Visual and Thermal Odometry with Deep Neural Networks

Muhamad Risqi Utama Saputra
Kellogg College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy

Hillary 2020

To my wife Mega, my son Awliya, and my parents ...

Acknowledgements

First and foremost, I would like to sincerely thank my supervisor Prof. Niki Trigoni and Dr. Andrew Markham for hosting me into their research group. Their invaluable guidance made me understand how to conduct a high-quality research and influenced my research outlook. Most of all, I thank them for their patience and understanding during difficult times, and keep believing in me even when I doubt myself.

I am grateful to my collaborator in the firefighter projects: Pedro, Yasin, Chris, Dai, Wei, and Johan. I learn a lot of technical stuff from them which helps me finish my study. I hope your hard work and sleepless night will pay off. I also want to thank all other members of Cyber Physical System (CPS) research group: Bo, Changhao, Stefano, Sen Wang, Ronnie, Prince, Peijun, Linhai, Asyraf, Shuyu, Marion, Rui, Bing, Ada, and Qingyong, which provides supportive and fun environment to conduct enjoyable researches and academic activities.

I greatly thank my lovely wife and son, Mega and Awliya, who always support me during our times in Oxford. Special thanks to my wife who is willing to sacrifice her time and career in our home country to accompany me pursuing my doctoral degree. I am forever grateful to my parents for their care and relentless support since I was child living in a small town in Indonesia, until now when I became a Doctor from the University of Oxford.

Finally, I would like to acknowledge Indonesia Endowment Fund for Education (LPDP), Ministry of Finance Indonesia, for their generosity to

fund my DPhil study. Part of this research was conducted within the context of "Pervasive, Accurate, and Reliable Location-Based Services for Emergency Responders" project, funded by the US National Institute of Standards and Technology (NIST) grant No. 70NANB17H185.

Abstract

Accurate camera ego-motion estimation, widely known as Visual Odometry (VO), remains a key prerequisite for many applications in computer vision and robotics. Conventional VO, which relies on hand-crafted feature engineering, is prone to drift and can easily lose track as the extracted features contain outliers and unknown noise. This is problematic for application which requires robustness and high level accuracy such as for tracking UAV position in underground tunnel or estimating firefighter position in emergency operation. To alleviate the problem of noisy feature engineering, machine learning algorithms, especially Deep Neural Networks (DNN), have been used in the past few years to automatically learn robust odometry features from large amounts of data. However, despite some promising results, several fundamental drawbacks still exist in terms of accuracy, efficiency, and applicability to visually-denied environments.

The work presented in this thesis tackles these shortcomings by proposing a novel network architecture and optimization strategy for DNN-based odometry estimation. To address issues of accuracy and long-term consistency, we propose to train DNN-based VO using both a windowed-based composite transformation loss and relative transformation loss through curriculum learning. With this approach, we can improve the generalization ability of the network for both translation and rotation by 21% and 16% respectively. We also propose the use of an attention network to conditionally re-weight image features such that the network can produce more accurate poses whilst being more amenable to interpretation. This

method improves translation and rotation estimation by 27.8% and 43.1% respectively over the model without attention.

The second contribution deals with the efficiency problem of DNN-based VO by proposing the first distillation approach for camera pose regression. We demonstrate that distilling knowledge from a deep pose regression network can be done effectively if we emphasize the knowledge transfer only when we trust the teacher network prediction. We also show that a distilled network can be further compressed with factorization and could be more generalizable due to low-rank constraints. Our proposed approach can reduce the number of student parameters by up to 92.95% (2.12× faster) whilst keeping the prediction accuracy very close to that of the teacher.

Finally, we deal with the issue of tracking in visually-denied environments by proposing the first DNN-based thermal-inertial odometry system. Since thermal images inherently lack robust features, we design the network to not only extract features from the thermal images, but to also hallucinate visual features given thermal image as the input. Through extensive evaluation across two datasets, we conclude that our proposed method can produces accurate odometry estimation with less than 2 m absolute trajectory errors on average.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Positioning systems remain a key enabler for many applications in computer vision and robotics. Primary examples include tracking mobile device position for augmented reality, robot navigation, autonomous vehicles, or pedestrian localization. Global Positioning System (GPS) is the de facto standard technology for location tracking in outdoor environments. However, GPS provides inadequate accuracy and robustness within indoor environments as building structures highly attenuate the GPS signal. Even an optimized GPS positioning solution for indoor environment developed by Qualcomm can only provide location accuracy between 5 to 50 m [100]. This low accuracy is unsuitable for application in GPS-denied environments such as for tracking UAV position in underground tunnel or estimating firefighter position in emergency operation which typically requires sub-metre level accuracy.

With the motivation to provide precise location tracking in GPS-denied environments, many alternative positioning systems have been developed especially in the form of infrastructure-free solutions. Infrastructure-free solution tracks the position of mobile agent without requiring external hardware but instead utilizing built-in (wearable) sensors to estimate the relative motion from the previous position. Imaging sensors (e.g. RGB or depth camera) or Inertial Measurement Unit (IMU) sensors are typically employed to estimate the relative motion since they are cheap and ubiquitous due to the omnipresence of mobile devices. Moreover, imaging sensor can be

used beyond position estimation, enabling better perception through scene understanding and semantic information.

Given IMU and image sensor data, relative transformation can be estimated using odometry algorithms. The odometry estimation is performed by exploiting the change of sensor observation from subsequent sensor data. Model-based approaches (e.g. Kalman filter family, bundle adjustment, etc.) are then used to calculate the agent transformation as a rigid body motion in which the motions are typically represented as 3 or 6 Degree-of-Freedom (DoF) poses. However, the accuracy and robustness of such system is often insufficient in particular environment and applications, especially when sub-metre accuracy is required such as in firefighter tracking. An IMU-based strapdown navigation system, for example, has exponential error growth due to double integration operations which leads to very poor position estimation performance [20]. Similarly, feature-based approaches used in Visual Odometry (VO) are very sensitive to outliers [17, 18, 41] and easily lose track in texture-less areas or when the field-of-view of the camera is obstructed by dynamic objects [81, 1, 79, 6]. Moreover, the performance of vision-based odometry estimation largely depends on the benign visibility. In adverse illumination conditions and/or in the presence of airborne particulates (e.g. dust, soot, smoke, etc.), visual odometry collapses [76].

In the past few years, machine learning algorithms, especially Deep Neural Networks (DNN), have revolutionized the field of computer vision and language understanding. DNNs achieve state-of-the-art results in many tasks such as image classification, object detection, or machine translation. Through multiple hierarchical layers of neural networks, a DNN learns a high-level abstraction of the data and is able to capture real-world complexities, which are typically very difficult to model by using hand-crafted feature engineering. Because of this ability to learn non-linear structures in the data, DNNs have recently started to be used in other fields like VO estimation, which indeed requires strong non-linear function estimation. By directly predicting the output from the input data, DNNs alleviate some problems in classical

VO (e.g noisy feature correspondences), which in turn can make the VO system more robust against input perturbation.

1.1 Challenges

Classical odometry estimation using RGB camera dates back to 2004 when Nister et al. [124] proposed "Visual Odometry" as an approach to estimate the camera ego-motion by observing the apparent changes in pixel position captured from a moving camera. By extracting key points (e.g. corners) and finding their correspondences among consecutive frames, the camera pose can be estimated by using 8-point [56] or 5-point [123] algorithms. Many researchers then improved this approach by using more robust and efficient key points extractor and matching algorithms (e.g. SIFT, SURF, FAST, ORB, etc.) [140, 24, 120], incorporating outliers rejection (e.g. RANSAC, PROSAC, etc.) [80, 138] and second order-based optimization (e.g. bundle adjustment) [117, 114, 86]. Despite their great performances, these feature-based visual odometry remain struggle in feature-less areas (e.g. corridor with flat/white wall) as it leads to insufficient number of point correspondences. Dynamic scenes in front of the camera are also problematic since it generates spurious correspondences [81, 1, 79, 6]. To alleviate the difficulty of matching key points in feature-less area, researchers developed visual odometry based on line segments. The algorithms in [44, 96] extract line segments as landmarks and use them to improve the robustness of point correspondences in texture-less scene. Nonetheless, this technique will fail if there are few line features in the observed scene as described by [105]. This technique also faces difficulty when the camera point of view abruptly changes or there is obstruction in front of the camera, in which the properties of the previously tracked line segments (e.g. the location of extremities) dramatically alters [96]. Ambiguous line segments might also be found in repetitive building structure, confusing the matching process. Although these ambiguous lines can be filtered by identifying perpendicular line segments, such structural information does not always persist in the observed

environment. Another point of view to tackle this problem is by using the whole image information (not limiting to several key points) as described in direct approach. The direct approaches [75, 142, 122, 40, 36] estimate and optimize the camera poses directly by minimizing the photometric errors. Nonetheless, this approach needs good (stable) initialization, sensitive to rolling shutter, and typically requires more computational time to perform whole image alignment than feature-based approach. Moreover, all these hand-engineered approaches require manual and careful tuning of the hyper-parameters (e.g. thresholding) to make it work well for different motion models and environment conditions.

To alleviate the weakness of hand-engineering approaches, in the last couple of years deep learning-based approaches have been developed for visual odometry estimation. As an end-to-end system, deep learning-based visual odometry learns a mapping function given consecutive image pairs as the input and 6-DoF camera poses as the output. The neural networks are trained to implicitly encode feature correspondences or optical flow information [161, 109] by minimizing the relative camera transformation loss between two consecutive images. Recent work [162] also employ recurrent network to model the long term dependencies of camera poses. This data driven approach has shown to be more robust in texture-less areas as it does not rely on key points extraction. However, in rich textured environment, the accuracy of such system is still less than the one produced by feature-based approach [70]. As the network also needs large amount of weights, it requires large computation time, preventing real time implementation in resource-constrained devices [162]. Finally, as most works developed for visible camera, it is unusable in darkness or in the presence of airborne particulates. It is also unclear whether existing network architecture can be used by another odometry modalities that are immune to darkness (e.g. thermal images).

1.2 Research Problem

As stated before, recent developments of DNN-based VO have shown promising results [161, 162, 118, 173, 176]. However, there are several fundamental drawbacks that make the VO system impractical.

DNN-based VO is typically trained by minimizing the relative transformation error between two images. However, only minimizing relative loss does not guarantee the consistency of the absolute transformation after composing the relative estimates. On the other hand, directly minimizing the absolute transformation error leads to degraded performance compared to only using relative loss. This shortcoming can make the VO system produce highly erroneous absolute transformation output. Another problem in the current implementation of DNN-based VO is that it employs the information of the entire image which often leads to sub-optimal performance. The design of geometry-based approaches suggests that not all information in the image space is useful or equally important. For instance, some areas are texture-less which make feature matching difficult and noisy, or some parts of the image belong to dynamic objects, and thus violate the epipolar constraints.

Another factor that makes DNN-based VO difficult to implement is the requirement for high computational resources. A DNN typically needs tens or hundreds of millions of weights to effectively capture the structure of the data. As a consequence, a DNN requires significant memory and computation, which prevents the model from being implemented on a small embedded device. On the other hand, a VO system is usually required to produce pose estimates in real-time. A DNN, despite being robust against noisy feature correspondences, cannot necessarily run in real-time in a resource-constrained environment. To compound the issue, even if we are able to reduce the computation time of DNN-based VO by sub-sampling the image frames, it compromises system accuracy.

Lastly, while standard DNN-based VO is useful in many circumstances, applications are limited to scenarios where the observed environment has sufficient illumi-

nation. There are, however, scenarios that require the VO system to be functional in visually-denied environments with limited or even no illumination. An example odometry application in this category is tracking a firefighter in emergency response scenarios or locating aerial robots in underground mines and tunnels. In such visually-denied environments, alternative modalities such as thermal images and Inertial Measurement Unit (IMU) are necessary to make odometry practical. While thermal cameras are commonly used in these scenarios, usage is usually limited to perception and inspection. It remains an open question as to whether we can estimate odometry accurately from a thermal imaging system, as the camera captures the temperature profile of the environment rather than the scene geometry.

1.3 Objectives

The main objectives of this research are summarized as follows:

1. A comprehensive **review** of the state of the art algorithms for visual and thermal odometry, either based on feature correspondences or machine learning models.

2. Development and evaluation of a novel DNN-based VO architecture and optimization strategy to enhance the **accuracy** of the VO system.

3. Construction of an **efficient** DNN-based VO model and evaluation of its accuracy and computation time in a resource constrained environment.

4. Investigation of alternative modalities for **visually-denied** environments, such as thermal imaging or IMU systems, development of DNN-based thermal-inertial odometry, and evaluation of the proposed system.

1.4 Contributions

The general contribution of this thesis is the investigation and design of novel neural network architectures and optimization strategies for accurate and efficient visual and thermal odometry. The details of the contributions are described as follows:

- **Contribution 1**: To enhance the accuracy of a DNN-based VO system, we take two different approaches. The first one is from the optimization point of view, while the second one is from the perspective of network architecture. From the optimization perspective, we propose a novel objective function by minimizing the relative and composite transformation error over small windows via bounded pose regression loss. As minimizing the composite loss is more difficult to make converge, we propose an optimization strategy based on curriculum learning to effectively learn the proposed objective. In this geometric-based curriculum learning strategy, we train the network by gradually increasing the difficulty of the objective function. To realize this system, we construct a DNN consisting of a cascade optical flow network and differentiable window composition layers. We perform extensive experiments on two public datasets and one self-collected dataset and show that the proposed system outperforms state of the art feature-based and learning-based approaches. From the architecture perspective, we propose a new attention network to place relative importance of image features such that the network can harness more important features for VO estimation. Two architectures are explored: a model that learns a joint spatial-wise attention map for translation and rotation; and a model that learns decoupled attention maps for translation and rotation.

- **Contribution 2**: For efficiency, we develop a novel Knowledge Distillation (KD) approach for VO, which is the first KD applied for odometry regression. As KD normally only works for classification due to the absence of dark knowledge in the regression network, we develop a new KD approach by introducing the idea of confidence scores in the knowledge transfer process. We use teacher loss as the confidence value to measure how much we can trust the teacher network during the learning process. With this confidence score, the student training process will rely more on easier examples that the teacher network is

good at. We also fuse together our proposed KD approach and existing factorization approaches to further generalize the system. An extensive evaluation was carried out on two public datasets in order to demonstrate the efficacy of the proposed technique.

- **Contribution 3**: We construct the first DNN-based thermal-inertial odometry for application in visually-denied environments. As thermal images are largely textureless and lack sufficient features for accurate odometry estimation, we propose to not only extract features from thermal images, but to additionally learn to hallucinate visual features similar to the ones extracted from a DNN-based VO. Then, the proposed network will produce three feature channels, i.e. thermal, hallucinated visual, and IMU features. Since each channel comes with their own limitations and strengths, we employ selective fusion to automatically learn the most suitable approach to feature fusion given particular input conditions. Experimental evaluation in our self-collected dataset, which consists of hand-held and mobile robot data in benign (e.g. good illumination), dark, and smoke-filled environments, shows the effectiveness of the proposed DNN model.

1.5 Thesis Structure

The remainder of this thesis is organized as follows:

- Chapter 2 discusses the state-of-the-art visual and thermal odometry approaches. We review existing techniques and systems including both features-based and learning-based approaches.

- Chapter 3 presents the first contribution of this thesis - a novel architecture and optimization strategy for training deep visual odometry models. Extensive evaluation of the proposed approach on the public and self-collected dataset is provided.

- Chapter 4 describes an approach to construct an efficient deep network for visual odometry. We present a novel knowledge distillation approach for odometry regression and discuss a fusion mechanism between distillation and compression. We evaluate the proposed techniques on two public benchmarks for visual odometry.

- Chapter 5 discusses odometry estimation in visually-denied environments using thermal and inertial data. We present the first DNN-based thermal-inertial odometry system which is tested on two types of data (i.e. hand-held and mobile robot) in variety of environment conditions (i.e. good lighting, darkness, and smoke-filled).

1.6 Publications

The following are first-authored publications describing the main contributions of this thesis:

1. M. R. U. Saputra, A. Markham, and N. Trigoni. "Visual SLAM and Structure from Motion in Dynamic Environments: A Survey". **ACM Computing Surveys (CSUR)**, 51(2), 2018.

2. M. R. U. Saputra, Pedro P. B. de Gusmao, S. Wang, A. Markham, and N. Trigoni. "Learning Monocular Visual Odometry through Geometry-Aware Curriculum Learning". In **IEEE International Conference on Robotics and Automation (ICRA)**, 2019.

3. M. R. U. Saputra, Pedro P. B. de Gusmao, Y. Almalioglu, A. Markham, and N. Trigoni. "Distilling Knowledge From a Deep Pose Regressor Network". In **IEEE/CVF International Conference on Computer Vision (ICCV)**, 2019.

4. M. R. U. Saputra, Pedro P. B. de Gusmao, C. Xiaoxuan Lu, Y. Almalioglu, S. Rosa, C. Chen, J. Wahlstrom, W. Wang, A. Markham, and N. Trigoni. "DeepTIO: A Deep Thermal-Inertial Odometry with Visual Hallucination". In **IEEE Robotics and Automation Letters (RA-L)**, Vol. 5, No. 2, April 2020, and was presented in **IEEE ICRA 2020**.

The following are other publications (and paper under review) that the author contributed to during his DPhil study:

1. Y. Almalioglu, **M. R. U. Saputra**, P. P. B. de Gusmao, A. Markham, and N. Trigoni. "GANVO: Unsupervised Deep Monocular Visual Odometry and Depth Estimation with Generative Adversarial Networks". In IEEE International Conference on Robotics and Automation (ICRA), 2019.

2. W. Wang, **M. R. U. Saputra**, P. Zhao, P. P. B. de Gusmao, B. Yang, C. Chen, A. Markham, and N. Trigoni. "DeepPCO: End-to-end Point Cloud Odometry through Deep Parallel Neural Network". In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2019.

3. Y. Almalioglu, M. Turan, A. E. Sari, **M. R. U. Saputra**, P. P. B. de Gusmao, A. Markham, and N. Trigoni. "SelfVIO: Self-Supervised Deep Monocular Visual-Inertial Odometry and Depth Estimation". Under review in IEEE Transactions on Robotics.

4. C. Xiaoxuan Lu, **M. R. U. Saputra**, P. Zhao, Y. Almalioglu, P. P. B. de Gusmao, C. Chen, K. Sun, N. Trigoni, and A. Markham. "Single-chip mmWave Radar Aided Egomotion Estimation via Deep Sensor Fusion". Under review in ACM SenSys 2020.

Chapter 2

Background

2.1 Visual Odometry

The term Visual Odometry (VO) was coined by Nister et al. [124] in 2004 as an approach to estimate the motion of an agent (e.g. robot, mobile device, autonomous vehicle) solely by using information from the camera. The term odometry itself is taken from its similarity with wheel odometry estimation, in which a vehicle estimates its motion by integrating over time the number of turns of its wheels [41]. VO utilizes a very similar idea but instead of using wheels to calculate the motion, VO employs the change of the scene captured from a moving camera. VO plays an important role as a main or supplementary modality to many positioning systems (e.g. global positioning system, IMU, laser odometry, etc.) especially for applications in GPS-denied environments.

Initially, VO techniques were mainly developed based on geometry. However, as a result of advances in learning algorithms, many learning based VO solutions have been proposed in the last three years. Based on the recent popularity of learning based methods, it is timely to consider learning-based approaches as an alternative class of techniques for VO. Thus, we can divide the VO algorithms into two main categories: based on geometry and those based on learning.

2.1.1 Geometry-based Approaches

Geometry-based VO estimates the camera motion based on hand-engineered feature extraction techniques and multiple-view geometry algorithms developed by the computer vision community. Most state-of-the-art VO algorithms fall into this family and have been developed for more than three decades. They typically consist of the following building blocks: camera calibration, feature extraction, feature matching, outlier rejection, pose estimation, and back-end optimization through Bundle Adjustment (BA) [61].

2.1.1.1 Calibration

Camera calibration, also referred to as camera resectioning, is the process of estimating the internal parameters of the camera. This process is fundamental to obtain accurate odometry estimation. There are two categories of camera parameters that can be extracted during the camera calibration step, namely intrinsic and extrinsic. Intrinsic parameters include focal length, image sensor format, and principal point. Extrinsic parameters represent the coordinate system transformation, which is usually used for transformation between the world coordinates and the camera coordinates. Although the camera ego-motion still can be estimated without a calibration step, with the knowledge of camera parameters, the pose can be calculated up to a similarity transformation (the rigidity and uniform scale are preserved). Nevertheless, for VO estimation, we only require the intrinsic values during the calibration since the extrinsic values will be estimated gradually as the camera moves over time.

Let $M = [X, Y, Z, 1]^T \in \mathbb{P}^3$ be a 3D world point in homogeneous coordinates. The projection of M to an image plane can be estimated as $m = [fX/Z, fY/Z, 1]^T = [u, v, 1]^T \in \mathbb{P}^2$ where f is the focal length of the camera. By using a pinhole camera model, the camera calibration step recovers the camera projection matrix $P \in \mathbb{R}^{3\times4}$ such that

$$m \simeq PM \tag{2.1}$$

where $\simeq$ means equal up to a scale factor. Using QR decomposition, P can be factorized into

$$P = \lambda K[R|t] \tag{2.2}$$

where $K \in \mathbb{R}^{3\times3}$ is the camera intrinsic matrix, $R \in SO(3)$ is the rotation matrix, $t \in \mathbb{R}^3$ is the translation vector, and λ is an unknown scale factor. The intrinsic matrix K, the only parameter that will be used in the next pipeline, has the following form

$$K = \begin{pmatrix} \alpha_x & \gamma & \mu_x \\ 0 & \alpha_y & \mu_y \\ 0 & 0 & 1 \end{pmatrix} \tag{2.3}$$

where $\alpha_x = fm_x$ and $\alpha_y = fm_y$ represent the focal length (in pixels), m_x and m_y represent the number of pixels per world unit in the x and y direction, μ_x and μ_y are the optical center in pixel, and γ is the skew parameter.

Early work in camera calibration needed two or three planes orthogonal to each other which requires expensive calibration apparatus and an elaborate setup [38]. An easier technique developed by Z. Zhang [175] only needs a checkboard pattern at few different orientations. The camera parameters can then be estimated by using homography. A complete description of Zhang's calibration method can be found in [175].

2.1.1.2 Feature Extraction and Matching

In geometry-based VO, salient features are extracted to find image correspondences: a set of features in an image that is similar to a set of features in another image. The computer vision community has developed a large number of feature extraction techniques. While early work such as [155] and [124] made use of Harris corner detector [60], more recent works [165, 141] employ robust feature detection techniques such as Scale Invariant Feature Transform (SIFT) [103] or its lightweight variants like Speeded Up Robust Features (SURF) [7]. SIFT employs Difference of Gaussian (DoG) to find scale invariant features as local extrema among image pyramids, whilst

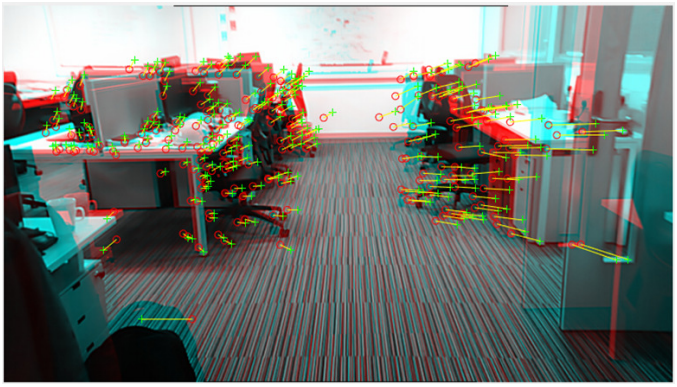


Figure 2.1: An example of feature extraction and matching process using SURF features and descriptor.

SURF uses a box filter as an approximation of the determinant of the Hessian to achieve faster computation time. However, since SIFT and SURF are still considered computationally expensive, a faster approach such as Features from Accelerated Segment Test (FAST) [133] is utilized for real-time applications [83, 99].

After features are extracted, feature matching techniques are employed in order to find the correspondences. The techniques can be divided by how far the distance between the optical centers of two cameras (termed *baseline/parallax*) are separated. For short baselines, optical flow-based techniques (e.g. Kanade-Lucas-Tomashi (KLT) tracker [106]) can be used for matching. On the contrary, for long baselines, highly discriminative feature descriptors (e.g. SIFT [103], SURF [7], BRIEF [19], etc.) are necessary to find correspondences by calculating dissimilarity between those descriptors. In order to find the candidate matches among salient features, exhaustive matching or Nearest Neighbor can be employed. Unfortunately, using these feature matching techniques does not guarantee perfect correspondences especially when the data contains outliers. Adoption of robust estimators (e.g. RANSAC [39], PROSAC [26], MLESAC [152], etc.) are useful to reject outliers and handle false correspondences. Figure 2.1 depicts an example of matched features using SURF feature descriptors.

2.1.1.3 Epipolar Geometry and Pose Estimation

Epipolar geometry describes geometric relations between two perspective views of the same 3D scene. Given m and m' as the location of feature point on the first and the second image respectively, the following epipolar constraint is satisfied

$$m'^T F m = 0 \tag{2.4}$$

where $F \in \mathbb{R}^{3\times3}$ is the fundamental matrix with $rank(F) = 2$ and has 7 Degrees-of-Freedom (DoF). The epipolar constraint tells that any 3D point M and its projection on the image plane (m and m') should lie on the epipolar plane. Furthermore, any image point m must also lie on the epipolar line on the corresponding image such that $l' = Fm$ is the epipolar line corresponding to m and $l = F^T m'$ is the epipolar line corresponding to m'. It implies that we can search for corresponding points on another image along a line instead of over a 2D image region which can significantly reduce the complexity of the matching algorithm.

To recover F, we need to convert the bilinear form of the epipolar constraint into a null-space problem by leveraging Kronecker product and vectorized operator (see [61] for detail). Then, by stacking 5 (using 5-point algorithm by [123]) or 8 (using 8-point algorithm by [56]) or more image points, we can solve the resulting linear equations by a least squares method. In practice, we might need to normalize the coordinate system by transforming m into $\tilde{m} = Tm$ where T is the normalization transformation consisting of a translation and scaling. Moreover, we might need to enforce a singularity constraint such that $det(F) = 0$ since the output matrix from the least squares method does not always follow the properties of the fundamental matrix. More details and variations of the method can be seen in [61].

As the fundamental matrix F describes the relationship between the corresponding image points in pixel coordinate, it can be decomposed into

$$F = K'^{-T}[t]_x R K^{-1} \tag{2.5}$$

where K and K' are the camera calibration matrix in the first and the second image respectively, R is rotation matrix, and $[t]_x$ are skew-symmetric matrix of translation vector t. If the camera calibration matrix is known, it is preferable to work with the normalized coordinates such that $\hat{m} = K^{-1}m$ and the epipolar constraint becomes

$$\hat{m}'^T E\hat{m} = 0 \tag{2.6}$$

where $E \in \mathbb{R}^{3\times3}$ is the Essential matrix, a singular matrix that has only 5 DoF. Comparing both epipolar constraints from the fundamental matrix and the essential matrix, it follows that $E = K'^T FK$. This essential matrix can also be decomposed into $E = [t]_x R$ by using Singular Value Decomposition (SVD). Unfortunately, there are four possible solutions, but we can select the valid one by projecting an image point using the camera matrix and observing which one lies in front of both images. If we have three views, a trifocal tensor [154] can be utilized to estimate the camera pose. In case some 3D points of the scene have been reconstructed, camera poses can be obtained with respect to the 3D model by solving the perspective-n-point problem (e.g. using P3P algorithm [43]).

2.1.1.4 Robust Estimator and Outlier Rejection

The estimated feature correspondences are potentially noisy and contain outliers. Computing the fundamental or essential matrix from this set of noisy image correspondences will not be accurate. A robust estimator such as Random Sample Concensus (RANSAC) is usually employed to obtain inlier points given the epipolar constraint. RANSAC samples n random points among all correspondences and computes F (e.g. by applying the 8-point algorithm) based on those n points. Then for every putative correspondence, RANSAC determines which one belongs to the inlier set by computing distance metric $d_\perp$ such that $d_\perp < \tilde{t}$ where $\tilde{t} \in \mathbb{R}$ is a threshold value. The process of sampling and computing inliers continues until F with the largest cardinality of inliers is found.

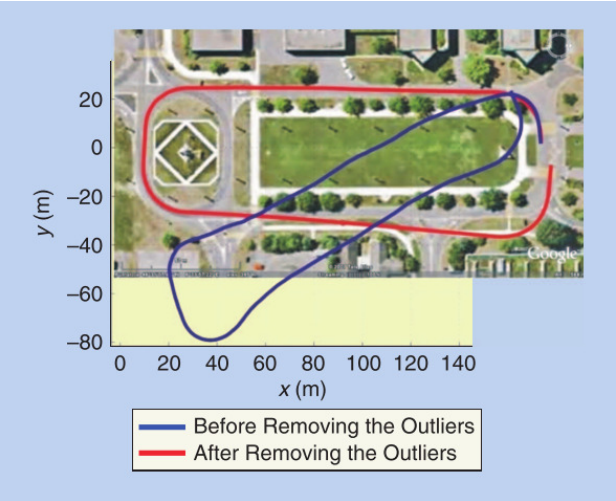


Figure 2.2: Comparison between VO trajectories estimated before and after outlier rejection [41].

For j number of image points, solving F under the RANSAC scheme is basically the same as finding the minimum of the cost function C defined as

$$C = \sum_j \rho(d_{\perp j}), \quad \text{where} \quad \rho(d_\perp) = \begin{cases} 0, & d_\perp < \tilde{t} \\ \text{constant}, & d_\perp > \tilde{t} \end{cases} \tag{2.7}$$

This implies that inliers score nothing while each outlier scores a constant penalty. In other words, if $\tilde{t}$ is large, then all solutions will have the same cost since all the matches will be considered as inliers. At no extra cost, this optimization function can be changed into

$$C = \sum_j \rho_2(d_{\perp j}), \quad \text{where} \quad \rho_2(d_\perp) = \begin{cases} d_\perp, & d_\perp < \tilde{t} \\ \tilde{t}, & d_\perp > \tilde{t} \end{cases} \tag{2.8}$$

This new cost function means that outliers are still given fixed penalty but inliers are scored based on how well they fit the data. This modification of RANSAC is called M-estimator Sample Consensus (MSAC). Another modification estimates the solution that maximizes the likelihood of observing inliers (dubbed MLESAC) [152]. Figure 2.2 shows the comparison between VO trajectories estimated before and after removing outliers [41].

2.1.1.5 Bundle Adjustment (BA)

The final stage of the VO pipeline is back-end optimization through Bundle Adjustment (BA). BA is an optimization technique originating from the photogrammetry community, which intends to adjust (optimize) the bundle of light rays from each 3D world point such that its reprojection error on image plane is small. BA refines both camera pose and 3D points to avoid the drifting problem by means of minimizing reprojection errors.

Specifically, for i number of images and j number of points, BA minimizes the following objective function

$$E = \sum_{i} \sum_{j} \theta_{ij} \left\| m_{ij} - \phi(R_i, t_i, M_j) \right\|^2 \tag{2.9}$$

where $\theta_{ij} = 1$ when point j is visible in image i and $\theta_{ij} = 0$ otherwise. m_{ij} denotes the tracked point feature j in image i. $\|.\|^2$ denotes the L2-norm and $\phi(R_i, t_i, M_j)$ represents the perspective projection of 3D point M_j after rotation by R_i and translation by t_i.

BA loss function is usually solved by using the Levenberg-Marquardt (LM) algorithm. LM is an iterative non-linear optimization method that combines the advantages of gradient descent and the Newton method. It changes the following Newton update function at $(t+1)$-th iteration

$$m_{t+1} = m_t - H^{-1} g \tag{2.10}$$

where H^{-1} is the inverse of Hessian and g is the gradient, into this damped form

$$m_{t+1} = m_t - (H + \lambda I)^{-1} g \tag{2.11}$$

where λ is a scalar value and I is the identity matrix. If $\lambda = 0$, it acts as the Newton method that promises fewer iterations when close to a local or global minimum. On the other hand, when $\lambda > 0$, it becomes gradient descent with step size $1/\lambda$ which guarantees to converge to a local or global minimum although it may require many

iterations. As the Hessian can be approximated as $H \approx 2J^T J$ and the gradient is $g = 2J^T r$, where J is the Jacobian matrix and r is the residual function of the input arguments (L2-norm in BA), then LM's iterative function becomes

$$m_{t+1} = m_t - (J^T J + \lambda I)^{-1} J^T r \tag{2.12}$$

This final form also avoids the computation of the inverse Hessian which is slow for a large matrix.

The choice of how to implement this non-linear least squares optimization depends heavily on the application, especially when considering the trade-off between accuracy and execution time. Optimizing the whole camera pose and 3D points of the scene through Global Bundle Adjustment (GBA) can be implemented if accuracy is a priority. On the other hand, when real-time implementation is desired, optimizing only over a small window of recent images by Local Bundle Adjustment (LBA) can be a method of choice although the pose estimation result is only locally optimized. Another option is carrying out BA in a multicore system as in [166].

2.1.1.6 Taxonomy of Geometry-based Approaches

As it has been described before, most geometry-based approaches are based on feature matching. However, there are variants that do not rely on feature correspondences but instead directly utilize the image appearance to match image pairs. The following provides a taxonomy of VO algorithms based on how the matching process is performed:

1. **Feature-based Approaches**. As explained in the previous section, feature-based approaches extract salient features (e.g. features, corner, etc.) from consecutive images and match those features based on hand-engineered feature descriptors (e.g. SIFT, SURF, etc.). VO algorithms that belongs to this category include the original work by Nister (2004) [124], followed by Maimone et al. (2007) [108], Howard (2008) [65], Parra et al. (2010) [125], Konolige et

al. [87], Naroditsky et al. (2011) [121], Badino et al. (2013) [5], and Huang et al. (2017) [66]. The evolution of this class of algorithms is mainly due to the development of robust and efficient feature extraction and matching (from Harris corner to SIFT/SURF/FAST/ORB, etc.) and robust outliers detection (e.g. the inclusion of RANSAC, PROSAC, etc. to the pipeline). In order to improve robustness, feature-based approaches have also been extended by considering fusion with other modalities like inertial [98], depth [66], or lidar point clouds [174]. Interesting applications demonstrated by the research community include locating ground vehicles in rough terrain when wheel odometry is not reliable [87] or tracking the position of Mars rovers as demonstrated by [108].

2. **Appearance-based (Direct) Approaches**. Instead of extracting important image features, appearance-based approaches (or widely known as direct approach) calculate the changes in appearance (represented by the pixel intensity) from two images to infer the odometry estimation. This method computes the displacement of brightness patterns similar to the one demonstrated by optical flow algorithms. Originally, this method was developed based on a template matching algorithm. However, modern direct approaches use non-linear optimization to minimize photometric error between the two images.

    - **Template Matching**. In a template matching-based approach, the algorithm selects a patch (small part of image) from the current image, which is the template, and attempts to find the corresponding patch in the next image frame. The algorithm computes the similarity between the two patches by using sum square differences or normalized cross correlation [3]. After finding the matches (patches with the highest correlation), the pixel displacements in x and y directions between the two correlated patches are then estimated. These displacements can be converted to a physical displacement (e.g. in metres) by using the camera intrinsic parameters. VO

work that belongs to this category includes Bellotto et al. [8], Lovegrove et al. [102], Gonzalez et al. [50], and Gonzalez et al. [51].

- **Photometric Error**. In a photometric error-based approach, the algorithm warps the target image using the information from the estimated camera poses and instrinsic parameters, such that the intensity errors between two (patch) images are minimized. In this case, the algorithm tries to directly calculate the camera ego motion that minimizes the sum square errors of the photometric loss without explicitly extracting feature correspondences. Compared to feature-based approaches, this direct method uses the whole image information to generate the odometry and is able to reconstruct the whole image instead of sparse image points. However, the computation of the photometric error is heavier than the standard reprojection error since it involves warping and integrating large image regions [40]. It has been shown in [122] that direct methods are more robust in scenes with little texture or in the case of camera-defocus and motion blur. However, despite the advantages, this method has several drawbacks: it is relatively slower compared to feature-based approaches, requires good initialization, and is not robust to rolling shutter. Work in this category includes Jin et al. [75], Silveira et al. [142], Newcombe et al. [122], Foster et al. [40], and Engle et al. [36].

