16-833 Course Project
Is Monocular Vision Sufficient For Multi-view Visual Odometry?

Ravi Tej Akella (rakella) Neha Boloor (nboloor) Anirudh Chakravarthy (achakrav)
Naveen Venkat (nvenkat)

Abstract

In scenarios such as autonomous driving, vehicles are often mounted with several camera sensors. As the robot navigates through the environment, we need to infer the odometry to accurately localize the robot's position on a global map. Several methods use complicated geometry or learning-based methods to do this. In this project, we aim to develop simple baselines for multi-view visual odometry, leaning on advances in monocular visual odometry. We propose some heuristics and learning-based approaches to fuse monocular visual odometry estimates from each camera and evaluate its performance. Code available at: https://github.com/neha-boloor/MVO-fusion.

1 Introduction

Visual SLAM has gained prominence in literature due to the ease of capturing images from autonomous robots and vehicles. Visual odometry (VO) is one of the first steps towards visual SLAM wherein given paired images between two timesteps, the objective is to predict the relative camera poses. This is slightly different from traditional structure-from-motion (SfM), in that VO predictions need to be online while SfM can also be performed in an offline setting. This way VO is a more realistic problem setting for autonomous navigation use cases.

While several learning-based and geometry-based VO methods have emerged in the literature, monocular VO methods have received significant attention. Under this challenging setting, we only assume access to a single camera and no inertial information. Intrinsically, this is a difficult problem due to issues such as scale drift over time. However, since this setting is well-studied and works well in practice, we base our method on existing state-of-the-art MVO methods.

In practice, we often have multiple cameras mounted onto a robot that can be used for more precise localization. This task is known as multi-view VO, which has garnered limited interest until recently in the literature. Existing works either heavily rely on the scene geometry or use complicated networks posing challenges for real-world generalization.

In our project, we aim to answer the question *Is Monocular Vision sufficient for Multi-view Visual Odometry?* by exploring whether MVO results can be stitched together to develop a simple yet robust multi-view VO method. Concretely, given C cameras, we aim to fuse the C MVO predictions to generate a single more robust pose estimate for the agent. We explore a variety of fusion mechanisms to identify a strong baseline.

2 Related Work

2.1 Datasets

In order to validate our hypothesis, we need datasets that consist of multiple calibrated cameras. Existing VO works benchmark on datasets such as KITTI [1], TartanAir [2], EuRoC [3], or TUM RGBD [4] which leverage stereo geometry to provide auxiliary information such as depth. However, these datasets are inappropriate for our use case because the baseline between the stereo pair is usually small, resulting in insufficient perceptual differences between the two views. In such cases, running MVO independently on both images is almost as good as using a single image.

We would like to have a dataset consisting of ≥ 3 cameras, which poses additional challenges compared to stereo image pairs. One such dataset we identified is the NuScenes [5] dataset which consists of 6 cameras (as illustrated in Fig. 1a, front-left, front, front-right, back-right, back, back-left). NuScenes is a large-scale autonomous driving dataset

