UNIVERSITY OF CALIFORNIA SAN DIEGO

An Investigation of Traditional and Deep Structure from Motion Methods on a Diverse Dataset

A Thesis submitted in partial satisfaction of the
requirements for the degree
Master of Science

in

Computer Science

by

Dewal Gupta

Committee in charge:

Professor Manmohan Chandraker, Chair
Professor David Kriegman
Professor Ravi Ramamoorthi

2019

The Thesis of Dewal Gupta is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California San Diego

2019

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to thank Professor Manmohan Chandraker for his guidance and support throughout the development of this work.

ABSTRACT OF THE THESIS

An Investigation of Traditional and Deep Structure from Motion Methods on a Diverse Dataset

by

Dewal Gupta

Master of Science in Computer Science

University of California San Diego, 2019

Professor Manmohan Chandraker, Chair

Visual odometry has become an important tool given the new popularity of mobile robotics. Camera pose estimation is a key part of visual odometry and has traditionally been computed by hand-engineered algorithms. Given the recent explosion of deep learning, learned networks are making headway in replacing such algorithms. They have proven to be very successful, and in certain instances are already better traditional methods like ORB-SLAM. However, such methods are usually trained and tested on a relatively homogeneous dataset that contains little variation in lighting, motion, or other cues that can prove challenging. In this work, a novel dataset is curated

to specifically introduce more realistic and diverse trajectories. This dataset is a combination of three other datasets: Advio, Aqualoc and Euroc MAV. Using this dataset, it is shown how traditional methods like ORB-SLAM and DSO still out-perform learned, structure from motion methods, and certain failure cases are identified as a ways to improve the learned algorithms. Such a dataset can be used in the future to truly ascertain the strength of a network's performance.

Chapter 1

Introduction

Accurate visual odometry has historically been a key challenge in computer vision due to its useful applications in robotics. More recently, mobile robotics have become extremely popular, finding applications in areas like drones, autonomous vehicles, augmented reality and more. They are useful for exploring areas that humans physically cannot such as disaster areas or other planets like Mars. Robots in such environments must be able to understand their position relative to their environment for navigation without any human input. An integral part of these robotic systems is the ability to locate themselves using only cameras providing images of their environment. Though traditional technologies such as GPS, radar, and lidar can aid in this pursuit, often times scenarios arise where these same technologies become unfeasible. This may be due to the fact that GPS signals either are not reachable, or the localization error would simply be too high. Inertial sensors may cause drift that over time becomes unacceptable and useless to the robot as well [PKK18]. Although visual odometry methods may not be as reliable as using traditional sensors, another advantage comes down to cost. Cameras are relatively cheap and

simple to set up whereas accurate sensors such as LIDAR or RADAR may be cost prohibitive. Even if they were to be used, often times they can introduce other problems such as rolling shutter and synchronization issues leading to more challenging problems. However, visual odometry can sometimes be combined with these traditional systems to provide the best of both worlds - the reliability and robustness of sensors like GPS or IMU, along with the localization accuracy of visual odometry systems.

Convolutional neural networks (CNNs) have revolutionized computer vision in the past decade and have enjoyed remarkable success in problems such as object detection, image classification, image depth estimation, and more. In the case of visual odometry, however, such models are only beginning to be built to deal with the end to end problem of tracking ego-motion. CNNs have performed extremely well at identifying interesting features in images, matching and find correspondences, estimating depth and even identifying dense, per-pixel motion between two frames - all important parts of successful, traditional visual odometry algorithms. The potential for deep learned methods to eventually out-perform non-learned methods is very real, and certain works have already shown significant progress towards such goals [ZBSL17] [VRS+17] [DMR18]. However, one class of models has become quite popular: unsupervised and monocular pose estimation. Though not real time as traditional visual odometry algorithms are, these models claim to successfully learn camera pose estimation using a monocular set up in an unsupervised manner. These methods hold a lot of potential due to their lack of reliance on extra sensory data, ease of setup, and independence from expensive ground truth data.

1.1 Project Goals

Too many learned models are tested on homogeneous datasets that lack variation in lighting, exposure and other photometric cues. As a result, it is very likely that such models over-fit with respect to the aforementioned cues, failing to generalize if any property was found to be different than the training set. In this work, a highly diverse dataset is carefully curated, and used to train and test deep learning methods. Their performance is evaluated under the novel context of this dataset, and the weaknesses and strengths of such models are empirically investigated. The diverse dataset contains different types of motions, lighting cues, geometries and textures, and violates assumptions such as the lambertian or rigid motion assertions to different degrees. Using such a dataset, the goal of this work is to present a solution to training more generalizable and adaptable deep learning methods.

Chapter 2

Related Work

2.1 Traditional Methods

2.1.1 Direct Methods

Traditional methods involve non-learned methods using hand-engineered techniques to conduct visual odometry. Of these methods, there are two major categories: direct and indirect. Direct methods attempt to directly use sensor values as measurements into a probabilistic model. This model then aims to predict the camera motion from these measurements, and as a result tries to optimize the photometric error. Jin et al. [JFS03] are one of the few to propose a direct method that is sparse as well. Their method relies on an extended Kalman filter and explicitly models illumination changes in a sequence of frames. Engel et al. [EKC18] built DSO by using a non-linear optimization framework instead of the Kalman filter. They also explicitly account for geometric and photometric calibrations necessary for optimal results in a direct

method. DTAM (Dense Tracking And Mapping) [NLD11] and LSD-SLAM (Large-Scale Direct SLAM) [ESC14] also use direct formulations, modeling camera motion off of images with no preprocessing. However, they are both dense methods, using geometry priors to build a dense 3D map of the environment, whereas DSO is a sparse method treating geometric features, such as keypoints, independent from each other with no notion of neighborhoods. DTAM builds off its predecessor PTAM (Parallel Tracking And Mapping) [KM07], an indrect method, which calculates camera motion and maps 3D features in a parallel fashion. DTAM, on the other hand, is a dense formulation and uses the 3D surface map to help calculate the camera trajectory as well. However, this is an expensive process and requires a GPU to run in real time. LSD-SLAM helps circumvent this requirement by simplifying the formulations used by DTAM.

2.1.2 Indirect Methods

Indirect methods use a preprocessing step and extract features from the frame before trying to build the model. As a result, these models optimize geometric error, relying on correspondences of key points or flow vectors. These types of models are generally more robust to different camera models as factors like auto exposure, gamma correction, vignetting. One of the most exemplar indirect models is PTAM, as previously mentioned, which works by parallelizing 3D mapping and tracking. It uses FAST corners that are matched making it useful for tracking. Strasdat et al. confront the challenges of monocular SLAM system by utilizing optical flow estimation and FAST feature mapping. Another method is ORB-SLAM [MAMT15] which utilizes ORB features to build its map and track the camera motion. It uses sparse features and can run in real time without the use of a GPU. Another method, SVO (Semi-direct Visual Odometry) [FPS14], uses

both direct and indirect formulations to its advantage. Although, no loop closure is performed, it uses a direct formulation for most frames allowing the method to skip expensive feature extraction. It does extract features to initialize novel 3D keypoints but only optimizes photometric error for all other frames.

2.2 Structure from Motion

Structure from motion is has long been a cornerstone in computer vision research in which models must elucidate the geometric and 3D properties of the scene given frames from different poses focused on the scenes. Structure from motion relies on accurate keypoint detection and matching, which are then geometrically verified by estimating the best fundamental matrix to describe the motion. From there the scenes are reconstructed and bundle optimization is iteratively applied to minimize any drift errors that would otherwise compound over time [SF16]. However, such methods rely on accurate correspondences which causes difficulties in sequences containing texture-less surfaces, occlusions, non-rigid motion, or other similar attributes. In the recent past, deep learning has attempted to mitigate such affects by focusing on parts of the structure from motion pipeline. Posenet [KGC15] is one such model as it attempts to directly regress the camera pose from a given image. The model works by using a deep convolutional neural network to build features in a high dimensional space using a standard VGG [SZ14] or Resnet [HZRS16] model. The relationship between these features and the camera pose is treated as a regression problem. However, many issues with absolute pose regression have been pointed out. Sattler et al. [SZPLT19] have shown how 3D structure and geometry based methods consistently outperform

absolute pose regression networks. They also show that regressing poses from image frames is more similar to image retrieval algorithms than pose estimating algorithms.

2.3 Unsupervised Models

More recently, unsupervised models have come up as way to avoid the building expensive ground truth data. Garg et al. [GVCR16] used an auto encoder architecture which utilized two stereo calibrated frames with known camera motion to build intermediate depth maps. These depth maps were then used to inverse warp the other frame back to the original frame. The photometric loss was calculated between the warped frame and true frame which then essentially served as a ground truth signal. However, Godard et al. [GMB17] showed that the photometric loss on reconstructed images is not an ideal supervision signal, and proposed instead an alternate approach to learning single-view depth. This approach made use of geometric constraints between left-right stereo pair images and introduced a novel left-right consistency loss to improve the quality of depth maps. However, both these methods relied on stereo image pairs with known camera motions.

Zhou et al. [ZBSL17] and Vijayanarasimha et al. [VRS$^+$17] borrow from structure from motion by learning to solve for depth and motion together in a monocular setting. Both use scene geometries and deep networks to build depth maps from single images in an unsupervised manner. These depth maps are used to inverse warp one frame into another, and a photometric loss between the frames can help the network learn. As a result, the networks require no ground truth, although Sfm-Net can optionally make use of it if it exists. One of the biggest challenges

in these methods is handling non-rigid movement in the sequences, as it can heavily impact the photometric loss and stop the network from learning. Zhou et al. do not explicitly handle non-rigid motion and instead use a separate explainability mask, also learned by their network, to account for it. However, Vijayanarasimhan et al. aim to learn depth, object masks and flow to explicitly model in frame movement. Mahjourian et al. [MWA18] build a network that instead utilizes a 3D loss function that make use of the geometry between adjacent frames. The authors of GeoNet [YS18] go further by introducing a residual flow learning module and enforcing geometric constraints as a type of consistency check for their supervision signal.