2.1.2 Learning-based Approaches

While geometry-based VO estimates the camera ego-motion by using hand-crafted features, learning-based VO automatically learns useful features for VO estimation given large amount of training data. Recent advances in Deep Learning (DL) and the availability of many large scale public datasets have made this possible. DL is a representation learning technique that attempts to learn high-level abstractions of data by using multiple hierarchical layers of Deep Neural Networks (DNNs) [53, 93].

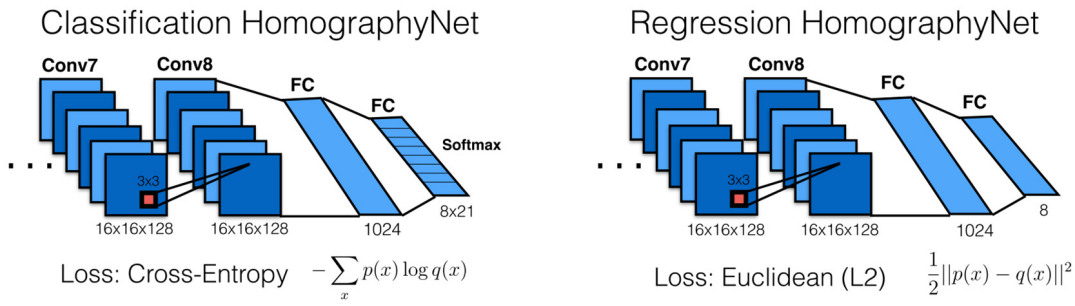


Figure 2.3: The difference between classification and regression networks on HomographyNet [33].

The main characteristic of DNNs is that they can process raw input data directly without the necessity of domain expertise to design a feature extractor. This technique has started to make significant changes in many research areas of computer vision and language understanding, including ones that were previously considered as not possible to cast as a learning problem due to the involvement of geometric transformations such as VO [59].

Depending on the availability of ground truth data, learning-based VO can be divided into supervised and unsupervised. Supervised VO learns the camera ego-motion by minimizing the relative transformation error between two images with respect to the given ground truth transformation. On the other hand, unsupervised VO leverages the intrinsic geometrical constraints between consecutive images to predict the camera poses based on a novel view synthesis paradigm.

2.1.2.1 Supervised VO

Supervised learning-based VO trains Convolutional Neural Networks (CNNs) by minimizing errors in predicting the ego-motion compared to the ground truth pose. As CNNs are best known for classification tasks, in early works, pose estimation was considered as a classification problem over the discretized space of translation and rotation of the camera. Konda and Memisevic [85] were probably the first to propose the estimation of VO using this principle. They utilized a stereo camera to predict the velocity and the direction of the camera. The network trains the rep-

resentation of motion and depth from stereo pairs by using synchrony autoencoders [84]. These motion and depth representations are fed into a CNN to estimate the velocities and orientations through softmax-based classification. Instead of estimating general motion similar to the fundamental matrix, DeTone et al. [33] proposed "HomographyNet" to train a CNN for computing homography between two frames using 4-point parameterization of homography. They proposed two different networks, one is a classification network based on cross-entropy loss function and the other one is a regression network based on Euclidean loss function (see Figure 2.3). They showed that the regression network is more accurate than the classification network due to the continuous nature of the prediction.

After realizing that a CNN can be used accurately for the regression problem, all recent techniques for pose estimation employ regression-based CNNs. Mohanty et al. [111] utilized a pre-trained AlexNet network [88] as the input of the regression network. Two consecutive images are fed into two parallel AlexNet networks and the outputs are concatenated for regressing the camera odometry through a fully connected layer. Based on the experiments, they observed that the extracted features from AlexNet are not generic for the problem of visual odometry, i.e. odometry only works well in a previously seen environment.

Since pre-trained convolutional layers for object detection and classification are not suitable for odometry estimation, researchers turned to optical flow based networks to generalize the learned parameters in different environments. Muller and Savakis [118] designed "Flowdometry", a network consisting of two sequential CNNs: the first one for predicting optical flow and the latter for estimating camera motion. The FlowNetS [34] architecture is used for both networks although the second network replaces the refinement part by a fully connected layer in order to incorporate inter-frame odometry computation. Melekhov et al. [110] developed an end-to-end CNN for computing ego-motion between two views. They stacked two parallel CNNs with weight sharing followed by spatial pyramid pooling (SPP) layer to tackle arbi-

trary input image while maintaining spatial information in the feature maps. The regression layer consists of two fully connected layer for predicting camera translation and rotation.

While the previous works only learn geometric feature representation of the scene through CNNs, Wang et al. [161] propose "DeepVO" as an end-to-end learning framework which is capable of learning sequential motion dynamics from image sequences through a combination of CNN and Recurrent Neural Network (RNN). RNNs are frequently used for learning sequential data such as speech or language since they maintain a history of all elements of the sequence in the network [93]. Their results show that by utilizing both a CNN and a RNN, the output odometry is more accurate than competing state-of-the-art methods (e.g. to VISO2 Monocular system [46]). Nonetheless, they stated that moving objects in front of the camera might reduce the accuracy of pose estimation and it is unclear how to deal with this challenge under a deep learning framework.

2.1.2.2 Unsupervised VO

In the unsupervised case, the CNN is trained without the availability of ground truth data. Instead, the network learns to predict the camera pose by minimizing the photometric error similar to LSD-SLAM [37]. Given I_{ref} as a reference image where $I : \Omega \rightarrow \mathbb{R}$ provides the color intensity, the photometric error minimizes the following objective function:

$$E(\xi) = \sum_{i \in \Omega_{ref}} \left(I_{ref}(x_i) - I_{new}(\omega(x_i, D_{ref}(x_i), \xi))\right)^2 \tag{2.13}$$

where $\omega(x_i, D_{ref}(x_i), \xi)$ is a warp function that projects the image point $x_i \in \Omega_{ref}$ in the reference image I_{ref} to the respective point in the new image I_{new} based on the inverse depth value of the reference image $D_{ref}(x_i)$ and the camera transformation $\xi \in se(3)$.

Zhou et al. [177] developed this unsupervised learning mechanism using the principle of novel view synthesis (the problem of synthesizing a target image with different

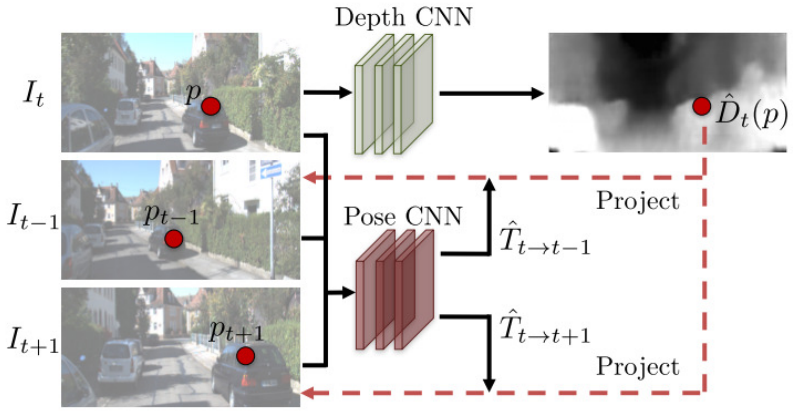


Figure 2.4: In unsupervised learning, the output from depth and pose network are used to inverse warp the source image in order to reconstruct the target view [177].

pose given a source image). They constructed two parallel CNN networks for predicting depth and estimating the camera pose. The predicted depth from the source image is used for synthesizing the target image given the camera transformation matrix and the source image (Figure 2.4). By minimizing the photometric error as in Equation (2.13), depth and camera pose can be jointly trained. Instead of generating the target image from depth prediction, Vijayanarasimhan et al. [158] constructed 3D scene flow based on depth prediction, camera motion, and dynamic object segmentation resulting from the convolutional/deconvolutional network. The scene flow is transformed by the camera motion and then back-projected to the current frame for evaluating the photometric error. Recent developments extends the novel view synthesis idea by leveraging Generative Adversarial Networks (GAN) [2]. GAN enables two neural networks, namely the generator and discriminator, to compete against each other in a zero-sum game. In the case of novel view synthesis for unsupervised VO, the generator synthesises the depth image of the current view, given the previous and next view, while the discriminator is trained to discriminate the real depth image and the synthesized depth image from generator. Experimental results show that optimizing novel view synthesis based on Equation (2.13) using GAN can improve the accuracy of the Zhou approach [177], either for depth map generation or VO estimation.

2.1.3 Visual Odometry Beyond Point Features

2.1.3.1 Visual Odometry using Line Segments

As discussed before, feature-based visual odometry approaches typically rely on point-based features (e.g. corners). While point-based features have shown good accuracy in some scenarios, there are cases when visual odometry easily lose track due to insufficient point features in the environment. This typically happens in indoor environment (e.g. office corridor) in which the scene mostly composes of texture-less wall. To alleviate this problem, researchers have developed alternative approach which is based on line-segments instead of relying on point-based features. In the earliest work, A. P. Gee and W. Mayol-Cuevas [44] proposed a model-based localization and mapping system by using line as the landmark. They employed Unscented Kalman Filter (UKF) to model the camera motion, while the line segments extracted from the images are used to initialized 3D line segments used for tracking. T. Lemaire and S. Lacroix [96] followed this idea but they utilized Plucker coordinates instead of using standard format (e.g. by using two endpoints/extremities and/or vector directions) to represent line features. They employed Plucker coordinates, which represents a line as a cross product between point in a line with the line direction, as it is well adapted to the projection through a pinhole camera. However, the proposed approaches found it difficult to initialize the line landmarks in feature-less area or when the camera is moving within the plane. Other improvement includes designing effective iterative closest multiple line approach to enable more accurate data association performance [164]. Nonetheless, this technique also faces difficulty when the camera move abruptly or there is obstruction in front of the camera as this changes the properties of the previously tracked line segments (e.g. the location of extremities) [96].

As Bundle Adjustment (BA) based approaches have shown to be more accurate than filtering based approach [145], recent line-based approaches typically employ BA as their main building blocks. P. Pumarola, et al. [128] construct line-based visual SLAM system which is built on top of ORB-SLAM [119]. The window-based

BA is used to optimize both camera poses and line segments through reprojection error from three frames. To improve the robustness due to the difficulty of matching line segments from different camera viewpoint, recent works combine together point-based features and line segments as seen in [47, 128, 48]. Nonetheless, the robustness of these techniques highly depend on the availability and consistency of both point and line features [105].

2.1.3.2 Visual Odometry on Pixel Processor Arrays

L. Bose, et al. [16] proposed visual odometry algorithms that can run on Pixel Processor Arrays (PPA). Different with standard feature-based approach that transfers images to another (central/graphical) processing units, their visual odometry algorithms executes directly in PPA. PPA has processor and data storage capability in each sensor pixel, enabling fast parallel computation on the image plane. It enables visual odometry to be executed at 1000 Hz. However, this also comes with limitation as running with very high frame rate requires sufficient illumination due to the associated low exposure time. Nevertheless, they show that visual odometry can run accurately on PPA especially in term of rotation although from relatively short sequences, while the translation estimation is less accurate and scaleless. C. Greatwood, et al. [52] improved this approach by proposing perspective correction algorithm, in which images of surface are warped to appear as it is captured directly facing the surface. This is done to enable consistent estimation of the ground plane despite any changes in roll and pitch of the MAV. Experiments on Quadrotor MAV shows that the approach can track the Quadrotor position accurately compared to ground truth position. Nevertheless, the translation estimation has to be aligned with the ground truth as translation estimation is accurate up to an unknown scale factor.

2.1.3.3 Commercial Odometry System

In the last couple of years, odometry system embedded in commercial devices has been made available by some technological companies. Typically, those devices em-

ploy multiple sensor systems including RGB camera, depth/range camera, and IMU. For example, Microsoft developed Hololens, a Mixed Reality (MR) smartglasses, in which an ego-motion estimation is performed by using IMU and 4 120° × 120° depth cameras. Google releases ARCore which allows mobile phone to perform motion tracking by using both RGB and IMU data. Intel also releases Intel RealSense Tracking Camera which is able to perform SLAM by using two fisheye sensors camera with 163° field of view and IMU. However, these systems are typically designed with tightly-synchronized hardware, making it difficult to customize it for a very specific purpose, in which industrial camera (e.g. IDS uEye) is typically preferred. They still drift or lose tracks in particular scenarios (e.g. Hololens provides handling mechanism when the device loses tracks[1], Intel RealSense easily drift in static or slow forward motion[2,3], Google AR Core loses track when the device moves/rotates quickly [4], etc.). Moreover, none of them are specifically designed to work in visually-denied scenarios such as by incorporating thermal or radar sensing.

2.1.4 Discussion of Advantages and Disadvantages

Tables 2.1 and 2.2 show a summary of existing geometry-based and learning-based approaches respectively. In general, a feature-based approach promises accurate odometry estimation in environments with rich texture as long as bundle adjustment is performed. However, in texture-less areas, feature-based approaches can sometimes lose tracks or produce inaccurate camera poses. Moreover, the methods are very sensitive to outliers and noisy correspondences, making them not robust in low texture area. To alleviate this issue, line-based visual odometry employs line segments to provide more robustness of feature correspondences in texture-less areas. However, this approach also face difficulty when the camera moves abruptly making the line

[1]https://docs.microsoft.com/en-gb/windows/mixed-reality/tracking-loss-in-unity

[2]https://support.intelrealsense.com/hc/en-us/community/posts/360036423993-T265-position-drifting-away

[3]https://github.com/IntelRealSense/librealsense/issues/3970

[4]https://github.com/google-ar/arcore-unity-sdk/issues/101

properties altered, or when the line segments are obstructed by dynamic objects. In repetitive building structure, ambiguous line segments might also be found which confuses the matching process. Although this ambiguity can be filtered by identifying perpendicular line segments, this structural information does not always persist in the observed environment.

To enable real-time estimation for feature-based approaches, bundle adjustment can be executed in a window-based scheme or additional constraints (e.g. non-holonomic, etc.) can be incorporated to enable lower point requirements for estimating poses (e.g. 1-point-RANSAC). Nevertheless, these constraints might limit the application to only specific scenarios (e.g. wheeled-vehicles). Direct approaches, on the other hand, utilize the whole image information by directly optimizing the camera poses and the per-pixel inverse depth through photometric errors. This makes direct approaches more robust in texture-less areas since they employ all parts of the images to estimate odometry even in low image gradient regions. Nonetheless, good initialization is needed to enable accurate tracking, and large computation time is required to achieve whole image alignment. Rolling shutter also becomes a problem as it distorts images making dense alignment difficult.

Learning-based approaches offer more robustness to noisy correspondences by training the model in an end-to-end fashion. However, without explicitly enforcing geometric information, it is potentially more difficult to generalize to new environments, showing lower accuracy compared to geometry-based approaches. Unsupervised approaches try to introduce geometry by using photometric error as the objective function. The photometric error is minimized between the warped target image and the source image based on the camera pose and depth prediction. However, this warping-based approach requires a pre-defined camera instrinsic parameters which may vary across datasets. This may prevent the algorithm to generalize well in different datasets, even if it is tested in the same domain (e.g. a model trained on KITTI might not work well when it is tested in Malaga dataset although both datasets

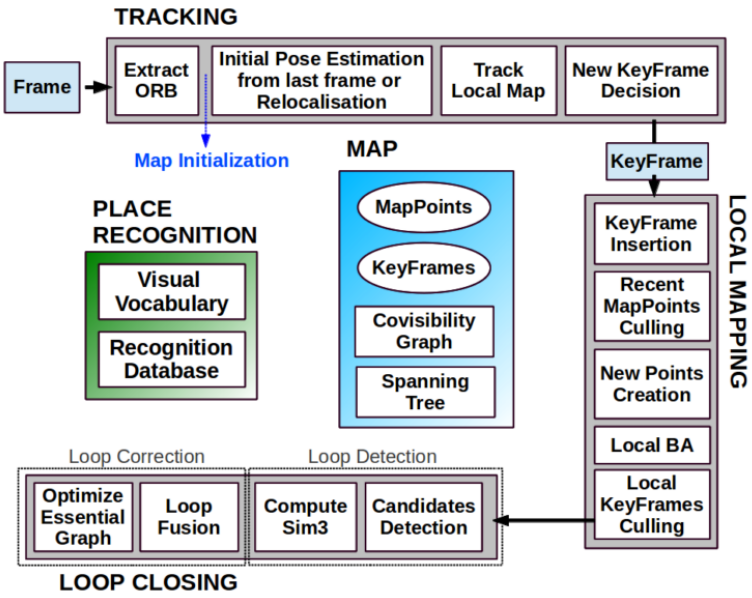


Figure 2.5: Architecture of ORB-SLAM, state-of-the-art feature-based visual SLAM system, which consists of front-end processing (tracking and local mapping) and back-end optimization (loop closing) [119].

represent the motion of vehicles). Finally, as most state-of-the-art learning-based approaches are based on DNNs, they require large weights (e.g. from tens to hundred millions) which prevents realtime execution in resource-constrained environments.

2.2 Visual SLAM and Structure-from-Motion

As a closely related concept with VO, both visual Simultaneous Localization and Mapping (vSLAM), which came from the robotic community, and Structure-from-Motion (SfM), which came from the computer vision community, can also be used to estimate the camera poses. The difference between VO, vSLAM, and SfM is the objective of vSLAM and SfM is not only to estimate the camera ego-motion but also to reconstruct a consistent map. To achieve this, both vSLAM and SfM provide other optimization techniques beyond what VO has, including loop closing methods or graph optimization. Moreover, SfM is more general than vSLAM since SfM can handle unordered image collections while vSLAM mainly deals with ordered image sequences or video for real-time navigation.

Table 2.1: Summary of Existing Geometry-based Visual Odometry Approaches

Year	Method	Advantages	Disadvantages
Geometry-based Approaches			
2004	Nister et al. [124]	Accurate odometry in short sequence	Easily lose tracks in texture-less area, sensitive to outliers and missing correspondences, large drift in long sequences
2006	Mouragnon et al. [114]	Reduced drift in long sequences due to windowed-bundle adjustment	Require more computation time than Nister approach, easily lose tracks in texture-less area, sensitive to outliers and missing correspondences
2009	Scaramuzza [139]	Real-time performance due to the usage of 1-point-RANSAC enabled by incorporating non-holonomic constraints	Can be only applied in wheeled-vehicle, easily lose tracks in texture-less area, sensitive to outliers and missing correspondences
2011	Lovegrove et al. [102]	More robust in texture-less environment	Not robust to small occlusion, require extensive computation time to find the matched template
2011	DTAM [122]	Use more complete information (whole image matching) to estimate odometry, can reconstruct the whole image, reconstruction more meaningful than point features, more robust in texture-less environment	Require good initialization, not robust to any photometric variations caused by rolling shutter, much slower than feature-based
2014	SVO [40]	Faster than DTAM as it only use direct method for tracking while the mapping part uses standard feature-based, more robust in texture-less environment	Require good initialization, not robust to any photometric variations caused by rolling shutter
2018	DSO [36]	Faster than DTAM as it uses sparse information through pixel sampling, more robust in texture-less region	Require good initialization, not robust to any photometric variations caused by rolling shutter

Table 2.2: Summary of Existing Learning-based Visual Odometry Approaches

Year	Method	Advantages	Disadvantages
Supervised Learning			
2017	DeepVO [161]	End-to-end trainable, accurate scale estimation implicitly learnt during training	Ground truth required, no explicit understanding of geometry, much slower compared to geometry-based (no real-time predictions), does not work in different domains
2018	ESP-VO [161]	More accurate than DeepVO, probabilistic estimates	Same disadvantages as DeepVO
2018	LS-VO [29]	Jointly estimates pose and optical flow, more robust to appearance changes, blur, and large camera speed changes	Ground truth required, lower accuracy in terms of rotation compared to geometry-based, no real-time predictions
Unsupervised Learning			
2017	SfMLearner [177]	No ground truth required, camera pose and depth image can be estimated jointly	Need camera calibration parameters, prediction correct up to scale, lower accuracy compared to geometry-based, difficult to generalize to new dataset (even in the same domain) as it requires camera parameters
2019	GANVO [2]	More accurate than SfMLearner in terms of depth and pose estimation	Need camera calibration parameters, prediction correct up to scale, difficult to generalize to new dataset as it requires different camera parameters

Although the pipeline of vSLAM and SfM is quite similar to VO, in practice there are some notable differences. For example, instead of optimizing the camera pose and 3D structure of the environment over all images, Mouragnon et al. [116, 115] propose to optimize over a few recent images by employing local bundle adjustment (LBA). Klein and Murray [82] introduce "PTAM" which shows that tracking and mapping can run in real-time if the pipeline is executed on different threads. Furthermore, PTAM also introduced the idea of choosing key frames, thus LBA can also be implemented over the selected key frames. On the other hand, Lim et al. [99] used binary descriptors and a metric topological mapping such that large scale mapping can operate in real-time without any parallel computation. Recent state-of-the-art techniques like ORB-SLAM [119] integrate hardware and algorithmic advances in the past decade by including parallel computing, ORB features [134], statistical model selection [153], loop closures based on bag-of-words place recognition [32], local bundle adjustment [115], and graph optimization [92]. Figure 2.5 describes the architecture of ORB-SLAM. For a more detailed review of ORB-SLAM or other standard feature-based techniques, interested readers can follow [42] or [171].

2.3 Alternative Modalities in Visually-denied Environments

While standard visual (visible spectrum or RGB) imaging systems work very well in environments with good illumination, they are unusable for applications in visually-denied environments. Alternative modalities that rely on the non-visible spectrum, such as thermal imaging systems, or non vision-based modalities, such as IMU, will be explored in this challenging scenario.

2.3.1 Thermal Imaging

2.3.1.1 Thermal Camera System

A thermal camera is different from a standard RGB camera as it captures the radiation emitted from objects in the form of Long-Wave Infrared (LWIR) signals which belong to the non-visible spectrum. Although it is non-visible to human eye, the radiometric data captured from the thermal camera is typically converted to a visible format (usually in grayscale) to ease human interpretation. This camera is commonly used in firefighting since it allows firefighters to see beyond thick smoke, darkness, or heat-permeable obstacles.

Unlike visible light cameras, the pixel sensor in thermal cameras can have different sensitivity with respect to neighboring pixels. This makes a thermal camera greatly affected by spatial non-uniformities in the form of fixed-pattern noise [15]. The fixed-pattern noise often noticeable during longer exposure which is shown by a brighter intensities above the general background noise for particular pixels. To compensate for this noise, a standard thermal camera performs a Non-uniform Correction (NUC) to re-calibrate the sensor during operation and reset the sensor noise. This calibration process freezes the camera for around 0.5 to 1 second during which the same thermal image is produced. Most thermal cameras perform NUC periodically depending on the sensor quality and application. NUC can be executed as many as 3 times a minute or every couple of minutes. Modern thermal imaging systems such as those produced by FLIR have an option to manually set up the frequency of NUC although using default NUC frequency set by the manufacturer is recommended.

2.3.1.2 Thermal Odometry

Accurately estimating camera ego-motion from a thermal imaging system remains a challenging problem. Nevertheless, some efforts have been made to realize thermal odometry although it is for relatively short distances and yields a sub-optimal performance compared to the visible camera systems. Existing works in thermal odometry

usually rely either on sparse feature-based or direct-based approaches. Mouats et al. [113] employed Fast-Hessian feature detector for UAV tracking using a stereo thermal camera. They designed an effective mechanism to calibrate the thermal imaging system using an aluminium checkboard pattern. They also used double dogleg algorithm for pose optimization and showed that it can be used as a viable alternative to the standard Levenberg–Marquadt approach. To enable practical odometry, [15] designed a thermal odometry system with an automatic mechanism to determine the appropriate time to perform NUC operation based on the current and the predicted camera poses. By using efficient NUC management, combined with road lane estimation to estimate the scale of the prediction, they show that practical thermal odometry is viable.

Recent work fused together a thermal imaging system and IMU for robust UAV tracking. Khattak et al. [78] developed a keyframe-based direct approach to thermal odometry which minimizes radiometric error (or photometric error in terms of RGB images) between consecutive frames. They used raw radiometric data instead of the normalized grayscale data to avoid the difficulty of matching features as the scene dynamically changes based on the observed environment temperature. Furthermore, by fusing thermal odometry with IMU, they improve the robustness of the thermal-inertial odometry system during the NUC operation.

2.3.2 Inertial Measurement Units (IMUs)

An Inertial Measurement Units (IMU) is an electronic device capable of measuring linear acceleration and angular velocity of the body using a combination of accelerometers, gyroscopes, and sometimes magnetometers. The most commonly used IMUs are based on low cost Micro-Electro-Mechanical System (MEMS) technology which is widely used in robotics, UAVs, or mobile devices. As a small, low cost device, MEMS-based IMU comes with high sensor noise [20]. The measurement of accelerometers

and gyroscopes, together with sensor biases, are typically modelled as follows

$$\hat{\omega} = \omega + b_{gyro} + \eta, \qquad \eta \sim N(0, \sigma^2_{gyro}) \tag{2.14}$$

$$\hat{a} = a + a_g + b_{acc} + \eta, \qquad \eta \sim N(0, \sigma^2_{acc}) \tag{2.15}$$

where ω and a are actual gyro and accelerometer measurements respectively, b_{gyro} and b_{acc} are unknown biases, a_g is the gravity force, and η is white gaussian noise.

2.3.2.1 IMU-based Odometry System

Odometry estimation using IMU is typically performed either with classical Strapdown Inertial Navigation System (SINS) or Pedestrian Dead Reckoning (PDR). SINS estimates the device motion by double integrating acceleration to position. This technique is straightforward and fast, but it is bounded by exponential error growth, especially if a noisy MEMS sensor is employed. One solution to tackle this problem is by attaching the IMU on the pedestrians feet to take advantage of zero velocity updates as shown in [143]. However, the application of zero velocity update is limited to foot-mounted navigation system.

Unlike SINS, PDR estimates the device position by repeatedly detecting steps, calculating stride length and direction of motion using an empirical formula [74]. Although this approach can alleviate some problem in SINS, a noticable drift still persists due to incorrect step displacement segmentation and inaccurate stride estimation [20]. Moreover, many parameters are required to be tuned, especially if the system is being implemented for general user with different walking patterns. Abandoning these two formulations (SINS and PDR), recent work for inertial odometry employed DNNs to directly predict position and orientation in the form of polar vector coordinates from raw IMU data [20].

2.4 Efficient Deep Neural Networks

A DNN model usually contains tens or hundreds of millions of parameters, which requires significant computation and memory resources. This typically prevents the

implementation of a DNN on resource-constrained hardware (e.g. mobile phones, smart watch, or other embedded devices). Over the past few years, many attempts have been made to construct more efficient DNNs. In general, we can divide the approaches into neural network compression and knowledge distillation.

2.4.1 Neural Network Compression

Compression techniques attempt to compress existing networks. We can divide current methods into quantization, network pruning, and network decomposition.

2.4.1.1 Quantization

Quantization compresses the network by reducing the number of bits required to represent each weight. Quantization could be applied by using 16-bit or 8-bit representation as proposed by [156, 55]. As an extreme case, a 1-bit representation (or binary network) has been explored in [31, 67]. By using only two possible values, such as -1 or 1, a binary network can replace many multiply-accumulate operations by simple accumulations. An example framework that uses this technique is BinaryConnect [31]. BinaryConnect utilizes the binary representation in forward and backward passes of the neural network during training. This approach boosts the training speed by a factor of 3. Using a deterministic version of BinaryConnect, the inference time can be pushed further by a factor of 16 or more. Similarly, the Binarized Neural Network (BNN) approach [67] trains a DNN with binary weights and is proven to significantly reduce the memory consumption during the forward pass, either at run time or at training time. Although many attempts have been made to improve the performance of binary networks, in general, such approaches suffer from inevitable loss of accuracy since relaxing the weights to binary reduces the representation capability of the original 32-bit weights of the network.

2.4.1.2 Network Pruning

Pruning, as the name implies, removes redundant and non-informative weights from the network. Pruning can be done by using the magnitude-based approach or the dropout-based approach. In the magnitude-based approach, Han et al. [58] proposed to remove the weight connections if the magnitude of the weights is less than a predefined threshold. Doing this approach iteratively with fine-tuning can reduce the storage and computation time by an order of magnitude without affecting accuracy. Figure 2.6 describes an example of iterative pruning methods presented in [58]. However, since pruning based on magnitude turns the network weights into a sparse matrix, it requires additional implementation of sparse matrix operations. Moreover, pruning the network aggressively might result in irretrievable network damage.

Another way of pruning the weights is based on dropout operations. Yao et al. [169] proposed "DeepIoT" as a unified approach to compress convolution, recurrent, and fully connected neural networks by finding the minimum number of non-redundant hidden elements through dropout learning. They designed a compressor neural network to produce dropout probability for each layer in the original network. By optimizing the compressor network and the original network jointly through the compressor-critic framework, DeepIoT can outperform other baseline compression algorithms by a large margin. Another dropout-based method employs the variational dropout technique to sparsify both fully connected and convolutional layers [112]. However, the authors only focus on sparsifying the network weights (which successfully reduces the number of parameters up to 280 times on LeNet and 60 times on AlexNet), without necessarily achieving a decrease in computation time.

2.4.1.3 Network Decomposition

The decomposition-based approach reduces the neural network complexity by exploiting low-rank constraints on the network weights. Bhattacharya et al. [11] proposed "SparseSep" to separate a fully connected or a convolutional layer into two weight

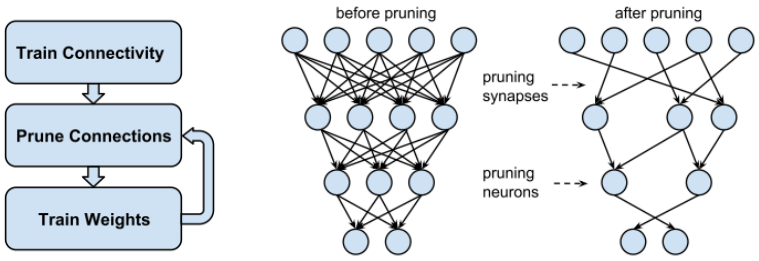


Figure 2.6: In pruning based on magnitude, the network needs to be trained iteratively to avoid accuracy loss [58].

matrices through matrix factorization. Singular Value Decomposition (SVD) is used to factorize the weight matrices such that the reconstruction error is minimized. In the case of convolutional layers, the separation process decomposes the original filter into two vertical and horizontal filters. By using this factorization technique, the number of matrix multiplications becomes smaller than the original network although the network is essentially deeper. However, the obtained compression is generally lower than that achieved by the pruning based approach and the low-rank constraint that is imposed on the network might impair the network performance.

2.4.2 Knowledge Distillation

Instead of compressing the network structure, Hinton et al. [62] proposed Knowledge Distillation (KD) as an approach to transfer the knowledge of a large teacher network to a smaller student network. The main idea of KD is to allow the student to capture the finer structure learned by the teacher instead of learning solely from the true labels. Let T be the teacher network where $\mathbf{O}_T = \text{softmax}(\mathbf{a}_T)$ is the teacher output probability and $\mathbf{a}_T$ is the teacher's logits (pre-softmax output). A student network S with $\mathbf{O}_S = \text{softmax}(\mathbf{a}_S)$ as the prediction and $\mathbf{a}_S$ as the logits is trained to mimic $\mathbf{O}_T$. Since $\mathbf{O}_T$ is usually very close to the one-hot representation of the class labels, a temperature $\tau > 1$ is used to soften the output probability distribution of T. The same temperature is used for training S such that $\mathbf{O}_T^\tau = \text{softmax}(\frac{\mathbf{a}_T}{\tau})$ and $\mathbf{O}_S^\tau = \text{softmax}(\frac{\mathbf{a}_S}{\tau})$, but $\tau = 1$ is then used for testing S. The KD objective function is

formed by minimizing both hard label (one-hot encoding of class labels) error and soft label error) as follows

$$\mathcal{L}_{KD} = \alpha\mathcal{H}(\mathbf{y}, \mathbf{O}_S) + (1-\alpha)\mathcal{H}(\mathbf{O}_T, \mathbf{O}_S) \tag{2.16}$$

where $\mathcal{H}$ is the cross-entropy, $\mathbf{y}$ is the one-hot encoding of the true labels, and α is a parameter to balance both cross-entropies. Figure 2.7 illustrates how KD is applied to classification problem.

The KD formulation with softened outputs ($\tau > 1$) in Equation (2.16) gives more information for S to learn, as it provides information about the relative similarity of the incorrect predictions, which is called dark knowledge [62], [132]. For example, T may mistakenly predict an image of a car as a truck, but that mistake still has a much higher probability than mistaking it for a cat. These relative probabilities of incorrect prediction convey how T tends to generalize to new data [62]. By training using this KD objective, S can emulate the generalization capability of T.

Extending Hinton's work, Romero et al. proposed "FitNets" [132] as a way to train a student network that is deeper but thinner than the teacher network by using a hint training approach. Hint training utilizes the intermediate representations learned by the teacher to teach the intermediate hidden layer of the student network. This technique allows one to train deeper students that can run faster or generalize better than the teacher network, a trade-off that is controlled by the chosen student capacity.

2.5 Curriculum Learning

Curriculum Learning (CL) was proposed by Bengio et al. [9] to formalize the idea of learning through a meaningful order of examples or concepts, which mimics how humans and animals learn. However, the basic idea of starting small or simple actually dates back to 1993 when Elman [35] successfully trained a DNN to recognize a simple grammar by increasing the complexity of the task. Bengio's work [9] confirmed Elman's findings and showed that a well chosen CL strategy can improve the generalization ability of a DNN model. This idea was further improved by [89] through

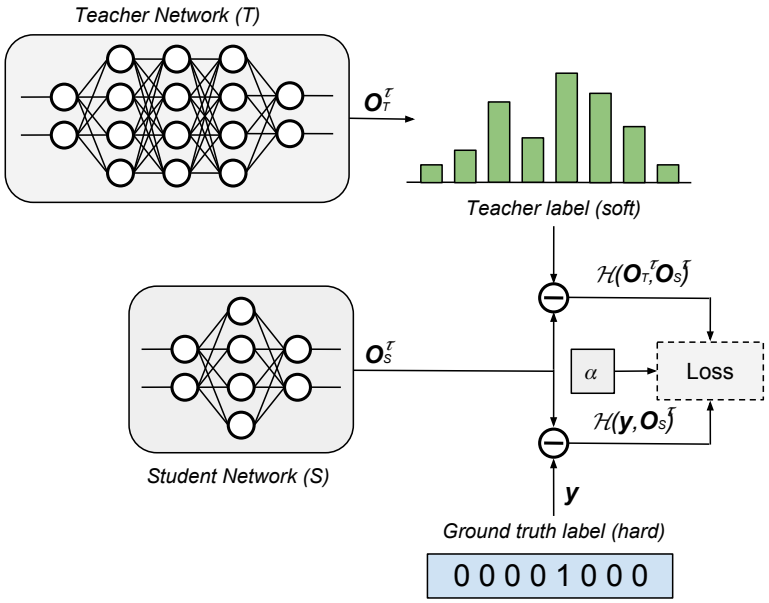


Figure 2.7: The illustration of how standard KD is applied to classification problem. Note that in classification KD can take advantage of "dark knowledge" provided by soft teacher labels.

Self-Paced Learning (SPL), in which the curriculum is learned during training rather than determined by prior knowledge. Jiang et al. [73] then combined both idea of CL and SPL through Self-Paced Curriculum Learning (SPCL). SPCL takes into account both prior knowledge and the learning progress during training in constructing the curriculum. The application of CL and its improvement includes action detection [72], dictionary learning [151], domain adaptation [150], and object tracking [147], but none of them tackle VO estimation where it is more difficult to differentiate between easy and hard examples or tasks.

