trajectory of sequence 07 from KITTI Odometry estimated by the baseline system with a 3-level pyramidal KLT tracker. With reduced pyramid level of the KLT tracker for stereo matching, the proposed system is able to survive from the failure point shown in Figure 5.4. However, there is a noticeable drift at that particular point.

successfully estimate a full trajectory on sequence 07 and acquire a smoother trajectory on sequence 01. However, it is worth mentioning that this is merely a transitory solution to the problem caused by dynamic objects and is not guaranteed to work any time.



(a) **(b)**

Figure 6.3 (a) Scenario excerpted from sequence 01 of KITTI Odometry dataset. In this scenario the ego-vehicle is driving on highway where the VO is difficult to estimate the pose because the ego-motion is fast and the scene lacks of useful textures. The estimated trajectory becomes zig-zag when a car passing with high speed on the left side. (b) With reduced pyramid level of the KLT tracker, keypoints on the fast-passing-by car are not tracked in the stereo image pairs. This result in a much smoother trajectory.

6.7 Future Research Direction

The ablation study above shows that the system with the reference setting has in general the best performance among all proposed variations. It adopts the deep optical flow as a prior and performs a nonlinear optimization to refine the flow vector. Compared to the original VIO of Basalt, it achieves better estimation regarding to both global and local accuracy on the the majority of the KITTI sequences despite the failure on the sequence 07 and on almost all EuRoC sequences.

As introduced in chapter 4, the models utilized in our system are pretrained models which are released in the official repository of RAFT and LiteFlowNet. Thus, in future work we should finetune these models or train a new model from scratch with a new dataset which is specifically constructed according to the requirement of the VIO task. To be more specific, in contrast to most of the existing datasets, the new one should consist of gray-scale images and it should include some static scenarios as well as image pairs with small displacement.

Moreover, a VIO system should be able to run at real-time, which is not the case of the presented system due to the high time consumption at the optical flow inference stage. Besides, only a tiny part of the dense optical field is taken into account and many likely useful informations are loss. Our system is not efficient regarding to run time and utilization rate of data. To improve the efficiency and make the system real-time capable, we can for example reduce the time spent on estimating optical flow by applying a smaller model with simpler structure and less parameters. We can also infer only the forward flow for each camera and applying other methods such as epipolar constraint and reprojection constraint to remove outliers instead of forward-backward consistency.

An other interesting direction for future works is improving robustness of the proposed system by alleviating problems caused by dynamic objects. To solve the failure on scenarios similar to sequence 07 of KITTI, a mask created by approaches such as Detectron2 [55] for dynamic objects such as vehicles and pedestrians can be useful. Another solution to this failure case is to re-identify the keypoints which are once occluded by dynamic objects.

Optical flow and stereo matching are similar problems. They can be solved not only by variational methods like the KLT tracker of the original Basalt VIO but also by neural networks. There are neural networks in particular for stereo matching which are developed based on optical flow networks such as [56] and [57]. Another interesting work direction to make advantage of neural networks on Basalt is to integrate additional networks for stereo matching, which then makes the front-end fully deep and avoids shortcomings in presence of occlusions and significant illumination changes of the traditional feature tracking methods.

7 Conclusion

In this semester thesis, we have proposed a visual-inertial odometry system extended from a state-of-the-art visual-inertial VIO, the Basalt VIO [1] by integrating deep optical flow in it for feature tracking. We replace the pyramidal KLT tracker in Basalt VIO with deep optical flow. To estimate optical flow for consecutive image pairs forwards and backwards, we adopted the pretrained models of LiteFLowNet and RAFT. To remove outliers, we employ a two-stage approach. At the first stage, we check the forward-backward-inconsistency of the predicted deep optical flow of correspondences. For further outliers filtering, we use epipolar constraint of the matched keypoints in stereo image pairs. The evaluation results of the proposed VIO system on KITTI Odometry dataset [2] show larger mean track length of the extracted keypoints and improved accuracy at pose estimation in most of the tested sequences compared to the original Basalt VIO. Nevertheless, in one particular sequence using deep optical flow results in failed pose estimation. One possible reason is, due to the strong robustness of deep optical flow to dynamic objects, keypoints on fast-moving objects are not identified as outliers with the adopted constraints and hence deteriorate the following pose estimation badly. On all sequences of the EuRoC MAV dataset, our integration acquires more accurate results than the original Basalt VIO. Although we can obtain more successfully tracked keypoints from frame to frame and therefore more accurate pose estimation by using deep optical flow, the inference time of the deep optical flow estimation strictly limits the real-time capability of our system. Additionally, only an extremely small amount of information in deep optical flow is in use. Therefore, the presented approach of integration is not efficient in general.