In most works, KITTI [GLSU13] or Cityscapes [COR+16] is the main dataset in which the models are trained and tested. This is problematic as the camera calibration for these datasets do not consider factors like gamma-correction, white balancing, or vignetting. Due to the fact that direct methods do not make use of any intermediary image features, they are most vulnerable to these uncalibrated parameters [SLZ+19]. The models, unlike non-learned algorithms such as DSO, do not explicitly account for these factors either. In this work, a more diverse dataset is used to highlight these issues, and show case major failure cases for the above mentioned models.

Chapter 3

Methods Employed

The diverse dataset was formed by utilizing existing datasets from various sources and combining them in a meaningful manner to create a dataset that provided a large variation in lighting cues, motion, resolution, frame rate, recording device, and other factors for the ideal training of deep learned models. The goal is to build a large enough dataset for it to become a standard in training and testing models. The dataset is composed of three different datasets: Advio [CSRK18], Aqualoc [FMT+18], and Euroc MAV [BNG+16] dataset. All three provide ground truth pose data measured by IMU sensors. Though Aqualoc does not provide stereo images, the other two datasets do.

3.1 Diverse Dataset

The Advio dataset contains a trajectory captured by an iPhone, recording at roughly 60 frames per second in a high resolution. The dataset also contains other calibrated measurements

recorded by the iPhone including accelerometer, gyroscope, magnetometer, and altimeter measurements. The same trajectories were also captured with a Google Tango using a fish-eye lens which also generated point clouds. Lastly, their set-up utilized a Google Pixel device to capture another view of the same trajectories to provide stereo images.

The original dataset contains a total of 23 trajectories spanning scenes of a mall, an office and an outdoor, urban environment. For the purposes of this project, the outdoor portion of the dataset was discarded. This is simply because there are only two outdoor sequences present in the entire dataset, and it is of scenes that are remarkably dissimilar to the rest of the dataset which is primarily indoors.

The Euroc MAV dataset is a relatively well known dataset used by many other researchers for different tasks. It consists of stereo frames obtained from an aerial drone (a micro aerial vehicle) along with ground truth IMU measurements and laser scans for true 3D structures and tracking. The authors run this set up through 11 different trajectories, all of which were utilized for this work. Five of the trajectories are located in the same room called the machine hall, where another six trajectories take place in two different vicon rooms which are essentially closed rooms with artificial features put up.

Lastly, the Aqualoc dataset was collected underwater using a robot and a monochromatic camera. The seven sequences vary in length but are mostly contain similar lighting and motion cues. As a result, only four sequences, chosen at random, were used from this dataset. Since a single sequence is decently long with around 2,000 to 5,000 frames, the inclusion of four sequences was justified as sufficient. The aqualoc dataset also provides ground truth pose that was generated by the authors using COLMAP [SF16]. There are also barometric measurements

provided.

3.2 Network Architectures

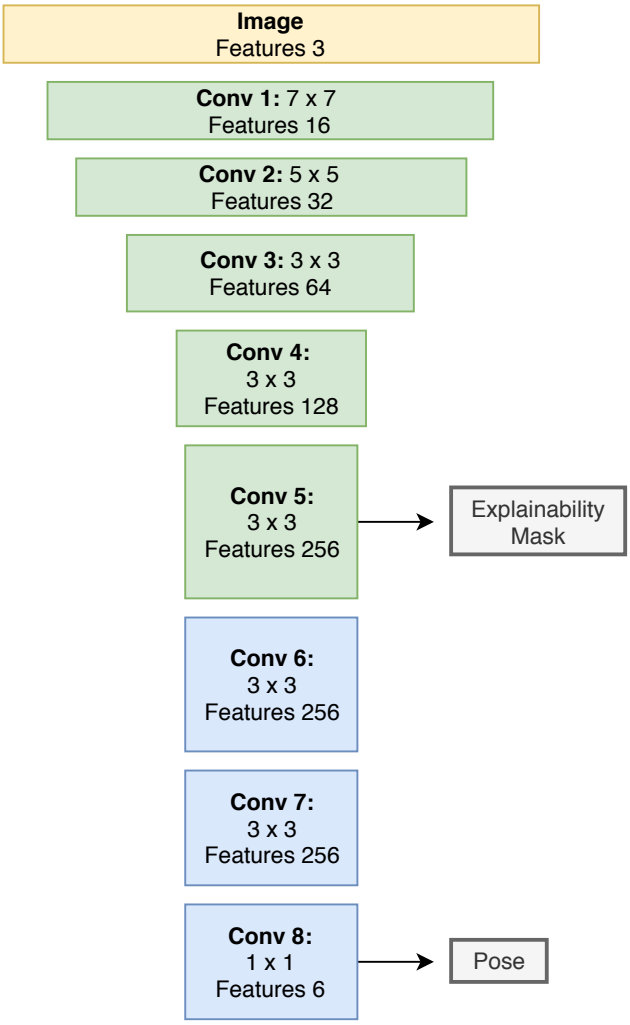


Figure 3.1: Network architecture for the pose generating network in Sfm-Learner.

The PoseNet architecture repurposes the GoogLeNet [SLJ+14] design that is slightly modified from the original by replacing all softmax classifiers with affine regressors. Also, another fully connected layer is put before the last layer with 2048 features. The authors use this as a local

feature vector that is investigated further in their experiments. The images are also scaled down so that the smallest dimension is 256 and then a crop of 224 by 224 pixels is taken at random during training. During the testing, the crop is still taken but it is always the center crop.

For Sfm-Learner, the network is detailed in Figure 3.1. The figure only shows the network architecture responsible for generating the pose. The authors use a different network, built off of DispNet [MIH+16]. The network used for pose is simpler than the one used by PoseNet. It consists of 8 convolution layers using 3x3 kernels except for the first two layers which use a 7x7 and 5x5 kernel respectively. The features after convolution 5 are used to generate the explainability masks by using up-convolutions. The same features are also further convolved three more times to generate the pose.

Chapter 4

Experiments and Results

4.1 ORB and DSO

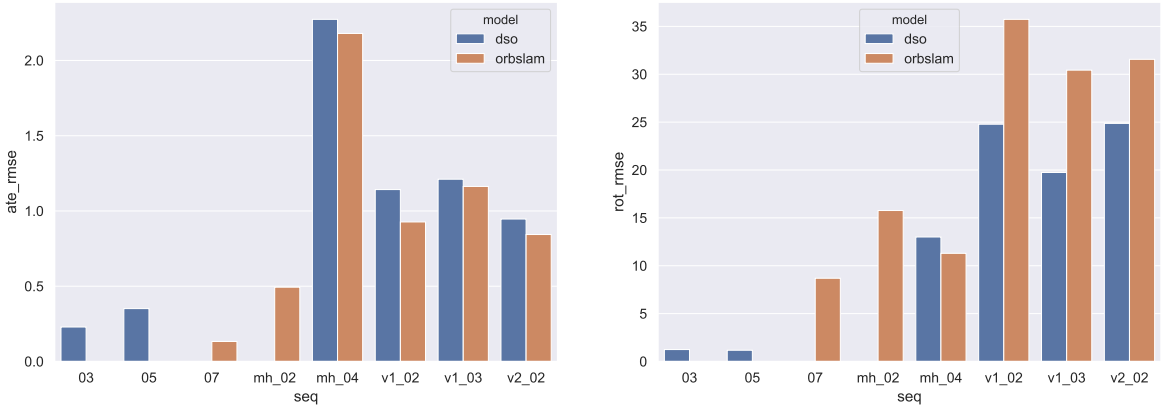


Figure 4.1: ORB-SLAM and DSO RMSE ATE and RE on all sequences they could be run on.

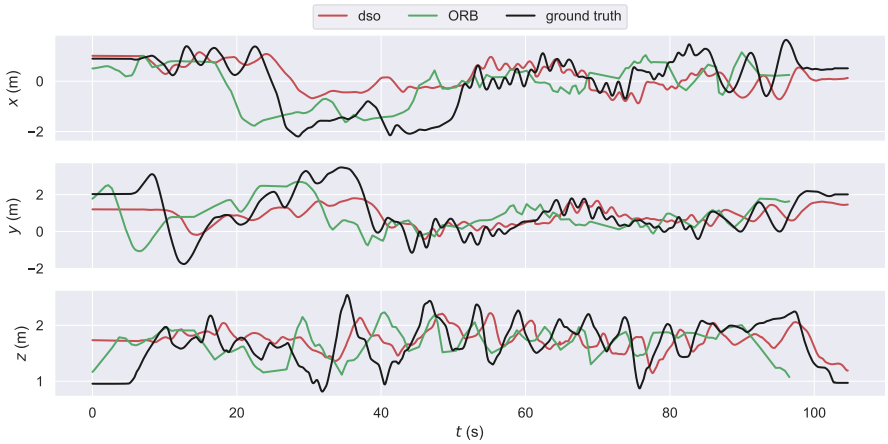
All sequences were all ran through the classic ORB-SLAM 2 [MAMT15] and DSO [EKC18] algorithms which provided interesting results. Both algorithms are simple to run, and require no learned or trained parameters. As a result they perform relatively well despite

lighting, motion and other key differences in the sequences. None of the image sequences were photometrically calibrated which is well known to affect the accuracy and robustness of both ORB-SLAM and DSO [YWGC18]. Sample trajectories are shown in Figure 4.2.

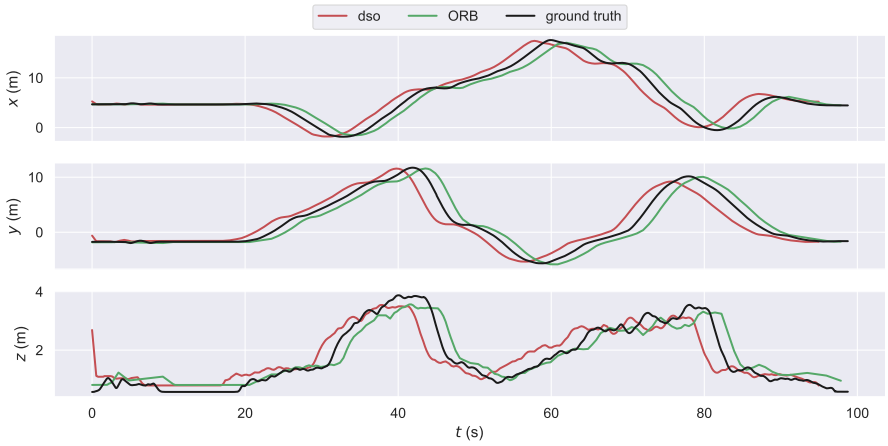
However, these algorithms are not without weaknesses themselves, and one of the major weaknesses of both algorithms is the case of pure rotational motion. Though both algorithms work differently, one being a direct formulation and the other indirect, they both fall victim to feature starvation introduced by pure rotational motion [LAD18]. Both algorithms rely on mapping keypoints into space in a sparse way which are tracked throughout the trajectory to determine the camera pose. However, in the case of pure rotational motion, the algorithm fails to triangulate new keypoints leading to keyframe starvation, and eventually tracking is lost.