2.6 Summary

In this chapter, we have reviewed existing approaches to visual odometry estimation, either based on geometry or based on machine learning. While geometry-based approaches show good accuracy and consistency in their estimation, they are not robust to outliers, noisy correspondences, dynamic scenes, and the effect of rolling shutters,

or they require good initialization. They also face difficulty in abrupt motion especially when the camera has limited field of view. On the other hand, DNN-based approaches can be more robust to those issues as they do not require explicit feature extraction and matching but implicitly learn features from training with large amounts of data. However, despite being more robust, DNN-based approaches can produce less accurate results compared to geometry-based approaches since they do not apply geometric constraints. They are very slow and inefficient requiring tens or hundred millions of weights, preventing their application in resource-constrained hardware. Moreover, DNN-based VO approaches have not been adequately explored in environments with limited illumination. To the best of our knowledge, there are no DNN-based visual/thermal odometry algorithms that can work well in visually-denied environments.

Chapter 3

Learning Accurate Visual Odometry

3.1 Introduction

As discussed in Chapter 2, visual odometry is a key enabler for many applications, with deep learning techniques emerging as promising approaches. State-of-the-art DNN-based VO [118, 161, 162, 29, 149] typically minimizes the relative transformation loss as the objective function during training. Minimizing the frame-to-frame relative transformation loss generally can provide reasonable trajectory estimation. However, this approach does not guarantee the consistency of the composed transformation when integrating those relative estimates into longer trajectories. Incorporating a compositional transformation loss in the objective function is a natural way to introduce this consistency into the network. However, our experiments suggest that training DNN-based VO using compositional transformation loss is hard to converge. Our intuition is that it is too difficult for the network to directly learn the complex geometry of composing the 6 Degree-of-Freedom (DoF) camera poses since the prediction errors are accumulated.

An intuitive way to reduce the difficulty of training this complex geometry problem is by starting the learning process from an easier task and then gradually increasing the difficulty of the task. The idea of learning from small or easy tasks and progressively increasing the difficulty has been studied in the context of Curriculum Learning

(CL). Inspired by the cognitive process of humans and animals, Bengio et al. [9] proposed CL as a strategy to improve the convergence speed and generalization ability of a machine learning model by learning through highly organized or meaningful order of examples. In this chapter, we study whether a similar learning strategy can be applied for estimating the complex geometry of monocular VO.

Another perspective to alleviate the lack of accuracy in existing DNN-based VO is to focus on a set of information in the input space that are more useful for VO estimation. Our understanding from geometry-based approaches tells us that not all information in the image space is equally important. For instance, some areas are texture-less which makes feature matching difficult and noisy, or some parts belong to dynamic objects which violate the epipolar constraints [137]. This motivates the need to treat the information in the image space with varying importance of relevant features. To this end, in this chapter, we also propose an attention network as an alternative way to improve the accuracy of DNN-based VO.

The rest of this chapter is organized as follows. Section 3.2 introduces the main contributions of presented in this chapter. Section 3.3 elaborates on the first contribution of this paper, explaining how to improve the accuracy of DNN-based VO from an optimization point of view. Section 3.4 introduces the attention network as another perspective to improve the accuracy of DNN-based VO from a network architecture perspective. Section 3.6 concludes this chapter.

3.2 Contributions

We summarize the research questions we tackle in this chapter as follows: (1) *how can we improve the accuracy of DNN-based VO from the optimization perspective, by focusing on the design of the objective function?* (2) *Can the attention mechanism help to increase the accuracy of DNN-based VO? If so, what type of visual attention could work effectively for VO?* Our specific contributions that arise from these questions are listed below:

- We propose a novel geometry-aware objective function by jointly optimizing relative transformation and its composition over small windows via bounded pose regression loss.
- We present a curriculum learning strategy to train the network using bounded pose regression loss by gradually making the learning objective more difficult during training.
- We design a visual attention network by generating the attention map conditioned on the current latent poses and performing a spatial-wise attention operation on the feature map.
- We explore different ways to incorporate the visual attention, either via a joint attention network or via a decoupled attention network for translation and rotation components. We show that the decoupled attention network is superior to the joint attention network.

Some technical contributions and experimental results in this chapter have been described in the following published paper:

- **M. R. U. Saputra**, Pedro P. B. de Gusmao, S. Wang, A. Markham, and N. Trigoni. "Learning Monocular Visual Odometry through Geometry-Aware Curriculum Learning". In IEEE International Conference on Robotics and Automation (**ICRA**), 2019.

3.3 Geometry-Aware Curriculum Learning

In this section, we describe the first part of our contribution, namely improving the accuracy of DNN-based VO from an optimization perspective. In particular, we employ bounded pose regression loss to train the network in an end-to-end manner and adopt the CL strategy to gradually learn the proposed objective from a simpler objective. We call the network as CL-VO and refer to the optimization approaches as

Geometry-Aware Curriculum Learning (GA-CL). We start this section by discussing the general approach to learning the camera ego-motion using a DNN, followed by describing the technique we proposed and the experimental results.

3.3.1 Approach

3.3.1.1 Learning Ego-motion with DNNs

Conventional VO methods require the use of hand-crafted features and multiple view geometry techniques. On the other hand, DNN approaches work directly with raw image sequences by training the network in an end-to-end manner. Formally, given two concatenated images $\mathbf{I}_{t-1,t} \in \mathbb{R}^{2\times(w\times h\times c)}$ at times $t-1$ and t, where w, h, and c are the image width, height, and channels respectively, DNNs learn the following mapping function to regress the 6-DoF camera pose:

$$\text{DNNs} : \{(\mathbb{R}^{2\times(w\times h\times c)})_{1:N}\} \rightarrow \{(\mathbb{R}^{6})_{1:N}\} \tag{3.1}$$

where N is the total number of consecutive image pairs. The 6-DoF camera poses represent relative pose transformation $\{\hat{\mathbf{p}}_{t-1}^{t}\} \subset \mathbf{SE}(3)$ from pairs of consecutive images $\{I_{t-1}, I_t\}$. The cumulative composition of these estimations generates a global trajectory with respect to the starting position

$$\hat{\mathbf{p}}^{t} = \hat{\mathbf{p}}_{t-1}^{t} \oplus ... \oplus \hat{\mathbf{p}}_{1}^{2} \oplus \hat{\mathbf{p}}^{1} \tag{3.2}$$

where $\oplus$ represents the $\mathbf{SE}(3)$ pose composition operation.

3.3.1.2 Enforcing Geometric Constraints

During the training process, standard DNN-based VO typically minimize the relative pose transformation error between two consecutive frames. However, the ground truth pose is usually available as the composition of these relative pose transformations defining a sequence of global poses. In order to fully exploit both relative and composite pose transformation information, we need to jointly optimize these terms. Instead of directly placing relative and composite terms together in the objective

function, we propose to utilize the composed transformation as a constraint for the relative loss term. We only add the composite loss when its value at time t is larger than it was at time $t-1$. This means that the network does not have to minimize the composite transformation loss when the integration of relative poses at time t yields an accurate absolute pose transformation. Moreover, in order to reduce the accumulative errors, we only minimize the composite loss over small, bounded windows. We refer to this loss function as *bounded pose regression loss*.

Equations (3.3)-(3.6) show this bounded loss where N is the number of images. L_{rel} is the relative transformation loss that measures pose errors between consecutive frames, while L_{com} is the composite transformation loss which accounts for errors over a small window. The coefficient α is used to balance both terms. The pose error defined in Equation (3.6) compares the estimated translation $\hat{\mathbf{t}}$ and rotation $\hat{\mathbf{r}}$ vectors (encapsulated in $\hat{\mathbf{p}}$) with their respective ground truth values. We also use δ and ζ to weigh the translation and rotation terms in relative loss as in [77, 161].

$$L_{total} = \sum_{t=1}^{N} \alpha L_{rel} + (1-\alpha) L_{com} \tag{3.3}$$

$$L_{rel} = L\left(\hat{\mathbf{p}}_{t-1}^{t}\right) \tag{3.4}$$

$$L_{com} = \begin{cases} L\left(\hat{\mathbf{p}}_{t-w}^{t}\right), & \text{if } L\left(\hat{\mathbf{p}}_{t-w}^{t}\right) > L(\hat{\mathbf{p}}_{t-w-1}^{t-1}) \\ 0, & \text{otherwise} \end{cases} \tag{3.5}$$

$$L\left(\hat{\mathbf{p}}_{i}^{j}\right) = \delta \left\| \hat{\mathbf{t}}_{i}^{j} - \mathbf{t}_{i}^{j} \right\|^{2} + \zeta \left\| \hat{\mathbf{r}}_{i}^{j} - \mathbf{r}_{i}^{j} \right\|^{2} \tag{3.6}$$

3.3.1.3 Curriculum Learning

The bounded pose regression loss blends together relative and composite transformation loss. However, we discover in our experiments that training DNN-based VO using composite transformation loss is difficult to converge due to the accumulative nature of prediction errors. Figure 3.1 shows normalized translation and rotation pose errors for different values of α in Equation (3.3) in the first training stage. It can be seen that training a DNN-based VO using only composite transformation loss

($\alpha = 0$) leads to very large translation and rotation errors compared to when relative transformation loss is also incorporated ($\alpha > 0$). The best performance is even achieved by training using relative transformation loss only ($\alpha = 1$), which indicates the difficulty in training with relative and composite losses right from the start. This motivates the utilization of Curriculum Learning (CL) where the learning process starts from the simplest objective and then increasing its difficulty. We refer to this mechanism as *Geometry-Aware Curriculum Learning* (GA-CL).

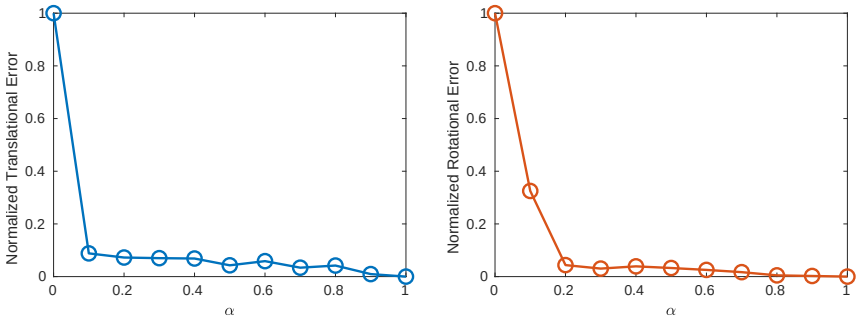


Figure 3.1: Normalized translation and rotation errors for different value of α.

In the first stage of GA-CL, we start the training process by predicting a reasonable relative transformation (as suggested from Figure 3.1). This can be seen as minimizing the bounded pose regression loss from Equations (3.3)-(3.6) with $\alpha = 1$. During the second stage, once the network has learned to produce reasonable relative transformations (as the validation loss no longer decreases), we may reveal more information to the network by gradually decreasing α so as to equalize relative and composite transformation loss ($\alpha = 0.5$). In the final stage, we put more emphasise on the composite loss $0 < \alpha < 0.5$ such that the network can learn consistent composite transformation.

3.3.1.4 Network Architecture

The network architecture, dubbed CL-VO, is depicted in Figure 3.2 and is mainly composed of a feature extractor and a pose regressor. The feature extractor is essentially a CNN designed to learn dense optical-flow for VO estimation. We construct

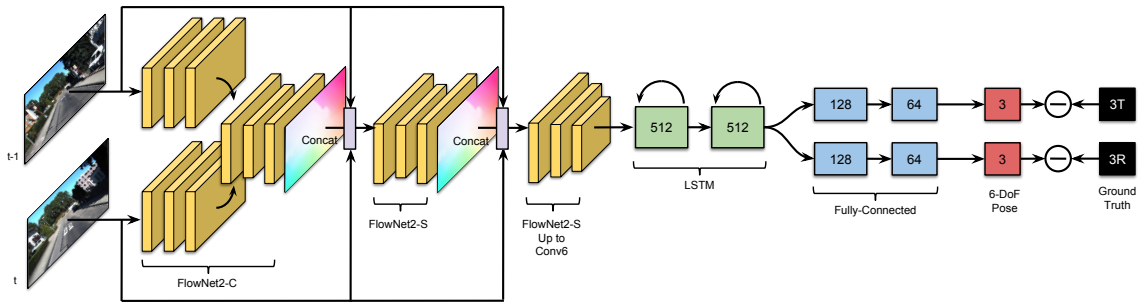


Figure 3.2: CL-VO architecture consists of cascade optical-flow networks followed by recurrent networks and fully connected layers.

a cascade optical flow network which refines optical flow estimation subsequently from the previous sub-network for providing more accurate flow estimation. To avoid training the network from scratch, we adopt FlowNet2-C [69] for the first network and FlowNet2-S [69] for the second and the third network. The output for each network is a dense optical flow in which the estimation is refined in subsequent network. However, for producing the latent variables that can be directly consumed by the pose regressor, we remove the refinement part from the last optical flow network.

The pose regressor part consists of two recurrent layers, in particular two Long Short Term Memory (LSTM) [63] layers, followed by fully connected layers to estimate 6-DoF camera poses. Compared to directly using a fully connected layer for pose regressor, as seen in [118] and [77], LSTM is more suitable to learn the long term dependencies of camera pose since it can maintain its hidden state over time. The LSTM operation can be formulated as follows:

$$\begin{bmatrix} \mathbf{i} \\ \mathbf{f} \\ \mathbf{o} \\ \mathbf{g} \end{bmatrix} = \begin{bmatrix} \text{sigm} \\ \text{sigm} \\ \text{sigm} \\ \tanh \end{bmatrix} \mathbf{W}_{\text{lstm}}^{(l)} \begin{bmatrix} \mathbf{h}_t^{(l-1)} \\ \mathbf{h}_{t-1}^{(l)} \end{bmatrix}, \tag{3.7}$$

$$\mathbf{c}_t^{(l)} = \mathbf{f} \odot \mathbf{c}_{t-1}^{(l)} + \mathbf{i} \odot \mathbf{g}, \tag{3.8}$$

$$\mathbf{h}_t^{(l)} = \mathbf{o} \odot \tanh(\mathbf{c}_t^{(l)}), \tag{3.9}$$

where $\mathbf{W}_{lstm}^{(l)} \in \mathbb{R}^{4n^{(l)} \times (n^{(l-1)} + n^{(l)})}$ is the weight matrix for layer l, n is tensor dimension, $t = 1, ..., T$ is the timestep, and the vector $\mathbf{h}_t^{(l)} \in \mathbb{R}^{n^{(l)}}$ is its hidden state at step t

and layer l. Vector $\mathbf{h}_t^{(0)}$ is equal to the input $\mathbf{x}_t$ at step t. Operators *sigm*, *tanh*, and $\odot$ denote sigmoid function, hyperbolic tangent, and element-wise multiplication respectively. For composing the relative transformation from a certain number of previous frames, we construct a differentiable custom *windowed composition layer* as seen in Figure 3.3. A windowed composition layer concatenates the current frame-to-frame camera ego motion with the previous ego motion for a predefined number of window w as follows

$$\hat{\mathbf{p}}_w^t = \hat{\mathbf{p}}_{t-1}^t \oplus ... \oplus \hat{\mathbf{p}}_{t-w}^{t-w+1} \oplus \hat{\mathbf{p}}^{t-w}. \tag{3.10}$$

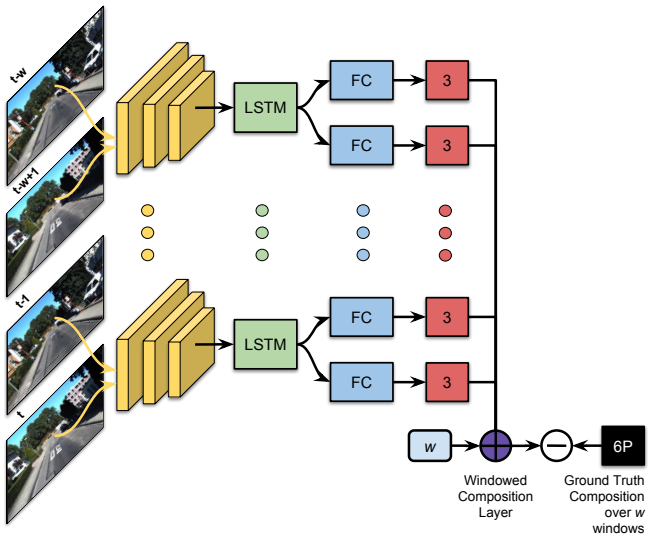


Figure 3.3: CL-VO architecture with a windowed composition layer to integrate relative estimates over small windows w.

3.3.2 Experimental Results

3.3.2.1 Datasets

We employ three datasets for the experiments which consist of two public datasets (i.e. KITTI and Malaga dataset) and one self-collected dataset imitating firefighter walking pattern.

1. *KITTI Dataset.* KITTI dataset [45] is a well-known autonomous driving dataset for evaluating VO and visual SLAM algorithms. KITTI dataset contains 21 tra-

jectories in which ground truth poses are provided for Sequences 00-10 while the remaining sequences (11-21) are typically used for benchmarking the alrogithms. For all sequences, stereo images captured by Point Gray Flea2 color cameras (10 Hz, resolution: 1392 $\times$ 512 pixels, field of view: 90° $\times$ 35°) are provided. The ground truth trajectory is generated by the GPS/IMU localization unit projected into the coordinate system of the left camera after performing rectification. Although the dataset provides stereo imagery, we only use the left image for testing monocular VO algorithms. In general, we use KITTI odometry data Sequences 00-10 for quantitative evaluation and Sequences 11-21 for qualitative evaluation.

2. *Malaga Dataset.* The second dataset is the Malaga urban dataset [13], which is also collected in a driving scenario. Similar to KITTI, stereo images are provided by using Point Grey Research Bumblebee 2 stereo camera with 1024 $\times$ 768 image resolution, 100° horizontal field of view, and running at 20 fps. The ground truth path is generated by the GPS data. Similar to KITTI, we only utilize the left camera for testing monocular VO methods. We only use this dataset to test a pre-trained model without training or fine-tuning.

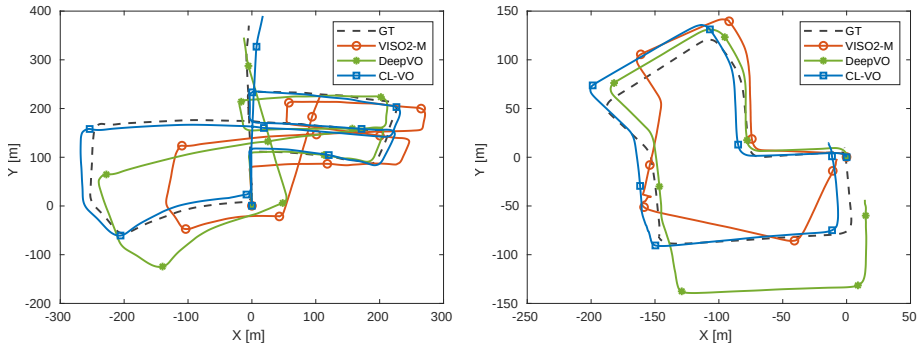
3. *Firefighter Walking Pattern.* The last dataset is our self-collected human motion data imitating a firefighter walking pattern. This dataset was collected in an indoor environment that consists of a corridor and a large room for approximately 1.5 hours. We used uEye global shutter camera mounted in a helmet, with VGA resolution (640 $\times$ 480), 65° $\times$ 45° field of views, and runs at 30 Hz. The ground truth was taken from a ViCon Motion Capture system with approximately 1mm accuracy. The firefighter walking motion contains sweeping hand and foot for inspecting obstacles in front of the user, which is very challenging for monocular VO since it creates a zigzag motion pattern. Moreover, the moving hand occasionally obstructs some parts of the image.

3.3.2.2 Competing Techniques

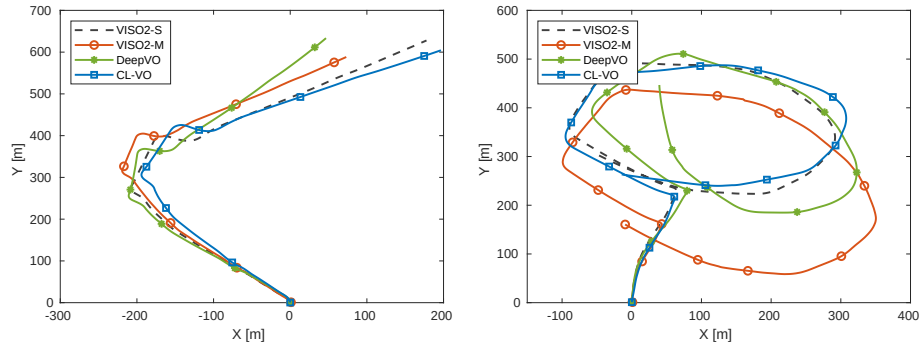
To evaluate the performance of CL-VO, we compare our method with the state-of-the-art feature-based and learning-based VO methods, namely VISO2 [46], ORB-SLAM [119], and DeepVO [161]. For VISO2, we use the monocular version (VISO2-M) for quantitative evaluation while we utilize the stereo version (VISO2-S) for qualitative comparison. We set the height of the camera in VISO2-M as described on each dataset paper to estimate the scale of the prediction. For ORB-SLAM, we used the result from [162] for quantitative evaluation. As for DeepVO, we constructed the DeepVO model with the same architecture and parameters as described in the paper. For each dataset, we trained DeepVO with the same settings as CL-VO (e.g. total training sequences, validation data, total epochs, optimizer, learning rate, etc.). We also train DeepVO with GA-CL to see how much improvement GA-CL can bring to DeepVO.

3.3.2.3 Implementation and Augmentation

We implemented CL-VO using Tensorflow and Keras, and ran the training code on a NVIDIA TITAN V GPU. Before training, we computed the dataset mean and used it to normalize the image intensity. In order to provide more trajectory variations, we generated sequences with random start and end points, and random lengths. In every epoch, we constructed 10 random trajectories for each training sequence. The training can extend to 200 epochs for each training stage which takes around 10 hours, or can be stopped earlier if the validation loss shows no improvement. We used the Adam optimizer with $1e-3$ as the initial learning rate. We also applied Dropout [144] with 0.2 dropout rate for regularizing the network. For parameter in Equations (3.3)-(3.6), we set $[\delta;\zeta] = [1;100]$ for the KITTI dataset, and $[\delta;\zeta] = [1;0.001]$ for the human motion dataset. For GA-CL setting, we mostly set the window $w = 2$ or 3 and $\alpha = 1$ for the 1st stage, $\alpha = 0.5$ for the 2nd stage, and $\alpha = 0.1$ for the 3rd stage as it get the best performance in KITTI dataset.



(a) Estimated trajectory from Sequences 05 and 07



(b) Estimated trajectory from Sequences 11 and 18

Figure 3.4: (a) Qualitative results from Sequences 05 and 07 and (b) Sequences 11 and 18 on KITTI dataset. Note that the ground truth pose is not available for KITTI Sequences 11-20.

3.3.2.4 Tests on KITTI Dataset

We performed two experiments on the KITTI dataset. The first experiment is conducted for KITTI Sequences 00-10 where precise ground truth is available such that quantitative evaluation can be conducted. The second experiment is aimed to test further the generalization of the network on KITTI testing Sequences 11-20. Since there is no ground truth available for KITTI Sequences 11-20, no quantitative evaluation is performed.

For the first experiment, we trained CL-VO on KITTI Sequences 00, 01, 02, 08, and 09, and tested on KITTI Sequences 03, 04, 05, 06, 07, and 10 as seen in [161]. Figure 3.4 (a) shows the qualitative results from Sequences 05 and 07. It can be seen

that all CL-VO predictions are relatively accurate and consistent against the ground truth. CL-VO significantly outerforms VISO2-M and DeepVO. As for VISO2-M, the VO estimation in Figure 3.4 (a) suggest that the scale estimation using fixed camera height is not robust against noise due to car jolts during driving [162]. Note that neither scale estimation nor post alignment to ground truth is conducted for CL-VO. The quantitative results can be seen in Figure 3.5 where CL-VO consistently yields better performance for both translation and rotation against the path length compared to VISO2-M and DeepVO. Table 3.1 details the frame-to-frame relative transformation errors of the compared algorithms for each testing sequences. The result indicates that CL-VO achieves more robust outputs than VISO2-M, ORB-SLAM, and DeepVO, although the performance is, as expected, worse than the stereo algorithm, i.e. VISO2-S. The table also shows that GA-CL can boost the performance of DeepVO by 21% and 16% for translation and rotation respectively. CL-VO achieves higher accuracy than DeepVO+GA-CL as it estimates more accurate optical flow through the cascade optical flow networks.

For the second experiment, we trained CL-VO on KITTI Sequences 00-10 and tested on KITTI testing Sequences 11-20. Qualitatively, we can see from Figure 3.4 (b) that CL-VO predictions are more similar to the stereo algorithm (VISO2-S) estimation than VISO2-M and DeepVO. This confirms that CL-VO can generalize well in new scenarios with different motion patterns and environments although it suffers from drift over time.

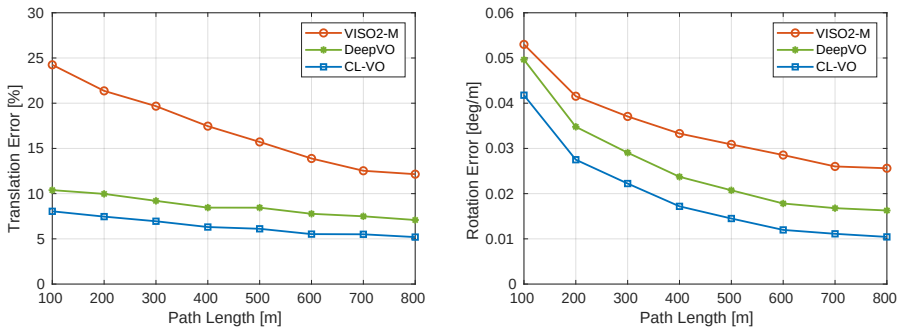


Figure 3.5: Translation and rotation errors against path length on KITTI dataset.

Table 3.1: Frame-to-frame relative translation and rotation errors on KITTI dataset among the competing approaches.

	Monocular VO										Stereo VO	
Seq	VISO2-M		ORB-SLAM		DeepVO		DeepVO+GA-CL (**ours**)		CL-VO (**ours**)		VISO2-S	
	trans(%)	rot(°)	trans(%)	rot(°)	trans(%)	rot(°)	trans(%)	rot(°)	trans(%)	rot(°)	trans(%)	rot(°)
03	28.14	**0.0230**	21.07	0.1836	10.71	0.0479	8.36	0.0353	**8.12**	0.0347	3.21	0.0325
04	33.92	**0.0177**	**4.46**	0.0560	9.95	0.0407	8.66	0.0308	7.57	0.0261	2.12	0.0212
05	14.65	0.0397	26.01	0.3427	8.02	0.0265	5.81	0.0210	**5.77**	**0.0200**	1.53	0.0160
06	19.54	0.0249	17.47	0.1717	**7.10**	0.0186	7.39	0.0183	7.66	**0.0166**	1.48	0.0158
07	12.69	0.0647	24.53	0.3890	16.20	0.0380	9.79	0.0413	**6.79**	**0.0300**	1.85	0.0191
10	30.39	0.0306	86.51	0.9890	9.04	0.0391	8.30	0.0303	**8.29**	**0.0294**	1.17	0.0130
avg	23.22	0.0334	30.01	0.3553	10.17	0.0351	8.05	0.0294	**7.37**	**0.0267**	1.89	0.0196

3.3.2.5 Generalization in Malaga Dataset

In order to further test the generalization ability of the proposed framework, we tested CL-VO on the Malaga dataset without any further training or fine-tuning. We used the CL-VO model which is trained on KITTI dataset Sequences 00-10 and tested directly on the Malaga image data. Since the image resolution in the Malaga dataset is different from KITTI, we cropped the images to the KITTI image size. Some image information is expected to get lost during this cropping process which might affect the final predictions.

Figure 3.6 depicts the test results on Malaga dataset Sequences 03, 04, and 09, superimposed on Google Map. Since the Malaga dataset does not have ground truth, a quantitative evaluation cannot be conducted. However, since frequent GPS data is available, we still can perform qualitative comparison. Moreover, since the dataset also contains stereo images, we can generate VO estimation from VISO2-S for comparison with stereo algorithms. As we can see from Figure 3.6, CL-VO predictions are close to GPS and VISO2-S in those three sequences. It is significantly better than VISO2-M and DeepVO, although it suffers from drift. This experiment further confirms that CL-VO generalizes to other datasets which are collected with different

cameras in different environments. This also shows that CL-VO generalizes better than DeepVO as the drift of DeepVO is larger on the test sequences.

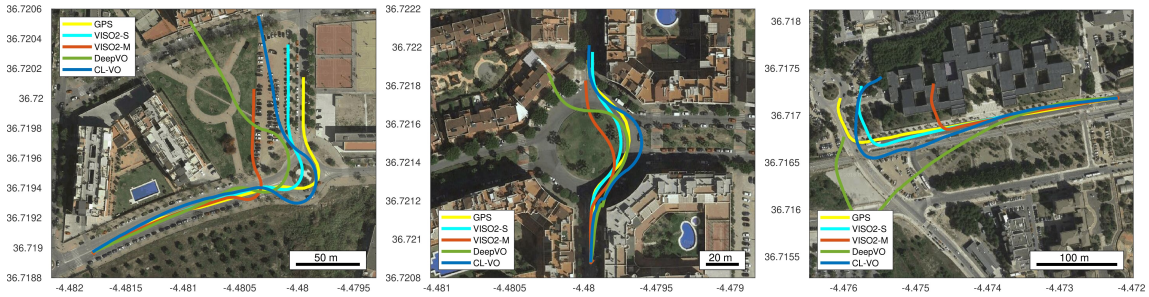
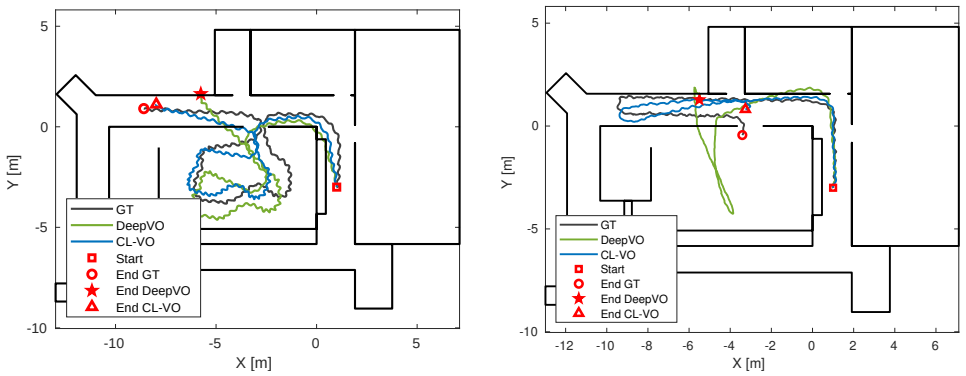


Figure 3.6: Generalization tests on Malaga Dataset from Sequences 03, 04, and 09 respectively, superimposed on Google Map. DeepVO and CL-VO are only trained on KITTI dataset Sequences 00-10 and tested it directly without fine-tuning.



(a) Sequence 20 (b) Sequence 21

Figure 3.7: Test on human walking data in an office building. (a) Test with the human walks in and out of room. (b) Test in corridor involving U-turn motion.

3.3.2.6 Tests on Human Motion Dataset

We divided the human motion dataset into train and test sets: 1 hour and 15 minutes for training and the remaining 15 minutes for testing. We subsample one frame for every six images to provide sufficient displacement between consecutive frames.

Figure 3.7 shows the qualitative results on one of the test sequences. It can be seen that CL-VO performs better than DeepVO as the prediction is closer to the ground truth. While CL-VO successfully tracks the camera movement, DeepVO fails to

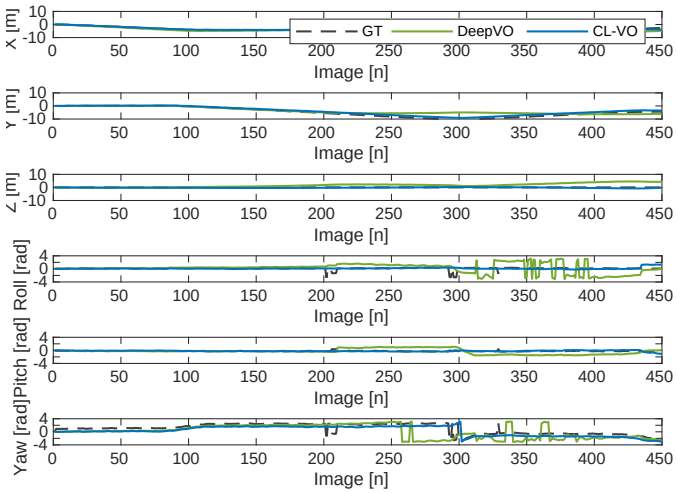


Figure 3.8: The 6-DoF camera poses compared to the ground truth poses in Seq 20 with human walking data.

perform turning accurately which leads to much larger drift. In particular, Figure 3.7 (b) shows that DeepVO prediction can drift quite significantly as the trajectory moves in the wrong direction. This indicates that training using relative transformation only without injecting any information about geometric (e.g. composite loss), can produce bias in pose estimation especially when there is not enough training data. Figure 3.8 shows the 6-DoF translation (x, y, z) and orientation (roll, pitch, yaw) of CL-VO compared with DeepVO and ground truth. It is clear that CL-VO tracks the changes on translation and orientation accurately. Figure 3.9 illustrates the distribution of the absolute errors (RMSE). CL-VO significantly outperform DeepVO, achieving less than 2 meters errors during 100% of testing time.

3.3.2.7 The Impact of Geometry-Aware Curriculum Learning

We performed an ablation study to understand the impact of the geometry-aware curriculum learning (GA-CL). We compare the performance of the proposed network when it is trained with curriculum, with reversed curriculum (anti-curriculum), and without curriculum. For training without curriculum, we use two loss functions, namely the standard relative loss and the bounded pose regression loss with $w = 2$ and $\alpha = 0.5$. For the anti-curriculum, the stages described in Section 3.3.1.3 are

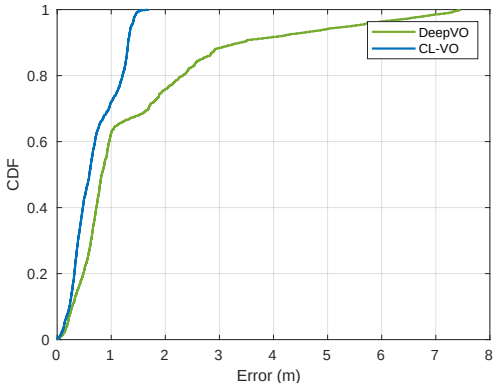


Figure 3.9: CDF of RMS absolute errors for all test sequences in human walking data.

reversed. All competing networks are trained with the same setting except GA-CL and anti-curriculum changes the parameter of the objective function at the end of each training stage.

Figure 3.10 depicts the key results of this study. As expected, directly training the network with the bounded loss is more difficult to converge although the performance gradually improves in later stages of training. On the other hand, the network trained with the relative loss already reaches a stable state in the first stages of training. It only improves slightly afterwards or can even lead to overfitting as the accuracy of the rotation part decreases. The anti-curriculum gets very low accuracy in the beginning although the performance improves after training with relative loss. Finally, the network trained with GA-CL can converge and generalize better which results in significantly lower translation and rotation errors in each training stage.

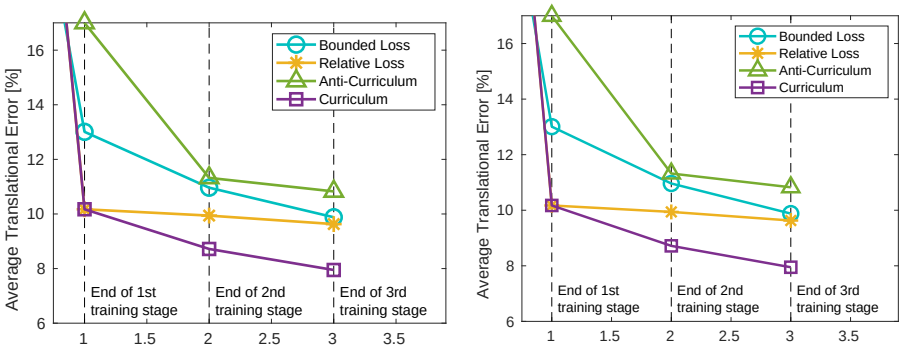


Figure 3.10: The impact of GA-CL algorithm on translation and rotation errors.