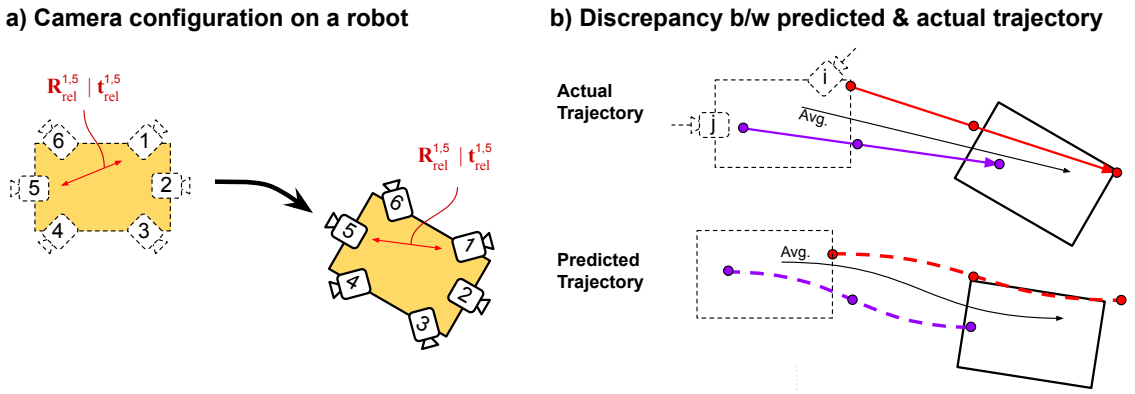

Figure 1: Illustration of MVO fusion. (a) Suppose we have an agent with 6 cameras. We consider the scenario where the relative configuration of the cameras $[\mathbf{R}_{rel}|\mathbf{t}_{rel}]$ remains fixed when the agent moves (*e.g.* as shown for cameras 1 and 5 here). We apply an MVO algorithm on each camera independently and obtain the camera motion estimates (as shown in (b)). **(b)** However, the trajectories predicted by the MVO algorithm (bottom) may not match the actual trajectory (top). Moreover, the noise in predictions would be different for each camera (here, the red and the purple trajectories are perturbed differently). Our goal is to fuse the multi-camera predictions to obtain one trajectory (black) which represents the motion of the agent more faithfully.

consisting of images, LiDAR scans, RADAR scans, sensor calibrations, and sensor poses. This is a challenging dataset consisting of dynamic urban scenes *i.e.*, several moving objects exist in the scene, where simple geometric approaches for Stereo VO would fail. Therefore, we choose the NuScenes dataset for our experiments.

2.2 Multi-View Visual Odometry

Before exploring approaches for monocular VO, we wish to study existing multi-view VO to gather insights. However, in contrast to Monocular and Stereo VO, we observe that this field is not well-explored and to the best of our knowledge, only one prior work exists. AFT-VO [6] uses a transformer architecture to learn a data-driven fusion from several asynchronized sensors. The authors note that probabilistic fusion techniques can be used for synchronized cameras, and we are motivated to validate this hypothesis. Therefore, for simplicity, we assume the presence of synchronized sensors in this project. This kind of synchronized sensor data is also made available in the NuScenes dataset.

2.3 Monocular Visual Odometry (MVO)

For this project, to build our method, we first study monocular VO under a fully-supervised setting. DPT-VO [7] uses a transformer network to resolve scale ambiguity in monocular VO. JPerceiver [8] uses a multi-task network for joint depth estimation, pose prediction, and BEV layout from an input image. DytanVO [9] jointly optimizes for motion segmentation and camera pose to achieve strong performance on dynamic urban scenes.

TartanVO [10] incorporates camera intrinsics into the model and introduces a scale-ambiguous loss function. The model is trained on synthetic dataset (TartanAir [2]) and shows strong generalization to real-world datasets such as KITTI and EuROC without additional fine-tuning. DPVO [11] introduces a two-stage network for MVO where the patch extraction module splits the image features into patches and the update module attempts to track these patches over time, while also optimizing for camera poses using differentiable bundle adjustment.

For the scope of our project, we work with TartanVO. We elaborate further details on its usage in Sec. 4.

3 Approach

We aim to develop an approach to fuse trajectory predictions obtained using MVO method on individual cameras installed on an agent as shown in Fig. 1. In Sec 3.1, we formally introduce Monocular Visual Odometry (MVO) and provide the motivation for fusing independent MVO predictions. Lastly, Sec 3.2 discusses four strategies for late-fusion of individual MVO predictions to obtain a robust estimation of relative motion.