Advio's method of capture exacerbates the problem as well. The iPhone camera provides a very narrow frame giving very little field of view, so it does not take much rotational motion to throw off the tracking algorithm. As a result, neither ORB SLAM nor DSO were able to be run on any of the Advio dataset sequences due to these issues. This is corroborated by the authors, as they were unable to obtain trajectories from DSO or ORB-SLAM either.

Another source of errors is the photometric calibration or lack thereof. DSO, being a direct method, relies on brightness constancy assumption which may or may not hold throughout these sequences. This is another reason the Advio dataset proves to be challenging as the lighting is not as consistent as the mall scenes especially contain a lot of non-lambertian surfaces that reflect light into the camera. There is also a lot of glass which can prove to be challenging as well. Lastly, the dataset contains sequences which have varying levels of people and non-rigid motion through the scenes which are known to also affect the performance of these algorithms.



(a) Euroc sequence V1 03



(b) Euroc sequence MH 04

Figure 4.2: Estimated trajectories for ORB-SLAM and DSO on two euroc sequences. The first three trajectories are from aqualoc while the rest are from Euroc. DSO nor ORB were able to be run on the Advio dataset.

4.2 PoseNet Experiments

Next, the PoseNet model was examined on the diverse dataset. PoseNet, in a sense, represents the simplest form of a learned model for determining camera pose due to the fact that it treats the problem as that of a simple regression. However, it is clearly evident that there are major issues with such a model. The model performs well on images it has already seen before

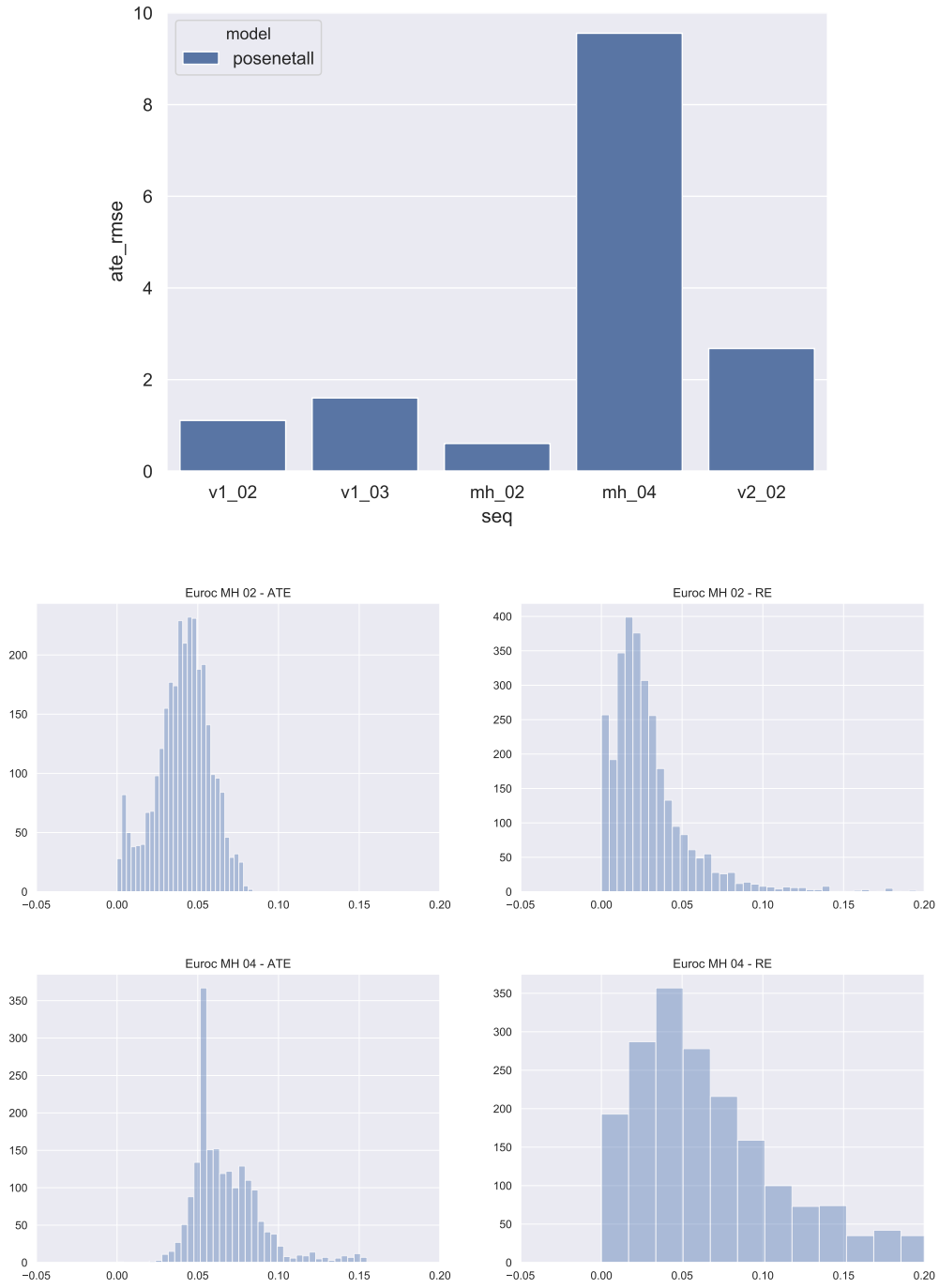
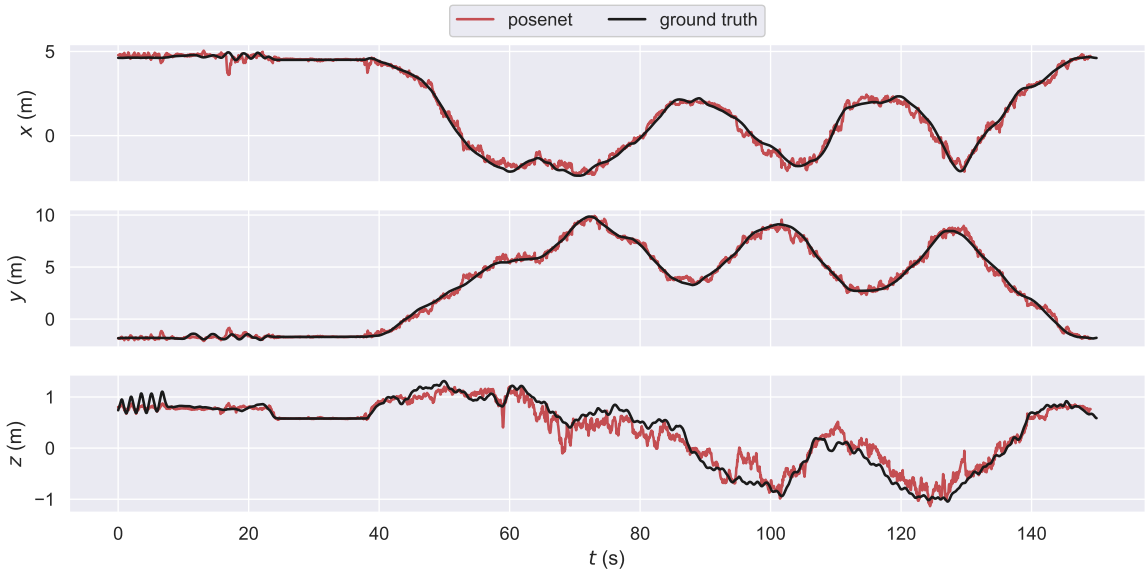
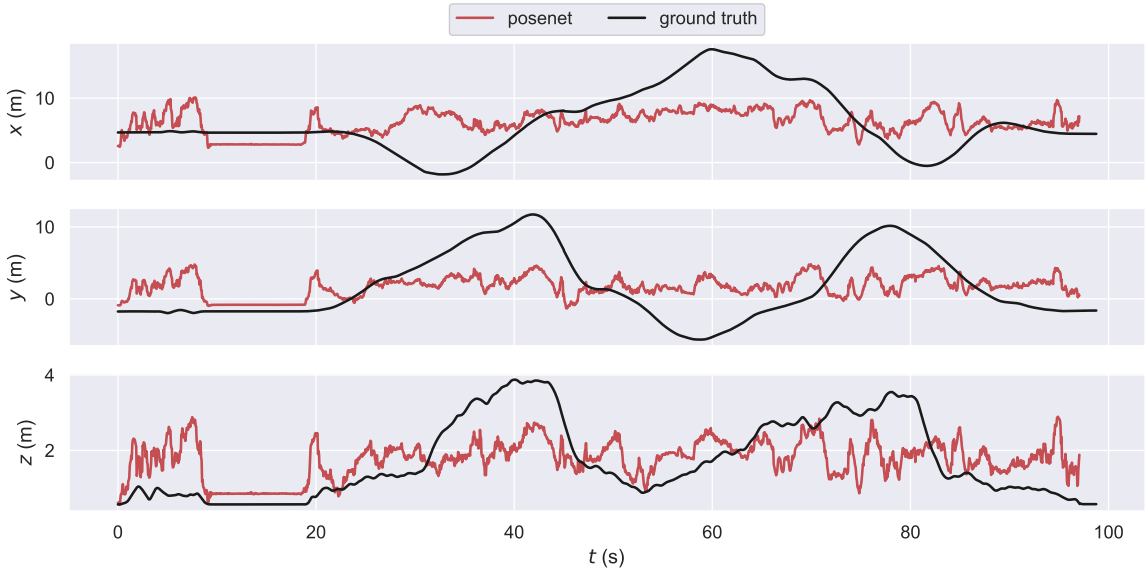


Figure 4.3: Top: EuroC RMSE ATE and RE on all sequences. Middle and Bottom rows: 5-frame window error distributions for ATE and RE for sequences MH 02 and MH 04. Note, the model was trained on sequence MH 02 but was tested on sequence MH 04.