Overall, with this thesis, we show that w.r.t. accuracy, deep optical flow is a promising alternative to the pyramidal KLT tracker applied in Basalt VIO. However, the integrated system shows its limitation in robustness to scenarios with fast moving objects as well as large occlusion. Moreover, the introduced integration approach possesses low efficiency and is not real-time capable. For future research, in order to achieve real-time and robust performance while maintaining the state-of-the-art accuracy, approaches for utilizing deep optical flow efficiently as well as methods to filter out dynamic objects are in interest.

Bibliography

[1] V. Usenko, N. Demmel, D. Schubert, J. Stuckler, and D. Cremers, "Visual-Inertial Mapping With Non-Linear Factor Recovery," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 422–429, Apr. 2020. [Online]. Available: https://ieeexplore.ieee.org/document/8938825/

[2] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, Sep. 2013. [Online]. Available: http://journals.sagepub.com/doi/10.1177/0278364913491297

[3] 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, Sep. 2016, publisher: SAGE Publications Ltd STM. [Online]. Available: https://doi.org/10.1177/0278364915620033

[4] T.-W. Hui, X. Tang, and C. C. Loy, "LiteFlowNet: A Lightweight Convolutional Neural Network for Optical Flow Estimation," *arXiv:1805.07036 [cs]*, May 2018, arXiv: 1805.07036. [Online]. Available: http://arxiv.org/abs/1805.07036

[5] Z. Teed and J. Deng, "RAFT: Recurrent All-Pairs Field Transforms for Optical Flow," *arXiv:2003.12039 [cs]*, Aug. 2020, arXiv: 2003.12039. [Online]. Available: http://arxiv.org/abs/2003.12039

[6] H. Zhan, C. S. Weerasekera, J.-W. Bian, R. Garg, and I. Reid, "DF-VO: What Should Be Learnt for Visual Odometry?" *arXiv:2103.00933 [cs]*, Mar. 2021, arXiv: 2103.00933. [Online]. Available: http://arxiv.org/abs/2103.00933

[7] K. Yousif, A. Bab-Hadiashar, and R. Hoseinnezhad, "An Overview to Visual Odometry and Visual SLAM: Applications to Mobile Robotics," *Intelligent Industrial Systems*, vol. 1, no. 4, pp. 289–311, Dec. 2015. [Online]. Available: http://link.springer.com/10.1007/s40903-015-0032-7

[8] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge: Cambridge University Press, 2004. [Online]. Available: https://www.cambridge.org/core/books/multiple-view-geometry-in-computer-vision/0B6F289C78B2B23F596CAA76D3D43F7A

[9] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, no. 1, pp. 185–203, 1981. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0004370281900242

[10] B. D. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," p. 10.

[11] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski, "A Database and Evaluation Methodology for Optical Flow," *International Journal of Computer Vision*, vol. 92, no. 1, pp. 1–31, Mar. 2011. [Online]. Available: http://link.springer.com/10.1007/s11263-010-0390-2

[12] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A Naturalistic Open Source Movie for Optical Flow Evaluation," in *Computer Vision – ECCV 2012*, D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, vol. 7577, pp. 611–625, series Title: Lecture Notes in Computer Science. [Online]. Available: http://link.springer.com/10.1007/978-3-642-33783-3_44

[13] N. Yang, R. Wang, J. Stuckler, and D. Cremers, “Deep Virtual Stereo Odometry: Leveraging Deep Depth Prediction for Monocular Direct Sparse Odometry,” 2018, pp. 817–833. [Online]. Available: https://openaccess.thecvf.com/content_ECCV_2018/html/Nan_Yang_Deep_Virtual_Stereo_ECCV_2018_paper.html

[14] N. Yang, L. von Stumberg, R. Wang, and D. Cremers, “D3VO: Deep Depth, Deep Pose and Deep Uncertainty for Monocular Visual Odometry,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA: IEEE, Jun. 2020, pp. 1278–1289. [Online]. Available: https://ieeexplore.ieee.org/document/9157454/

[15] Y. Huang, B. Zhao, C. Gao, and X. Hu, “Learning Optical Flow with R-CNN for Visual Odometry,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, May 2021, pp. 14 410–14 416, iSSN: 2577-087X.

[16] T. Brox and J. Malik, “Large Displacement Optical Flow: Descriptor Matching in Variational Motion Estimation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 3, pp. 500–513, Mar. 2011. [Online]. Available: http://ieeexplore.ieee.org/document/5551149/

[17] D. Sun, S. Roth, and M. J. Black, “A Quantitative Analysis of Current Practices in Optical Flow Estimation and the Principles Behind Them,” *International Journal of Computer Vision*, vol. 106, no. 2, pp. 115–137, Jan. 2014. [Online]. Available: http://link.springer.com/10.1007/s11263-013-0644-x

[18] A. Bruhn, J. Weickert, C. Feddern, T. Kohlberger, and C. Schnorr, “Variational optical flow computation in real time,” *IEEE Transactions on Image Processing*, vol. 14, no. 5, pp. 608–615, May 2005, conference Name: IEEE Transactions on Image Processing.