3.1 Preliminaries

Monocular Visual Odometry. We consider having access to C cameras each with intrinsics $\{K^c \in \mathbb{R}^{3\times3}\}_{c=1}^{C}$ calibrated with a certain fixed relative configuration (Fig. 1a). From each camera c we receive a video stream (sequence of images) $\{\mathbf{I}_n^c \in \mathbb{R}^{H\times W\times 3}\}_{n=1}^{N}$. For each camera c, the MVO algorithm outputs a relative pose prediction $\{[\Delta\mathbf{R}_n^c|\Delta\mathbf{t}_n^c]\}_{n=1}^{N-1}$ between each consecutive pair of frames, where $\Delta\mathbf{R}_n^c$ and $\Delta\mathbf{t}_n^c$ are the relative rotation and translation between n^{th} and $(n+1)^{th}$ frames. Given the predictions, the pose of the camera c can be updated by:

$$\mathbf{R}_{n+1}^c = (\Delta\mathbf{R}_n^c)\mathbf{R}_n^c \quad ; \quad \mathbf{t}_{n+1}^c = \Delta\mathbf{t}_n^c + \mathbf{t}_n^c \tag{1}$$

MVO Fusion. For simplicity, we assume that the video feeds across the cameras are temporally aligned (*i.e.* $\mathbf{I}_n^i$ and $\mathbf{I}_n^j$ belong to the same time instant n). Our goal is to use MVO to predict the trajectory of each camera $\{[\mathbf{R}_n^c|\mathbf{t}_n^c]\}$ simultaneously and fuse the predictions to obtain a single trajectory. Our fusion function can be expressed as:

$$\text{fuse}(\{[\Delta\mathbf{R}_n^c|\Delta\mathbf{t}_n^c]\}) \rightarrow \{[\Delta\mathbf{R}_n|\Delta\mathbf{r}_n]\} \tag{2}$$

Note that Eq. 1 assumes consistent noise-free outputs from the MVO algorithm. However, this may not hold in practice. For instance, if the MVO algorithm uses a deep neural network that is not robust to domain shifts, we would lose precision in poses. Some methods depend on auxiliary data or tasks (*e.g.* depth prediction, loop closure, etc.) to improve the predictions. Instead, we ask *can we improve the performance of MVO if we have multiple cameras?* Formally, we assume that the errors in the relative motion predictions of each individual camera are independent when conditioned on the frames from the given camera. Our intuition is that through the fusion of multiple cameras, one may be able to mitigate the noise and obtain more accurate predictions.

Late fusion. We adopt a late fusion strategy. In theory, this leads to an approach agnostic to the underlying MVO algorithm. For instance, one can directly fuse the pose predictions as shown in Eq. 2 allowing flexibility to run the MVO algorithm of choice for each camera. More sophisticated deep-learning approaches can fuse high-level features from a neural network trained for MVO. This allows the model developer to employ supervisory signals beyond those used by the backbone model leading to more accurate predictions. In the next section, we discuss four different strategies to fuse the individual noisy predictions to obtain a robust estimate of the robot's relative motion.

3.2 Late-Fusion Strategies

Given per-camera rotation and translation estimates, the objective is to aggregate the predictions into a single relative motion estimate for the agent. Assuming that the prediction errors are independent and identically distributed, this problem reduces to finding a central tendency. We propose the following strategies for fusing MVO predictions. Here, local fusion refers to fusion at each timestep, and global fusion refers to fusing trajectories globally.

(a) Local Fusion. In the following strategies, we fuse the relative motion predictions at each time step.

1. **Euler angle fusion.** We convert each of the C rotation predictions into their Euler angle representations and then find their central tendency using (i) mean and (ii) median. The translations are fused by taking their mean across each Cartesian dimension in the robot's body frame.