(a) MH 02



(b) MH 04

Figure 4.4: Posenet trajectories on EuroC sequences MH 02 and MH 04. The 3D trjactory is decomposed into the x, y and z axis with red showing the predicted posenet path and the black representing the actual, ground truth trajectory.

or very similar images, but fails to provide evenly remotely accurate localization on images of the same scene from novel poses. This hints at the fact that the model is prone to over-fitting and fails to generalize to any other sequences. This is evident from the fact that PoseNet makes no use of any geometric information present in the sequences and simply focuses on regressing pose from the frame. Though it runs extremely quickly and is trainable with a large dataset, the results are coarse at best, with localization failing for frames that are outside what the network has already seen. This can be empirically seen in Figure 4.4 where PoseNet was trained on the Euroc sequence MH 02 and tested on MH 04. Both sequences take place in the machine hall room but are just different trajectories. As seen in the figure the model is able to over-fit on the training dataset really well except for minor fluctuations that are within reasonable bounds of error. However, examining the MH 04 sequence shows that the network was not able to learn anything useful as the trajectory does not even follow the general rotations of the ground truth. It also contains a lot more noise and perturbations that are beyond reasonable.

It is interesting to note that PoseNet, though simple, is most similar to direct methods as it does not extract any specific features from the frames and rather regresses the pose directly from the image. This makes it highly susceptible to changes in occlusion, lighting and motion conditions. To overcome such issues, the dataset would need to compensate as the original author's dataset did by providing the same images under different lighting or occlusion conditions. In this case, this was not possible nor is it reasonable to assume such data will exist, but it is a major reason why the network did not generalize as well as it could have.

By using the diverse dataset, it becomes apparent that Posenet fails to even generalize beyond the same dataset. It fails to handle any variations, even differing trajectories in the same

rooms within the EuroC dataset, highlighting a major flaw. The network is not designed in a way to generalize well because it fails to make any use of photometric or geometric cues. The regression model would most likely fail in simple cases such as when exposure times vary by a large margin. The model would also most likely fail when tested on novel views of the same scene. Even if the model has been trained on images of nearby poses, it fails to use any geometric cues to extrapolate the correct pose. This results in larger errors as the test trajectory moves farther and farther away from the training trajectory. Although training PoseNet on the diverse dataset does not improve the performance of the method, it is able to highlight major flaws in the design of the network.

4.3 Sfm-Learner Experiments

4.3.1 Aqualoc Experiments

The Aqualoc dataset is interesting because it is the most consistent and homogeneous of the diverse datasets. This is simply because all the sequences are underwater with very similar lighting conditions and features. Although, some sequences do tend to have more feature-friendly frames than others - something which ends up playing a big role in the performance of the models.

Sfm-Learner performs rather well on sequence 3 - the only sequence it does not see in the training dataset. The full trajectory has a RMSE of about roughly 1.0 although DSO outperforms it by a large factor. However, when the actual trajectory is visually examined, pose scaling and alignment, it is easily observed that Sfm-Learner's model does a good job approximating the actual trajectory with a majority of its error coming from the z-axis motion in the beginning and

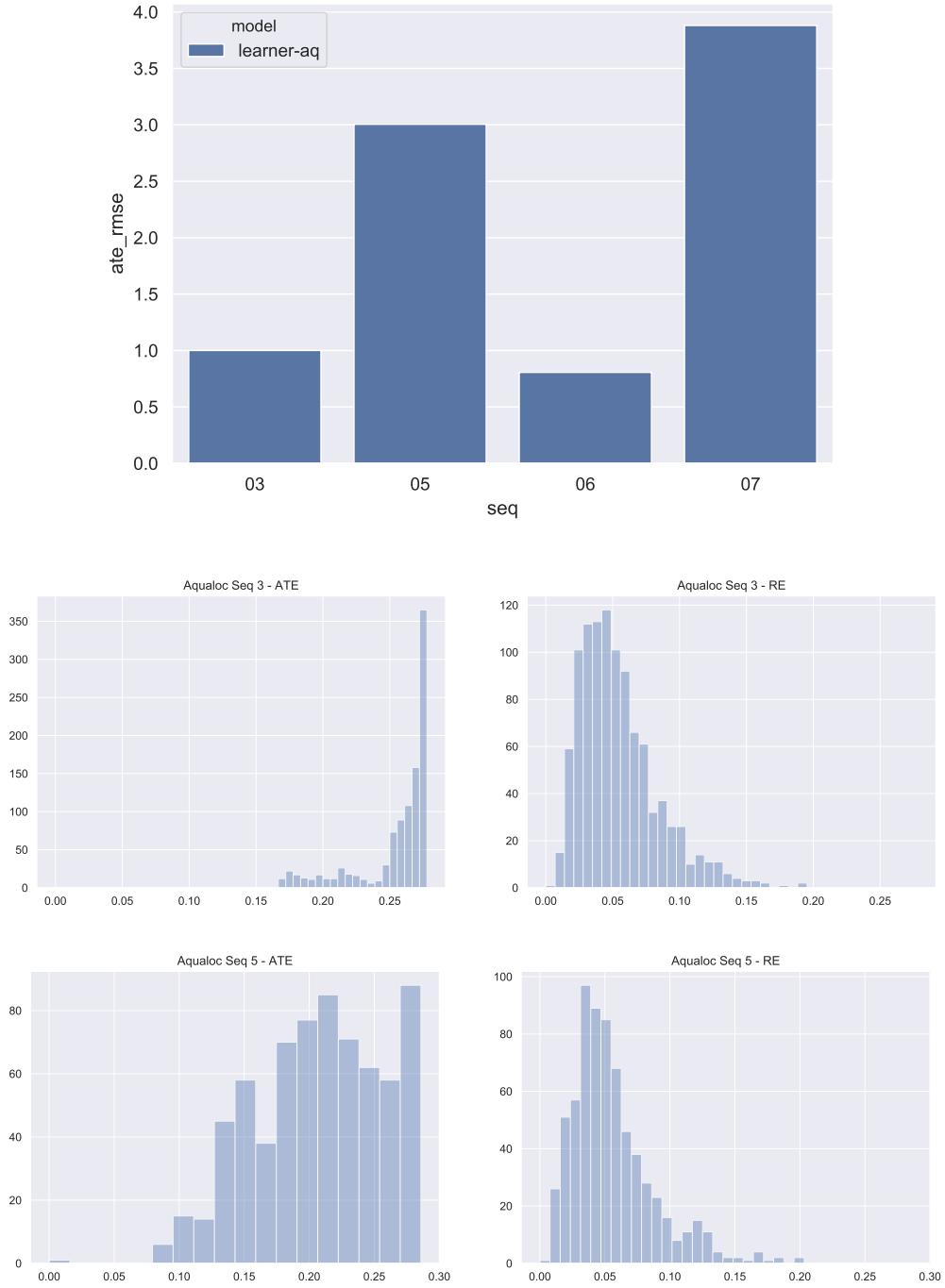
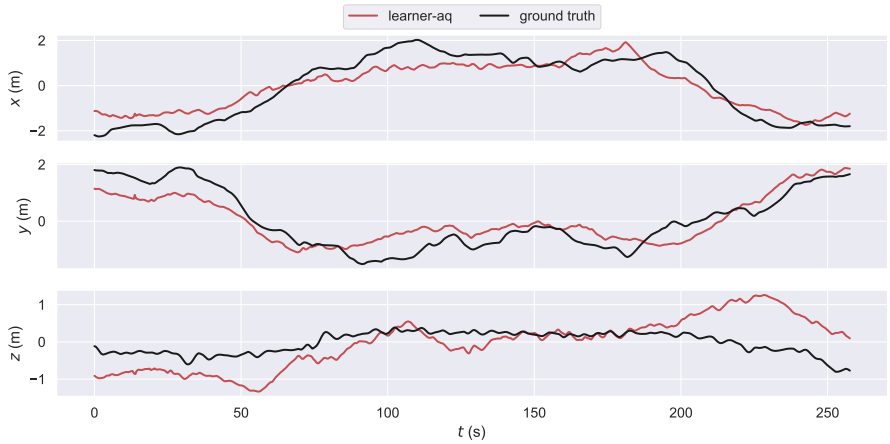
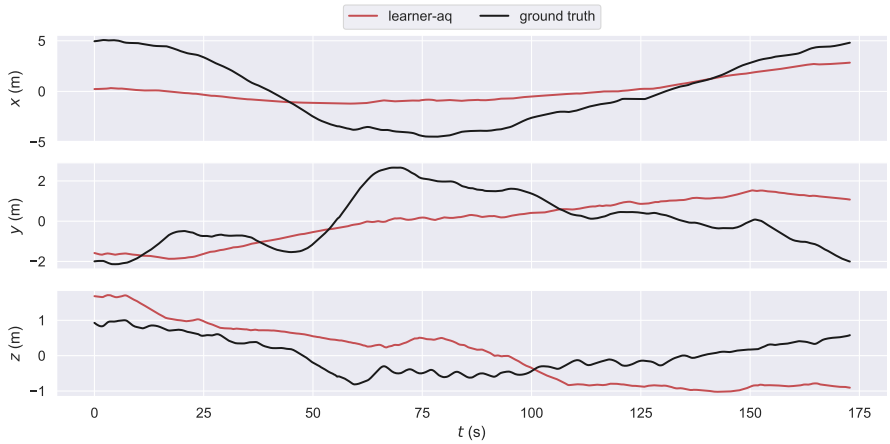


Figure 4.5: Top: Aqualoc RMSE ATE and RE on all sequences. Middle and Bottom rows: 5-frame window error distributions for ATE and RE for sequences 3 and 5. Note, the model was trained on sequence 3 but was tested on sequence 5.



(a) Sequence 03



(b) Sequence 05

Figure 4.6: Estimated trajectories for the learner-aq model on two sequences.

end of the trajectory.