[19] P. Fischer, A. Dosovitskiy, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox, “FlowNet: Learning Optical Flow with Convolutional Networks,” *arXiv:1504.06852 [cs]*, May 2015, arXiv: 1504.06852. [Online]. Available: http://arxiv.org/abs/1504.06852

[20] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, “FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks,” *arXiv:1612.01925 [cs]*, Dec. 2016, arXiv: 1612.01925. [Online]. Available: http://arxiv.org/abs/1612.01925

[21] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, “PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Salt Lake City, UT, USA: IEEE, Jun. 2018, pp. 8934–8943. [Online]. Available: https://ieeexplore.ieee.org/document/8579029/

[22] A. Ranjan and M. J. Black, “Optical Flow Estimation Using a Spatial Pyramid Network,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, HI: IEEE, Jul. 2017, pp. 2720–2729. [Online]. Available: http://ieeexplore.ieee.org/document/8099774/

[23] Z. Liang, Y. Feng, Y. Guo, H. Liu, W. Chen, L. Qiao, L. Zhou, and J. Zhang, “Learning for Disparity Estimation through Feature Constancy,” *arXiv:1712.01039 [cs]*, Mar. 2018, arXiv: 1712.01039. [Online]. Available: http://arxiv.org/abs/1712.01039

[24] H. Zhou, B. Ummenhofer, and T. Brox, “DeepTAM: Deep Tracking and Mapping,” 2018, pp. 822–838. [Online]. Available: https://openaccess.thecvf.com/content_ECCV_2018/html/Huizhong_Zhou_DeepTAM_Deep_Tracking_ECCV_2018_paper.html

[25] J. Wulff, D. J. Butler, G. B. Stanley, and M. J. Black, “Lessons and Insights from Creating a Synthetic Optical Flow Benchmark,” in *Computer Vision – ECCV 2012. Workshops and Demonstrations*, D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, A. Fusiello, V. Murino, and R. Cucchiara, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg,

2012, vol. 7584, pp. 168–177, series Title: Lecture Notes in Computer Science. [Online]. Available: http://link.springer.com/10.1007/978-3-642-33868-7_17

[26] E. Ilg, T. Saikia, M. Keuper, and T. Brox, "Occlusions, Motion and Depth Boundaries with a Generic Network for Disparity, Optical Flow or Scene Flow Estimation," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, vol. 11216, pp. 626–643, series Title: Lecture Notes in Computer Science. [Online]. Available: http://link.springer.com/10.1007/978-3-030-01258-8_38

[27] N. Mayer, E. Ilg, P. Häusser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, "A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation," Dec. 2015. [Online]. Available: https://arxiv.org/abs/1512.02134v1

[28] S. Poddar, R. Kottath, and V. Karar, "Evolution of Visual Odometry Techniques," *arXiv:1804.11142 [cs]*, Apr. 2018, arXiv: 1804.11142. [Online]. Available: http://arxiv.org/abs/1804.11142

[29] A. Geiger, J. Ziegler, and C. Stiller, "StereoScan: Dense 3d reconstruction in real-time," in *2011 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2011, pp. 963–968, iSSN: 1931-0587.

[30] R. Mur-Artal and J. D. Tardos, "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, Oct. 2017, arXiv: 1610.06475. [Online]. Available: http://arxiv.org/abs/1610.06475

[31] 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, Aug. 2018, arXiv: 1708.03852. [Online]. Available: http://arxiv.org/abs/1708.03852

[32] G. Silveira, E. Malis, and P. Rives, "An efficient direct approach to visual SLAM," *Robotics, IEEE Transactions on*, vol. 24, pp. 969–979, Nov. 2008.

[33] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-Scale Direct Monocular SLAM," in *Computer Vision – ECCV 2014*, ser. Lecture Notes in Computer Science, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 834–849.

[34] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 15–22, iSSN: 1050-4729.

[35] J. Engel, V. Koltun, and D. Cremers, "Direct Sparse Odometry," *arXiv:1607.02565 [cs]*, Oct. 2016, arXiv: 1607.02565. [Online]. Available: http://arxiv.org/abs/1607.02565

[36] 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)*, May 2017, pp. 2043–2050.

[37] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised Learning of Depth and Ego-Motion from Video," *arXiv:1704.07813 [cs]*, Jul. 2017, arXiv: 1704.07813. [Online]. Available: http://arxiv.org/abs/1704.07813

[38] C. Tomasi and T. Kanade, "Detection and Tracking of Point Features," p. 22.