One possible explanation for this performance gain is that GA-CL can be regarded as a special form of transfer learning, where the initial tasks (i.e. minimizing relative transformation loss) are used to guide the learner such that it can perform better on the final task (i.e. minimizing bounded pose regression loss). While the motivation of conventional transfer learning is to improve the generalization by sharing model weights across tasks, GA-CL introduces the idea of guiding the optimization process, either for faster convergence or better local minima [9]. Another perspective is GA-CL can be seen as a way of gradually injecting domain knowledge into DNNs by progressively revealing more information to the network over time via objective function alteration.

3.4 Attention Network

In this section, we describe an alternative way of improving the performance of DNN-based VO by only processing important information from the input image through an attention network. We call the proposed network as SalientDVO. We first explain in detail the structure of the attention network, followed by the experimental results to validate the devised attention network.

3.4.1 Approach

As stated in Section 3.3, the standard DNN architecture for VO estimation typically consists of a feature extractor and a pose regressor network. The former extracts features by employing CNN as the main building blocks. The latter regresses the 6-DoF camera poses through fully connected or recurrent layers. We will first describe the feature extractor and the pose regressor of SalientDVO, followed by the proposal of the attention mechanism.

3.4.1.1 Feature Extractor and Pose Regressor Network

The feature extractor network is designed to automatically learn geometric features necessary for estimating VO. Inspired by [162] we utilize FlowNet [34] to extract

optical flow-like features since they are more suitable for tackling geometry problems that do not rely on appearance. We directly use the FlowNet model except that we remove the refinement part since we want to produce the latent fatures that can be directly consumed by the pose regressor network. For the input, we concatenate two consecutive images such that the network can learn pixel displacements. Formally, given input image $\mathbf{I}_t \in \mathbb{R}^{w \times h \times 2c}$ at time t, where w, h, and c are the image width, height, and channels respectively, the feature extractor network performs the following operation:

$$\mathbf{x}_t = cnn(\mathbf{I}_t) \tag{3.11}$$

where $cnn(.)$ is a series of convolutional operations and $\mathbf{x}_t \in \mathbb{R}^{\hat{\mathbf{w}} \times \hat{\mathbf{h}} \times \hat{\mathbf{c}}}$ is the output feature map at time t. $\hat{\mathbf{w}}$, $\hat{\mathbf{h}}$, and $\hat{\mathbf{c}}$ are the feature map weight, height, and channels respectively. As we use FlowNet, $\hat{\mathbf{c}}$ is equal to 1024 and $\hat{\mathbf{w}} = w/64$ and $\hat{\mathbf{h}} = h/64$.

The pose regressor part consists of Fully Connected (FC) layers and optionally RNN layers to model long-term motion dynamic of the camera. The output features $\mathbf{x}_t$ will be fed to a series of FC layers to produce 6-DoF poses as follows:

$$\mathbf{O}_t = f_c(\mathbf{x}_t) \tag{3.12}$$

where $f_c(.)$ is a series of linear operation performed by the FC layers and $\mathbf{O}_t = [\mathbf{t}; \mathbf{r}] \in \mathbb{R}^6$. Terms $\mathbf{t}$ and $\mathbf{r}$ are the translation and rotation of the camera represented in Euler angle. The main difference from standard DNN-based VO is that we add attention network that will be described in the following section.

3.4.1.2 Network Architecture

The main objective of the attention network is to improve the network performance by placing relative importance on image features. For the network to generate this meaningful attention, our insight is to generate an attention map by conditioning on the previous camera poses. However, instead of directly using the camera poses, we propose to use the latent features that represent the sequential motion dynamics of the

poses from current sequences using ConvLSTM [167]. Unlike a standard LSTM which eliminates the spatial structure of the data, ConvLSTM keeps the spatial structure of the input by replacing FC layers inside the network with convolution operation. Keeping the spatial structure of the input is important to generate a meaningful attention map. Given the input features $\mathbf{x}_t$ at time t, the previous hidden state $\mathbf{h}_{t-1}$ and memory cell $\mathbf{C}_{t-1}$, the ConvLSTM updates can be formulated as:

$$\begin{aligned}
\mathbf{i}_t &= \sigma(\mathbf{W}_{xi} * \mathbf{x}_t + \mathbf{W}_{hi} * \mathbf{h}_{t-1} + \mathbf{W}_{ci} \circ \mathbf{C}_{t-1} + b_i) \\
\mathbf{f}_t &= \sigma(\mathbf{W}_{xf} * \mathbf{x}_t + \mathbf{W}_{hf} * \mathbf{h}_{t-1} + \mathbf{W}_{cf} \circ \mathbf{C}_{t-1} + b_f) \\
\mathbf{g}_t &= \tanh(\mathbf{W}_{xg} * \mathbf{x}_t + \mathbf{W}_{hg} * \mathbf{h}_{t-1} + b_g) \\
\mathbf{C}_t &= \mathbf{f}_t \circ \mathbf{C}_{t-1} + \mathbf{i}_t \circ \mathbf{g}_t \\
\mathbf{o}_t &= \sigma(\mathbf{W}_{xo} * \mathbf{x}_t + \mathbf{W}_{ho} * \mathbf{h}_{t-1} + \mathbf{W}_{co} \circ \mathbf{C}_t + b_o) \\
\mathbf{h}_t &= \mathbf{o}_t \circ \tanh(\mathbf{C}_t)
\end{aligned} \tag{3.13}$$

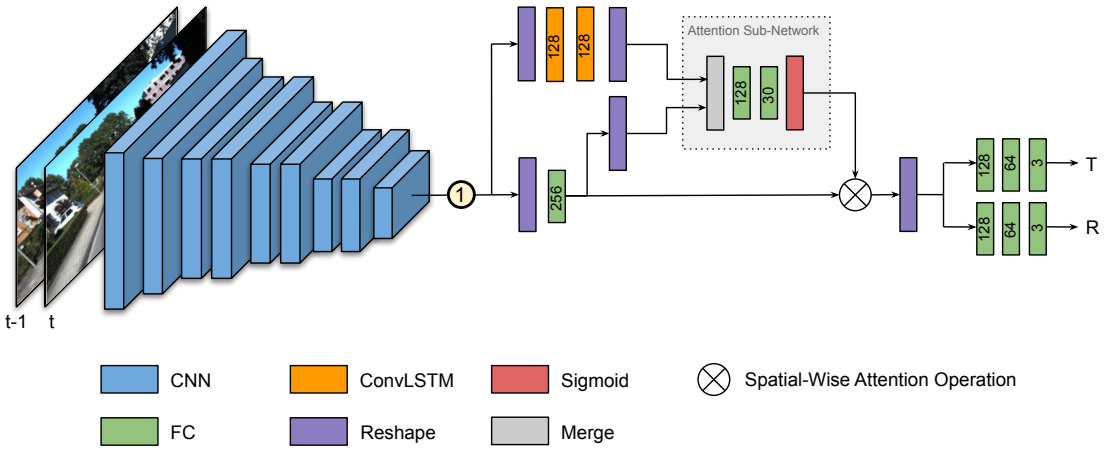
where $*$ is the convolution operation and $\circ$ is the hadamard product. The attention map is then learnt from the concatenation of the hidden state of ConvLSTM with the current, reshaped feature vectors $\hat{\mathbf{x}}_t \in \mathbb{R}^{\hat{\mathbf{w}}\hat{\mathbf{h}} \times \hat{\mathbf{c}}}$. However, for ease of modelling and weight efficiency, we transform $\hat{\mathbf{x}}_t$ into $\mathbf{v}_t$ through a single layer perceptron. The full steps to generate the attention map $\alpha_t \in \mathbb{R}^{\hat{\mathbf{w}}\hat{\mathbf{h}}}$ is described as follows:

$$\begin{aligned}
\mathbf{v}_t &= \text{relu}(\mathbf{W}_{xv}\hat{\mathbf{x}}_t) \\
\mathbf{z}_t &= \text{relu}(\mathbf{W}_z \text{merge}(\mathbf{h}_t, \mathbf{v}_t)) \\
\alpha_t &= \text{sigmoid}(\mathbf{W}_\alpha \mathbf{z}_t)
\end{aligned} \tag{3.14}$$

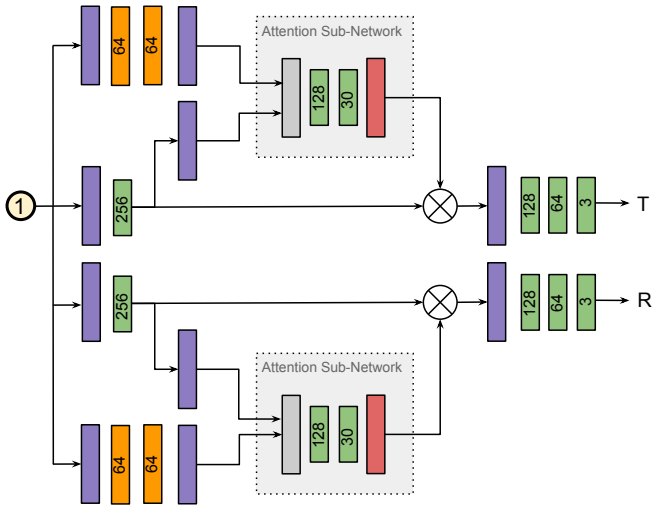
where $\mathbf{W}_{xv}$, $\mathbf{W}_z$, and $\mathbf{W}_\alpha$ are the learnt weights.

In image captioning problems [168], [104], α_t is usually used to blend together the feature maps via a weighted average operation. However, performing weighted average operations on the feature maps might erase the spatial information related to salient and non-salient regions which is important for VO. Thus we instead propose a spatial-wise attention mechanism which performs an element-wise multiplication $\otimes$ across all channel dimensions as follows:

$$\mathbf{c}_t = \alpha_t \otimes \mathbf{v}_t \tag{3.15}$$



(a) Joint Attention Network



(b) Parallel Attention Network

Figure 3.11: Two architectures of SalientDVO with (a) joint attention network (b) parallel attention network.

where $\mathbf{c}_t$ are the output context features (borrowing the term from image captioning). The final pose is learnt from these context features such that Equation (3.12) becomes $\mathbf{O}_t = f_c(\mathbf{c}_t)$.

Figure 3.11 shows the architecture of the proposed attention network (SalientDVO) which also describes all formulations to generate context features. Figure 3.11 (a) describes the network with joint attention network for translation and rotation estimation (denotes SalientDVO-j), while Figure 3.11 (b) shows parallel attention network which decouples the attention for translation and rotation (denotes SalientDVO-

p). We propose the decoupled attention network as we conjecture that the network might need to learn different attention maps for translation and rotation. This conjecture is supported by previous works for conventional VO (e.g. ORB-SLAM2 [120]) which states that salient features in the image space have different sensitivity with respect to translation and rotation. For the parallel attention network, the final output pose is generated by $\mathbf{t} = f_c(\mathbf{c}_{1,t})$ and $\mathbf{r} = f_c(\mathbf{c}_{2,t})$ where $\mathbf{c}_{j,t}$ are the context features for sub-network $j \in \{1, 2\}$.

3.4.1.3 Objective Function

As the network is trained by supervision, the objective is to minimize the relative error between the prediction and the ground truth. The loss function is usually composed of Mean Square Error (MSE) of translations $\mathbf{t}$ and rotations $\mathbf{r}$ as follows:

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \left(\left\| \hat{\mathbf{t}}_i - \mathbf{t}_i \right\|^2 + \zeta \left\| \hat{\mathbf{r}}_i - \mathbf{r}_i \right\|^2 \right) \tag{3.16}$$

where $\hat{\mathbf{t}}$ and $\hat{\mathbf{r}}$ are the ground truth translation and rotation respectively, and ζ is a weight used to balance translation and rotation terms as in [77, 161].

3.4.2 Experimental Results

3.4.2.1 Implementation and Training Details

We implemented SalientDVO using Keras with Tensorflow backend and ran the training code on a NVIDIA TITAN V GPU. We trained and evaluated the network on the KITTI VO/SLAM benchmark, which consists of 11 sequences with ground truth. Before training, we scaled each image into 192×640 and subtracted the dataset mean. We used sequences 00, 01, 02, 08, and 09 for training and sequences 03, 04, 05, 06, 07, and 10 for testing. The training runs for up to 200 epochs or can be stopped earlier if the validation loss does not improve anymore. Since we employed FlowNet as the feature extractor, we only trained the pose regressor network. We used Adam optimizer with $1e-3$ learning rate and Dropout [144] with 0.2 dropout rate for reg-

ularizing the network. $K = 10$ is used for the parameter of MDN. To balance the translation and rotation term during training, we set ζ to 100.

3.4.2.2 Competing Algorithms

We compare SalientDVO with state-of-the-art monocular VO algorithms that use geometry-based and learning-based approaches. For conventional geometry-based approaches, we use the monocular version of VISO2 [46], ORB-SLAM2 [120], and DynaSLAM [10]. We employ VISO2 as the most basic but popular implementation of feature-based monocular VO. For VISO2's scale estimation, we set the height of the camera as 1.65m. We choose ORB-SLAM2 and DynaSLAM as the representation of visual SLAM system that performs clustering of the image features based on how they effect translation and rotation components. DynaSLAM is the improvement of ORB-SLAM2 by incorporating a mechanism to segment dynamic objects. For both SLAM systems, we turned off the loop closure function to enable fair comparison with VO. For learning methods, we compare with DeepVO [161] model that we trained with the same setting as our SalientDVO.

3.4.2.3 The Impact of Attention Network

To inspect the impact of the attention network, we compare the performance of the network with attention (where $\mathbf{O}_t = f_c(\mathbf{c}_t)$) and without attention (where $\mathbf{O}_t = f_c(\mathbf{v}_t)$). We measure the performance using Root Mean Square (RMS) Relative Pose Error (RPE) and Absolute Trajectory Error (ATE) [146]. Table 3.2 shows the average results for all test sequences. It can be seen that incorporating relative importance among image features (as $\mathbf{c}_t = \alpha_t \otimes \mathbf{v}_t$) boosts significantly the network performance either on RPE or ATE as the network harnesses more important features for VO estimation. In particular, it improves translation and rotation estimation by 27.8% and 43.1% respectively.

Table 3.2: The Impact of Attention Network on Accuracy

Methods	RMS RPE		RMS ATE
	t (m)	**r** (°)	
Without Attention	0.1434	0.1879	43.14
With Attention	**0.1035**	**0.1069**	**26.11**

3.4.2.4 Comparison with State-of-the-art

Figures 3.12 and 3.13 depict the qualitative evaluation of the proposed and the competing approaches. As it can be seen, SalientDVO produces a trajectory closer to the ground truth than the competing approaches. Table 3.3 shows the comparison of SalientDVO with the competing approaches in terms of ATE. As ORB-SLAM2 and DynaSLAM do not have a way of estimating scale, we need to align and scale with respect to the ground truth using the closed-form Horn approach as in [146]. To perform a fair comparison with other competing approaches, we also perform alignment and scaling for other approaches. It can be seen that SalientDVO-p considerably outperforms competing approaches. Without loop closure, ORB-SLAM2 and DynaSLAM experience a large drift and require a large scaling value to fit the ground truth. Note that as seen in Table 3.3, both ORB-SLAM2 and DynaSLAM require a large scaling value (around 13.5 multiplier) to align with the ground truth. On the contrary, SalientDVO implicitly learns the scale from the training data such that the predicted scale of the estimated trajectory is already very close to the ground truth (consistently uses small scaling multiplier) and generalizes very well. VISO2 also requires a reasonable scaling factor as the scale of the prediction can be calculated based on the height of the car that has been specified before. The result in Table 3.3 also shows that performing attention separately for translation and rotation yields better performance (by 26.87%) than performing it jointly. This confirms our conjecture that each component of translation and rotation require different parts of the image to attend.

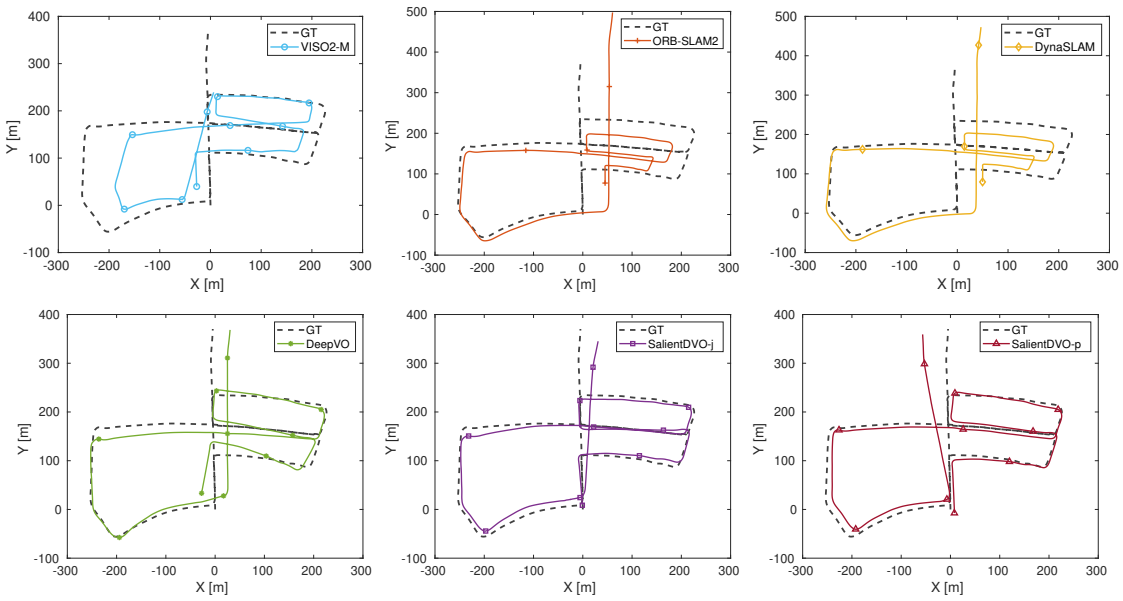


Figure 3.12: Output trajectories for KITTI Seq 05.

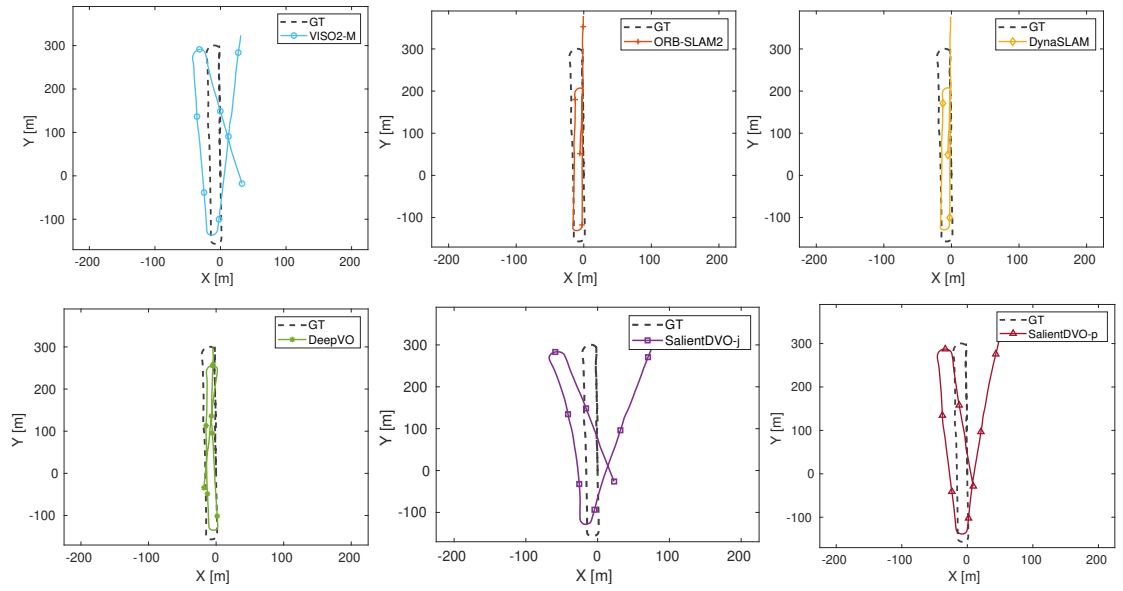


Figure 3.13: Output trajectories for KITTI Seq 06.

3.4.2.5 Visualization of the Attention Mask

To better understand the network behaviour, we visualize the attention map by up-sampling and overlaying it on the input image. Table 3.4 (row 1) shows the attention produced by SalientDVO-p when the camera moves forward. In this condition, despite the presence of some noise, the translation network places attention on the lower part

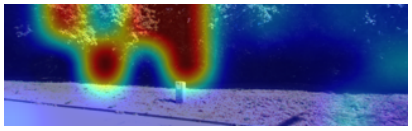
Table 3.3: Absolute Trajectory Errors (ATE) among Competing Methods

Methods	S	Sequences						Mean
		03	04	05	06	07	10	
VISO2	1.2	2.05	2.34	63.17	28.05	19.93	52.08	28.01
ORB-SLAM2	13.5	1.06	1.21	42.66	47.32	15.26	5.99	18.92
DynaSLAM	13.5	**0.86**	**1.20**	36.17	46.62	16.54	**4.81**	17.69
DeepVO	1.11	7.55	3.45	22.14	28.77	21.37	13.08	16.06
SalientDVO-j	**1.01**	5.33	4.59	22.03	36.59	19.17	21.90	18.27
SalientDVO-p	1.03	6.07	3.45	**19.63**	**27.19**	**11.67**	12.17	**13.36**

*S indicates the average scale multiplier required to fit the ground truth.
The colour in red shows approaches that require large scaling multiplier.

of the image which corresponds to closer feature points. These close feature points are important for accurate translation estimation as found by ORB-SLAM2 [120] or [94]. On the other hand, the rotation network attends to the top area around the vanishing point (boundary between sky and buildings). These points are usually further apart and good for rotation estimation which is consistent with the findings in the feature-based approach [120]. Table 3.4 (row 2) depicts the attention when the camera turns left. In the top figure, the camera rotates and translates at the same time such that the translation network attend to almost all area of the images as translational flow is observed in the whole image. The rotation network, however, still attends to distant points on the top image although it also focuses the attention to the left area which is the direction of where the camera moves. Nevertheless, we also observe some conditions when the translation network produces zero attention everywhere as seen in Table 3.4 (row 3). This happens mostly in conditions when the camera performs pure rotation or with very small changes of translation such that the pose is sufficiently determined by the rotation network. This also indicates that the network has the capability to selectively choose features from translation or rotation networks depending on the input.

Table 3.4: Visualization of Attention Map

Motion	Translation Sub-Network	Rotation Sub-Network
Forward		
Left		
Pure Rotation		

3.5 Discussion

Despite its great performance, our model seems to produce larger error in particular scenarios. Fig. 3.14 shows that the model tends to generate higher error when part of images are obstructed by dynamic objects. This happens in the human motion data with head-mounted camera in which we try to imitate firefighter walking pattern by sweeping feet and hand to inspect obstruction. While in most cases only small part of images are obstructed by the hand, in some conditions the hand precludes a significant part of images as seen in Fig. 3.14, making the relative pose estimation erroneous. Nevertheless, the errors are still considerably small such that an accurate trajectory can be generated as depicted in Fig. 3.7 (a).

Another limitation is related to domain adaptation. It is widely known in machine learning community that deep learning has some sort of domain adaptation problem. The same problem also found in our model. The model that we trained on KITTI dataset will not work well when we test it on our human walking dataset. Because of this reason, we trained the model separately for KITTI and human walking dataset. On the other hand, conventional approaches do not have this problem when the algorithm will be implemented to different environment although the camera intrinsic and extrinsic parameters have to be re-calculated.

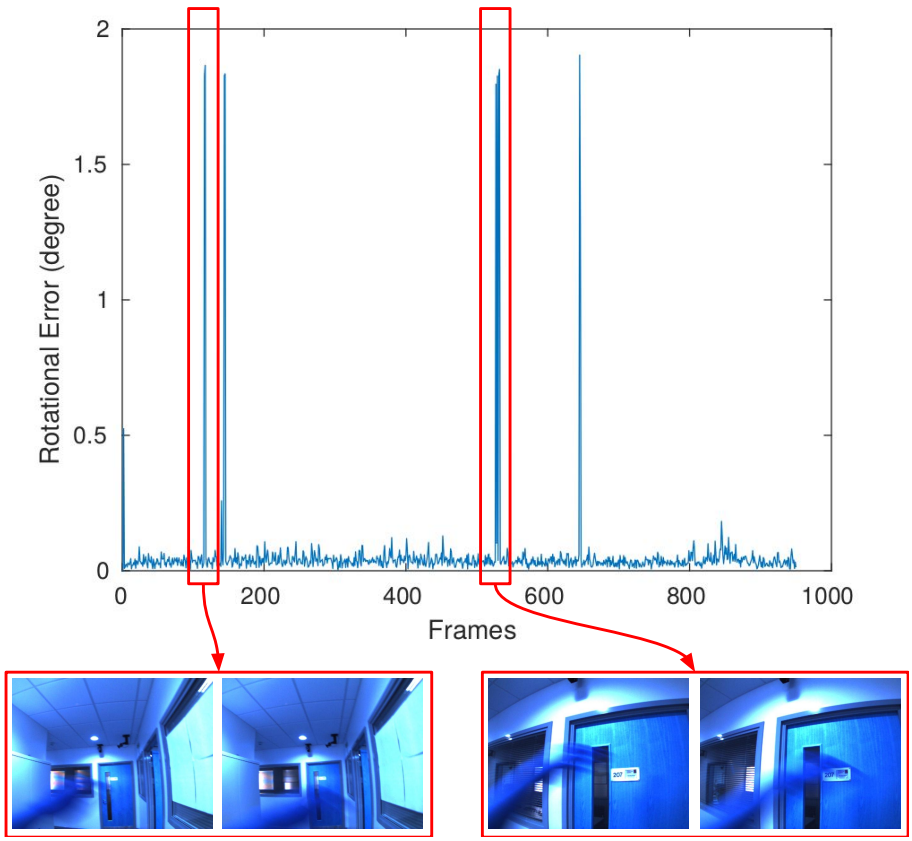


Figure 3.14: Our model produces larger error when a significant part of images are obstructed by dynamic objects (hand motion when performing sweeping). The data is taken from a sequence in Fig. 3.7 (a).

3.6 Conclusion

In this chapter, we have presented a novel optimization strategy for DNN-based VO using Geometry-Aware Curriculum Learning (GA-CL). We have shown that better objective function (GA-CL) which explicitly incorporates geometric consistency among window of frames is one of the key success to realize accurate DNN-based VO. In particular, we have shown that GA-CL can blend together the windowed-based composite loss and relative transformation loss in one objective function, and can train them successfully using the principle of curriculum learning. We show that this strategy can significantly improve the generalization ability of the network for both translation (by 21%) and rotation (by 16%), compared to a network that is trained

without GA-CL. We have also presented a novel attention network for odometry estimation by conditioning the attention network on the current latent poses. We showed that decoupling the attention network for translation and rotation gives better results (by 26.87%) than joining them. Visualization of the attention map gives insight into what the network learn to attend during training and what the network actually sees when making predictions. Our experiments show that we are one step forward to achieve robust and accurate DNN-based VO. However, the limitation of this method is that the DNN-model is not transferable to different domains (e.g. the model trained on autonomous car motion will be largely drifted when it is tested in human motion). This will be addressed in the future work.

Chapter 4

Efficient Deep Neural Odometry

4.1 Introduction

In Chapter 3, we have shown that the accuracy of DNN-based VO can be significantly improved by training the model using Geometry-Aware Curriculum Learning (GA-CL) or by incorporating visual attention in the network. Despite this success, the proposed networks typically require tens or even hundreds of million weights. This huge computational and space requirement brings the efficiency problem to the table. If not addressed, DNN-based VO models cannot be widely implemented in resource-constrained environments (e.g. mobile phones, robotic platform, quadcopter, etc.). To compound the issue, these applications typically require near real-time inference.

Within the last few years, there have been tremendous efforts towards compressing DNNs. State-of-the-art approaches for network compression such as quantization [49, 31, 67], pruning [57, 54, 169], or low-rank decomposition [148, 11] can yield significant speed-ups but at the cost of accuracy. On the other hand, an approach called Knowledge Distillation (KD) proposed by Hinton et al. [62] offers to recover the accuracy drop by transferring the knowledge of a large teacher model to a small student model. Some recent works show that a small network trained by KD could match or even exceed the accuracy of a large network if it is trained with careful optimization [132]. Moreover, the compressed network typically has similar characteristics to the original network except it has fewer layers or less weights. This makes KD a

compelling approach as the distilled network can be further compressed with other compression techniques.

In this chapter, we will explore the KD approach in order to resolve the efficiency problem of DNN-based VO. While other compression techniques have shown great results in reducing the network size, each of them has its own drawbacks. Network quantization can dramatically reduce the number of weights but at the cost of significant reduction in accuracy [23]. Pruning requires additional implementation of sparse matrix multiplication which needs more resource consumption and specialized hardware and software [107]. Low-rank decomposition generally obtains lower compression ratio although extreme decomposition can be performed with high reduction in accuracy. In this sense, we choose KD over other compression techniques as it has been shown in [62] that KD could reduce the network weights dramatically while at the same time be able to keep the performance similar to the original network or sometimes even surpass them as displayed by [132]. Moreover, all other compression approaches are orthogonal (independent) methods compared to KD, in which each of them technically can be combined with KD. However, state-of-the-art KD approaches [62, 101, 170, 160, 127] typically demonstrated the performance only for classification problem which raises a question as to whether KD can be applied to VO, a regression problem. Finally, we also investigate whether KD can be fused with other compression techniques such as factorization.

The rest of this chapter is organized as follows. Section 4.2 introduces the research problem we tackle in this chapter. Section 4.3 present the research questions and contributions. Section 4.4 explains the proposed approach to apply KD to pose regression. Section 4.5 describes the implementation details of the proposed KD approach. Section 4.6 evaluates the performances and provides the comparison with competing approaches. Section 4.8 concludes this chapter.

4.2 Research Problem

Most KD approaches [62, 132, 101, 170, 160, 95, 127] and other compression techniques [156, 31, 58, 107, 131] focus on the problem of classification. KD works very well in the context of classification since it has the advantage of "dark knowledge" which refers to the softened logits output of the teacher. This provides more information than mere one-hot encoding of the class label and contains hidden knowledge about the correlations of class labels [62]. By using the logits output for training, the student network can emulate the generalization capability of the teacher network. However, this advantage does not exist in pose regression, or even regression problem in general. In the regression problem, a deep regression network predicts sequential, continuous, values which have the exact same characteristics as the ground truth, with the exception of being plagued with an unknown error distribution. Without access to any dark knowledge, it is unclear how KD could help in compressing a regression network. In recent surveys, it is even stated that the main drawback of KD is that it only works for classification problems [23].

KD methods for classification rely solely on the teacher prediction without considering the error made with respect to the ground truth. In regression however, the real-valued predictions are unbounded, and hence the teacher network can give highly erroneous guidance to the student network. Previous work [22] alleviated this issue by using teacher loss as an upper bound. However, it was designed for standard bounding box regression which has different characteristics to pose regression as it belongs to $SE(3)$ (Lie Groups). Moreover, they directly transferred the knowledge from the teacher network to the student network without filtering good and bad example.

4.3 Contributions

Based on the research problem elaborated in the previous section, we define the following research questions: (1) *Can Knowledge Distillation (KD) be applied to the DNN-based pose regression problem despite the apparent lack of dark knowledge? If*

yes, what kind of KD approach is suitable for DNN-based VO? (2) *How much efficiency can KD bring to DNN-based VO?* (3) *Can we fuse KD with another compression technique such as factorization?* The following contributions explain at a high level how the questions above have been addressed:

- We study different ways to apply KD to DNN-based VO by investigating various methods to blend the loss of the student both with respect to ground truth and with respect to teacher.
- We propose to transfer the knowledge from the teacher network to the student network only when we trust the teacher. We achieve this by using the normalized teacher loss as a confidence score to attentively learn from examples that the teacher is good at predicting. We apply this in the final objective function as Attentive Imitation Loss (AIL) and in the intermediate layer as Attentive Hint Training (AHT).
- We perform extensive experiments on KITTI and Malaga datasets which show that our proposed approach can reduce the number of student parameters by up to 92.95% (2.12× faster) whilst keeping the prediction accuracy very close to that of the teacher.
- We demonstrate that our proposed distillation approach can be combined with other compression techniques, such as low-rank factorization, to further boost the parameter reduction. In our experiment on KITTI dataset, we showed that this combination can reduce the teacher parameters by up to 97.39% with the same or even slightly better performance.

Some part of the contributions and experimental evaluations in this chapter have been described in the following published paper:

- **M. R. U. Saputra**, Pedro P. B. de Gusmao, Y. Almalioglu, A. Markham, and N. Trigoni. "Distilling Knowledge From a Deep Pose Regressor Network". In IEEE/CVF International Conference on Computer Vision (**ICCV**), 2019.

4.4 Approach

4.4.1 Blending Teacher, Student, and Imitation Loss

As mentioned above, it is unclear how KD can be applied to the pose regression problem. In classification problems, KD only takes into account the teacher prediction in the knowledge transfer. It does not consider the teacher's prediction error since all information are already available in the teacher (logits) prediction. Since this is not the case for the pose regression problem, our intuition tells us that we should utilise the teacher's error with respect to ground truth to provide more information during the knowledge transfer. To this end, we study different ways to blend together the loss of the student's prediction both with respect to ground truth and with respect to the teacher's prediction. For simplicity, we refer to the teacher network as T and to the student network as S. We refer to the error of S with respect to ground truth as *student loss* and the loss of T with respect to ground truth as *teacher loss*. We refer to *imitation loss* ($\mathcal{L}_{imit}$) for the errors of S with respect to T because S tries to imitate T's prediction. The following outlines different formulations and rationale of blending teacher, student, and imitation loss.

1. **Minimum of student and imitation loss**. In the first formulation, we make a strong assumption that teacher network T has good prediction accuracy in all conditions. In this case, as T's prediction will be very close to the ground truth, it does not really matter whether we minimize error of student network S with respect to ground truth or with respect to T. However, since we do not know whether S's prediction will be closer to T's prediction or to the ground truth, the simplest way to improve S's prediction is to minimize whichever one is smaller as follows:

$$\mathcal{L}_{reg} = \frac{1}{n}\sum_{i=1}^{n} \min\left(\left\|\mathbf{p}_S - \mathbf{p}_{gt}\right\|_i^2, \left\|\mathbf{p}_S - \mathbf{p}_T\right\|_i^2\right) \tag{4.1}$$

where $\mathbf{p}_S$, $\mathbf{p}_T$, and $\mathbf{p}_{gt}$ are S's prediction, T's prediction, and ground truth labels respectively. Note that $i = 1, ..., n$ indicates the image sample.

2. **Imitation loss as an additional loss**. Instead of seeking the minimum between the student and imitation loss, we can use the imitation loss as an additional loss term for the student loss. In this case, we regard the imitation loss as another way to regularize the network and prevent the network from overfitting [62]. Then, the objective function becomes

$$\mathcal{L}_{reg} = \frac{1}{n}\sum_{i=1}^{n}\left(\alpha \left\|\mathbf{p}_S - \mathbf{p}_{gt}\right\|_i^2 + (1-\alpha)\left\|\mathbf{p}_S - \mathbf{p}_T\right\|_i^2\right) \tag{4.2}$$

where α is a scale factor used to balance the student and imitation loss. This formulation is similar to the original formulation of KD for classification as seen in Equation (2.16) except the cross-entropy loss is replaced by the regression loss.