2. **Quaternion averaging.** In this strategy, we convert the rotation predictions into quaternion representations and then average them using the technique proposed in [12], which is given as:

$$\bar{q} = \text{argmax}_{q\in\mathcal{S}^3} \; q^\top \left(\sum_{i=1}^{n} q_i q_i^\top\right) q, \tag{3}$$

where $\{q_i\}_{i=1}^n$ are the independent quaternion predictions that need to be averaged. While this objective seems hard to decipher at first glance, it can be rewritten as follows:

$$\bar{q} = \text{argmax}_{q\in\mathcal{S}^3} \; \sum_{i=1}^{n} < q, q_i >^2 . \tag{4}$$

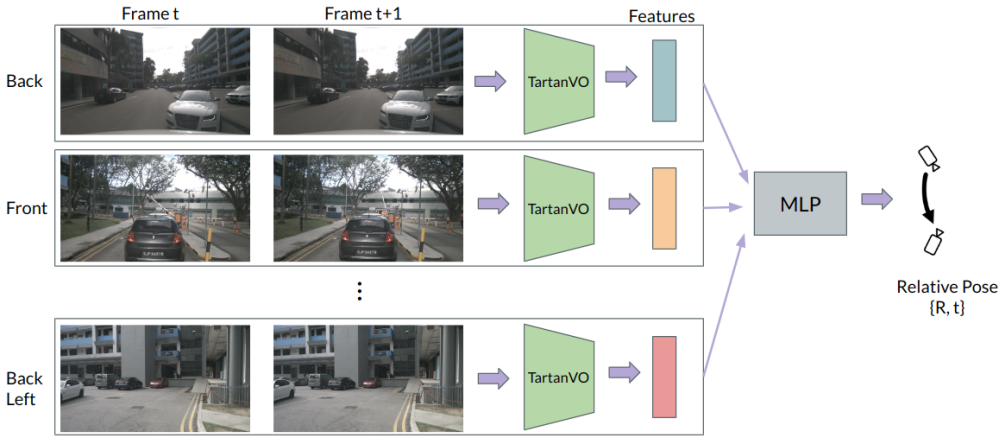

Figure 2: Learning-based late fusion. Given images from several cameras at time t and $t+1$, we concatenate the per-camera latent features obtained from TartanVO and use an MLP to predict a single relative pose, aggregating information across views.

Essentially, we are trying to find the quaternion that is closest to all the prediction quaternions in terms of cosine similarity. The translation predictions are fused similar to (1) by averaging over each Cartesian dimension.

(b) Global Trajectory Fusion. Since the trajectory predictions are generated by integrating the relative predictions over time, they are susceptible to drift, and simple methods such as mean or median not work well. Unlike the aforementioned methods, we now focus on fusing entire unrolled trajectories in the inertial frame.

1. **Robust polynomial fitting through RANSAC.** We fit a polynomial through the trajectory points using squared error loss and run this optimization using RANSAC for robustness against outliers.
2. **Refined mean via iterative outlier removal.** A more direct approach to getting rid of the outliers is by iteratively pruning trajectory points with high error margins w.r.t the estimated mean trajectory. Initially, the mean trajectory is obtained using local fusion such as quaternion averaging. Then, at each iteration, we find the farthest point from the mean trajectory and nullify its contribution to the mean by deleting it.

(c) Learning-based Fusion. While the aforementioned methods use domain knowledge to find the central tendency, we can also task a neural network to aggregate the predictions. This requires very little prior knowledge and relies on data-driven supervision for estimating the robot's motion. As illustrated in Fig. 2, we use TartanVO, the neural network trained for MVO in our case, to extract latent features capturing the essence of the relative motion independently. and perform a learning-based fusion.

4 Experiments

4.1 Preliminary Experiments

TartanVO. We set up the environment and codebase for TartanVO, and ran inference using the pre-trained model on sequence 10 from the KITTI dataset (Fig. 3). We could replicate performance as reported in the paper, hence verifying the correctness of the set-up.

DPVO. Next, we set up the environment for DPVO, and using a pre-trained model, we ran inference on KITTI, NuScenes (Fig. 4), and self-recorded iPhone videos.

The architecture used in DPVO includes a recurrent neural network component to track each patch, extracted by the transformer-based neural network in the first stage, through time. Its complex modules make it hard to choose the right feature per image pair in the sequence. Hence, we decided to only use TartanVO for further analysis.