However, this is in stark contrast with sequence 5 - a sequence that was actually used in the training for the model. Despite that, the trajectory learned by the model is very poor and does not appear to be comparable to the ground truth in any way. At best, it is a course approximation of the actual trajectory and lacks the significant rotations highlighting a possible lack of awareness of that rotational motion. This is an interesting failure case because the model performs relatively well on the test sequence indicating that the network has indeed learned useful information about

the data.

The failure case in this scenario is probably due to the fact that sequence 5 contains relatively few features, and the image is very homogeneous in terms of pixel intensity. This is an issue because of the gradient locality issue that the authors of Sfm-Learner explicitly mention and aim to overcome. While the network is learning, it is most likely that incorrect warps were calculated to be optimal despite being incorrect. This resulted in the network getting stuck in a local minima that results in average trajectories missing even the strongest rotation cues. Due to the nature of scene, it is highly unlikely that the author's solution of multi-scale and smoothness losses is able to overcome these issues.

Another issue present inherently in the dataset images is the natural vignetting due to the lighting of the ocean floor which is often not able to permeate throughout the entire frame. As a result, the same pixel locations could have very different pixel intensities which could negatively affect the warping. This is where DSO holds an advantage as it aims to explicitly model the photometric properties of the data. Despite its crucial role played in direct methods, Sfm-Learner has no such modules, learned or non-learned, which leave it to fall victim to poorly calibrated images.

In the case of Aqualoc, it is noted that the Sfm-Learner is capable of learning such trajectories, but can as easily fail on sequences that are slightly less than ideal. Despite using a more diverse dataset to train the model, it does not show any performance or accuracy benefits. Despite this, the dataset is able to identify areas of critical failure cases. Since the model does not make use of the ground truth, it shows the network's inability to learn from difficult sequences.

4.3.2 Advio Experiments

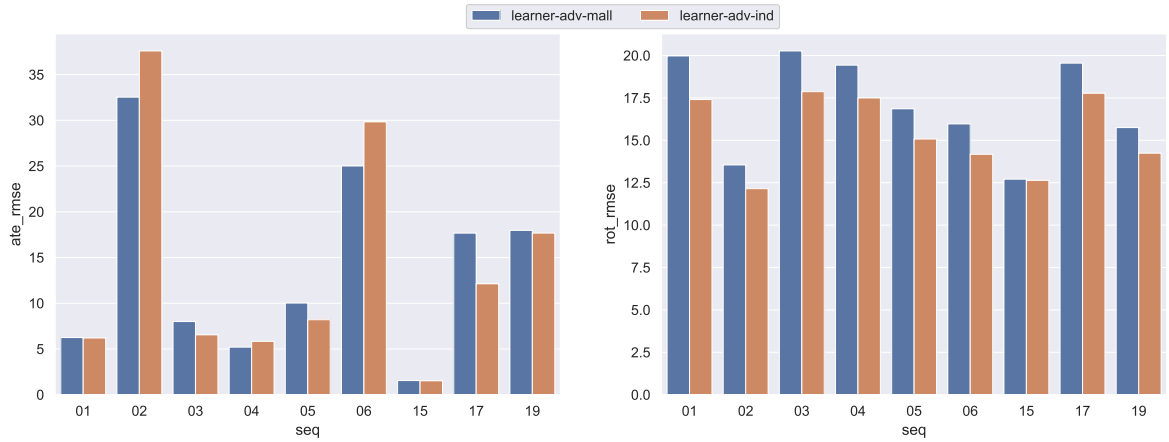


Figure 4.7: RMSE ATE and RE for Advio sequences.

The Advio dataset is the most natural dataset used in this project due to the fact that it is all shot on a hand-held iPhone camera. For the same reason, this makes this dataset also one of the most challenging. Another reason it is difficult to estimate accurate odometry on is due to the non-rigid motion present in the scenes of the mall. Two models were trained for this specific dataset using the Sfm-Learner network: advio-mall (learner-adv-mall) and advio-indoor (learner-adv-ind), each trained on their respective dataset splits. When examining their performance in Figure 4.7, some interesting patterns emerge. As expected, advio-mall performs better on mall scenes (scenes 01-06) generally speaking. The advio-indoor model performs either on par with the advio-mall model or better on the office sequences. However, it is interesting to see the advio-mall model, which was not trained on any office room scenes, still perform comparably well to the advio-indoor model on sequences 15 and 19. The opposite is also true as advio-indoor model performs as well as the advio-mall model in certain sequences, even beating

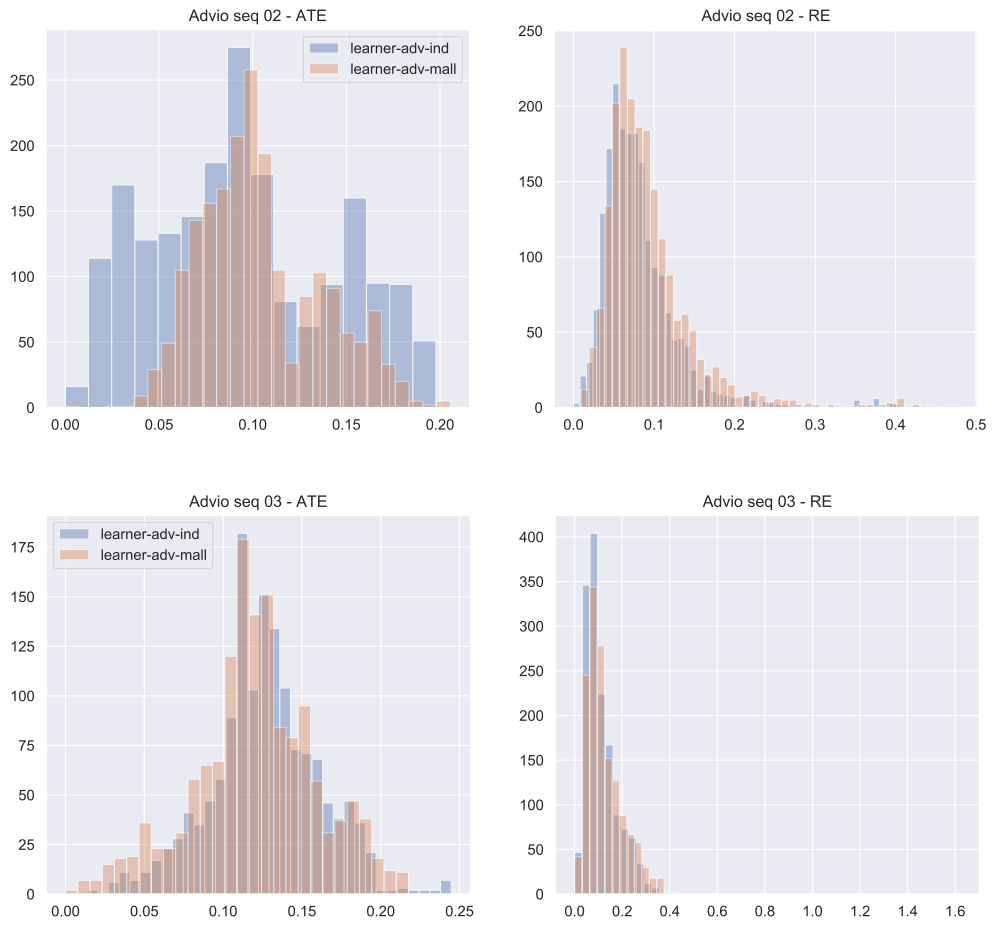
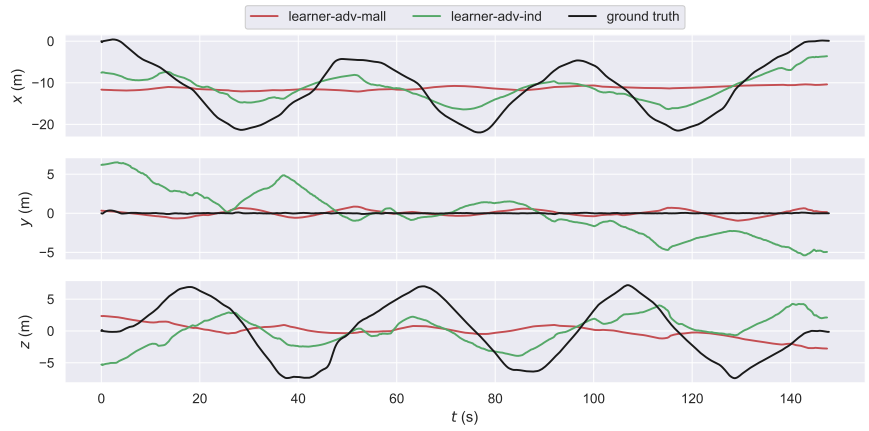


Figure 4.8: 5-frame window error distributions for ATE and RE for sequences 02 and 03.

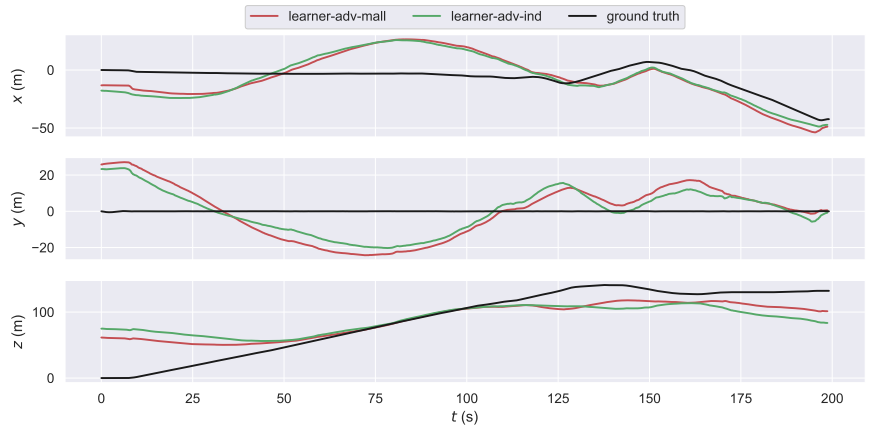
it in some. This is not expected as there is a larger variation between the mall and office scenes, and a model trained on just the mall scenes would be expected to outperform one trained on both. This suggests that there is some level of learning that is not happening where it should.