3. **Teacher loss as an upper bound**. Equations (4.1) and (4.2) assume that T has very good generalization capability in most conditions. However in practice, T can give very erroneous guidance for S. There is a possibility that in adverse environments, T may predict camera poses that are far from to the ground truth pose. Hence, instead of directly minimizing S with respect to T, we can utilize T as an upper bound. This means that S's prediction should be as close as possible to the ground truth pose, but we do not add additional loss for S when its performance surpasses T [22]. In this formulation, Equation (4.2) becomes the following equation

$$\mathcal{L}_{reg} = \frac{1}{n}\sum_{i=1}^{n}\left(\alpha \left\|\mathbf{p}_S - \mathbf{p}_{gt}\right\|_i^2 + (1-\alpha)\mathcal{L}_{imit}\right) \tag{4.3}$$

$$\mathcal{L}_{imit} = \begin{cases} \left\|\mathbf{p}_S - \mathbf{p}_T\right\|_i^2, & \text{if } \left\|\mathbf{p}_S - \mathbf{p}_{gt}\right\|_i^2 > \left\|\mathbf{p}_T - \mathbf{p}_{gt}\right\|_i^2 \\ 0, & \text{otherwise} \end{cases} \tag{4.4}$$

4. **Probabilistic imitation loss (PIL)**. As stated before, T is not always accurate in practice. Since there is some degree of uncertainty in T's prediction, we can explicitly model this uncertainty with a parametric distribution. For

example, we can model the imitation loss using Laplace's distribution

$$\mathbb{P}\left(\mathbf{p}_S|\mathbf{p}_T,\sigma\right) = \frac{1}{2\sigma}\exp\frac{-\left\|\mathbf{p}_S-\mathbf{p}_T\right\|}{\sigma} \tag{4.5}$$

where σ is an additional quantity that S should predict. In this case, the imitation loss is turned into minimizing the negative log likelihood of Equation (4.5) as follows

$$-\log\mathbb{P}\left(\mathbf{p}_S|\mathbf{p}_T,\sigma\right) = \frac{\left\|\mathbf{p}_S-\mathbf{p}_T\right\|}{\sigma} + \log\sigma + \text{const.} \tag{4.6}$$

The final objective is obtained by replacing $\mathcal{L}_{imit}$ in Equation (4.3) with Equation (4.6) as follows:

$$\mathcal{L}_{reg} = \frac{1}{n}\sum_{i=1}^{n}\left(\left\|\mathbf{p}_S-\mathbf{p}_{gt}\right\|_i^2 + \frac{\left\|\mathbf{p}_S-\mathbf{p}_T\right\|_i}{\sigma} + \log\sigma\right) \tag{4.7}$$

The constant α in Equation (4.2) is removed in Equation (4.7) as its function to balance the hard and soft loss can be substituted with σ. We can view Equation (4.7) as a way for S to learn a suitable coefficient (via σ) to down-weigh unreliable predictions of teacher network T. Besides Laplacian distribution, other parametric distributions like Gaussian could be used instead.

5. **Attentive imitation loss (AIL)**. The main objective of modeling the uncertainty in the imitation loss is that we could then adaptively down-weigh the imitation loss when a particular T prediction is not reliable. However, modeling T's predictions with a parametric distribution may not accurately reflect the error distribution of T's prediction. Hence, instead of relying on S to learn a quantity σ to down-weigh unreliable T predictions, we can use the empirical error of T's prediction with respect to ground truth (which is the teacher loss)

to do the job. Then, the objective function becomes

$$\mathcal{L}_{reg} = \frac{1}{n}\sum_{i=1}^{n} \alpha \left\|\mathbf{p}_S - \mathbf{p}_{gt}\right\|_i^2 + (1-\alpha)\Phi_i \left\|\mathbf{p}_S - \mathbf{p}_T\right\|_i^2 \tag{4.8}$$

$$\Phi_i = \left(1 - \frac{\left\|\mathbf{p}_T - \mathbf{p}_{gt}\right\|_i^2}{\eta}\right) \tag{4.9}$$

$$\eta = \max\left(e_T\right) - \min\left(e_T\right) \tag{4.10}$$

$$e_T = \{\left\|\mathbf{p}_T - \mathbf{p}_{gt}\right\|_j^2 : j = 1, ..., N\} \tag{4.11}$$

where Φ_i is the normalized teacher loss for each i sample, e_T is a set of teacher loss from entire training data, and η is a normalization parameter that we can retrieve from subtracting the maximum and the minimum of e_T. Note that $\|.\|_i$ and $\|.\|_j$ are not p-norm symbol. Instead we use i and j in $\|.\|_i$ and $\|.\|_j$ as index to differentiate which loss is computed from the batch samples ($i = 1, ..., n$) and which loss is calculated from the entire training data ($j = 1, ..., N$).

Figure 4.1 shows how each component in Equations (4.8)-(4.11) blends together. Note that we still keep α to govern the relationship between student and imitation loss. In this case, Φ_i's role is to place different relative importance, hence it is called *attentive*, for each component in the imitation loss as seen in the weighted sum operation. Notice that Equation (4.8) can be rearranged into the following equation

$$\mathcal{L}_{reg} = \frac{\alpha}{n}\sum_{i=1}^{n} \left\|\mathbf{p}_S - \mathbf{p}_{gt}\right\|_i^2 + \frac{1-\alpha}{n}\sum_{i=1}^{n} \Phi_i \left\|\mathbf{p}_S - \mathbf{p}_T\right\|_i^2. \tag{4.12}$$

As Φ_i is computed differently for each image sample and is intended to down-weigh unreliable T prediction, we could also say that by multiplying the imitation loss with Φ_i, we rely more on the example data which T is good at predicting in the process of knowledge transfer between T and S.

4.4.2 Learning Intermediate Representations

In the previous section, we have set the objective function for the main KD task by blending the teacher, student, and imitation loss. Another important aspect in KD's

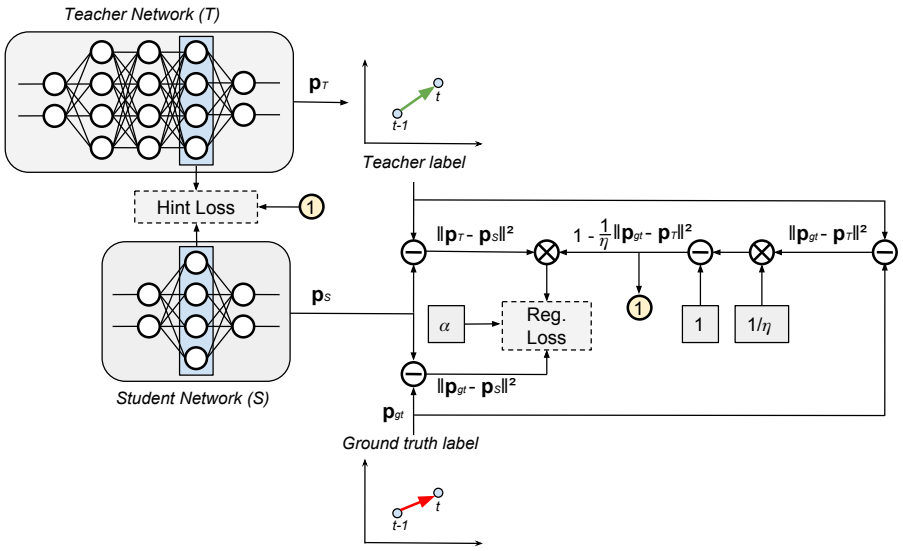


Figure 4.1: Our KD approach applied to regression problem. Note that in regression, we are unable to use the "dark knowledge" provided by soft teacher labels.

transfer process is Hint Training (HT). HT is the process of training the intermediate representation of S such that it could mimic the latent representation of T. It was designed as an extension of the original KD [62] and formulated by [132] to transfer the knowledge of T to S with deeper but thinner architecture. Even if it is devised to help training S with deeper layers than T, we would argue that it is also an important step for training a regressor network with a shallow architecture. HT could act as another way to regularize S such that it could better mimic the generalization capability of T [132].

In Hint Training, a *hint* is defined as a layer in T that is used to train a *guided* layer in S. Let $\mathbf{W}_{guided}$ and $\mathbf{W}_{hint}$ be the parameters of S and T up to their guided and hint layers respectively. With the standard HT formulation, we can train S up to the guided layer by minimizing the following objective function

$$\mathcal{L}_{hint} = \frac{1}{n}\sum_{i=1}^{n} \|\Psi_T(\mathbf{I}; \mathbf{W}_{hint}) - \Psi_S(\mathbf{I}; \mathbf{W}_{guided})\|_i^2 \tag{4.13}$$

where Ψ_T and Ψ_S are T's and S's deep neural functions up to their respective hint or guided layers. The drawback with this formulation is that it does not take into account the fact that T is not a perfect function estimator and can give incorrect

guidance to S. While in Section 4.4.1 we describe how to tackle this issue through down-weighing unreliable T predictions by multiplying with the normalized teacher loss, we argue that this step is also required for HT. Then, we propose a modification of HT termed *Attentive Hint Training* (AHT) as follows:

$$\mathcal{L}_{hint} = \frac{1}{n}\sum_{i=1}^{n} \Phi_i \left\| \Psi_T(\mathbf{I}; \mathbf{W}_{hint}) - \Psi_S(\mathbf{I}; \mathbf{W}_{guided}) \right\|_i^2 \tag{4.14}$$

where Φ_i is the normalized teacher loss as seen in Equation (4.9). While Equations (4.8) and (4.14) can be trained jointly, we found out that training separately yields superior performance especially in absolute pose error. Then, the knowledge transfer between T and S becomes a two stages optimization process. The first stage trains S up to the guided layer with Equation (4.14) as the objective. The second stage trains the remaining layer of S (from guided until the last layer) with Equation (4.8) as the objective.

4.4.3 Compressing Distilled Networks with Low-Rank Factorization

Since S has similar characteristics with T with the exception of the depth of the network, S could be further compressed with other compression techniques. We propose to further reduce the parameters of S, especially the CNN weights, with a factorization technique. We employed Singular Value Decomposition (SVD) to factorize the CNN weights of S as seen in [131] and [71], termed Low-Rank Separable Filters (LRSF). This decomposition yields a deeper network since the original filter at layer k becomes a horizontal and vertical filter as follows

$$g_k * \mathbf{I} = v_k * (h_k * \mathbf{I}) \tag{4.15}$$

where $*$ is convolutional operation, and $g_k \in \mathbb{R}^{f \times f}$, $v_k \in \mathbb{R}^{f \times 1}$, $h_k \in \mathbb{R}^{1 \times f}$. Using this filter factorization, the convolutional operation can be performed in $\mathcal{O}(2fhw)$ instead of $\mathcal{O}(f^2hw)$ [71].

Let $\mathbf{W}^k_{S_{cnn}} \in \mathbb{R}^{c\times f\times f\times d}$ be the weight tensor of CNN in S for layer k, where c, f, and d are output channel, filter, and input channel dimension respectively. To decompose $\mathbf{W}^k_{S_{cnn}}$, we reshape it into a matrix (2-D tensor) $\hat{\mathbf{W}}^k_{S_{cnn}} \in \mathbb{R}^{(cf)\times(fd)}$ where (cf) and (fd) are the row and column size respectively. Then, by applying SVD, we get $\hat{\mathbf{W}}^k_{S_{cnn}} = \mathbf{USV}^T$ where $\mathbf{S} \in \mathbb{R}^{(cf)\times(fd)}$ is diagonal matrix containing singular values of $\hat{\mathbf{W}}^k_{S_{cnn}}$, and $\mathbf{U} \in \mathbb{R}^{(cf)\times(cf)}$ and $\mathbf{V} \in \mathbb{R}^{(fd)\times(fd)}$ are their respective eigenvectors. To obtain parameter efficiency, we select the m highest singular values to approximate $\hat{\mathbf{W}}^k_{S_{cnn}} \approx \tilde{\mathbf{U}}\tilde{\mathbf{S}}\tilde{\mathbf{V}}^T$, in which $\tilde{\mathbf{U}} \in \mathbb{R}^{(cf)\times(m)}$, $\tilde{\mathbf{S}} \in \mathbb{R}^{(m)\times(m)}$, and $\tilde{\mathbf{V}} \in \mathbb{R}^{(m)\times(fd)}$. Now, we can construct two weight tensors from $\mathbf{W}^k_{S_{cnn}}$ as

$$\mathbf{v}_k = \tilde{\mathbf{U}}\tilde{\mathbf{S}}^{\frac{1}{2}}, \mathbf{h}_k = \tilde{\mathbf{S}}^{\frac{1}{2}}\tilde{\mathbf{V}}^T. \tag{4.16}$$

After this factorization process, we usually need to fine-tune the decomposed filters to recover from a loss in accuracy.

4.5 Implementation Details

4.5.1 Camera Pose Regression with DNNs

DNN-based VO learns the camera ego-motion directly from raw image sequences by training the network in an end-to-end manner. Let $\mathbf{I}_{t-1,t} \in \mathbb{R}^{2\times(w\times h\times c)}$ be two concatenated images at times $t-1$ and t, where w, h, and c are the image width, height, and number of channels respectively. DNNs essentially learn a mapping function to regress the 6-DoF camera poses $\{(\mathbb{R}^{2\times(w\times h\times c)})_{1:N}\} \rightarrow \{(\mathbb{R}^6)_{1:N}\}$, where N are the total number of image pairs. In the supervised case, learning 6-DoF camera poses can be achieved by minimizing the discrepancy between the predicted poses $\mathbf{p}_{pr} \in \mathbb{R}^6$ and the ground truth poses $\mathbf{p}_{gt} \in \mathbb{R}^6$ as follows

$$\mathcal{L}_{reg} = \frac{1}{n}\sum_{i=1}^{n} \left\|\mathbf{p}_{pr} - \mathbf{p}_{gt}\right\|_i^2, \tag{4.17}$$

given n number of sample images. However, since translation and rotation have different constraints, we usually decompose $\mathcal{L}_{reg}$ into

$$\mathcal{L}_{reg} = \frac{1}{n}\sum_{i=1}^{n}\left(\beta \left\|\mathbf{t}_{pr} - \mathbf{t}_{gt}\right\|_i^2 + (1-\beta)\left\|\mathbf{r}_{pr} - \mathbf{r}_{gt}\right\|_i^2\right) \tag{4.18}$$

where $\mathbf{t} \in \mathbb{R}^3$ and $\mathbf{r} \in \mathbb{R}^3$ are the translation and rotation components in x, y, and z axes. $\beta \in \mathbb{R}$ is used to balance $\mathbf{t}$ and $\mathbf{r}$. Here the rotation part is represented as an Euler angle. Another representation such as quaternion or rotation matrix can be used as well [161].

In order to apply our distillation approach to DNN-based VO, we decompose each translation and rotation component in Equation (4.18) into Equation (4.8) such that it becomes the following formula

$$\begin{aligned}\mathcal{L}_{reg} = \frac{1}{n}\sum_{i=1}^{n} \beta(\underbrace{\alpha_t \left\|\mathbf{t}_S - \mathbf{t}_{gt}\right\|_i^2 + (1-\alpha_t)\Phi_i^t \left\|\mathbf{t}_S - \mathbf{t}_T\right\|_i^2}_{\text{translation component}}) + \\ (1-\beta)(\underbrace{\alpha_r \left\|\mathbf{r}_S - \mathbf{r}_{gt}\right\|_i^2 + (1-\alpha_r)\Phi_i^r \left\|\mathbf{r}_S - \mathbf{r}_T\right\|_i^2}_{\text{rotation component}}).\end{aligned} \tag{4.19}$$

Note that with Equation (4.19), we have different α and Φ_i for each translation (i.e. α_t , Φ_i^t) and rotation (i.e. α_r , Φ_i^r) components. For the hint training, we also require to decompose AHT formulation in Equation (4.14) into translation and rotation representation as in Equation (4.19). As the teacher loss distribution is also different for translation and rotation (as seen in Figure 4.2), η in Equation (4.9) is computed differently for each of them.

4.5.2 Network Architecture

We employ ESP-VO [162] for the teacher network T in which the architecture is depicted in Figure 4.3 (left part). It consists of two main parts, namely the feature extractor network and the pose regressor network. The feature extractor is composed from a series of Convolutional Neural Networks (CNNs) to extract salient features for VO estimation. Since VO estimates the camera pose between consecutive frames, optical-flow like feature extractor network (FlowNet [34]) is used to initialize the

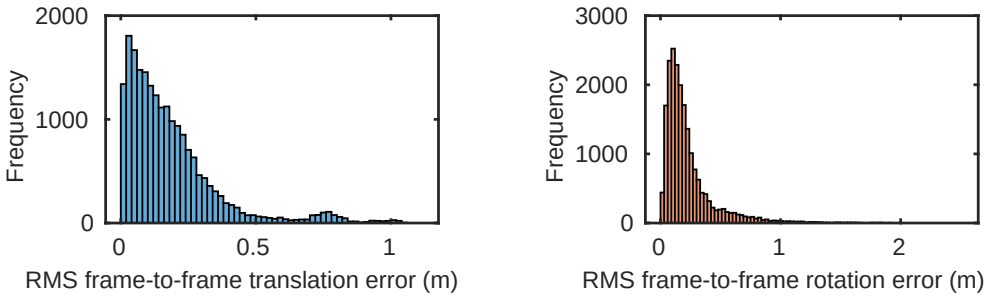


Figure 4.2: Empirical error distribution of the teacher network for translation and rotation on KITTI dataset Seq 00-08.

CNNs. The pose regressor consists of Long-Short Term Memory (LSTM) Recurrent Neural Networks (RNNs) and Fully Connected (FC) Layers to regress 6-DoF camera poses. The LSTM is utilized to learn long-term motion dependencies among image frames [161].

Figure 4.3 (right part) depicts S with 92.95% distillation rate (d_{rate}). The main building blocks of S are essentially the same as T except we remove a number of layers from T to construct a smaller network. To specify the structure of S, in general, we can remove the layers from T which contribute the most to the number of weights, but S should still consist of a feature extractor (CNN) and a regressor (LSTM/FC). In the feature extractor, the largest number of weights usually corresponds to the few last layers of CNNs, while in the regressor part it corresponds to the LSTM layers. Thus, for $d_{rate} = 92.95\%$, we remove the last five layers of the CNN and the two RNN-LSTM layers. However, we still initialize the CNN with the FlowNet's weight as in ESP-VO. To compensate for the loss of removing the CNN and LSTM layers, we add 1 FC layer in the regressor part for $d_{rate} < 75\%$ and 2 FC layers for $d_{rate} > 75\%$. This corresponds to fewer than 1% additional parameters.

4.5.3 Training Details

As stated in Section 4.4.2, we employ two stages of optimization. The first stage is training the intermediate representation of S through AHT. As seen in Figure 4.3, we

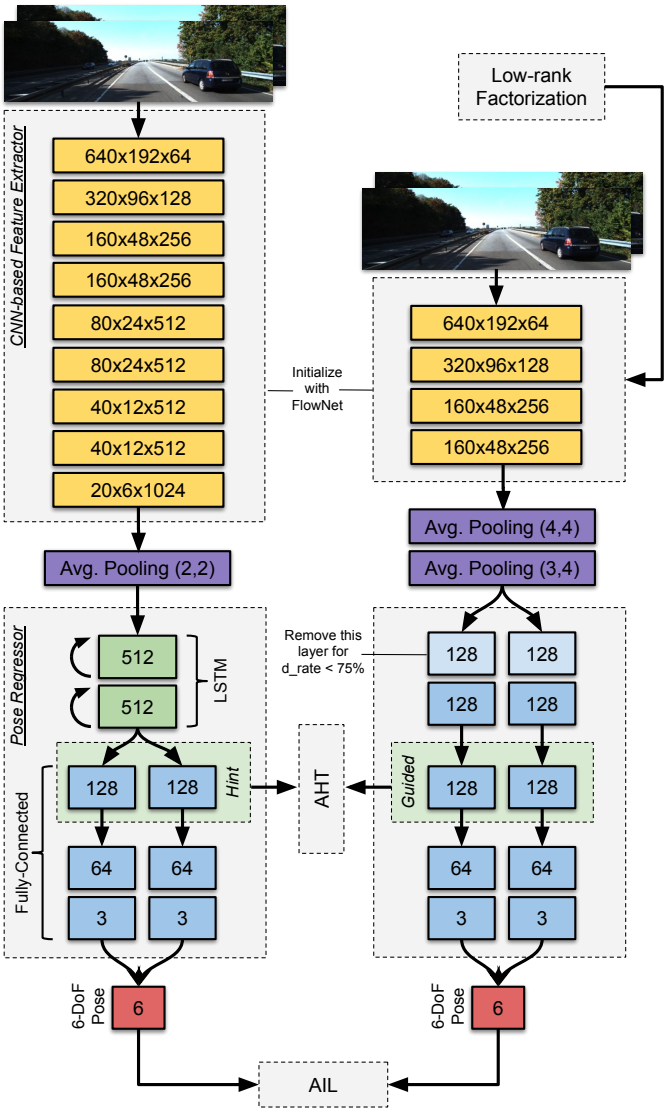


Figure 4.3: Details of the network architecture for teacher network (left) and student network with 92.95% distillation rate (right). Note that Low-Rank Separable Filters (LRSF) are only applied in the convolutional networks.

select the 1st FC layer of T as a hint and the 3rd FC layer of S (or the 2nd FC layer for $d_{rate} < 75\%$) as the guided layer. We used the FC layer of T as a hint not only to provide easier guidance for training S, since both FC layers in T and S have the same dimensions, but also to transfer the ability of T to learn the long-term motion dynamics of camera poses as the FC layer of T is positioned after the RNN-LSTM layers. In the second stage, we freeze weights of S trained from the first stage and train the remaining layers of S using Equation (4.8) as the objective. After transferring the

knowledge from T to S, we can optionally compress S further with LRSF as described in Section 4.4.3. For this training stage, we first decompose S using SVD. Then, we fine tune the decomposed filter while freezing the pose regressor. We also experiment with reversing the order to see whether this combination of distillation and LRSF is commutative.

4.6 Experimental Results

4.6.1 Experiment Environments

We implemented T and S in Keras. We employed NVIDIA TITAN V GPU for training and NVIDIA Jetson TX2 for testing. The training for each stage goes up to 30 epochs. For both training stages, we utilize Adam Optimizer with $1e-4$ learning rate. We also applied Dropout [144] with 0.25 dropout rate for regularizing the network. For the data, we used KITTI [45] and Malaga odometry dataset [13]. We utilized KITTI Seq 00-08 for training and Seq 09-10 for testing. Before training, we reduced the KITTI image dimension to 192×640. We only use Malaga dataset for testing the model that has been trained on KITTI. For this purpose, we cropped the Malaga images to the KITTI image size. Since there is no ground truth in Malaga dataset, we perform qualitative evaluation against GPS data. For data augmentation, we randomly generated sequences with different starting and ending points.

4.6.2 Metrics

In this work, we want to measure the trade-off between accuracy and parameter reduction. In VO, accuracy can be measured by several metrics. We use Root Mean Square (RMS) Relative Pose Error (RPE) for translation ($\mathbf{t}$) and rotation ($\mathbf{r}$) and RMS Absolute Trajectory Error (ATE) as they has been widely used in many VO or SLAM benchmarks [146]. For parameter reduction, we measure the percentage (%) of S's parameters with respect to T's parameters. We also measure the associated computation time (ms) and the model size (MB) for each reduction rate.

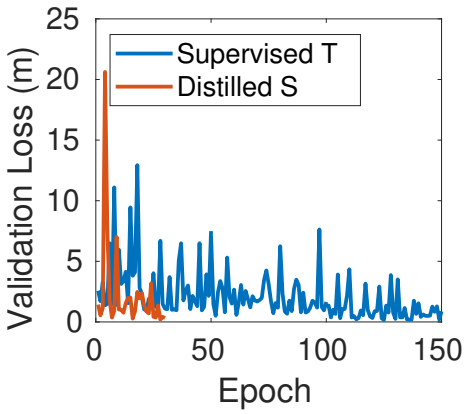


Figure 4.4: Training convergence between supervised T and distilled S (for the second stage).

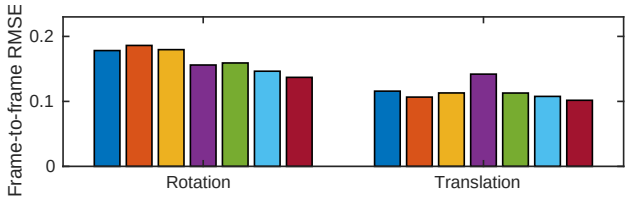
4.6.3 Training Time and Convergence

The training time for our approach takes around 3.5 hours for each stage in TITAN V (2x faster than training T). The training time is relatively the same for different d_{rate} as we initialize S with FlowNet [34] and only train the remaining layers using our proposed approach. Figure 4.4 shows the validation loss during training between supervised T and distilled S for the second stage of training. Note that we can only show the validation loss for the second stage of training as the first stage of training is intended to learn the intermediate representation which is not comparable with the validation loss for the final task. Note also that T was trained for a maximum of 200 epochs or was stopped earlier if validation loss showed no improvement, while S was only trained for 30 epochs for each stage. Figure 4.4 shows that S converges faster than T as the knowledge is transferred effectively from T using attentive knowledge transfer.

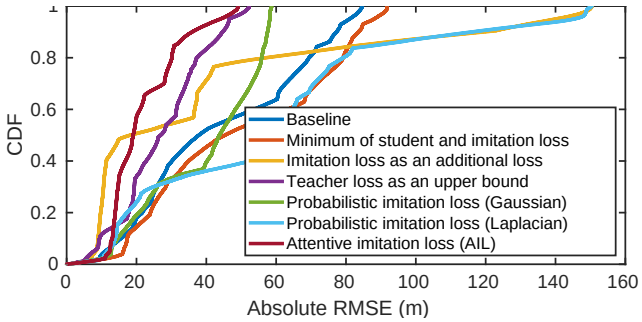
4.6.4 Sensitivity Analysis

The Impact of Different Methods for Blending Teacher, Student, and Imitation Loss. In this experiment, we want to understand the impact of different approaches to blending teacher, student, and imitation loss as described in Section

4.4.1. We used S with $d_{rate} = 72.78\%$ constructed from removing the last 3 CNNs and replacing 2 LSTMs with 1 FC layer. In order to get a fair comparison without having bias from the AHT process, we trained S with standard HT approach in the first stage. Then, we trained the remaining layer(s) with all formulations described in Section 4.4.1 in the second stage. For additional comparison, we add a baseline approach, in which we only minimize the student loss.



(a) Frame-to-frame relative RMSE



(b) CDF of Absolute Trajectory Errors

Figure 4.5: The impact of different ways in blending teacher, student, and imitation loss to (a) RPE and (b) ATE. Same legend is used for both graphs.

Figure 4.5 (a) and (b) depicts the RPE and the CDF of ATE of different methods in blending the losses. It can be seen that AIL has the best accuracy in both RPE and ATE. This indicates that distilling knowledge from T to S only when we trust T does not reduce the quality of knowledge transfer, but instead improves the generalization capability of S. The two approaches (minimum of student and imitation loss; imitation loss as additional loss) that rely on the assumption that T's prediction is always accurate have inferior performance even if compared to the baseline. PIL, either using Laplacian or Gaussian, yields good accuracy in RPE, but lacks robustness since they have larger overall drift (as seen in Figure 4.5 (b)). This is probably

Table 4.1: The impact of using Attentive Imitation Loss (AIL) and Attentive Hint Training (AHT) algorithm

	Network Architecture [a]	Distillation Rate (d_{rate})	HT	Final Objective	Reconstruction Error[b]	ATE (m)
1	6 CNN - 2 FC	72.88%	-	student loss	0.6242	80.842
2	6 CNN - 2 FC	72.88%	HT	student loss	0.0252	65.251
3	6 CNN - 2 FC	72.88%	HT	AIL	0.0252	36.459
4	6 CNN - 2 FC	72.88%	AHT	student loss	**0.0166**	52.320
5	6 CNN - 2 FC	72.88%	AHT	AIL	**0.0166**	**32.259**
6	5 CNN - 3 FC	79.69%	-	student loss	0.1341	68.350
7	5 CNN - 3 FC	79.69%	HT	student loss	0.0177	53.661
8	5 CNN - 3 FC	79.69%	HT	AIL	0.0177	29.751
9	5 CNN - 3 FC	79.69%	AHT	student loss	**0.0168**	37.645
10	5 CNN - 3 FC	79.69%	AHT	AIL	**0.0168**	**25.857**

[a] Total FC layers until intermediate layer used for HT and AHT.
[b] Reconstruction error of S's output intermediate representation with respect to T's output intermediate representation.

due to the failure of the parametric distribution function to model the teacher error distribution accurately. The upper bound objective has good balance between RPE and ATE but the performance is inferior to AIL.

The Impact of Attentive Hint Training. As we want to inspect the effect of the proposed AHT approach, we trained the model with 3 different procedures: without HT, with HT, and with AHT. We also alternate between using the student loss and AIL to see the effect of applying the attentive transfer mechanism in both intermediate (as AHT) and final layer (as AIL), or only in one of them. We used the same model architecture as the previous ablation to conduct this experiment. We compare the RMS Reconstruction Error of S's output latent representation with respect to T's representation and ATE with respect to ground truth.

Table 4.1 lists the results of this study which clearly show that as soon as HT is applied, S's reconstruction error with respect to T reduces dramatically (see rows [1, 2] or [6, 7]). This shows that without having guidance in the intermediate layer, it is very difficult for S to imitate the generalization capability of T. AHT then further reduces the reconstruction error of HT by giving different relative importance to T's representation and placing more emphasis on representations that produce accurate

T predictions. Figure 4.6 visualizes the output latent representation for different training procedures. It can be seen that AHT's output representation is very close to that of T. Slight differences with T's representation are due to different relative importance placed on T's predictions. However, even if AHT does not try to blindly imitate T's representation, the average reconstruction error is still lower than the HT approach which attempts to perfectly imitate T's representation (see Table 4.1 rows [2, 4] or [7, 9]).

The last column of Table 4.1 shows ATE for different combinations of applying attentive knowledge transfer. As it can be seen in rows [2, 4] (or [7, 9]) that applying attentive loss in the intermediate layer (AHT) can significantly reduce the ATE of S. However, the reduction rate is not as large as when applying it in the final task (AIL) (see Table 4.1 rows [2, 3] or [7, 8]) as it can reduce the ATE up to 1.8× smaller. This is sensible because the accuracy of a DNN model depends on the output from the final task. Better guidance in the final layer (main task) can yield stronger performance than better guidance in the intermediate layer. Finally, applying attentive loss in both intermediate (AHT) and final layers (AIL) consistently gives the best result for 6 and 5 CNNs architecture (see Table 4.1 row 5 and 10).

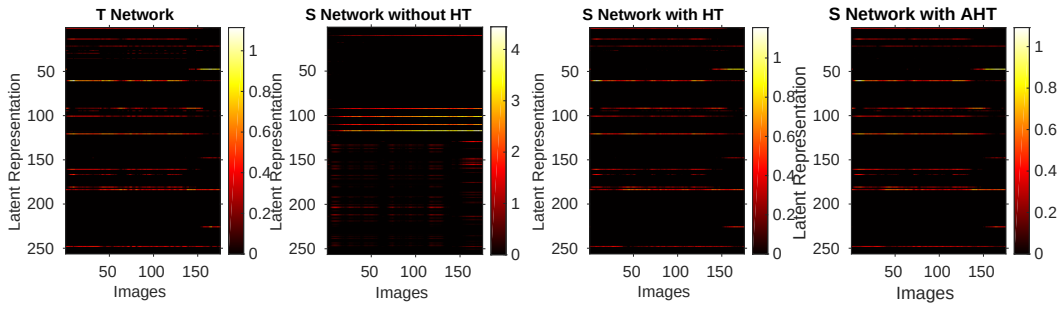


Figure 4.6: The difference of latent feature representation between T and S, trained without HT, with HT, and with AHT.

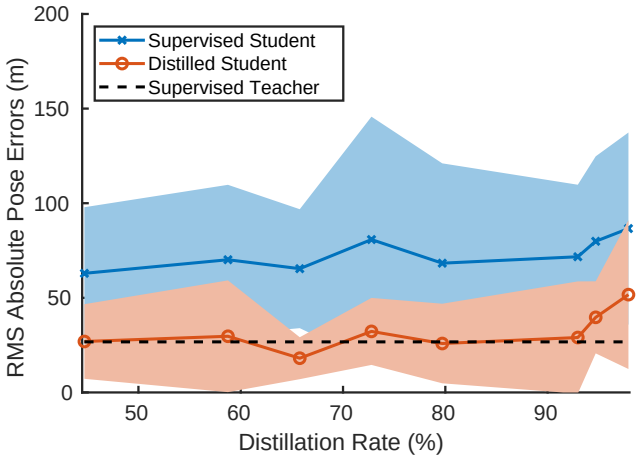


Figure 4.7: RMS absolute pose errors between Supervised and Distilled Student for different d_{rate}. Note that we trained the supervised student from scratch.

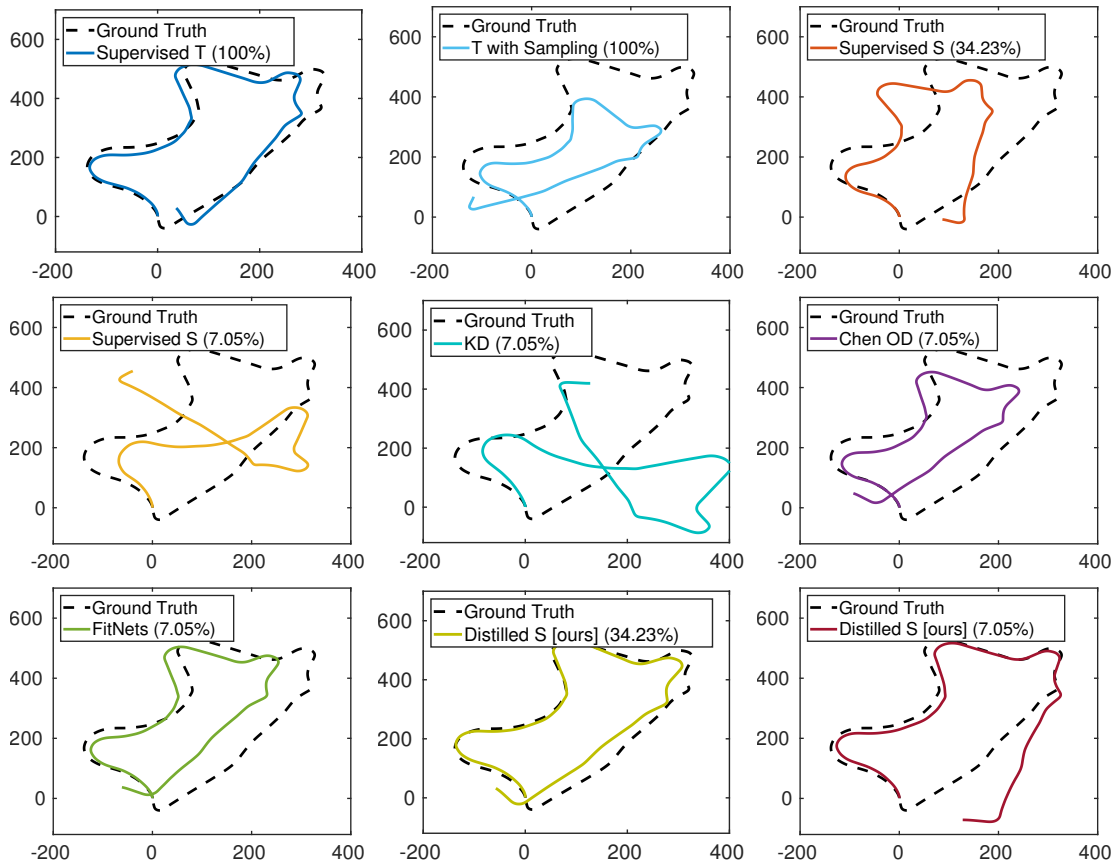


Figure 4.8: Trajectory prediction from T and S trained with various distillation approaches in KITTI Seq 09. The number in the bracket indicates the percentage of S parameters with respect to T.

Table 4.2: Trade-off between the number of parameters, model size, computation time, and accuracy (ATE)

Network (Weights %)	Parameters (millions)	Size (MB)	Execution Time (ms)	ATE (m)
T (100%)	33.64	286.9	87	26.74
S (55.28%)	18.59	74.6	82	26.92
S (41.25%)	13.88	55.7	71	29.69
S (34.23%)	11.52	46.3	62	18.09
S (27.22%)	9.16	36.8	58	32.26
S (20.30%)	6.83	27.5	47	25.86
S (7.05%)	2.37	7.3	41	29.03
S+LRSF (2.61%)	0.88	4.8	45	21.80

4.6.5 Trade-off between Accuracy, Model Size, and Execution Time

In this experiment, we want to understand the trade-off between the model size, execution time, and accuracy for different distillation/compression rate values (d_{rate}). We compare our distillation with the student network S trained from scratch to understand how much improvement our method can brings. It can be seen in Figure 4.7 that our proposed distillation approach always improve the performance of student network S by large margin for every (d_{rate}). Figure 4.7 also shows that our proposed distillation approach can keep S very close to T up to $d_{rate} = 92.95\%$. It can even achieve better performance than the teacher for $d_{rate} = 65.77\%$ and 79.69% as T might be over-parameterized (see also the output trajectory in Figure 4.8). For $d_{rate} > 92.95\%$, the performance starts to degrade more rapidly as it becomes too difficult to transfer the knowledge from T to S without other constraints. It can also be seen that if S is trained directly to fit the ground truth with hard loss (supervised student), it shows very poor performance.