[39] Jianbo Shi and Tomasi, "Good features to track," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition CVPR-94*. Seattle, WA, USA: IEEE Comput. Soc. Press, 1994, pp. 593–600. [Online]. Available: http://ieeexplore.ieee.org/document/323794/

[40] C. Harris and M. Stephens, "A Combined Corner and Edge Detector," in *Procedings of the Alvey Vision Conference 1988*. Manchester: Alvey Vision Club, 1988, pp. 23.1–23.6. [Online]. Available: http://www.bmva.org/bmvc/1988/avc-88-023.html

[41] E. Rosten and T. Drummond, “Machine Learning for High-Speed Corner Detection,” in *Computer Vision – ECCV 2006*, ser. Lecture Notes in Computer Science, A. Leonardis, H. Bischof, and A. Pinz, Eds. Berlin, Heidelberg: Springer, 2006, pp. 430–443.

[42] D. Lowe, “Object recognition from local scale-invariant features,” in *Proceedings of the Seventh IEEE International Conference on Computer Vision*. Kerkyra, Greece: IEEE, 1999, pp. 1150–1157 vol.2. [Online]. Available: http://ieeexplore.ieee.org/document/790410/

[43] “SURF: Speeded Up Robust Features | SpringerLink.” [Online]. Available: https://link.springer.com/chapter/10.1007/11744023_32

[44] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “ORB: An efficient alternative to SIFT or SURF,” in *2011 International Conference on Computer Vision*, Nov. 2011, pp. 2564–2571, iSSN: 2380-7504.

[45] N. Roma, J. Santos-Victor, and J. Tomé, “A Comparative Analysis Of Cross-Correlation Matching Algorithms Using a Pyramidal Resolution Approach,” May 2002.

[46] J. Reinders, *Intel threading building blocks: outfitting C++ for multi-core processor parallelism*, 1st ed. Beijing ; Sebastopol, CA: O’Reilly, 2007, oCLC: ocn156818622.

[47] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid, “DeepFlow: Large Displacement Optical Flow with Deep Matching,” in *2013 IEEE International Conference on Computer Vision*, Dec. 2013, pp. 1385–1392, iSSN: 2380-7504.

[48] J. Wulff and M. J. Black, “Efficient sparse-to-dense optical flow estimation using a learned basis and layers,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Boston, MA, USA: IEEE, Jun. 2015, pp. 120–130. [Online]. Available: http://ieeexplore.ieee.org/document/7298607/

[49] D. Kondermann, R. Nair, K. Honauer, K. Krispin, J. Andrulis, A. Brock, B. Gussefeld, M. Rahimimoghaddam, S. Hofmann, C. Brenner, and B. Jahne, “The HCI Benchmark Suite: Stereo and Flow Ground Truth with Uncertainties for Urban Autonomous Driving,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. Las Vegas, NV, USA: IEEE, Jun. 2016, pp. 19–28. [Online]. Available: http://ieeexplore.ieee.org/document/7789500/

[50] M. Jaderberg, K. Simonyan, A. Zisserman, and k. kavukcuoglu, “Spatial Transformer Networks,” in *Advances in Neural Information Processing Systems*, vol. 28. Curran Associates, Inc., 2015. [Online]. Available: https://papers.nips.cc/paper/2015/hash/33ceb07bf4eeb3da587e268d663aba1a-Abstract.html

[51] “RAFT,” Jan. 2022, original-date: 2020-03-27T03:18:25Z. [Online]. Available: https://github.com/princeton-vl/RAFT

[52] 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*. Vilamoura-Algarve, Portugal: IEEE, Oct. 2012, pp. 573–580. [Online]. Available: http://ieeexplore.ieee.org/document/6385773/

[53] D. Prokhorov, D. Zhukov, O. Barinova, A. Vorontsova, and A. Konushin, “Measuring robustness of Visual SLAM,” *arXiv:1910.04755 [cs]*, Oct. 2019, arXiv: 1910.04755. [Online]. Available: http://arxiv.org/abs/1910.04755

[54] S. Umeyama, “Least-squares estimation of transformation parameters between two point patterns,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 4, pp. 376–380, Apr. 1991, conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

[55] “facebookresearch/detectron2,” Jan. 2022, original-date: 2019-09-05T21:30:20Z. [Online]. Available: https://github.com/facebookresearch/detectron2

[56] L. Lipson, Z. Teed, and J. Deng, “RAFT-Stereo: Multilevel Recurrent Field Transforms for Stereo Matching,” *arXiv:2109.07547 [cs]*, Sep. 2021, arXiv: 2109.07547. [Online]. Available: http://arxiv.org/abs/2109.07547

[57] P. Liu, I. King, M. Lyu, and J. Xu, “Flow2Stereo: Effective Self-Supervised Learning of Optical Flow and Stereo Matching,” *arXiv:2004.02138 [cs]*, Apr. 2020, arXiv: 2004.02138. [Online]. Available: http://arxiv.org/abs/2004.02138