This is further exaggerated by the learned trajectories where it is highlighted how poorly some sequences are estimated. Sequence 2 is a good example and despite the advio-mall model being trained on this sequence, it is still valuable to understand the networks understanding. Despite there not being any strong rotation in the sequence except one major turn near the end, both models behave extremely similarly and estimate strong rotational motion. The error is a little

biased due to the fact that drift accumulates over time and the trajectory naturally varies between roughly 50 meters - one of the larger sequences distance wise. As a result, the distribution of 5-window errors can be examined instead.



(a) Sequence 03



(b) Sequence 05

Figure 4.9: Estimated trajectories for the learner-indoor and learner-mall models on two sequences.

In Figure 4.9 it can be seen how the error distributions for most frames are actually quite similar between the models. This is slightly worrying as it means that when training, the network was unable to further refine its estimate of the trajectory using the information in the sequence.

The fact that a model trained on this sequence performs so similarly to a model not trained on this sequence is a double edged sword. On one hand, it indicates the network is able to learn a decent representation of the scenes and motions and is able to apply it to unseen sequences with decent success. On the other hand, the poor estimation of the actual trajectory points to the fact that the network is failing to properly learn from this sequence. Due to the unsupervised nature of the model, this indicates that the supervision signal is not strong enough to provide reasonable estimates - even when that motion is extremely simple.

Another interesting sequence to examine is the mall sequence 03. In this sequence advio-indoor actually is able to model certain characteristics of the ground truth trajectory. The ground truth motion is in a circle where the x and y trajectory are sinusoidal and the advio-indoor model is able to model that motion relatively well. However, this learned behavior is not well reflected in any error metric due to the fact that the error shows up in the poorly estimated y-axis trajectory. Though the network is able to decently estimate x and z-axis motion, the y-axis motion is not a great fit resulting in a higher error.

These errors associated with both sequences 2, 3, and other sequences can be attributed to a host of factors but is most likely associated with the non-rigid scene motion in the scenes. Some of the mall scenes contain varying levels of humans walking through the video which at times are probably quite detrimental to the pose estimation. Despite the author's claims of having built an 'explainability mask' into the network to account for such motions, it does a poor job of identifying pedestrians and other sources of non-rigid motion resulting in poorly localized poses. Another reason this failure can be attributed to the non-rigid motion is due to the fact that sequences 2 and 6 have the most people in the frames as they are crowded mall scenes.

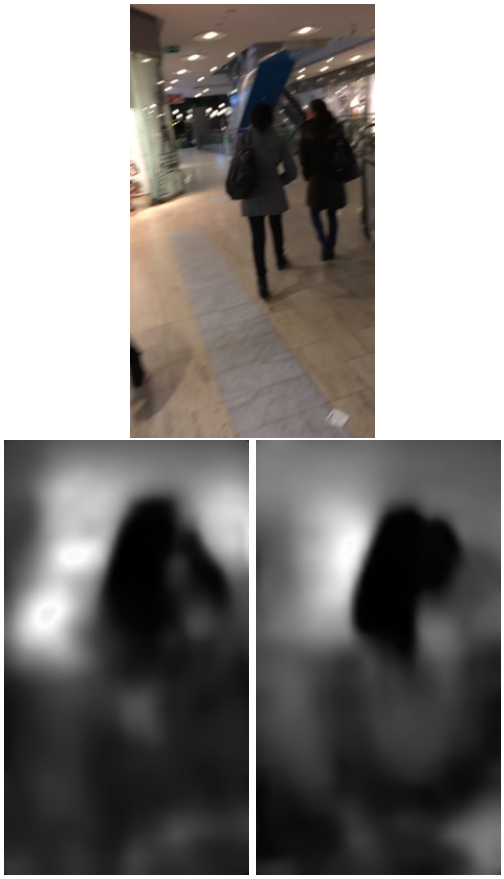


Figure 4.10: Depth maps for a random frame in sequence 3. Top row: original frame. Bottom row: learner-ind (left) and learner-mall (right). Both mistake the escalator for a much closer object, and the reflected lights for close objects.

Lastly, another major source of error in the mall and office scenes are the non-lambertian surfaces that are prevalent. These scenes are very well lit and having reflective tiles or large glass window show-cases make for a very challenging scene in which pixel intensities can easily change depending on the pose. This can be seen in the poorly generated depth maps that contain relatively little information about the true depth of the scene (Fig 4.10). These scenes violate one of the key assumptions of the model, but given that ORB-SLAM and DSO already fail on these sequences, the model does decently. However, this remains a weak area for all direct methods and leaves plenty room for improvement.

The diverse dataset does not help Sfm-Learner learn better representations when trained on the Advio dataset. However, it is known that the Advio dataset contains challenging trajectories very similar to real world data. The fact that Sfm-Learner fails to properly learn from this dataset shows that it was not designed to handle non-rigid motion despite the author's use of the explainability mask. Although the original authors explicitly state the model is not able to handle large occlusions and non-rigid motion, the original experiments are unable to ascertain the drop in performance given such challenges. Using this dataset, it becomes evident that the model cannot handle such challenges at all.

4.3.3 EuroC Experiments

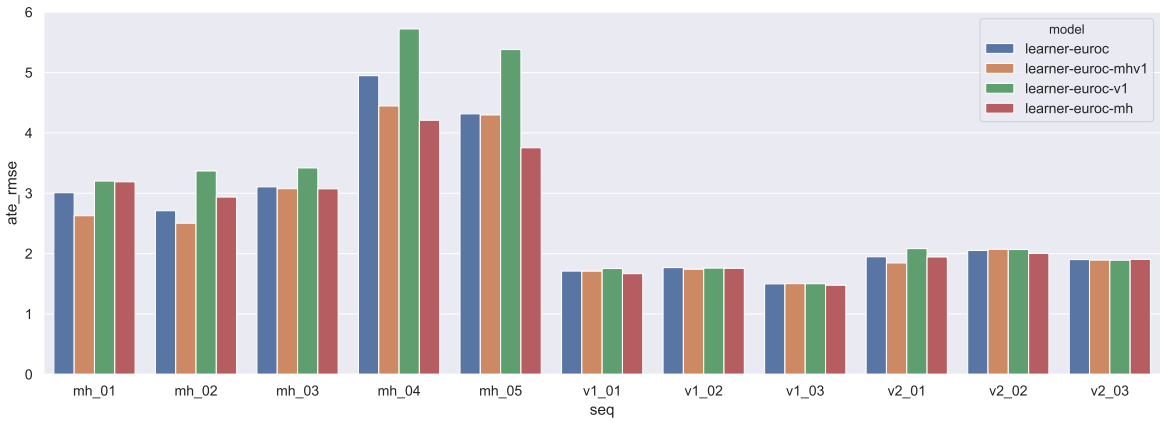


Figure 4.11: RMSE ATE and RE for euroc Sequences

The Euroc MAV dataset is one of the more well known odometry datasets and provides sequences of three different rooms from the perspective of a drone. These sequences provide unique motions in all three axis that are only capable by a drone. Four different models were trained on the four different splits that correspond to the different rooms. Two models were

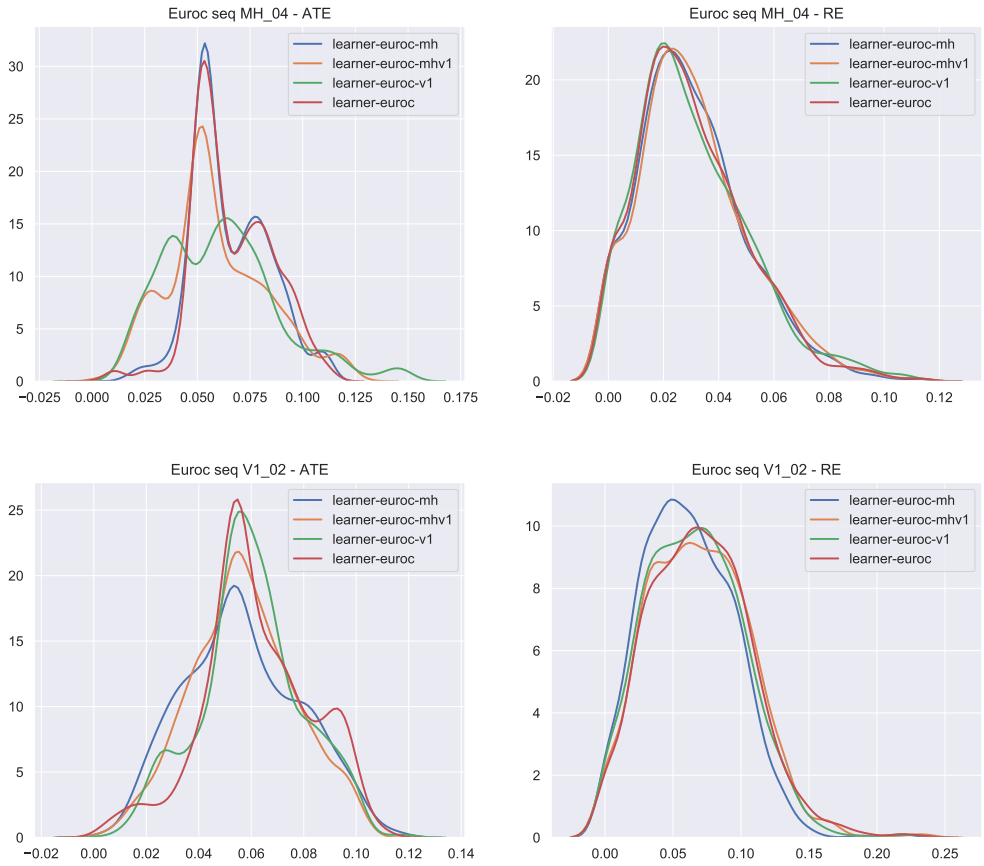
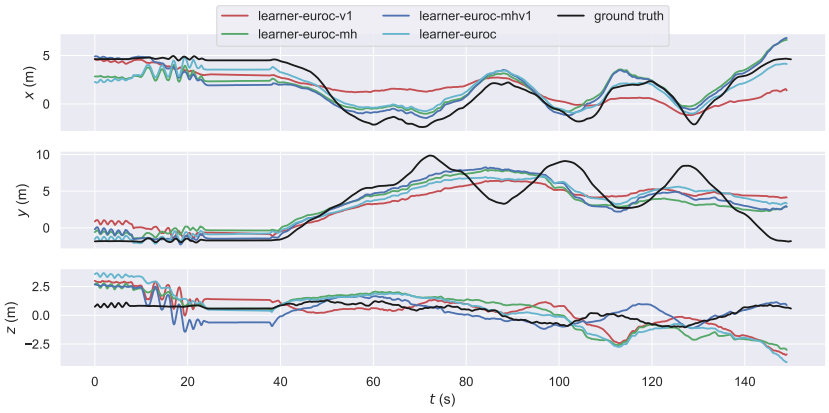