Table 4.2 shows the comparison between T and S in terms of number of parameters, model size, and computation time. As we can see, with $d_{rate} = 92.95\%$ we can reduce the model size from 286.9MB to 7.3MB (2.5%). Removing 2 LSTMs, which are responsible for 44.72% of T's parameters can already reduce T's model size to

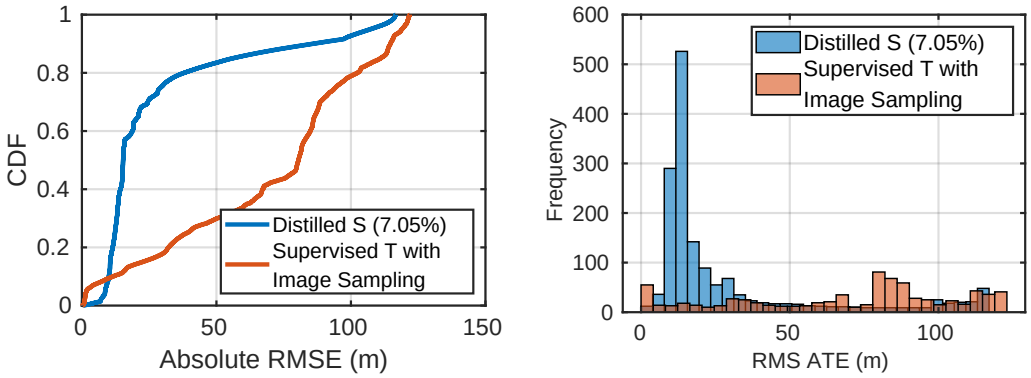


Figure 4.9: Distribution and histogram of ATE between distilled S and supervised T with image sampling.

78MB (27%) but it has less impact in the computation time as the LSTM has been implemented efficiently for NVIDIA cuDNN. With $d_{rate} = 92.95\%$, we reduce the computation time from 87ms to 41ms (2.12×), effectively doubling the frame rate. This has significant practical implication. If we re-train T given subsampled images such that the frame rate is similar to S with $d_{rate} = 92.95\%$, T's prediction accuracy will degrade 160% (see the trajectory in Figure 4.8). This is probably due to the difficulty of estimating accurate optical flow representation in large stereo baseline. Meanwhile with the same computation budget, the distilled S yields stronger performance than the subsampled T with only 8.63% accuracy drop with respect to supervised T as seen in Figure 4.9.

4.6.6 Comparison with Other KD Approaches

Since there is no specific KD for pose regression, to compare our proposed KD with other related works, we used some well known KD approaches for classification and object detection. However, we modify their objective function to fit our regression problem. We used three baselines for this experiment: KD [62], FitNets [132], and Chen's Object Detection (OD) model [22]. For KD [62], we trained the network with standard training procedure (without HT or AHT) and replaced the objective function with Equation (4.2). For FitNets [132], we used HT approach for the first

Table 4.3: Comparison with other distillation approaches for $d_{rate} = 92.95\%$

Method	RMS RPE (**t**)	RMS RPE (**r**)	RMS ATE
Supervised T	0.1197	0.2377	26.7386
Supervised S	0.1367	0.1627	71.7517
KD [62]	0.1875	0.1439	165.2182
Chen's OD [22]	0.1197	0.1416	46.2320
FitNets [132]	0.1450	0.1409	31.9624
Ours	**0.1053**	**0.1406**	**29.0294**

Table 4.4: Comparison with other distillation approaches for $d_{rate} = 65.77\%$

Method	RMS RPE (**t**)	RMS RPE (**r**)	RMS ATE
Supervised T	0.1197	0.2377	26.7386
Supervised S	0.1499	0.1187	65.4179
KD [62]	0.2979	0.1468	120.4039
Chen's OD [22]	0.1567	0.1560	40.6333
FitNets [132]	0.1031	0.1434	28.5408
Ours	**0.1023**	**0.1262**	**18.0915**

stage of training and utilize Equation (4.2) as the objective in the second stage. For Chen's OD Equation [22], we also used standard HT for the first stage and employ Equation (4.3) as the objective in the second stage. For all competing models, we used two compression rates for the comparison, namely $d_{rate} = 65.77\%$ which represents condition when the networks are expected to maximally perform (it can be seen from Figure 4.7 that our proposed approach reaches the maximum performance with $d_{rate} = 65.77\%$) and $d_{rate} = 92.95\%$ which represents extreme condition with large compression. We also compared using $d_{rate} = 92.95\%$ to show that even in extreme conditions (large compression), our proposed approach still yields reasonable performances for VO, and better than the competing approaches.

Table 4.3 shows the result of this experiment with $d_{rate} = 92.95\%$. It can be seen that our proposed approach have better accuracy for both RPE and ATE. Even if most of the competing approaches have better RPE than the supervised T, it has huge bias in the relative pose prediction such that the integration of these relative poses yields very large ATE. T tackle this bias to some extent by using LSTM layers

which are supposed to learn the long-term motion dynamic of the camera poses [161]. Since S removes the LSTM layers, most approaches fail to recover this knowledge from T but our proposed approach is able to reduce ATE by focusing to learn the good predictions from T. Table 4.4 shows the result when the competing methods are expected to maximally perform ($d_{rate} = 65.77\%$). It can be seen that our proposed approach also yields the best performance both in terms of RPE or ATE although the competing approaches also show improvement compared to the result in Table 4.3. Figure 4.8 shows the comparison of the output trajectory between the competing approaches on KITTI dataset. It can be seen that our distillation output trajectory is closer to T than the competing approaches.

Figure 4.10 depicts the qualitative evaluation in Malaga dataset. Note that we have not trained on this dataset, demonstrating generalization capacity. We can see that our proposed approach yields a closer trajectory to GPS, even when it is compared to T. This signifies that our distillation approach yields good generalization ability even when it is tested on a different dataset. This result also shows that T may overfit the training data as it has many redundant parameters. However, this redundancy seems necessary for the initial stage of training as a DNN requires large degree-of-freedom to find better weights and connections [62]. Meanwhile, directly training S without any supervision from T seems to be very difficult. Our results show that we can have better generalization when distilling large degree-of-freedom T to small degree-of-freedom S if we transfer the knowledge from T to S only when we trust T.

4.6.7 Results on Fusing Distillation and LRSF

In this experiment, we investigated whether we could gain additional parameter reduction without sacrificing accuracy by further compressing S using LRSF. To understand whether this fusion between distillation and LRSF is sensitive to order, we performed two experiments. The first is distillation-LRSF: we trained S with

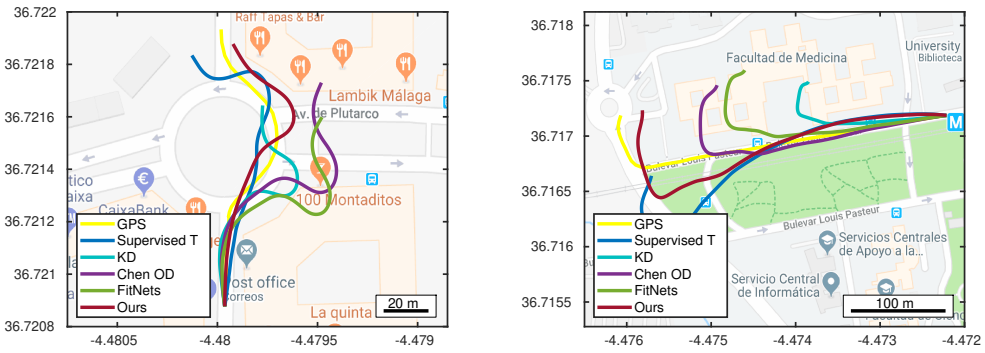


Figure 4.10: Qualitative evaluation in Malaga dataset Seq 04 and Seq 09. All model are only trained on KITTI Seq 00-08.

$d_{rate} = 92.95\%$ using our distillation approach, factorized the distilled S using LRSF ($d_{rate} = 97.39\%$), and finally fine-tune the decomposed filters. The second is LRSF-distillation: we decomposed S's CNN which has been initialized with FlowNet, trained the whole network to fine-tune the CNN, and finally applied distillation to the network.

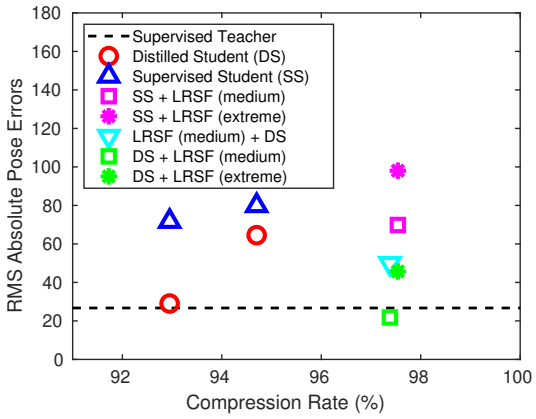


Figure 4.11: RMS absolute pose errors between Supervised and Distilled Student for different d_{rate} (left). Results on fusing distillation and LRSF (right).

Figure 4.11 depicts the result of this study. As we can see, distillation-LRSF clearly has better accuracy than LRSF-distillation. When we apply LRSF after distillation, we already have a good regressor such that we can recover the loss from the decomposition process through fine-tuning the filters while freezing the regressor. On the other hand, when we performed LRSF first, we alter the FlowNet weights significantly without having the ability to recover the weights from a good regressor

which results in suboptimal performance. Since applying LRSF to either supervised S or distilled S improves the performance (up to 18.41% as seen in Figure 4.11), this also indicates that low-rank factorization could improve the generalization of S as the low-rank constraints force the network to extract more effective and generalized features for VO estimation. Figure 4.12 depicts the predicted trajectory of S trained with distillation-LRSF approach which clearly shows the superior performance than T. Despite having more parameter reduction and increased accuracy, applying LRSF to distilled S does not improve the actual computation time (as seen in Table 4.2) as the depth of S's CNN is doubled. If we performed factorization aggressively (labelled as extreme in Figure 4.11 with $d_{rate} = 97.54\%$) we get some improvement in computation time but at the cost of larger accuracy reduction.

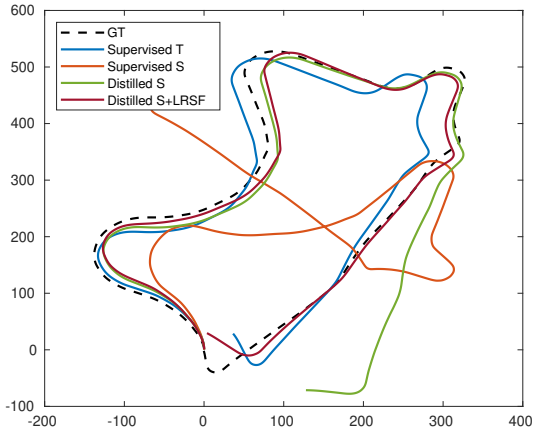


Figure 4.12: Trajectory prediction between T, S, and distilled S.

4.7 Discussion and Limitations

Our experiments show that our approach is an effective approach to compress deep visual odometry model while at the same time maintaining the accuracy. However, we note that our approach might not be the only solution to compress visual odometry model. Other compression techniques such as pruning or quantization can be explored in the future research directions to compress deep pose regressor network. Nonetheless, our distillation approach is an orthogonal (independent) method compared to

pruning and quantization, which means that our method can be easily combined with other compression technique. For example, we can use binarization to compress the deep visual odometry model but the training stages and objective functions follow our attentive KD approach. Note that some other distillation approaches can also perform very well by surpassing the teacher as described in [132]. However, it is demonstrated only for classification problem and it is not necessarily transferable to pose regression problem as we have been described in our experiments in Section 4.6.6.

4.8 Conclusion

In this chapter, we present an approach to distill the knowledge from a deep pose regressor network to a smaller network (92.95% fewer parameters) with a small loss of accuracy. We have shown that distilling a pose regressor network is possible if we emphasize the knowledge transfer only when we trust the teacher. We have also shown that the teacher loss can be used as an effective attentive mechanism to transfer the knowledge between teacher and student. We have displayed that we could compress further the distilled network (up to 97.39% parameter reduction) through the combination of distillation and Low-Rank Separable Filters (LRSF) while at the same time recovering the accuracy loss or even achieving better performance than the teacher. Possible future research directions include combination with other compression techniques (e.g. pruning, quantization, etc.) to boost further the actual computation time. Investigation of whether a distillation approach can also transfer the capability of the teacher to estimate uncertainty is also another interesting research direction.

Chapter 5

Alternative Odometry Modalities in Visually-denied Environment

5.1 Introduction

In Chapters 3 and 4, we have seen that we can improve the accuracy and efficiency of DNN-based VO in a principled way. With this improvement, DNN-based VO can be practically applied in a wide range of applications in robotics and computer vision. However, the applications will still be limited to those with sufficient illumination. DNN-based VO systems will not work if they are deployed in visually-denied environments such as in underground tunnels (as had been demonstrated by [76]) or in environment filled with airborne particulates (e.g. smoke and soot). In this scenario, alternative modalities, e.g. based on thermal imaging cameras, are required to enable accurate odometry estimation. Unlike conventional RGB cameras, thermal cameras are illumination-agnostic since they are not affected by lighting conditions or airborne particulates. With this characteristic, thermal cameras become a promising sensing alternative to the standard RGB cameras.

In this chapter, we will investigate the usage of thermal camera for odometry estimation in visually-denied environment. We also explore a fusion mechanism between thermal and other environment-agnostic sensing modality such as Inertial Measurement Unit (IMU) to improve the robustness of the system. In particular, we investigate a suitable DNN model architecture for accurate odometry estimation using

thermal and inertial data.

The rest of this chapter is organized as follows. In Section 5.2, we introduce the research problem we try to solve in this chapter. Section 5.3 formulates the research questions and contributions. Section 5.4 explains the proposed approach, consisting of the description of the network architecture and the learning mechanism. Section 5.5 provides the experimental results and the comparison with competing approaches. Finally, Section 5.6 concludes this chapter.

5.2 Research Problem

Thermal cameras have been commonly used in visually-denied environments, although their usage is largely limited to perception and inspection [126, 130]. The main hindrance preventing their use in odometry estimation is the lack of features (e.g. edges and textures) in the imaging system. Thermal cameras capture the radiation emitted from objects in the Long-Wave Infrared (LWIR) portion of the spectrum. These raw radiometric data are then converted to a temperature profile represented in a visible format (e.g. grayscale) to ease human interpretation [172]. As the camera captures the environmental temperature rather than the scene appearance and geometry, it is difficult to extract sufficient hand-engineered features to accurately estimate pose. Moreover, even for the same scene, the extracted features are dependent on the temperature gradient. This issue is further compounded by the fact that every thermal camera is plagued with fixed-pattern noise and requires frequent re-calibration during operation through Non-Uniformity Correction (NUC) which periodically freezes the images for about half to one second [4] every 30-150 seconds.

The last decade has witnessed a rapid development in the use of deep learning for automatically extracting salient features by directly learning a non-linear mapping function from abundant data. We believe that, with sufficient training data, a DNN can also learn to infer the 6 Degree-of-Freedom (DoF) camera poses from a sequence of thermal images. However, despite the DNN's ability to model this complexity, as

stated before, thermal images are largely textureless and inherently lack sufficient features for accurate odometry estimation. A novel DNN model that can harness the limited information from thermal data is desirable to enable accurate ego-motion estimation.

5.3 Contributions

Based on the research problem we previously mentioned, we can define our research questions as follows: (1) *What kind of DNN architecture is suitable to exploit thermal data effectively for odometry estimation?* (2) *How do we effectively fuse all available information from the sensing modalities (e.g. thermal, IMU, etc.) for accurate 6-DoF pose calculation?* (3) *How accurate is the devised DNN-based thermal-inertial odometry when it is tested in various scenarios and environments?* In summary, we made the following key contributions to answer those research questions:

- We present a novel deep neural odometry architecture by incorporating a hallucination network which forces the network to not only extract features from thermal images, but to additionally learn to hallucinate visual features similar to the ones extracted from a DNN-based VO, which have been proven to work well. The hallucination network acts as complimentary information provider for the thermal network.

- We present a new application of selective fusion to fuse effectively the input from three feature channels (i.e. thermal, IMU, and hallucinated visual features).

- Putting together thermal, hallucinated visual, and IMU network together, we believe that this is the first end-to-end trainable Deep Thermal-Inertial Odometry (DeepTIO) model.

- We perform extensive experiments and analysis in our self-collected hand-held and mobile robot dataset in benign and smoke-filled environments. We show

that DeepTIO is comparable with state-of-the-art visual-inertial odometry algorithms when it is tested in benign environment and outperforms them when it is tested in environment with low visibility (e.g. darkness, smoke-filled).

Parts of this work have been described in the following paper:

- **M. R. U. Saputra**, Pedro P. B. de Gusmao, C. Xiaoxuan Lu, Y. Almalioglu, S.Rosa, C. Chen, J. Wahlstrom, W. Wang, A. Markham, and N. Trigoni. "DeepTIO: A Deep Thermal-Inertial Odometry with Visual Hallucination". In **IEEE Robotics and Automation Letters (RA-L)**, and presented in **IEEE ICRA 2020**.

5.4 Approach

In this section we describe our proposed DeepTIO model for estimating thermal-inertial odometry. We start this section by describing the overall network architecture including the proposed hallucination network. Then, we will explain how to train this network effectively to generate accurate ego-motion estimation.

5.4.1 Network Architecture

Figure 5.1 illustrates the general architecture of DeepTIO model at inference time. It is composed by a feature encoder, a selective fusion module, and a pose regressor. The feature encoder extracts salient features from each modality. We use a CNN for encoding thermal data and hallucinating visual features from thermal images. To extract features from the IMU data stream we employ a RNN, as RNN works better to model temporal dependencies of time-series data [28]. The feature vectors generated from the IMU, thermal, and hallucination encoder networks are input into the selective fusion module, attentively selecting certain features that are necessary for pose regression. The reweighted features are further fed into the pose regression module to infer 6-DoF relative camera poses. The details of each module are described below.

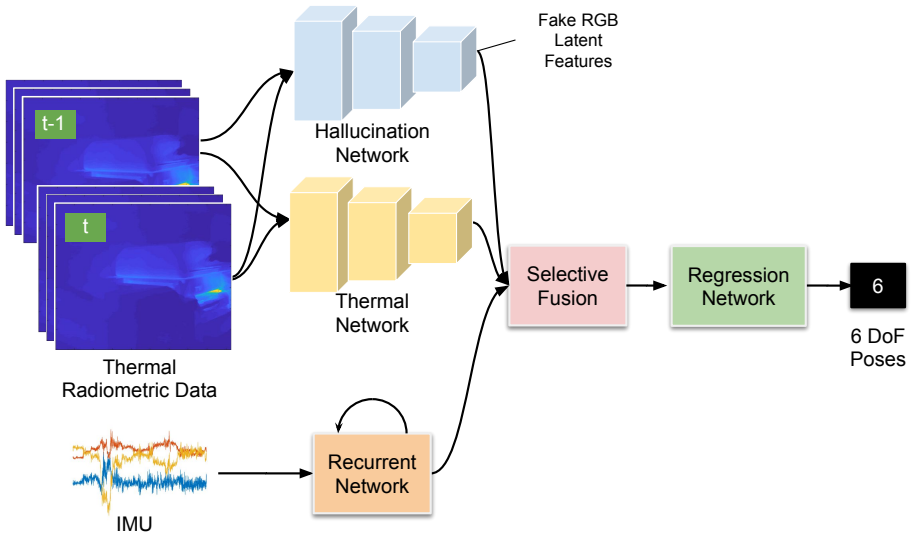


Figure 5.1: The architecture of DeepTIO at test time. DeepTIO not only extracts thermal features but also hallucinates visual features to provide additional information for accurate odometry.

5.4.1.1 Feature Encoder

Given a pair of consecutive thermal images $\mathbf{x}_T \in \mathbb{R}^{2\times(w\times h\times c)}$, the purpose of the thermal encoder network is to extract geometrically meaningful features for movement estimation (e.g. optical flow captured between moving edges). To this end, both thermal encoder Ψ_T and hallucination encoder Ψ_H are implemented and pre-initialized with FlowNetSimple structure [34]. As the observed temperature profile (in grayscale) fluctuates when the camera captures hotter objects, we directly use the 16 bit raw radiometric data to obtain more stable inputs. Since raw radiometric data are only represented by one channel, we duplicate it into three channels for feeding into the FlowNet structure. We use the last output activation from both Ψ_T and Ψ_H as our thermal $\mathbf{a}_T$ and visual hallucinated $\mathbf{a}_H$ features

$$\mathbf{a}_T = \Psi_T(\mathbf{x}_T), \quad \mathbf{a}_H = \Psi_H(\mathbf{x}_T). \tag{5.1}$$

We employ a single LSTM layer with 256 hidden states as IMU encoder Ψ_I. The 6-dimensional inertial data with a sequence of 20 frames $\mathbf{x}_I \in \mathbb{R}^{6\times 20}$ are fed into IMU

encoder Ψ_I to produce IMU features

$$\mathbf{a}_I = \Psi_I(\mathbf{x}_I). \tag{5.2}$$

To balance the number of features, we perform average pooling for $\mathbf{a}_T$ and $\mathbf{a}_H$, such that the final dimensions for all features are $\mathbf{a}_T \in \mathbb{R}^{2048}$, $\mathbf{a}_H \in \mathbb{R}^{2048}$, and $\mathbf{a}_I \in \mathbb{R}^{5120}$.

5.4.1.2 Selective Fusion

In DNN-based VIO, a standard way to fuse feature vectors coming from different modalities is by concatenation. However, a direct fusion of all feature modalities using concatenation results in sub-optimal performance, as not all features are useful and necessary [21]. The situation is even more emphasized by the intrinsic noise distribution of each modality. In our case, thermal data are plagued by fixed-pattern noise, while IMU data are affected by white random noise and sensor bias. On the other hand, the hallucination network might produce erroneous visual features. Moreover, in real applications there is high chance that different modalities, as well as the ground truth poses, will not be tightly synchronized.

To this end, we employ selective fusion [21] to let the network automatically learn the best suitable feature combination given feature inputs. Specifically, a deterministic soft fusion is employed to attentively fuse features from three sources with compensation for possible misalignment between inputs and ground truth. The fusion module will learn to re-weight each feature by conditioning on all channels. The corresponding mask for thermal $\mathbf{m}_T$, hallucination $\mathbf{m}_H$, and inertial feature $\mathbf{m}_I$ are learnt via:

$$\begin{aligned} \mathbf{m}_T &= \sigma(\mathbf{W}_T[\mathbf{a}_T; \mathbf{a}_H; \mathbf{a}_I]) \\ \mathbf{m}_H &= \sigma(\mathbf{W}_H[\mathbf{a}_T; \mathbf{a}_H; \mathbf{a}_I]) \\ \mathbf{m}_I &= \sigma(\mathbf{W}_I[\mathbf{a}_T; \mathbf{a}_H; \mathbf{a}_I]), \end{aligned} \tag{5.3}$$

where $[\mathbf{a}_T; \mathbf{a}_H; \mathbf{a}_I]$ denotes the concatenation of all channels features, $\sigma(x) = 1/(1 + e^{-x})$ is the sigmoid function and $\mathbf{W}_T$, $\mathbf{W}_H$, and $\mathbf{W}_I$ are the learnable weights for each feature modality. These masks are used to weight the relative importance of

the features modalities by multiplying them via element-wise operation $\odot$ with their corresponding masks:

$$\mathbf{a}_{fused} = [\mathbf{a}_T \odot \mathbf{m}_T; \mathbf{a}_H \odot \mathbf{m}_H; \mathbf{a}_I \odot \mathbf{m}_I]. \tag{5.4}$$

Finally, the merged features $\mathbf{a}_{fused}$ are fed to the pose regressor network to estimate 6-DoF poses.

5.4.1.3 Pose Regressor

The pose regressor consists of LSTM layers followed by two parallel Fully Connected (FC) layers that estimate relative translation and rotation respectively. We use an LSTM to model the long-term temporal dependencies of camera ego-motion as seen in [161, 136]. Each LSTM has 512 hidden states and takes the reweighted features $\mathbf{a}_{fused}$ as input. The output latent vectors from the LSTMs are then fed into three parallel FC layers with 128, 64, 3 units respectively. We decouple the FC layers for translation and rotation as it has been shown to work better separately as in [163]. We also use a dropout [144] rate of 0.25 between FC layers to help regularization.

5.4.2 Learning Mechanism

This section introduces the mechanism to train the hallucination network and learn odometry regression.

5.4.2.1 Learning Visual Hallucination

The visual hallucination network Ψ_H is intended to provide additional information along with the thermal encoder Ψ_T. Given original thermal images $\mathbf{x}_T$ as an input, this module produces visual hallucination vectors $\mathbf{a}_H$ that imitate the visual features $\mathbf{a}_V$ from real RGB image input encoded by a visual encoder Ψ_V. To train the hallucination network, we use the output embedding vectors from a deep Visual-Inertial Odometry (VIO) model as the pseudo ground truth. In this case, we employ a modified VINet [27] architecture for the VIO model. The only difference is that we utilize

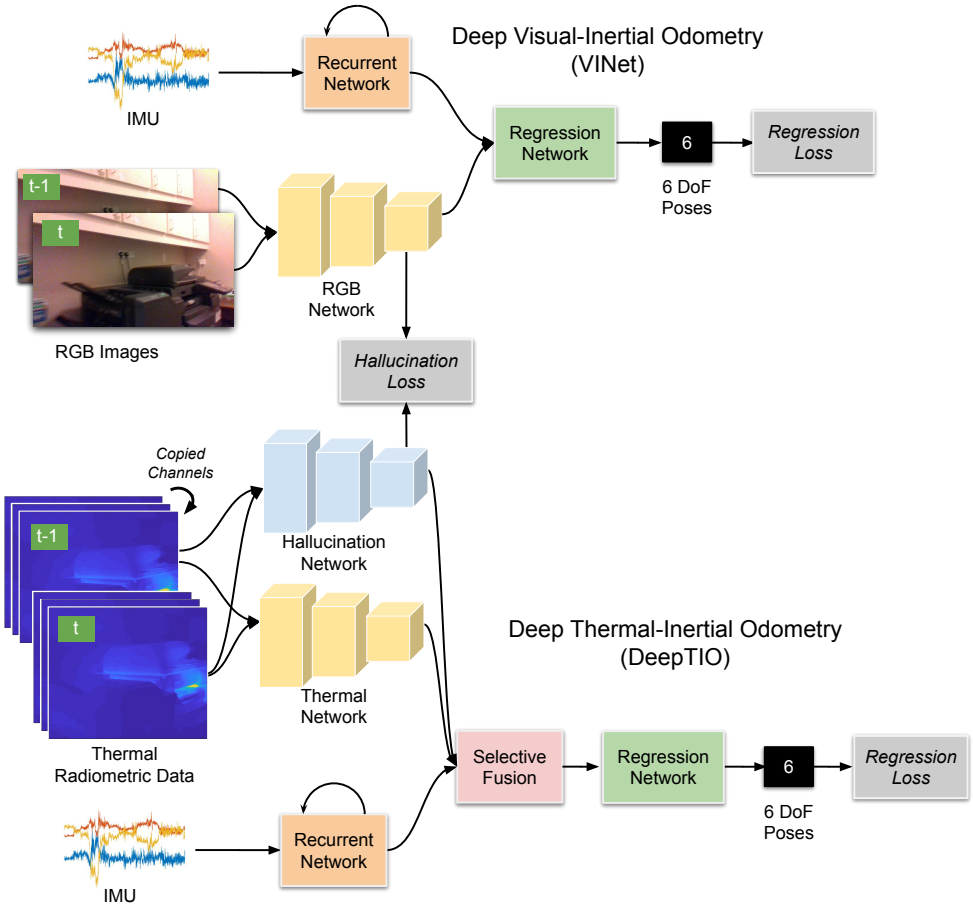


Figure 5.2: The architecture of DeepTIO at training time. Note how RGB images are used to guide the visual hallucination.

FlowNetSimple as the feature extractor instead of FlowNetCorr [34] used in the original VINet. This modification allows hallucination features $\mathbf{a}_H$ and visual features $\mathbf{a}_V$ to have the same dimension, simplifying the training process. After training the VINet model, the weights $\mathbf{W}_V$ in the visual encoder Ψ_V are frozen during the training of the hallucination network, while the hallucination encoder Ψ_H's weights $\mathbf{W}_H$ are trainable.

Figure 5.2 illustrates the architecture of our visual hallucination model in the training process. We train the hallucination network by minimizing the discrepancy ξ between the output activation from Ψ_H and Ψ_V. Standard $\mathcal{L}_2$ norm is generally used for minimizing ξ in benign cases [64, 135]. However, thermal camera requires

periodic NUC calibration, during which time the same image will be output for between half to one second. NUC will force several identical thermal features to be matched with different visual features during network training. This process might produce an erroneous mapping between $\mathbf{a}_H$ and $\mathbf{a}_V$ and contaminate ξ with outliers. Since the $\mathcal{L}_2$ loss is very sensitive to outliers, encountering some during training will impact gradient back-propagation as the outliers will dominate the loss, impacting convergence. To improve robustness against outliers, we instead propose to use the Huber Loss $\mathcal{H}$ [68] to minimize ξ. Then, our hallucination loss $\mathcal{L}_{\text{hallucinate}}$ is formally defined as follows:

$$\begin{aligned} &\mathcal{L}_{\text{hallucinate}} = \frac{1}{n}\sum_{i=1}^{n}\mathcal{H}_i(\xi), \\ &\text{with} \quad \xi = \Psi_H(\mathbf{x}_T;\mathbf{W}_H) - \Psi_V(\mathbf{x}_V;\mathbf{W}_V), \\ &\text{and} \quad \mathcal{H}(\xi) = \begin{cases} \frac{1}{2}\left\|\xi\right\|^2 & \text{for } \|\xi\| \leq \delta, \\ \delta(\|\xi\| - \frac{1}{2}\delta) & \text{otherwise} \end{cases} \end{aligned} \tag{5.5}$$

where δ is a threshold and n is the batch size during training. By using Huber loss, when ξ is larger than δ, it will have a linear effect instead of quadratic, making it less sensitive to outliers. Loss values below δ will still be minimized using quadratic loss to enable fast convergence. During training, we use $\delta = 1.0$.

5.4.2.2 Learning Odometry Regression

We train the network to estimate odometry by minimizing the loss between the predicted pose and the ground truth pose. This task is essentially learning a mapping function from the input to the output $\{(\mathbf{x}_T;\mathbf{x}_I)_{1:N}\} \rightarrow \{(\mathbb{R}^6)_{1:N}\}$ where N is size of the whole training data. The pose regressor network, together with all other networks except the hallucination part, are trained using the following regression loss

$$\mathcal{L}_{regress} = \frac{1}{n}\sum_{i=1}^{n}\mathcal{H}_i(\hat{\mathbf{t}} - \mathbf{t}) + \alpha\mathcal{H}_i(\hat{\mathbf{r}} - \mathbf{r}) \tag{5.6}$$

where $\mathcal{H}$ is the Huber Loss as in Equation (5.5). $[\mathbf{t},\ \mathbf{r}]$ and $[\hat{\mathbf{t}},\ \hat{\mathbf{r}}]$ are a pair of translation and rotation component for the predicted poses and the ground truth

poses respectively. We use Euler angles to represent rotation since it is faster to converge as it is free from constraints unlike other representations (e.g. rotation matrix or quaternion). We also use $\alpha = 0.001$ to balance the loss between translation and rotation.

5.4.2.3 Training Details

The network is trained in two stages. In the first stage we train the hallucination network, while in the second stage we train the remaining networks. Note that, in the second stage, we freeze the hallucination network such that the unstable learning process in the beginning of training the other networks does not alter the learnt hallucination weights that have been trained in the first stage. We use the Adam optimizer with a 0.0001 learning rate to train the hallucination network for 200 epochs. For training the remaining networks in the second stage we employ RMSProp with a 0.001 initial learning rate, dropping by 25% every 25 epochs for a total of 200 epochs. We normalize the input radiometric data by subtracting the mean over the dataset. We randomly cut the training sequence into small batches of consecutive pairs ($n = 8$) to obtain better generalization. We also sub-sample the input such that the frame rate is around 4-5 fps to provide sufficient parallax between consecutive frames. To further fine-tune the network, we alternately freeze and train the selective fusion and the pose regressor.

5.4.3 Relation to Learning with Side Information

Related to our work is the concept of learning with side information. Hoffman et al. [64] introduced this concept by incorporating a depth hallucination network to increase the accuracy of object detection in RGB images. This concept was then adopted in other applications such as learning hand articulations [25] or face recognition [97]. Our work introduces this concept to odometry regression and trains the whole network with the non-trivial Huber loss. We are the first to hallucinate visual features from thermal images for odometry regression.

5.5 Experimental Results

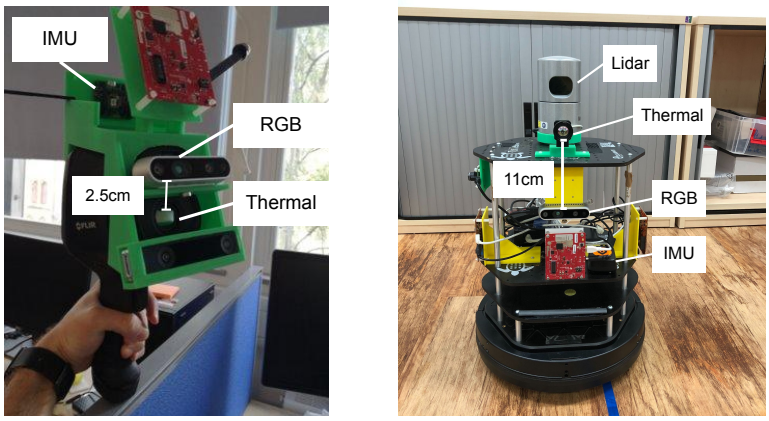
5.5.1 Dataset

We collected our data in two different scenarios. In the first scenario, the sensors are placed in a mobile device such that human can easily carried it as seen in Figure 5.3 (a). In the second scenario, the thermal and IMU sensors are placed in a mobile robot as seen in Figure 5.3 (b). The tools and the dataset for both scenarios are described as follows:

1. **Hand-held data**. The thermal images in hand-held data were collected using FLIR E95 camera at 60 fps with 464×348 image resolution and 24 × 18° field of view. We use a XSens MTI-1 Series for the IMU data. We collected RGB-D data to train the hallucination network using Intel RealSense D435 depth camera at 30 fps and at 848×480 image resolution. We place the thermal and RGB-D camera at a spacing of 2.5 cm such that there is sufficient overlap between the thermal and RGB image. The setup can be seen in Figure 5.3 (a). In total, we collected 19 sequences in five different buildings including a library, open office, apartment, underground storage, and an actual smoke-filled environment in firefighter training facility. We use 13 sequences for training and use the remaining data for testing. As we gathered the data mostly in public spaces, real ground truth poses are not available. Instead, we utilize VINS-Mono (visual-inertial SLAM) [129] for the comparison with the expectation that DeepTIO at least can be as accurate as visual-inertial system.

2. **Mobile robot data**. The mobile robot data was collected using a Turtlebot 2. Thermal images are captured from a Flir Boson 640 thermal camera operating at 60 fps with spatial resolution of 640×512 (95° field of view), while we utilize the same IMU device as with the hand-held data. We equip the robot with a Velodyne HDL-32E LiDAR (captures around 60,000 3D points) and an Intel RealSense Depth Camera with 680 × 480 RGB resolution. The distance

between thermal and RGB camera is 11 cm (see Figure 5.3 (b)) and there is at least 2/3 spatial correspondence. In total, we have 30 sequences collected in three different buildings. We use 23 sequences for training and 7 sequences for testing. For training, we employ inertial-assisted wheel odometry (from the Turtlebot) as the pseudo ground truth. For testing, we use VICON Motion Capture system (1mm accuracy) as the ground truth for the data collected at Oxford Cyber Physical Systems (CPS) Lab while Lidar Gmapping[1] is used for other sequences. For data collected in CPS Lab, we decorated the room with several obstacles, used different lighting condition (sufficient lighting, poor, or dark), and occasionally had a person walking in the camera frame. Figure 5.4 depicts the sample images collected by our mobile robot.