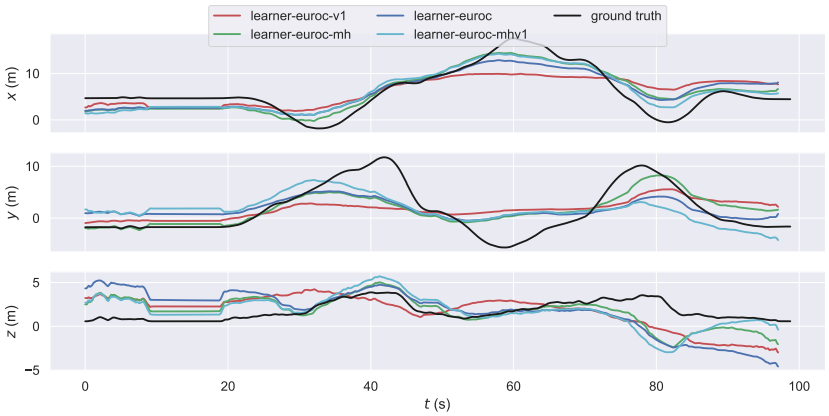
Figure 4.12: 5-frame window error distributions for ATE and RE for sequences MH 04 and V1 02.

trained on rooms MH (euroc-mh) and V1 (euroc-v1), another on the combination of the two (euroc-mh-v1) and lastly, a model was trained on all the rooms (euroc-all).

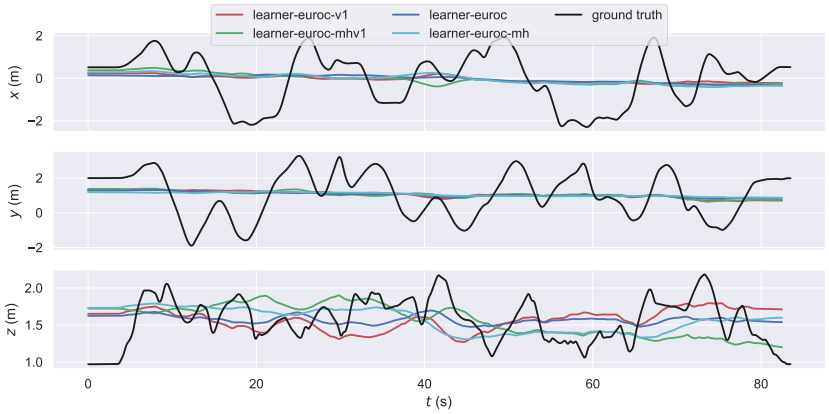
In the machine hall room (MH), the euroc-mh performs the best or on par with the other models. This is expected since it is only trained in the MH room and it would be expected to outperform other models in the room. Again, as expected the euroc-v1 model has the worst performance in the MH room since it has never seen those rooms before and was not trained on them. It is interesting to see the trajectories of all the models on sequences 02 and 04 in MH and how they all are able to capture the rotation and translations of the trajectory to varying degrees



(a) Sequence MH 02



(b) Sequence MH 04



(c) Sequence V1 02

Figure 4.13: Estimated trajectories for the learner-indoor and learner-mall models on two sequences.

of success Figure 4.13. Though there are cases of imperfect estimation, eg. in sequence MH 04, all networks do a relatively good job of staying close to the ground truth.

However, it is more interesting to examine the results of the V1 and V2 rooms. Although euroc-v1 failed to perform as well on MH rooms, the opposite was not found to be true. Oddly enough, the euroc-mh model performs just as well as the euroc-v1 and euroc-mh-v1 models on V1 and even V2 sequences. This is another scenario where the model is failing to learn useful pose and depth cues from the sequence. Also, these sequences don't suffer from the same catastrophic rotational motion or photometric issues that the Advio dataset contains, and decent results are produced by both ORB-SLAM and DSO algorithms. For the V1 and V2 rooms, it is the result of quick, short motion in all 3 dimensions that makes it difficult for the Sfm-Learner models to learn accurate depth and pose. Combined with this erratic motion is a problem with the scene itself. The fact that it is a relatively small, enclosed space with texture-less walls makes creating accurate depths a challenge as well. As a result, movement relative to the wall might not change a majority of the depth map causing an inability to learn accurate pose as well. This may be why the estimated pose is the same for all models whether or not they were trained on this sequence. The models are unable to learn usable depth maps forcing them to assume there is little or no motion in particular axes. This can be seen in Figures 4.14 and 4.15. The MH frame is handled fine by most of the learned models, except for learner-v1 which has never seen any of the MH sequences before. However, none of the networks are able to produce a cohesive depth map for the frame from V1.

Another reason might be the varying exposures in the sequence due to the windows in the room. At times the frames can be very dark as there are no windows in the frame. However,

whenever a window does pop up, it shines a lot of light and as a result the exposure changes making everything slightly brighter. As Sfm-Learner is a direct method working to optimize the photometric loss, this is likely to have a huge impact on the depth and pose estimating capabilities of the network.

As an aside, it is interesting to note that the error metrics currently employed often times hide such biases since window errors can overlook such problems. Even the results from the models on the MH 04 dataset look very similar (Fig 4.12) making it difficult to ascertain the true performance differences between euroc-v1 and euroc-mh.

Figure 4.14: Depth maps for a random frame in MH 04 sequence. Top row: original frame. Middle row: learner-mh (left) and learner-v1 (right). Bottom row: learner-mhv1 (left) and learner-euroc (right).

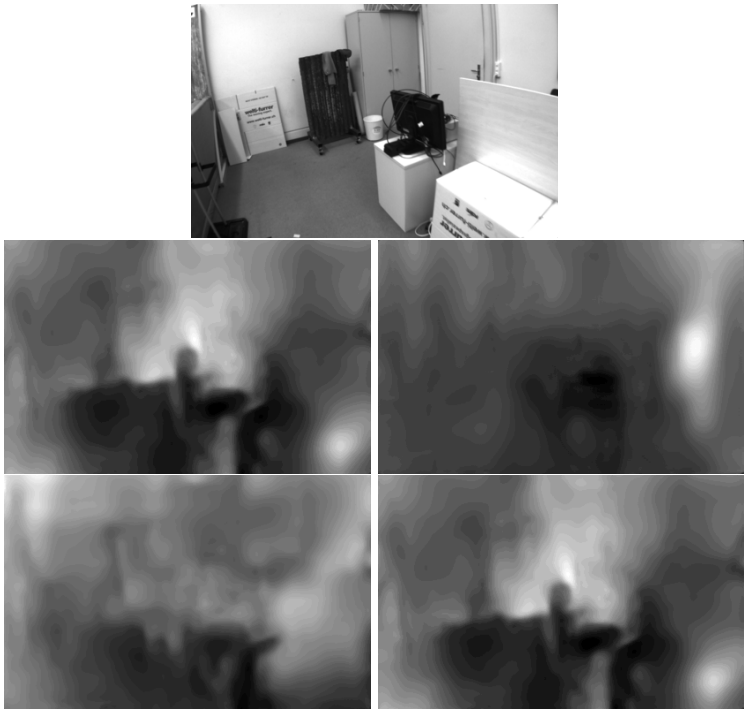


Figure 4.15: Depth maps for a random frame in V1 02 sequence. Top row: original frame. Middle row: learner-mh (left) and learner-v1 (right). Bottom row: learner-mhv1 (left) and learner-euroc (right). Interestingly, none of the models are able to produce a decent depth map for the frame.

4.4 Model Comparison

When comparing all the models together, it becomes apparent that traditional methods such as DSO or ORB-SLAM are able to outperform any learned method if they are able to be run on the sequence. Looking at tables 4.1 and 4.2, certain observations can be made. Although DSO and ORB-SLAM outperform both posenet and Sfm-Learner on the aqualoc sequences, the trajectories are not suitable for the algorithms. Tracking is often lost due to the homogeneous features, or the lack of features, and the two algorithms become unreliable. Sfm-Learner works, but works a lot better on certain sequences than others. Aqualoc is an interesting dataset with very similar looking features which may have resulted in Sfm-Learner failing to learn proper motion

cues from certain sequences. Also, due to the lighting conditions, the images have vignetting effects that remain unaccounted for by the model as well. As a result, there will always be some level of error in the model due its inability to match pixels around the edges.

On the Advio dataset, DSO and ORB-SLAM are both unable to be run due to the strong rotational motion and limited field of view. Sfm-Learner also struggles on these sequences, with both models performing similarly highlighting the difficulty the models face when learning. Despite the different splits for the two models, they learn similar information, but the overall performance remains lackluster for both models. This is attributable to the presence of non-lambertian surfaces such as glass, reflective tiles, and bright lights, along with a varying amount of moving people. These two properties of the dataset make it very demanding for any model to learn from. However, as per the original goal of this work, including a dataset like Advio allows for the evaluation of models under difficult circumstances and an opportunity to learn from challenging features. The fact that Sfm-Learner is unable to properly learn from either training split indicates a critical flaw in the model that fails to account for certain properties that are often prevalent in more real-life scenarios.

Lastly, in the EuroC dataset, ORB-SLAM out-performs every model on every sequence. Again, though Sfm-Learner and Posenet do slightly better on this dataset than they did on Advio, they are outperformed. The reason for this could be due to the exposure issues present in the sequences where the lighting affects exposure times in certain frames. This results in some frames being slightly over or under exposed. Though not as bad for the Machine Room (MH) sequences since they are indoors with overhead lighting, the Vicon Room (V1 and V2) are notoriously bad. These rooms are small, enclosed spaces with one window lighting the room. As a result, whenever

Table 4.1: ATE RMSE on all sequences for all models. A dash indicates the model was not run or tested on that sequence.