(a) Hand-held device (b) Mobile robot

Figure 5.3: The setup of (a) hand-held device and (b) mobile robot platform for data collection and testing.

5.5.2 Evaluation Metrics

To evaluate the proposed model, we utilize Mean Square (MS) of Relative Pose Error (RPE) and Absolute Trajectory Error (ATE), since they have been widely used for measuring VO or visual SLAM accuracy [146]. As we have different (pseudo) ground

[1]https://openslam-org.github.io/gmapping.html

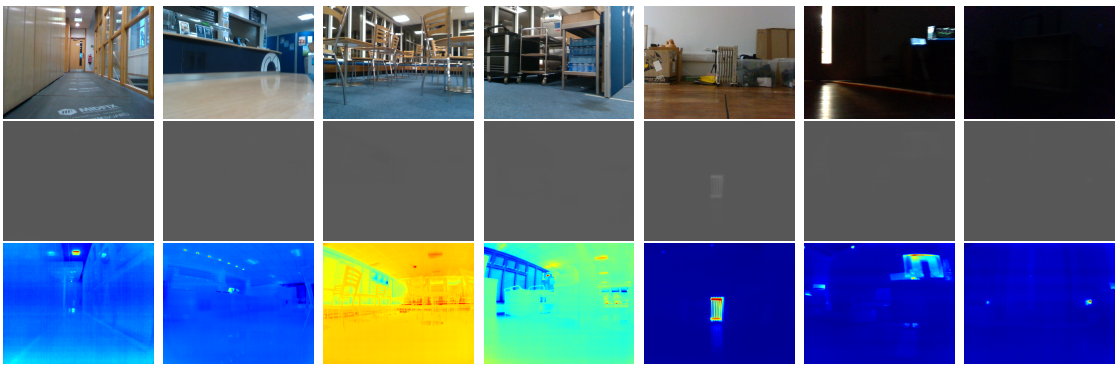


Figure 5.4: Sample images from the dataset. From top to bottom: RGB images, raw radiometric thermal images (14 bit), and normalized thermal images (8 bit).

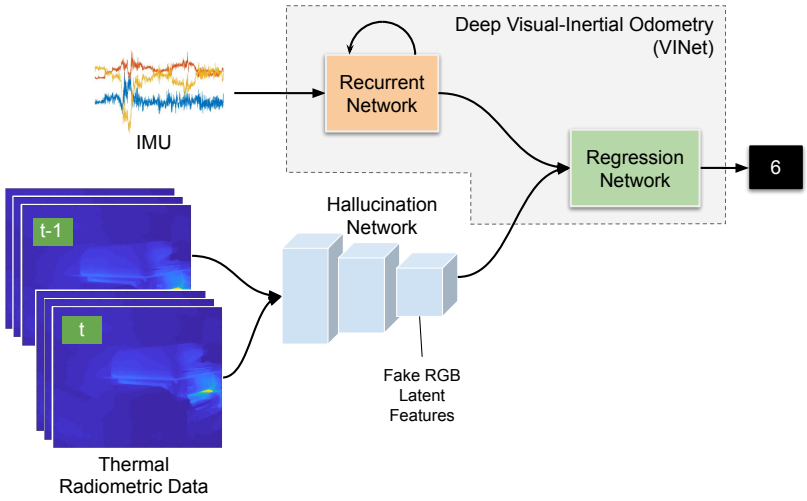


Figure 5.5: The hallucination network is validated by feeding the fake RGB features to VINet and measuring the pose estimation discrepancy.

truth format to compare with (VICON, Lidar Gmapping, or VINS-Mono), we align the predicted poses with the (pseudo) ground truth using Horn approaches and evaluate only the poses that are closest in time. We use the evaluation tools from the TUM RGB-D dataset to do this[2].

5.5.3 Sensitivity Analysis

To understand the influence of the hallucination network, we perform a sensitivity analysis in the following section.

[2]https://vision.in.tum.de/data/datasets/rgbd-dataset

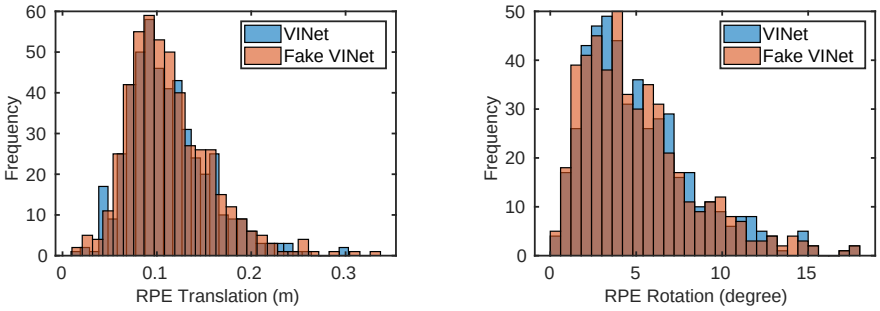


Figure 5.6: Relative Pose Error (RPE) distribution between VINet and Fake VINet for both translation and rotation.

5.5.3.1 Validating the Hallucination Network

To validate the hallucination network, we replace the visual decoder network from VINet with the hallucination network from DeepTIO as seen in Figure 5.5. By feeding the hallucinated visual features (fake RGB features) to the original VINet, we can measure how accurate the learnt representation produced by the hallucination network is. Figure 5.6 shows the distribution of RPE between VINet and Fake VINet (VINet with input from fake RGB features). It can be seen that the error distribution for both translation and rotation are very similar, showing the success of training the hallucination network. Table 5.1 shows how close the average RPE between VINet and Fake VINet are. Surprisingly, the Fake VINet got a slightly better result for rotation estimation, showing the efficacy of training using Huber Loss. Figure 5.7 illustrates the visualization of the output features from VINet and from hallucination network in the test sequence. It can be seen that the network can hallucinate visual features accurately (top) despite the lack of features in thermal domain. We presume that as long as there are spatial correspondences between thermal and RGB image, the network will learn to associate similar features and interpolate the missing ones. However, there are also cases when the hallucination network produces erroneous features (bottom) due to blurriness or lack of thermal edges. In this case, selective fusion plays an important role for selecting only relevant information from the hallucination network. It can be seen that in the erroneous case example, the DeepTIO's selective

Table 5.1: RPE between VINet and Fake VINet

Model	**t** (m)	**r** (°)
VINet	0.1124	5.1954
Fake VINet ($\mathcal{L}_2$)	0.1197	5.1926
Fake VINet (Huber)	**0.1128**	**5.0739**

fusion produces less dense fusion masks, indicating less features are being used.

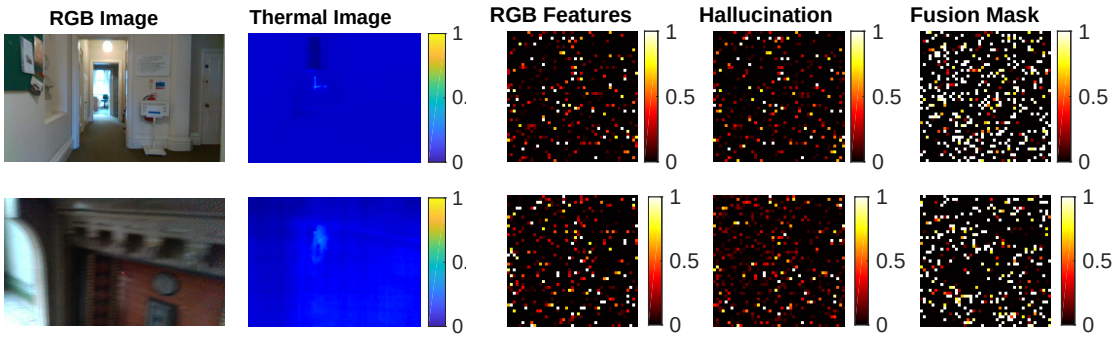


Figure 5.7: Comparison between RGB features produced by VINet and fake RGB features produced by DeepTIO's hallucination network in Corridor 2. Top: example of accurate hallucination. Bottom: example of erroneous hallucination. From left to right: RGB image, thermal image, original RGB features, hallucinated RGB features, and fusion mask for the hallucination features generated by DeepTIO.

5.5.3.2 The Influence of Each Feature Modality

To understand the influence of each feature modality, we decouple each feature modality, train it separately, and measure accuracy. The result can be seen in Table 5.2 and shows that thermal alone achieved the worst accuracy, implying the difficulty of estimating odometry solely based on temperature profile information. IMU alone clearly shows much stronger performance although the optimal solution may only produce only 3-DoF poses (instead of 6-DoF) as seen in [20] since there is not enough information from the IMU data to produce accurate 6-DoF poses. Incorporating IMU with thermal features or fake RGB features improves the accuracy (ATE) as the thermal or the visual features constrains the IMU error growth. Adding Fake RGB features to the model with IMU+Thermal further reduces the ATE, indicating

Table 5.2: The Impact of Each Feature Modality and Selective Fusion

Features	SF†	**t** (m)	**r** (°)	ATE (m)
Thermal	-	0.1497	6.5839	6.8347
IMU	-	0.1204	5.0151	1.7779
IMU+Thermal*	-	0.1133	5.3112	1.4731
IMU+Thermal$^+$	✓	0.1192	5.0461	0.7122
IMU+Fake RGB*	-	0.1153	5.2378	1.5021
IMU+Fake RGB$^+$	✓	0.1090	5.2300	1.0280
IMU+Thermal+Fake RGB*	-	0.1080	5.2127	1.2824
IMU+Thermal+Fake RGB$^+$	✓	**0.1074**	**4.8826**	**0.5267**

* has 52 M weights, while $^+$ has 136 M weights.

†Whether Selective Fusion (SF) is employed or not.

that the hallucinated visual features help generate more accurate poses. Note that all feature fusions (with the same mark in Table 5.2) have the same network capacity, indicating that the improved accuracy is due to more useful information, rather than increased network capacity.

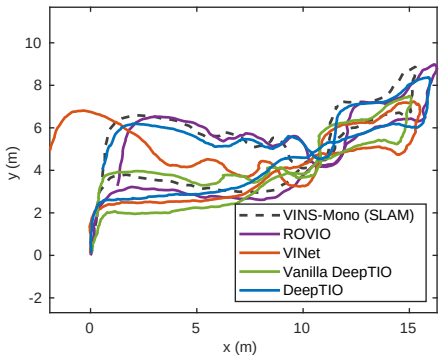
5.5.3.3 The Influence of Selective Fusion

As seen in Table 5.2, incorporating selective fusion to the combined features consistently reduces the ATE over the network without selective fusion. This shows that selective fusion plays an important role in producing accurate results as each feature modality comes with intrinsic noises, the hallucination network may produce erroneous visual features, and there is time misalignment between the sensors and the ground truth. Finally, putting together all feature modalities with selective fusion yields the strongest performance for both RPE and ATE.

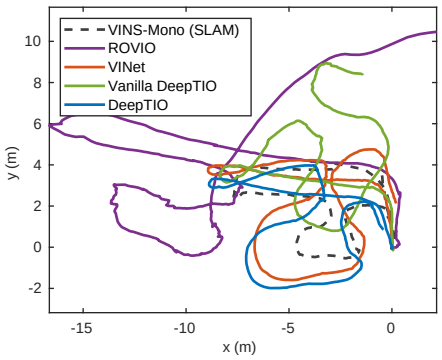
5.5.4 Evaluation on Hand-held Data

5.5.4.1 Test in Benign Environment

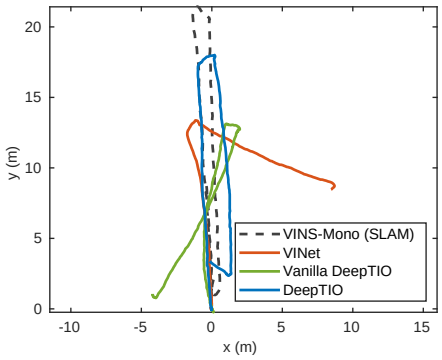
We test our model across different buildings and compare it with the state-of-the-art VIO frameworks to show that our DeepTIO solution is comparable. For comparison



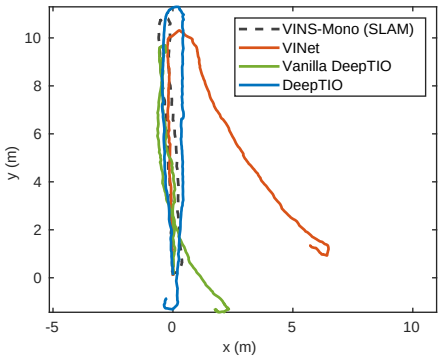
(a) Hand-held: Corridor 1



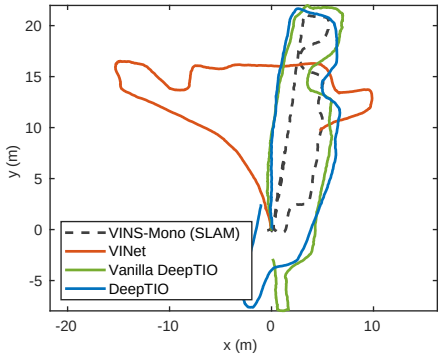
(b) Hand-held: Corridor 2



(c) Hand-held: Library 1



(d) Hand-held: Library 2



(e) Hand-held: Large Office

Figure 5.8: Qualitative evaluation on hand-held data in benign environment.

with a conventional approach we employ ROVIO [14], which tightly fuses IMU and visual data with an iterated extended Kalman filter. For comparison with deep learning based approaches, we use VINet [27] which fuses IMU and visual features in the intermediate layer. Both ROVIO and VINet use RGB as the input as they easily lose

track when we use thermal as the input. We also compare with Vanilla DeepTIO, a version of DeepTIO without the visual hallucination network. Figure 5.8 (a)-(e) depicts the qualitative results in this scenario.

Table 5.3 shows the numerical evaluation results in terms of ATE. ROVIO provides good accuracy in Corridor 2, although it suffers from the scaling problem and loses tracking in Corridor 1 due to lack of visual features when the camera faces white, flat walls. In misaligned sequences, ROVIO completely fails to initialize, since it requires tightly synchronized inputs. VINet also performs well when good alignment is available but suffers from large drift in presence of time misalignment. This shows that directly concatenating features may lead to sub-optimal performance. Nevertheless, VINet can still produce odometry where ROVIO completely fails, showing that deep learning approaches are more robust to the sensor alignment issue. However, the best results are achieved by Vanilla DeepTIO and DeepTIO as they employ selective fusion which is proven to be more robust to time synchronization issues [21]. Note that Vanilla DeepTIO and DeepTIO use a smaller thermal image resolution (464x348) compared to the RGB images used by ROVIO and VINet (848x480).

Vanilla DeepTIO achieves excellent results in Corridor 2, Large Office, and Library 2, but suffers from drift in Corridor 1 and Library 1. DeepTIO, on the other hand, produces better results due to the additional information provided by the hallucination network. Nonetheless, estimating an accurate scale is a problem in some sequences. As seen in the Large Office sequence, both Vanilla DeepTIO and DeepTIO give inaccurate scale, possibly due to a large variation of walking speeds. This scaling problem is very common in VO or VIO (as seen in ROVIO test in Corridor 1) and remains an open problem. Overall, DeepTIO yields the best ATE against the competing approaches, with an average ATE of 1.67 m.

5.5.4.2 Test in Smoke-filled Environment

In the smoke-filled environment, none of the VIO frameworks can work as the camera only captures black frames. Even Lidar odometry does not work well as near-visible

Table 5.3: ATE (m) For Experiment with Hand-held Data

	ROVIO	VINet	Vanilla DeepTIO (**ours**)	DeepTIO (**ours**)
Corridor 1	**0.3343**	1.1036	0.7122	0.5267
Corridor 2	6.2496	**1.8825**	2.1975	1.9333
Large Office*	failed	4.4359	3.3088	**3.2648**
Library 1*	failed	5.2647	2.5698	**2.0532**
Library 2*	failed	1.6812	1.5741	**0.5735**
Mean	3.2920	2.8736	1.7502	**1.6703**

*There is time misalignment among sensors for about 1-2 second.

light is blocked by the smoke [12]. In this case, we cannot provide quantitative evaluation with any (pseudo) ground truth. We instead provide a qualitative comparison with a zero-velocity-aided Inertial Navigation System (INS) [159], which is not impacted by visibility. This navigation system utilizes foot-mounted inertial sensors to detect Zero Velocity Updates (ZUPT) and thereby mitigates the fast error growth of stand-alone inertial navigation. Figure 5.9 shows the output trajectory together with the floor plan generated by FARO Lidar collected before the experiment with smoke. It can be seen that DeepTIO yields a similar trajectory shape to ZUPT. This shows that our model, despite being trained in a benign environment, can generalize to a smoke-filled environment as the thermal camera is not affected by the smoke. However, there is scaling issue which probably due to different speed of the camera (as an effect of different walking speed) or different temperature profile compared to the one observed in the training data. If we adjust the scale of DeepTIO, it can be seen that the prediction is very close to ZUPT. This shows that our model is promising for odometry estimation in smoke-filled environments.

5.5.4.3 Memory and Execution Time

The network was trained on an NVIDIA TITAN V GPU and required around 6-18 hours for training the hallucination network and 6-20 hours to train the remaining networks. The network contains around 136 millions weights, requiring 847 MB of

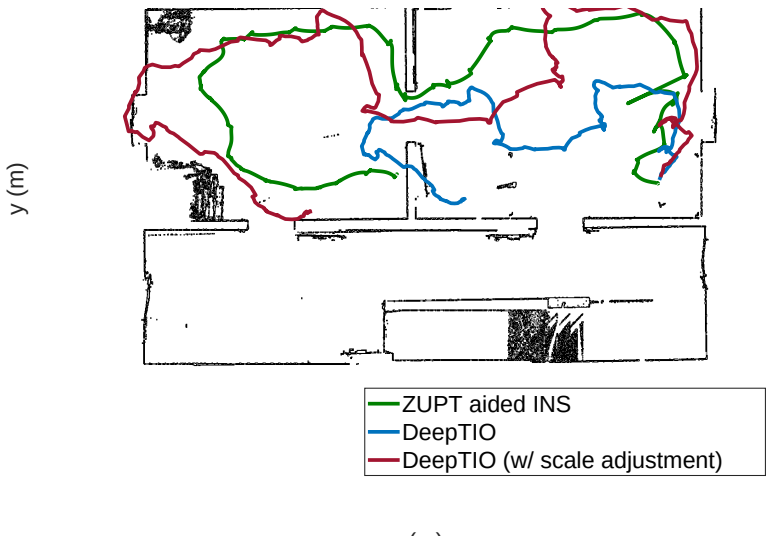
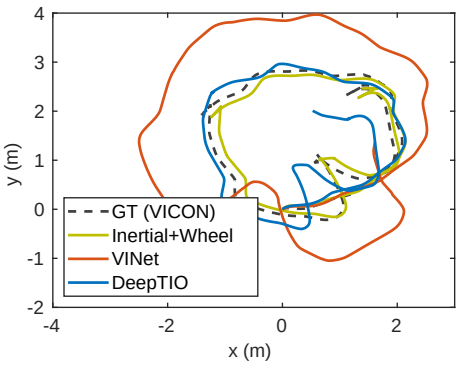


Figure 5.9: Test in real emergency scenario with smoke-filled environment. We qualitatively compared DeepTIO with ZUPT aided INS as VIO, VI-SLAM, or even Lidar odometry does not work in this visually-denied scenario.

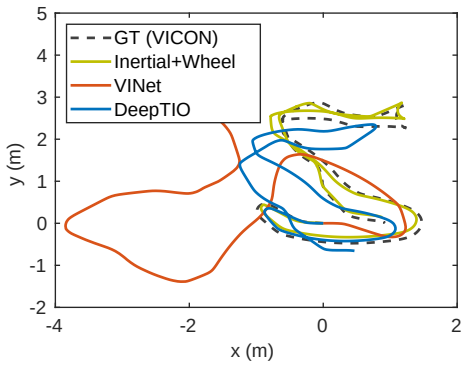
space. Neglecting the time to load and normalize the input, the model can run at 40 fps on a TITAN V and 5 fps on a standard CPU.

5.5.5 Evaluation on Mobile Robot Data

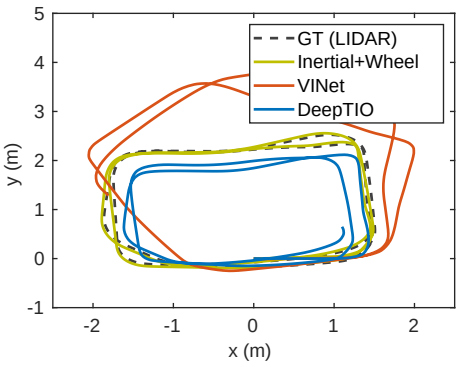
Figure 5.10 (a)-(f) depicts the test results of mobile robot data collected in two different buildings. As can be seen, DeepTIO yields very accurate result in all test sequences, outperforming VINet and generating a similar trajectory to inertial-assisted wheel odometry which we use as pseudo ground truth in training. In difficult motion when the mobile robot stops for short period of time (Figure 5.10 (a) and (b)), DeepTIO can still produce a relatively accurate trajectory. In the sequence with poor to no lighting (Figure 5.10 (c) and (d)), VINet experiences significant drift due to the low RGB illumination. On the contrary, DeepTIO can produce an accurate trajectory despite varying lighting conditions. DeepTIO also works very well in long sequences with challenging motions involving U-turns and with people walking within the camera frame as can be seen in Figure 5.10 (e) and (f). It indicates that using three modalities (although the RGB is a fake one) can yield more robust predictions in challenging environments compared to just using two modalities (e.g. VINet).



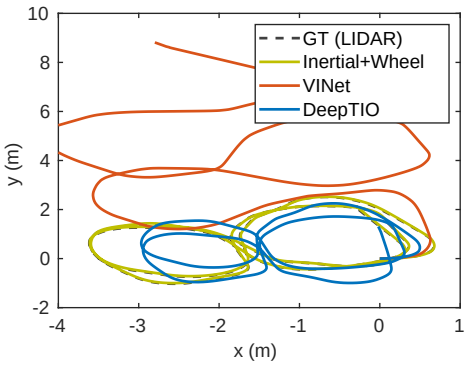
(a) Turtlebot: CPS Lab 1



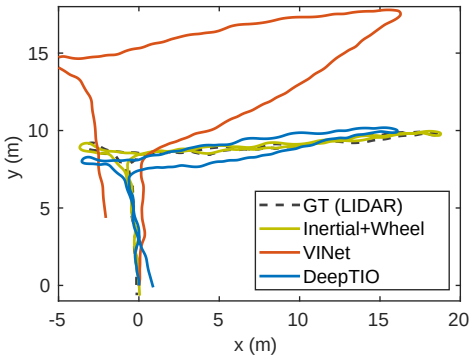
(b) Turtlebot: CPS Lab 2



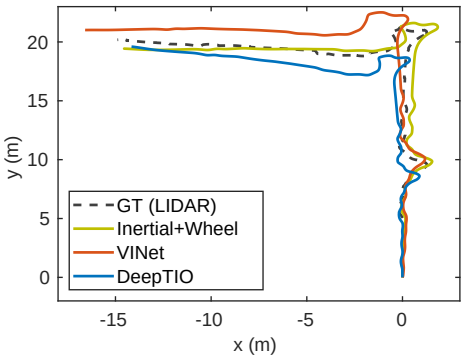
(c) Turtlebot: CPS Lab 3 (poor)



(d) Turtlebot: CPS Lab 4 (dark)



(e) Turtlebot: Corridor 1



(f) Turtlebot: Corridor 2

Figure 5.10: Qualitative evaluation in mobile robot data.

Table 5.4 describes the quantitative result for all test sequences. DeepTIO achieves a very low ATE (around 0.5 m in average). Comparing this result with the hand-held data, it implies that learning mobile robot odometry is easier than learning hand-held odometry. This is most likely because the robot movement is more stable in terms of

Table 5.4: ATE (m) For Experiment with Mobile Robot Data

Sequences	VINet	DeepTIO (**ours**)	Inertial +Wheel
CPS Lab 1	1.3619	**0.3824**	0.0931
CPS Lab 2	1.1440	**0.2785**	0.0805
CPS Lab 3 (poor lighting)	1.2225	**0.2250**	0.0033
CPS Lab 4 (darkness)	1.8379	**0.3807**	0.0036
Corridor 1	2.4497	**0.8988**	0.1279
Corridor 2	0.8294	**0.7433**	0.3595
Corridor 3	1.4629	**0.7307**	0.2297
Mean	1.4726	**0.5199**	0.1282

speed and planar constraints.

5.5.6 What the Hallucination Network Learns

In the previous section, we have shown that the hallucination network still produces very similar feature maps to the ones generated by the RGB network (e.g. from VINet) despite the fact that there are many objects (e.g. poster, signage) in the RGB image that are not differentiable from the background (e.g. wall) in the thermal domain. We presume that, via the hallucination loss, the hallucination network learns to associate similar features in the training data, amplify weak gradients, and probably interpolate missing features such that the two feature maps (i.e. thermal and visual) become very similar. During training, as long as the hallucination network still can find similar features (e.g. edges) between RGB and thermal (although it is just a fraction), it will interpolate the remaining differences. Note that this is possible as not all feature information in the early layers of CNN is useful, since the CNN structure (Flownet) gradually performs subsampling/pooling, leaving only the most important features in the final layer. In the earlier layers, the hallucination network might produce less similar hallucinated features. However, the final output from the hallucination network is very similar to the original RGB network as the pooling layers retains only the strongest features that are important for the task.

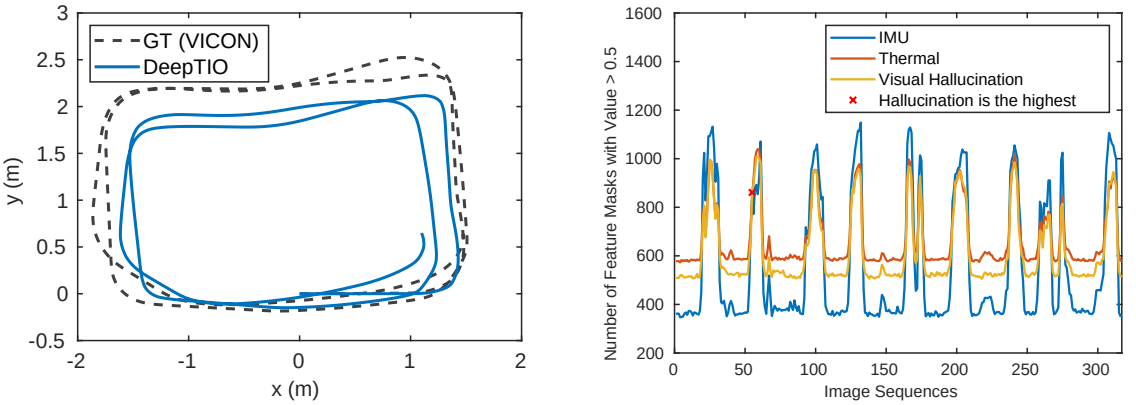
5.5.7 Interpreting Selective Fusion

In order to have better understanding of what the selective fusion module learns, we plot the total number of fusion masks for each feature modality with value higher than 0.5. We set 0.5 as the limit to determine the importance of each feature modality. The higher the number of masks with value > 0.5, the more important that feature modality is in that particular setting. The result can be seen from Figure 5.11. As we can see, in most cases, both thermal and hallucination features are usually more important for 6-DoF camera pose estimation than the IMU features. However, everytime the camera rotates sharply, more IMU features are used and the importance of IMU surpasses that of other modalities. This can be seen clearly in Figure 5.11 (a). Note that the total number of peaks in Figure 5.11 (a) is the same as the number of times the robot makes a sharp turn (around 90^0) in the corresponding trajectory, indicating more information is required to generate accurate odometry in difficult motion cases. In a case when the camera moves more randomly as seen in the Sequences CPS Lab 1, the hallucination features show their dominance, even demonstrating higher importance than thermal and IMU. This possibly happens as there are times when the camera views an area with less thermal gradient. In that case, the hallucination network clearly can provide complementary information alongside the other modalities.

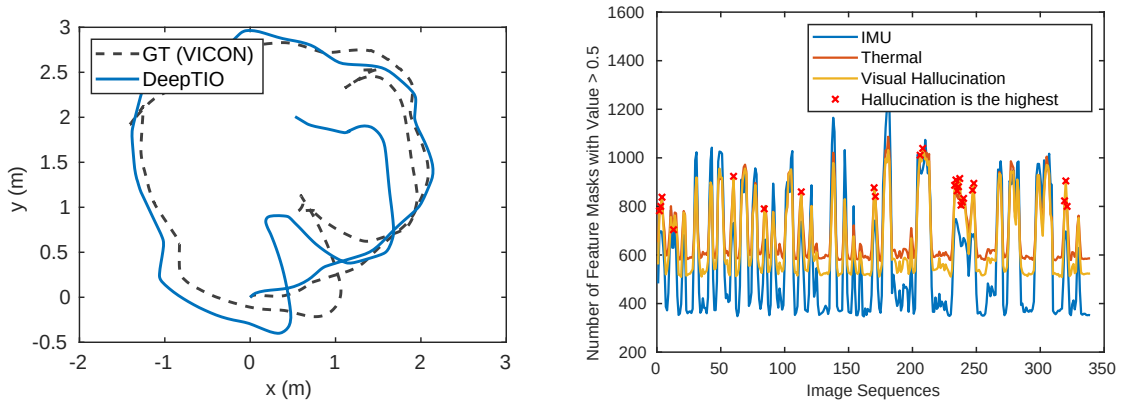
5.5.8 Limitations

Despite the fact that DeepTIO can work well in our test scenarios, there are some limitations:

1. *Sensitivity to abrupt motion and thermal reflection.* Similar to other odometry techniques applied to RGB camera, when the camera moves abruptly, the images blur and it does impact the accuracy of the model. As you can see in Figure 5.12, DeepTIO produces the largest rotational errors when the robot performs abrupt U-turn, blurring the images. This is compounded by thermal reflection



(a) Turtlebot: CPS Lab 3



(b) Turtlebot: CPS Lab 1

Figure 5.11: Selective fusion mask for mobile robot data in Sequences (a) CPS Lab 3 and (b) CPS Lab 1. We plot the total number of masks for each feature modality with value higher than 0.5, indicating the importance of the features. Left figures indicates the corresponding trajectories.

from a glass surface which can be regarded as dynamic objects (noise). This is interesting case which only happens in thermal imaging as the glass surfaces block infrared signals yielding reflection of the object temperature while at the same time allow the visible light to go through. However, note that even if the network typically produces larger error in this challenging scenario, the network still generates reasonably accurate odometry estimation as seen in Figure 5.10.

2. *Sensitivity to sampling rate.* As we trained DeepTIO with a frame rate of 4-5 fps, the network will only perform well by using that frame rate. When inferring with lower or faster fps, the accuracy will degrade as seen in Figure

5.13. Training with multiple fps at the same time might be possible to obtain robustness against different sampling rates. This may also alleviate the problem of scaling as the network would be trained with more variations of parallax. However, this might require a (pseudo) ground truth with constant fps, i.e. not irregularly sampled by a key frame selection process as in VINS-Mono.

3. *Robustness against distributional shift.* DNNs are usually vulnerable to distributional (covariate) shift which occurs when the test data are sampled from a different distribution than the training data. As we train our model in a benign environment, when we test it in smoke-filled environment, it is expected that we will experience some covariate shift as the temperature profile will be different. Development of an odometry approach robust against this distributional shift might be necessary to enable practical odometry in adverse environments.

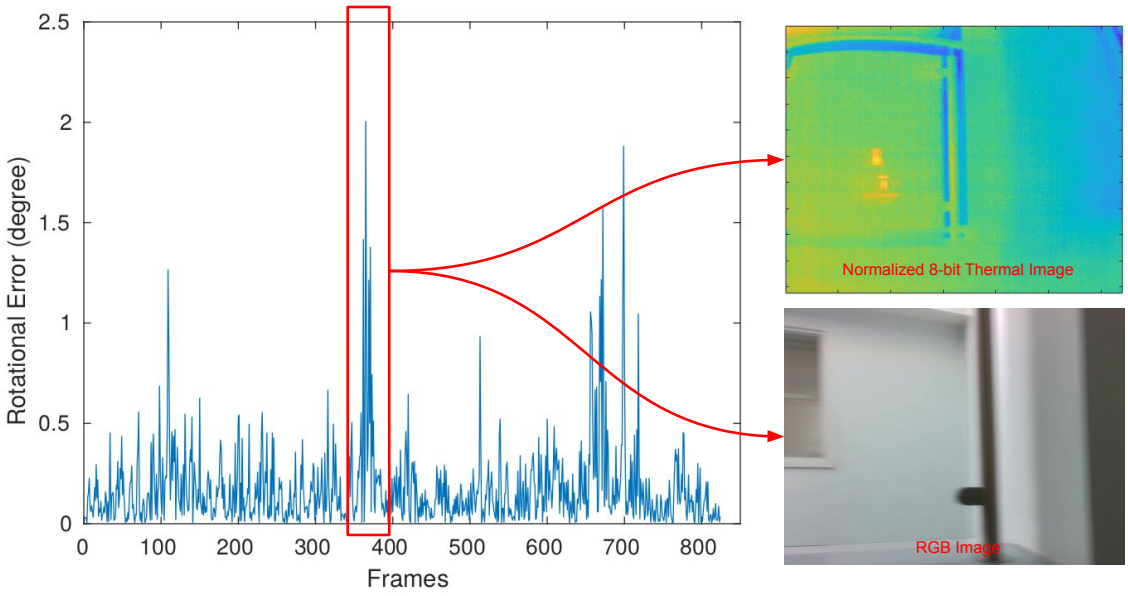


Figure 5.12: DeepTIO produces the largest relative rotational error when the camera moves abruptly in U-turn while at the same time there is thermal reflection from the glass surface (as it blocks infrared signals) which can be regarded as dynamic objects. Note that the reflection is not visible in RGB image. The relative rotational error is taken from mobile robot data in Corridor 1 sequence.

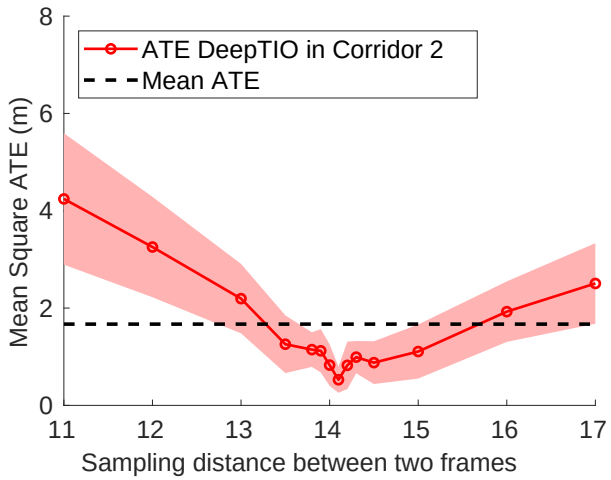


Figure 5.13: Sensitivity towards sampling rate (fps). By using 14 sampling distance between two frames (optimum performance), it keeps the prediction rate around 4.2 (for 60 thermal fps), which is still in the range of 4-5 fps used during training.

5.6 Conclusion

In this chapter, we have presented a novel DNN-based method for thermal-inertial odometry, termed DeepTIO, using hallucination networks. We demonstrated that the hallucination network can provide side information for the thermal network and, combined with a selective fusion mechanism, is able to produce accurate odometry estimation. We have also shown that DeepTIO can work well in various scenarios and environment conditions including smoke-filled environment. We note that our model still have some limitations with respect to abrupt motion, thermal reflection, distributional shift, or sensitive towards sampling rate variation. For future works, we can incorporate other sensor modalities to estimate more accurate scale in diverse scenarios and developing robust techniques to cope with thermal reflection or distributional (covariate) shifts in the test data. Another interesting future development includes the development of loop closure detection or a thermal SLAM system to alleviate the problem of drift in longer sequences.

Chapter 6

Conclusion and Future Work

In this chapter, we conclude the thesis and discuss possible future research directions. We summarize the main contributions and the conclusions that can be drawn from this work in Section 6.1. The directions for future research will be discussed in Section 6.2.

6.1 Conclusion

In this thesis, we have tackled the complex problem of odometry estimation using deep neural networks. Different vision-based modalities and techniques are explored for a variety of scenarios including application in visually-denied environments. We summarize the conclusions of this work as follows:

1. We introduced an optimization strategy for DNN-based visual odometry in Chapter 3 by incorporating a windowed-based composite transformation loss to the standard relative transformation loss via bounded pose regression loss. We trained the bounded pose regression loss by gradually increasing the difficulty of the objective function through Geometry-Aware Curriculum Learning (GA-CL). We have shown that GA-CL can improve the translation and rotation estimation of standard DNN-based visual odometry by around 21% and 16% respectively. From this result, we now understand that an objective function that enforces consistency in the estimated poses can improve the accuracy

of DNN-based visual odometry. We also proposed an attention network for DNN-based visual odometry by using the current latent poses to help generate sensible attention mask. We have shown that decoupling the attention network for translation and rotation yields 26.87% improvement than joining them, supported by a better interpretation of the attention masks when we visualize them. From this experiments, we also understand that we can selectively use the most important features from DNN-based visual odometry to generate more accurate trajectory.

2. In Chapter 4, we presented the first successful distillation approach to the pose regression problem by emphasizing the knowledge transfer between the teacher and the student network only when we trust the teacher network. We have shown that the normalized teacher loss can be used as an effective attentive mechanism in this knowledge transfer. We demonstrated this attentive mechanism via Attentive Imitation Loss (AIL), which is applied in the last layer as the final objective function, and via Attentive Hint Training (AHT) approach, which is applied in the intermediate layer. With this mechanism, we can keep the student prediction accuracy close to the teacher (or even better) for up to 92.95% parameter reduction. We also proposed a fusion mechanism between distillation and Low-Rank Separable Filters (LRSF) to further reduce the number of network weights. We concluded that performing decomposition via LRSF after distillation can improve the generalization capability of the network while enabling further compression of the network parameters for up to 97.39%. From this result, we know that distilling pose regressor network is possible if we emphasize the knowledge transfer only when we trust the teacher.

3. In Chapter 5, we proposed the first DNN-based thermal-inertial odometry, termed DeepTIO, which incorporates a visual hallucination network to alleviate the lack of robust features in thermal images. We have shown that the

hallucination network can provide side information for the thermal feature extractor network. By combining the hallucination network with selective fusion over three modalities (i.e. thermal, hallucinated visual, and IMU features), we can generate accurate odometry estimation for two different scenarios (i.e. hand-held motion and mobile robot) in diverse environment conditions (i.e. good illumination, darkness, and smoke-filled environments). The experimental results indicate that DeepTIO has comparable performance with visual-inertial odometry algorithms when they are tested in benign environments and better performance than them when the evaluations are performed in dark or smoke-filled environments. From this experiments we know that we can generate accurate odometry from thermal images if we incorporate the network with hallucinated visual features.

6.2 Directions for Future Work

The work in this thesis opens many interesting future research directions including:

1. **A Deep Multi-body Motion Estimation**. We have shown that accurate and consistent odometry estimation can be realized with DNNs. However, the current application is limited to static scenes or low dynamics (i.e. small number of dynamic objects in front of the camera). In the case of highly dynamic scenes, it might be very challenging for the current approach to generate accurate pose estimation. In conventional Structure from Motion (SfM) techniques, there are a category of algorithms called multi-body SfM (e.g. [30], [157], [90]) which try to not only calculate the camera ego-motion but also to estimate the motion of dynamic objects in front of the camera. This multi-body motion estimation is not only helpful to improve the accuracy of the visual odometry system in dynamic environments but also useful when developing obstacle avoidance and path planning algorithms for autonomous vehicles and mobile robots. Devising

a deep multi-body motion estimation is a natural progression to improve the DNN-based visual odometry estimation in highly dynamic environments.

2. **Distillation for Multi-modal Sensor Fusion**. In this thesis, we have demonstrated the first distillation approach for the pose regression problem with a single modality. Future work could consider designing a distillation approach that can transfer the knowledge from a large, multi-modal teacher network, to a small, multi-modal student network. This is a challenging problem especially when the teacher network is equipped with a selective fusion module. It will be interesting to explore if we can eliminate selective fusion from the student network (as it contributes to almost half of the total weights in DeepTIO) without compromising the student performance. Designing a knowledge transfer mechanism that can mimic the ability to attentively fuse multi-modal sensor data without actually including the selective fusion module might be the key to achieve this goal. A sophisticated hint training approach might need to be developed to enable this knowledge transfer.

3. **Thermal-Inertial Re-localization and SLAM**. This thesis focuses on developing a DNN-based thermal-inertial odometry (DeepTIO). While the trajectory estimation from DeepTIO is quite accurate, it is still subject to cumulative drift, especially if it is tested in long sequences. A potential solution to alleviate the drift problem is to develop a full thermal-inertial SLAM system equipped with re-localization or loop closure detection capability. By incorporating loop closure detection, the SLAM back-end can further optimize the estimated trajectory, correcting the accumulated drift. Both odometry estimation and loop closure detection might be constructed based on DNNs, while existing graphical model-based optimizers, such as g2o [91], can be used for the SLAM back-end.

Finally, in this thesis, we have demonstrated that accurate and efficient odometry estimation can be realized with carefully crafted deep neural networks. Advances in

this area are and will continue to fuel a range of exciting applications from autonomous robotic systems to location-based services for humans in both benign and challenging emergency environments.

Bibliography

[1] Pablo F Alcantarilla, José J Yebes, Javier Almazán, and Luis M Bergasa. On combining visual slam and dense scene flow to increase the robustness of localization and mapping in dynamic environments. In *2012 IEEE International Conference on Robotics and Automation*, pages 1290–1297. IEEE, 2012.

[2] Yasin Almalioglu, Muhamad Risqi U Saputra, Pedro PB de Gusmao, Andrew Markham, and Niki Trigoni. Ganvo: Unsupervised deep monocular visual odometry and depth estimation with generative adversarial networks. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[3] Mohammad OA Aqel, Mohammad H Marhaban, M Iqbal Saripan, and Napsiah Bt Ismail. Review of visual odometry: types, approaches, challenges, and applications. *SpringerPlus*, 5(1):1897, 2016.

[4] Amir Averbuch, Gabi Liron, and Ben Zion Bobrovsky. Scene based non-uniformity correction in thermal images using kalman filter. *Image and Vision Computing*, 25(6):833–851, 2007.

[5] Hernán Badino, Akihiro Yamamoto, and Takeo Kanade. Visual odometry by multi-frame feature integration. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 222–229, 2013.

[6] Dan Barnes, Will Maddern, Geoffrey Pascoe, and Ingmar Posner. Driven to distraction: Self-supervised distractor learning for robust monocular visual odome-

try in urban environments. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1894–1900. IEEE, 2018.

[7] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-Up Robust Features (SURF). *Comput. Vis. Image Underst.*, 110(3):346–359, 2008.

[8] Nicola Bellotto, Kevin Burn, Eric Fletcher, and Stefan Wermter. Appearance-based localization for mobile robots using digital zoom and visual compass. *Robotics and Autonomous Systems*, 56(2):143–156, 2008.

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

[10] Berta Bescos, José M Fácil, Javier Civera, and José Neira. Dynaslam: Tracking, mapping, and inpainting in dynamic scenes. *IEEE Robotics and Automation Letters*, 3(4):4076–4083, 2018.

[11] Sourav Bhattacharya and Nicholas D. Lane. Sparsification and Separation of Deep Learning Layers for Constrained Resource Inference on Wearables. In *ACM Conf. Embed. Netw. Sens. Syst. CD-ROM - SenSys*, pages 176–189, 2016.

[12] Mario Bijelic, Fahim Mannan, Tobias Gruber, Werner Ritter, Klaus Dietmayer, and Felix Heide. Seeing through fog without seeing fog: Deep sensor fusion in the absence of labeled training data. *arXiv preprint arXiv:1902.08913*, 2019.

[13] José-Luis Blanco-Claraco, Francisco-Ángel Moreno-Dueñas, and Javier González-Jiménez. The málaga urban dataset: High-rate stereo and lidar in a realistic urban scenario. *The International Journal of Robotics Research*, 33(2):207–214, 2014.

[14] Michael Bloesch, Sammy Omari, Marco Hutter, and Roland Siegwart. Robust visual inertial odometry using a direct ekf-based approach. In *IEEE/RSJ IROS*, pages 298–304, 2015.

[15] Paulo Vinicius Koerich Borges and Stephen Vidas. Practical infrared visual odometry. *IEEE Transactions on Intelligent Transportation Systems*, 17(8):2205–2213, 2016.

[16] Laurie Bose, Jianing Chen, Stephen J Carey, Piotr Dudek, and Walterio Mayol-Cuevas. Visual odometry for pixel processor arrays. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4604–4612, 2017.

[17] Martin Buczko and Volker Willert. How to distinguish inliers from outliers in visual odometry for high-speed automotive applications. In *2016 IEEE Intelligent Vehicles Symposium (IV)*, pages 478–483. IEEE, 2016.

[18] Martin Buczko and Volker Willert. Monocular outlier detection for visual odometry. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 739–745. IEEE, 2017.

[19] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. BRIEF : Binary Robust Independent Elementary Features. In *Eur. Conf. Comput. Vis.*, pages 778–792, 2010.

[20] Changhao Chen, Xiaoxuan Lu, Andrew Markham, and Niki Trigoni. Ionet: Learning to cure the curse of drift in inertial odometry. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[21] Changhao Chen, Stefano Rosa, Yishu Miao, Chris Xiaoxuan Lu, Wei Wu, Andrew Markham, and Niki Trigoni. Selective sensor fusion for neural visual-inertial odometry. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 10542–10551, 2019.

[22] Guobin Chen, Wongun Choi, Xiang Yu, Tony Han, and Manmohan Chandraker. Learning efficient object detection models with knowledge distillation. In *Advances in Neural Information Processing Systems (NIPS)*, pages 742–751, 2017.

[23] Yu Cheng, Duo Wang, Pan Zhou, and Tao Zhang. Model compression and acceleration for deep neural networks: The principles, progress, and challenges. *IEEE Signal Processing Magazine*, 35(1):126–136, 2018.

[24] Hsiang-Jen Chien, Chen-Chi Chuang, Chia-Yen Chen, and Reinhard Klette. When to use what feature? sift, surf, orb, or a-kaze features for monocular visual odometry. In *2016 International Conference on Image and Vision Computing New Zealand (IVCNZ)*, pages 1–6. IEEE, 2016.

[25] Chiho Choi, Sangpil Kim, and Karthik Ramani. Learning hand articulations by hallucinating heat distribution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3104–3113, 2017.

[26] O Chum and J Matas. Matching with PROSAC-Progressive Sample Consensus. In *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, number I, pages 220–226, 2005.

[27] Ronald Clark, Sen Wang, Hongkai Wen, Andrew Markham, and Niki Trigoni. Vinet: Visual-inertial odometry as a sequence-to-sequence learning problem. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[28] Jerome T Connor, R Douglas Martin, and Les E Atlas. Recurrent neural networks and robust time series prediction. *IEEE transactions on neural networks*, 5(2):240–254, 1994.

[29] Gabriele Costante and Thomas A Ciarfuglia. LS-VO: Learning Dense Optical Subspace for Robust Visual Odometry Estimation. *IEEE Robotics and Automation Letters*, 3(3), 2018.

[30] Joao Costeira and Takeo Kanade. A multi-body factorization method for motion analysis. In *Proceedings of IEEE International Conference on Computer Vision*, pages 1071–1076. IEEE, 1995.

[31] Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. BinaryConnect: Training Deep Neural Networks with binary weights during propagations. In *Adv. Neural Inf. Process. Syst.*, pages 1–9, 2015.

[32] M. Cummins and P. Newman. FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance. *Int. J. Rob. Res.*, 27(6):647–665, 2008.

[33] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Deep Image Homography Estimation. In *arXiv:1606.03798*, 2016.

[34] Alexey Dosovitskiy, Philipp Fischery, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. FlowNet: Learning Optical Flow with Convolutional Networks. In *IEEE Int. Conf. Comput. Vis.*, volume 11-18-Dece, pages 2758–2766, 2016.

[35] Jeffrey L Elman. Learning and development in neural networks: The importance of starting small. *Cognition*, 48(1):71–99, 1993.

[36] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct Sparse Odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 40(3):611–625, 2018.

[37] Jakob Engel, Thomas Sch, and Daniel Cremers. LSD-SLAM: Direct Monocular SLAM. In *Eur. Conf. Comput. Vis.*, pages 834–849, 2014.

[38] O Faugeras. Three-dimensional Computer Vision: a Geometric Viewpoint. In *Artif. Intell.* MIT Press, 1993.

[39] Martin a Fischler and Robert C Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Commun. ACM*, 24:381–395, 1981.

[40] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. SVO: Fast Semi-Direct Monocular Visual Odometry. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 15–22, 2014.

[41] Friedrich Fraundorfer and Davide Scaramuzza. Visual odometry: Part II - Matching, Robustness, Optimization, and Applications. *IEEE Robot. Autom. Mag.*, 19(2):78–90, 2012.

[42] Jorge Fuentes-Pacheco, Jose Ruiz-Ascencio, and Juan Manuel Rendon-Mancha. Visual Simultaneous Localization and Mapping: a Survey. *Artif. Intell. Rev.*, 43(1):55–81, 2012.

[43] Xiao Shan Gao, Xiao Rong Hou, Jianliang Tang, and Hang Fei Cheng. Complete Solution Classification for the Perspective-Three-Point Problem. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(8):930–943, 2003.

[44] Andrew P Gee and Walterio Mayol-Cuevas. Real-time model-based slam using line segments. In *International Symposium on Visual Computing*, pages 354–363. Springer, 2006.

[45] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pages 3354–3361, 2012.

[46] Andreas Geiger, Julius Ziegler, and Christoph Stiller. StereoScan: Dense 3D Reconstruction in Real-time. In *IEEE Intell. Veh. Symp.*, pages 1–9, 2011.

[47] Ruben Gomez-Ojeda and Javier Gonzalez-Jimenez. Robust stereo visual odometry through a probabilistic combination of points and line segments. In *2016*

IEEE International Conference on Robotics and Automation (ICRA), pages 2521–2526. IEEE, 2016.

[48] Ruben Gomez-Ojeda, Francisco-Angel Moreno, David Zuñiga-Noël, Davide Scaramuzza, and Javier Gonzalez-Jimenez. Pl-slam: a stereo slam system through the combination of points and line segments. *IEEE Transactions on Robotics*, 35(3):734–746, 2019.

[49] Yunchao Gong, Liu Liu, Ming Yang, and Lubomir Bourdev. Compressing Deep Convolutional Networks using Vector Quantization. In *arXiv:1412.6115*, 2015.

[50] Ramon Gonzalez, Francisco Rodriguez, Jose Luis Guzman, Cedric Pradalier, and Roland Siegwart. Combined visual odometry and visual compass for off-road mobile robots localization. *Robotica*, 30(6):865–878, 2012.

[51] Ramon Gonzalez, Francisco Rodriguez, Jose Luis Guzman, Cedric Pradalier, and Roland Siegwart. Control of off-road mobile robots using visual odometry and slip compensation. *Advanced Robotics*, 27(11):893–906, 2013.

[52] Colin Greatwood, Laurie Bose, Thomas Richardson, Walterio Mayol-Cuevas, Jianing Chen, Stephen J Carey, and Piotr Dudek. Perspective correcting visual odometry for agile mavs using a pixel processor array. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 987–994. IEEE, 2018.

[53] Yanming Guo, Yu Liu, Ard Oerlemans, Songyang Lao, Song Wu, and Michael S Lew. Deep Learning for Visual Understanding : A Review. *Neurocomputing*, 187:27–48, 2015.

[54] Yiwen Guo, Anbang Yao, and Yurong Chen. Dynamic Network Surgery for Efficient DNNs. In *Adv. Neural Inf. Process. Syst.*, 2016.

[55] Suyog Gupta, Ankur Agrawal, Kailash Gopalakrishnan, and Pritish Narayanan. Deep learning with limited numerical precision. In *International Conference on Machine Learning (ICML)*, pages 1737–1746, 2015.

[56] H. C. Longuet-Higgins. A Computer Algorithm for Reconstructing a Scene from Two Projections. *Nature*, 293:133 – 135, 1981.

[57] Song Han, Huizi Mao, and William J. Dally. Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. In *Int. Conf. Learn. Represent.*, 2016.

[58] Song Han, Jeff Pool, John Tran, and William J. Dally. Learning both Weights and Connections for Efficient Neural Networks. In *Adv. Neural Inf. Process. Syst.*, pages 1–9, 2015.

[59] Ankur Handa, Michael Bloesch, Viorica Patraucean, Simon Stent, John McCormac, and Andrew Davison. gvnn: Neural Network Library for Geometric Computer Vision. In *arXiv:1607.07405*, 2016.

[60] Chris Harris and Mike Stephens. A Combined Corner and Edge Detector. In *Alvey Vis. Conf.*, pages 147–151, 1988.

[61] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edi edition, 2004.

[62] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network. pages 1–9, 2015.

[63] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[64] Judy Hoffman, Saurabh Gupta, and Trevor Darrell. Learning with side information through modality hallucination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 826–834, 2016.

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

[66] Albert S Huang, Abraham Bachrach, Peter Henry, Michael Krainin, Daniel Maturana, Dieter Fox, and Nicholas Roy. Visual odometry and mapping for autonomous flight using an rgb-d camera. In *Robotics Research*, pages 235–252. Springer, 2017.

[67] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized Neural Networks. In *Adv. Neural Inf. Process. Syst.*, 2016.

[68] Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics*, pages 492–518. Springer, 1992.

[69] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *IEEE conference on computer vision and pattern recognition (CVPR)*, volume 2, page 6, 2017.

[70] Ganesh Iyer, J Krishna Murthy, Gunshi Gupta, Madhava Krishna, and Liam Paull. Geometric consistency for self-supervised end-to-end visual odometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 267–275, 2018.

[71] Max Jaderberg, Andrea Vedaldi, and Andrew Zisserman. Speeding up convolutional neural networks with low rank expansions. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2014.

[72] Lu Jiang, Deyu Meng, Shoou-I Yu, Zhenzhong Lan, Shiguang Shan, and Alexander Hauptmann. Self-paced learning with diversity. In *Advances in Neural Information Processing Systems*, pages 2078–2086, 2014.

[73] Lu Jiang, Deyu Meng, Qian Zhao, Shiguang Shan, and Alexander G Hauptmann. Self-paced curriculum learning. In *AAAI*, volume 2, page 6, 2015.

[74] Antonio R Jimenez, Fernando Seco, Carlos Prieto, and Jorge Guevara. A comparison of pedestrian dead-reckoning algorithms using a low-cost mems imu. In *2009 IEEE International Symposium on Intelligent Signal Processing*, pages 37–42. IEEE, 2009.

[75] Hailin Jin, Paolo Favaro, and Stefano Soatto. A semi-direct approach to structure from motion. *The Visual Computer*, 19(6):377–394, 2003.

[76] Christoforos Kanellakis and George Nikolakopoulos. Evaluation of visual localization systems in underground mining. In *2016 24th Mediterranean Conference on Control and Automation (MED)*, pages 539–544. IEEE, 2016.

[77] Alex Kendall, Matthew Grimes, and Roberto Cipolla. PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization. In *IEEE Int. Conf. Comput. Vis.*, pages 2938–2946, 2015.

[78] Shehryar Khattak, Christos Papachristos, and Kostas Alexis. Keyframe-based direct thermal-inertial odometry. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[79] Deok-Hwa Kim and Jong-Hwan Kim. Effective background model-based rgb-d dense visual odometry in a dynamic environment. *IEEE Transactions on Robotics*, 32(6):1565–1573, 2016.

[80] Bernd Kitt, Andreas Geiger, and Henning Lategahn. Visual odometry based on stereo image sequences with ransac-based outlier rejection scheme. In *2010 ieee intelligent vehicles symposium*, pages 486–492. IEEE, 2010.

[81] Bernd Kitt, Frank Moosmann, and Christoph Stiller. Moving on to dynamic environments: Visual odometry using feature classification. In *2010 IEEE/RSJ*

International Conference on Intelligent Robots and Systems, pages 5551–5556. IEEE, 2010.

[82] Georg Klein and David Murray. Parallel Tracking and Mapping for Small AR Workspaces. In *IEEE ACM Int. Symp. Mix. Augment. Real.*, 2007.

[83] Georg Klein and David Murray. Parallel Tracking and Mapping on a Camera Phone. In *8th IEEE Int. Symp. Mix. Augment. Real.*, pages 83–86, 2009.

[84] Kishore Konda and Roland Memisevic. Unsupervised Learning of Depth and Motion. In *arXiv:1312.3429*, 2013.

[85] Kishore Konda and Roland Memisevic. Learning Visual Odometry with a Convolutional Network. In *Int. Conf. Comput. Vis. Theory Appl.*, pages 486–490, 2015.

[86] Kurt Konolige and Motilal Agrawal. Frameslam: From bundle adjustment to real-time visual mapping. *IEEE Transactions on Robotics*, 24(5):1066–1077, 2008.

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

[88] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Adv. Neural Inf. Process. Syst.*, pages 1–9, 2012.

[89] M Pawan Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems*, pages 1189–1197, 2010.

[90] Suryansh Kumar, Yuchao Dai, and Hongdong Li. Multi-body non-rigid structure-from-motion. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 148–156. IEEE, 2016.

[91] Rainer Kümmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. g 2 o: A general framework for graph optimization. In *2011 IEEE International Conference on Robotics and Automation*, pages 3607–3613. IEEE, 2011.

[92] Rainer Kummerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. G2o: A General Framework for Graph Optimization. In *IEEE Int. Conf. Robot. Autom.*, pages 3607–3613, 2011.

[93] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep Learning. *Nature*, 2016.

[94] Chang-Ryeol Lee and Kuk-Jin Yoon. Exploiting feature confidence for forward motion estimation. *arXiv preprint arXiv:1704.07145*, 2017.

[95] Seung Hyun Lee, Dae Ha Kim, and Byung Cheol Song. Self-supervised knowledge distillation using singular value decomposition. In *European Conference on Computer Vision (ECCV)*, pages 339–354. Springer, 2018.

[96] Thomas Lemaire and Simon Lacroix. Monocular-vision based slam using line segments. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 2791–2796. IEEE, 2007.

[97] José Lezama, Qiang Qiu, and Guillermo Sapiro. Not afraid of the dark: Nir-vis face recognition via cross-spectral hallucination and low-rank embedding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6628–6637, 2017.

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

[99] Hyon Lim, Jongwoo Lim, and H Jin Kim. Real-Time 6-DOF Monocular Visual SLAM in a Large-Scale Environment. In *IEEE Int. Conf. Robot. Autom.*, 2014.

[100] Hui Liu, Houshang Darabi, Pat Banerjee, and Jing Liu. Survey of Wireless Indoor Positioning Techniques and Systems. *IEEE Trans. Syst. Man, Cybern. C Appl. Rev.*, 37(6):1067–1080, 2007.

[101] David Lopez-Paz, Léon Bottou, Bernhard Schölkopf, and Vladimir Vapnik. Unifying distillation and privileged information. In *International Conference on Learning Representations (ICLR)*, 2016.

[102] Steven Lovegrove, Andrew J Davison, and Javier Ibanez-Guzmán. Accurate visual odometry from a rear parking camera. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 788–793. IEEE, 2011.

[103] David G Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.*, 60(2):91–110, 2004.

[104] Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 375–383, 2017.

[105] Yan Lu and Dezhen Song. Robustness to lighting variations: An rgb-d indoor visual odometry using line segments. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 688–694. IEEE, 2015.

[106] Bruce D. Lucas and Takeo Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision, 1981.

[107] Jian-Hao Luo, Jianxin Wu, and Weiyao Lin. Thinet: A filter level pruning method for deep neural network compression. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 5058–5066, 2017.

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

[109] Steve Mann, Jason Huang, Ryan Janzen, Raymond Lo, Valmiki Rampersad, Alexander Chen, and Taqveer Doha. Blind Navigation With a Wearable Range Camera and Vibrotactile Helmet. In *19th ACM Int. Conf. Multimed. - MM '11*, page 1325, New York, New York, USA, 2011. ACM Press.

[110] Iaroslav Melekhov, Juha Ylioinas, Juho Kannala, and Esa Rahtu. Relative Camera Pose Estimation Using Convolutional Neural Networks. In *arXiv:1702.01381*, 2017.

[111] Vikram Mohanty, Shubh Agrawal, Shaswat Datta, Arna Ghosh, Vishnu Dutt Sharma, and Debashish Chakravarty. DeepVO: A Deep Learning Approach for Monocular Visual Odometry. In *arXiv:1611.06069*, 2016.

[112] Dmitry Molchanov, Arsenii Ashukha, and Dmitry Vetrov. Variational Dropout Sparsifies Deep Neural Networks. In *Int. Conf. Mach. Learn.*, 2017.

[113] Tarek Mouats, Nabil Aouf, Lounis Chermak, and Mark A Richardson. Thermal stereo odometry for uavs. *IEEE Sensors Journal*, 15(11):6335–6347, 2015.

[114] E Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd. Real Time Localization and 3D Reconstruction. In *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pages 1–8, 2006.

[115] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd. Generic and Real-time Structure from Motion using Local Bundle Adjustment. *Image Vis. Comput.*, 27(8):1178–1193, 2009.

[116] E. Mouragnon, M. Lhuillier, Michel Dhome, Fabien Dekeyser, and P. Sayd. Generic and Real-Time Structure from Motion. In *Br. Mach. Vis. Conf.*, pages 64.1–64.10, 2007.

[117] Etienne Mouragnon, Maxime Lhuillier, Michel Dhome, Fabien Dekeyser, and Patrick Sayd. Monocular Vision Based SLAM for Mobile Robots. In *18th Int. Conf. Pattern Recognit.*, 2006.

[118] Peter Muller and Andreas Savakis. Flowdometry: An Optical Flow and Deep Learning Based Approach to Visual Odometry. In *IEEE Winter Conf. Appl. Comput. Vis.*, 2017.

[119] Raul Mur-Artal, J. M M Montiel, and Juan D. Tardos. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Transaction on Robotics*, 31(5):1147–1163, 2015.

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

[121] Oleg Naroditsky, Xun S Zhou, Jean Gallier, Stergios I Roumeliotis, and Kostas Daniilidis. Two efficient solutions for visual odometry using directional correspondence. *IEEE transactions on pattern analysis and machine intelligence*, 34(4):818–824, 2011.

[122] Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. Dtam: Dense tracking and mapping in real-time. In *2011 international conference on computer vision*, pages 2320–2327. IEEE, 2011.

[123] David Nister. An Efficient Solution to the Five-Point Relative Pose Problem. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(6):756–770, 2004.

[124] David Nistér, Oleg Naroditsky, and James Bergen. Visual Odometry. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 652–659, 2004.

[125] Ignacio Parra, Miguel Angel Sotelo, David Fernández Llorca, and Manuel Ocaña. Robust visual odometry for vehicle localization in urban environments. *Robotica*, 28(3):441–452, 2010.

[126] Thierry Peynot, James Underwood, and Steven Scheding. Towards reliable perception for unmanned ground vehicles in challenging conditions. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1170–1176. IEEE, 2009.

[127] Antonio Polino, Razvan Pascanu, and Dan Alistarh. Model compression via distillation and quantization. In *International Conference on Learning Representations (ICLR)*, 2018.

[128] Albert Pumarola, Alexander Vakhitov, Antonio Agudo, Alberto Sanfeliu, and Francese Moreno-Noguer. Pl-slam: Real-time monocular visual slam with points and lines. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 4503–4508. IEEE, 2017.

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

[130] Paolo Bellezza Quater, Francesco Grimaccia, Sonia Leva, Marco Mussetta, and Mohammadreza Aghaei. Light unmanned aerial vehicles (uavs) for cooperative inspection of pv plants. *IEEE Journal of Photovoltaics*, 4(4):1107–1113, 2014.

[131] Roberto Rigamonti, Amos Sironi, Vincent Lepetit, and Pascal Fua. Learning separable filters. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2754–2761, 2013.

[132] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. FitNets: Hints for Thin Deep Nets. In *Int. Conf. Learn. Represent.*, 2015.

[133] Edward Rosten and Tom Drummond. Machine Learning for High-Speed Corner Detection. In *Eur. Conf. Comput. Vis.*, volume 1, pages 430–443, 2006.

[134] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An Efficient Alternative to SIFT or SURF. In *IEEE Int. Conf. Comput. Vis.*, pages 2564–2571, 2011.

[135] Muhamad Risqi U Saputra, Pedro PB de Gusmao, Yasin Almalioglu, Andrew Markham, and Niki Trigoni. Distilling knowledge from a deep pose regressor network. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 263–272, 2019.

[136] Muhamad Risqi U. Saputra, Pedro P. B. de Gusmao, Sen Wang, Andrew Markham, and Niki Trigoni. Learning monocular visual odometry through geometry-aware curriculum learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[137] Muhamad Risqi U Saputra, Andrew Markham, and Niki Trigoni. Visual slam and structure from motion in dynamic environments: A survey. *ACM Computing Surveys (CSUR)*, 51(2):37, 2018.

[138] Davide Scaramuzza. 1-Point-RANSAC Structure from Motion for Vehicle-Mounted Cameras by Exploiting Non-holonomic Constraints. *Int. J. Comput. Vis.*, 95(1):74–85, 2011.

[139] Davide Scaramuzza, Friedrich Fraundorfer, and Roland Siegwart. Real-Time Monocular Visual Odometry for On-Road Vehicles with 1-Point RANSAC. In *IEEE Int. Conf. Robot. Autom.*, pages 4293–4299, 2009.

[140] Davide Scaramuzza and Roland Siegwart. Appearance-guided monocular omnidirectional visual odometry for outdoor ground vehicles. *IEEE transactions on robotics*, 24(5):1015–1026, 2008.

[141] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-Motion Revisited. In *IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 4104–4113, 2016.

[142] Geraldo Silveira, Ezio Malis, and Patrick Rives. An efficient direct approach to visual slam. *IEEE transactions on robotics*, 24(5):969–979, 2008.

[143] Isaac Skog, Peter Handel, John-Olof Nilsson, and Jouni Rantakokko. Zero-velocity detection—an algorithm evaluation. *IEEE transactions on biomedical engineering*, 57(11):2657–2666, 2010.

[144] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[145] Hauke Strasdat, José MM Montiel, and Andrew J Davison. Visual slam: why filter? *Image and Vision Computing*, 30(2):65–77, 2012.

[146] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 573–580. IEEE, 2012.

[147] James S Supancic and Deva Ramanan. Self-paced learning for long-term tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2379–2386, 2013.

[148] Cheng Tai, Tong Xiao, Yi Zhang, Xiaogang Wang, and Weinan E. Convolutional neural networks with low-rank regularization. In *Int. Conf. Learn. Represent.*, 2016.

[149] J. Tang, J. Folkesson, and P. Jensfelt. Geometric correspondence network for camera motion estimation. *IEEE Robotics and Automation Letters*, 3(2):1010–1017, April 2018.

[150] Kevin Tang, Vignesh Ramanathan, Li Fei-Fei, and Daphne Koller. Shifting weights: Adapting object detectors from image to video. In *Advances in Neural Information Processing Systems*, pages 638–646, 2012.

[151] Ye Tang, Yu-Bin Yang, and Yang Gao. Self-paced dictionary learning for image classification. In *Proceedings of the 20th ACM international conference on Multimedia*, pages 833–836. ACM, 2012.

[152] Philip H. S. Torr and Andrew Zisserman. MLESAC: A New Robust Estimator with Application to Estimating Image Geometry. *Comput. Vis. Image Underst.*, 78(1):138–156, 2000.

[153] P.H.S. Torr. Geometric Motion Segmentation and Model Selection. *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.*, pages 1321–1340, 1998.

[154] P.H.S Torr and A Zisserman. Robust Parameterization and Computation of the Trifocal Tensor. *Image Vis. Comput.*, 15(8):591–605, 1997.

[155] P.H.S. Torr and Andrew Zisserman. Feature based Methods for Structure and Motion Estimation. In *Int. Work. Vis. Algorithms*, 1999.

[156] Vincent Vanhoucke, Andrew Senior, and Mark Z Mao. Improving the speed of neural networks on cpus. In *Proc. Deep Learning and Unsupervised Feature Learning NIPS Workshop*, volume 1, page 4. Citeseer, 2011.

[157] René Vidal, Yi Ma, Stefano Soatto, and Shankar Sastry. Two-view multibody structure from motion. *International Journal of Computer Vision*, 68(1):7–25, 2006.

[158] Sudheendra Vijayanarasimhan, Susanna Ricco, Cordelia Schmid, Rahul Sukthankar, and Katerina Fragkiadaki. SfM-Net: Learning of Structure and Motion from Video. In *arXiv:1704.07804*, 2017.

[159] Johan Wahlström, Isaac Skog, Fredrik Gustafsson, Andrew Markham, and Niki Trigoni. Zero-velocity detection – A Bayesian approach to adaptive thresholding. *IEEE Sensors Letters*, 3(6), 2019.

[160] Hui Wang, Hanbin Zhao, Xi Li, and Xu Tan. Progressive blockwise knowledge distillation for neural network acceleration. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2769–2775, 2018.

[161] Sen Wang, Ronald Clark, Hongkai Wen, and Niki Trigoni. DeepVO: Towards End-to-End Visual Odometry with Deep Recurrent Convolutional Neural Networks. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[162] Sen Wang, Ronald Clark, Hongkai Wen, and Niki Trigoni. End-to-end, sequence-to-sequence probabilistic visual odometry through deep neural networks. *The International Journal of Robotics Research (IJRR)*, pages 1–30, 2018.

[163] Wei Wang, Muhamad Risqi U. Saputra, Peijun Zhao, Pedro Gusmao, Bo Yang, Changhao Chen, Andrew Markham, and Niki Trigoni. Deeppco: End-to-end point cloud odometry through deep parallel neural network. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[164] Jonas Witt and Uwe Weltin. Robust stereo visual odometry using iterative closest multiple lines. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4164–4171. IEEE, 2013.

[165] Changchang Wu. Towards Linear-time Incremental Structure from Motion. In *Int. Conf. 3D Vis.*, pages 127–134, 2013.

[166] Changchang Wu, Sameer Agarwal, Brian Curless, and Steven M. Seitz. Multicore Bundle Adjustment. In *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, number 1, pages 3057–3064, 2011.

[167] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, pages 802–810, 2015.

[168] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057, 2015.

[169] Shuochao Yao, Yiran Zhao, Aston Zhang, Lu Su, and Tarek Abdelzaher. DeepIoT: Compressing Deep Neural Network Structures for Sensing Systems with a Compressor-Critic Framework. In *15th ACM Conf. Embed. Networked Sens. Syst.*, number 17, 2017.

[170] Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4133–4141, 2017.

[171] Georges Younes, Daniel Asmar, and Elie Shammas. A Survey on Non-filter-based Monocular Visual SLAM Systems. In *arXiv:1607.00470*, 2016.

[172] Zhang Yu, Shen Lincheng, Zhou Dianle, Zhang Daibing, and Yan Chengping. Camera calibration of thermal-infrared stereo vision system. In *2013 Fourth International Conference on Intelligent Systems Design and Engineering Applications*, pages 197–201. IEEE, 2013.

[173] Huangying Zhan, Ravi Garg, Chamara Saroj Weerasekera, Kejie Li, Harsh Agarwal, and Ian Reid. Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 340–349, 2018.

[174] Ji Zhang and Sanjiv Singh. Visual-lidar odometry and mapping: Low-drift, robust, and fast. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2174–2181. IEEE, 2015.

[175] Zhengyou Zhang. A Flexible New Technique for Camera Calibration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(11):1330–1334, 2002.

[176] Huizhong Zhou, Benjamin Ummenhofer, and Thomas Brox. Deeptam: Deep tracking and mapping. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 822–838, 2018.

[177] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. Unsupervised Learning of Depth and Ego-Motion from Video. In *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2017.