	Aqualoc		**Advio**			**EuroC**			
model	3	7	3	5	19	MH 02	MH 04	V1 02	V2 02
orbslam	-	**0.133**	-	-	-	**0.494**	**2.18**	**0.927**	**0.844**
dso	**0.229**	-	-	-	-	-	2.273	1.142	0.946
posenet	-	4.244	-	-	-	-	9.557	-	2.679
learner-aq	1.001	3.88	-	-	-	-	-	-	-
learner-adv-mall	-	-	8.002	10.022	17.968	-	-	-	-
learner-adv-ind	-	-	**6.551**	**8.202**	**17.672**	-	-	-	-
learner-euroc-mh	-	-	-	-	-	2.937	4.209	1.757	2.003
learner-euroc-v1	-	-	-	-	-	3.369	5.722	1.76	2.068
learner-euroc-mhv1	-	-	-	-	-	2.504	4.446	1.741	2.072
learner-euroc	-	-	-	-	-	2.711	4.948	1.768	2.052

Table 4.2: Photometric issues present in the test sequences of all 3 datasets used.

	Aqualoc		**Advio**			**EuroC**			
model	3	7	3	5	19	MH 02	MH 04	V1 02	V2 02
exposure issues	✗	✗	✗	✗	✗	✓	✓	✓	✓
non-lambertian surfaces	✗	✗	high	high	low	✗	✗	✗	✗
moving objects	✗	✗	moderate	low	✗	✗	✗	✗	✗
vignetting	✓	✓	✗	✗	✗	✗	✗	✗	✗

the MAV travels near the window, it fails to adjust for the lighting resulting in over-exposed frames. However, the next frames might then adjust, and produce normal frames adjusted for the lighting. When the MAV moves away from the window the opposite issue occurs where the frames can be under-exposed for a small period of time before getting rectified. This can have devastating effects on direct methods that do not account for such photometric inconsistencies. Since it is only for a few frames, and not the entire sequence like Advio, the effects are not as harmful and Sfm-Learner does ok. However, it can be seen from the trajectories and the depth maps that Sfm-Learner still struggles to learn.

Chapter 5

Conclusion

5.1 Summary

In this work, the performance of several models, including traditional, non-learned models and more modern deep learned models are examined under the context of a novel dataset. This dataset is built out of the existing Advio, Aqualoc and EuroC MAV datasets for the purposes of introducing more diversity of features and scenarios. Diversity in datasets is critical to ensuring models are able to properly learn from the right features, and generalize to other scenarios. Both DSO and ORB-SLAM perform well when they are able to in this dataset. One of the critical flaws is the loss of tracking when faced with strong rotational motion, and as a result neither model is able to be run on the Advio dataset. Since these are not learned methods, their weaknesses can only be hand engineered and, though already known, this dataset reaffirms this major weakness of traditional visual odometry methods.

As for the learned methods that were tested, PoseNet proved to be too simple of a model

that was incapable of adapting to different trajectories within even the same room or area. Despite training it on several sequences, it was unable to generalize beyond the small area it had already seen. Sfm-Learner, a much larger and more complex network, fared slightly better. However, it had critical flaws as well, namely the inability to learn under complex lighting or motion scenarios. This is not a surprise seeing how closely Sfm-Learner is to direct visual odometry methods since it directly aims to optimize a photometric loss. Yet, unlike DSO, it fails to account for frame exposure differences, gamma correction or vignetting issues. Though it is able to learn the general trend of some trajectories, it is observed that Sfm-Learner largely fails to learn poses within a reasonable error.

Unfortunately, training and testing the models on the diverse dataset does not improve their performance nor indicate that the models are able to generalize better to differing scenes. However, these are flaws of the models, not necessarily of the dataset, and by training and testing on the dataset, it is able to highlight these critical flaws of the models.

5.2 Conclusion

ORB-SLAM and DSO are amazing algorithms that perform extremely well in scenarios where they can, but unfortunately fail completely when they encounter specific scenarios like pure camera rotation. Deep learned models using convolutional neural networks can help mitigate these issues and create more accurate models that are able to generate better depth and camera poses.

The Sfm-Learner model performs well given the its key assumptions hold, though that

is very unlikely in most real-world scenarios. It is able to outperform traditional methods in only the most ideal conditions like KITTI. However, the popularity of this model has created an entire class of similar monocular, unsupervised models focused on generating depth and pose through image synthesis. These models all suffer from the same weaknesses of lacking proper photometric calibration, lacking a strong supervisory signal for learning depth maps and over-fitting on homogeneous datasets that are then used to claim strong results.

The Sfm-Learner model can be improved however. It needs a better and stronger supervision signal due to the fact that if depth maps are never accurately learned, then pose estimations will always suffer. As a result, it may be time to reconsider photometric error as the sole supervision signal and incorporate geometric loss functions into the model as well. Also, similar to how DSO explicitly handles photometric calibrations, it would be interesting to see similar, possibly learned, models to account for these parameters in these networks. This would be extremely helpful since the loss is based on photometric consistency, and ensuring that warped images do not face vignetting or exposure issues would result in better and more accurate models.

Many of these models are only trained and tested on the KITTI dataset for pose on specifically 5 window frames due to the error build up in longer sequences. As a result, it is hard to truly judge how good these models can behave and to what extent they can generalize. Using a dataset like the one this project has curated will help these models not only train on a wider set of problems, but also highlight the limitations of such models. Hopefully, learned models can eventually make use of this larger and more diverse dataset and learn to adapt to different scenarios whether they be under the water, in a mall, or even on Mars.

5.3 Future Direction

Currently, the models were trained and tested on each dataset individually and independently. Though not discussed in this work, transfer learning was attempted by training a model on two or more datasets and testing on the respective sequences. However, this showed no advantage over training the model on each dataset individually, likely due to the catastrophic forgetting problem. In the future, to build more robust, learned visual odometry systems, it will be necessary to overcome such domain adaptation issues and build a model that is able to use transfer learning to identify camera motion irrespective of the particular scene. This is an avenue to explore in the near future, as learned models still tend to over-fit to particular scenes and environments. Due to the diverse nature of the environments present in this dataset, it is an ideal dataset to train and test novel methods capable of successfully predicting camera motion.

Bibliography

[BNG+16] Michael Burri, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W Achtelik, and Roland Siegwart. The euroc micro aerial vehicle datasets. *The International Journal of Robotics Research*, 2016.

[COR+16] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. *CoRR*, abs/1604.01685, 2016.

[CSRK18] Santiago Cortés, Arno Solin, Esa Rahtu, and Juho Kannala. ADVIO: An authentic dataset for visual-inertial odometry. *arXiv e-prints*, page arXiv:1807.09828, Jul 2018.

[DMR18] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Self-Improving Visual Odometry. 2018.

[EKC18] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct Sparse Odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(3):611–625, 2018.

[ESC14] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *European conference on computer vision*, pages 834–849. Springer, 2014.

[FMT+18] Maxime Ferrera, Julien Moras, Pauline Trouvé-Peloux, Vincent Creuze, and Denis Dégez. The Aqualoc Dataset: Towards Real-Time Underwater Localization from a Visual-Inertial-Pressure Acquisition System. *arXiv e-prints*, page arXiv:1809.07076, Sep 2018.

[FPS14] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. Svo: Fast semi-direct monocular visual odometry. In *2014 IEEE international conference on robotics and automation (ICRA)*, pages 15–22. IEEE, 2014.

[GLSU13] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[GMB17] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, volume 2017-Janua, pages 6602–6611, 2017.

[GVCR16] Ravi Garg, B. G. Vijay Kumar, Gustavo Carneiro, and Ian Reid. Unsupervised CNN for single view depth estimation: Geometry to the rescue. Technical report, 2016.

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2016-Decem, pages 770–778, 2016.

[JFS03] Hailin Jin, Paolo Favaro, and Stefano Soatto. A semi-direct approach to structure from motion. *The Visual Computer*, 19(6):377–394, 2003.

[KGC15] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE international conference on computer vision*, pages 2938–2946, 2015.

[KM07] Georg Klein and David Murray. Parallel tracking and mapping for small AR workspaces. In *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*, Nara, Japan, November 2007.

[LAD18] Annamalai Lakshmi, Faheema Agj, and Dipti Deodhare. Robust Direct Visual Odometry Estimation for a Monocular Camera under Rotations. *IEEE Robotics and Automation Letters*, 3(1):367–372, jan 2018.

[MAMT15] Raul Mur-Artal, J. M.M. Montiel, and Juan D. Tardos. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[MIH+16] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4040–4048, 2016.

[MWA18] Reza Mahjourian, Martin Wicke, and Anelia Angelova. Unsupervised Learning of Depth and Ego-Motion from Monocular Video Using 3D Geometric Constraints. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 5667–5675, 2018.

[NLD11] 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.

[PKK18] Shashi Poddar, Rahul Kottath, and Vinod Karar. Evolution of Visual Odometry Techniques. 2018.

[SF16] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4104–4113, 2016.

[SLJ+14] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going Deeper with Convolutions. *arXiv e-prints*, page arXiv:1409.4842, Sep 2014.

[SLZ+19] Tianwei Shen, Zixin Luo, Lei Zhou, Hanyu Deng, Runze Zhang, Tian Fang, and Long Quan. Beyond Photometric Loss for Self-Supervised Ego-Motion Estimation. 2019.

[SZ14] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. 2014.

[SZPLT19] Torsten Sattler, Qunjie Zhou, Marc Pollefeys, and Laura Leal-Taixe. Understanding the Limitations of CNN-based Absolute Camera Pose Regression. 2019.

[VRS+17] Sudheendra Vijayanarasimhan, Susanna Ricco, Cordelia Schmid, Rahul Sukthankar, and Katerina Fragkiadaki. SfM-Net: Learning of Structure and Motion from Video. 2017.

[YS18] Zhichao Yin and Jianping Shi. GeoNet: Unsupervised Learning of Dense Depth, Optical Flow and Camera Pose. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1983–1992, 2018.

[YWGC18] Nan Yang, Rui Wang, Xiang Gao, and Daniel Cremers. Challenges in monocular visual odometry: Photometric calibration, motion bias, and rolling shutter effect. *IEEE Robotics and Automation Letters*, 3(4):2878–2885, 2018.

[ZBSL17] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, volume 2017-Janua, pages 6612–6621, 2017.