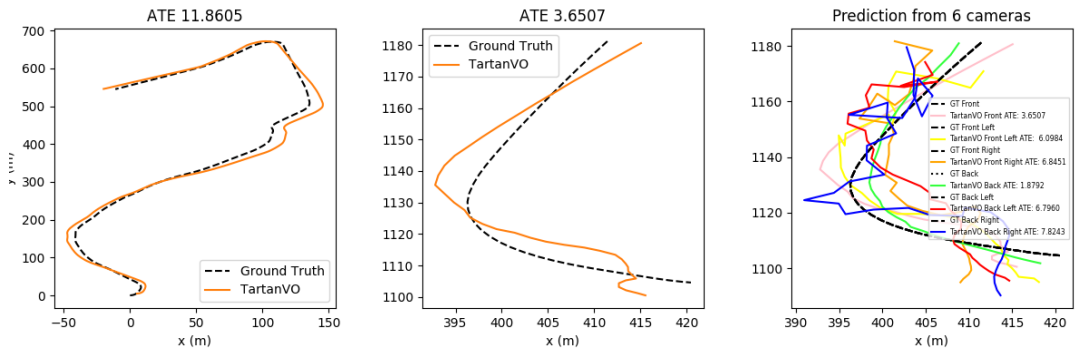

Figure 3: This figure shows the inference of TartanVO on **(a)** Sequence 10 from KITTI dataset (left), and, **(b)** One of the 6 Cameras of Scene 1 of NuScenes (center). **(c)** Running TartanVO on 6 cameras in NuScenes gives inconsistent results across cameras (right).

Figure 4: Using pre-trained DPVO for inference on NuScenes leads to realistic and temporally consistent pose prediction

4.2 Late-Fusion Strategies

4.2.1 Local trajectory fusion

Euler Angle Fusion and Quaternion Averaging. The results for local trajectory-based fusion approaches are visualized in Fig. 5. We plot individual trajectories of each camera (*e.g.* `CAM_FRONT`), and also the three averaging techniques - Quaternion averaging (`AVG_QUAT`), mean Euler angles (`AVG_EUL`), median Euler angles (`MED_EUL`). Clearly, the rolled-out trajectory fits close to the ground-truth trajectory. A key takeaway from our analysis is that the rotation is more faithfully captured by averaging quaternions, compared to euler angles [12]. However, the absolute trajectory error is still quite high when compared to other works in the literature. Since fusion approaches are limited by the performance of the MVO method, we are unable to improve this baseline further.

4.2.2 Global trajectory fusion

Refined Mean via Iterative Outlier Removal. In this approach, we iteratively prune data points with high error margins w.r.t the estimated mean trajectory to get rid of outliers and refine the mean trajectory. In Fig. 6, we plot the mean trajectory (a) before refinement, (b) after 100 iterations of refinement, (c) after 200 iterations of refinement. The corresponding paths of the 6 cameras are plotted in (d). Additionally, plot (e) shows the RMSE for the mean trajectory w.r.t the ground truth as the number of pruning iterations increases. We see that initially, pruning helps in refining the mean estimate, bringing it closer to the ground truth. However, as we keep pruning further, we lose track of the correct points as can be seen by the increased RMSE beyond 150 iterations. Choosing the sweet spot for the number of iterations without using ground truth is an interesting future work.

Polynomial Fitting with RANSAC The results for polynomial fitting are visualized in Fig. 7. It can be seen that polynomial fitting with RANSAC can be very effective against outliers despite heavy noise in MVO predictions. However, this method fails when the ground-truth trajectory is non-smooth or has sharp variations. Further, the degree of the polynomial is a hyperparameter for this method which can significantly affect the predictive performance of the approach. Since the performance of this approach relies on how well the polynomial can explain the ground-truth trajectory, it is hard to deploy this approach in practice.

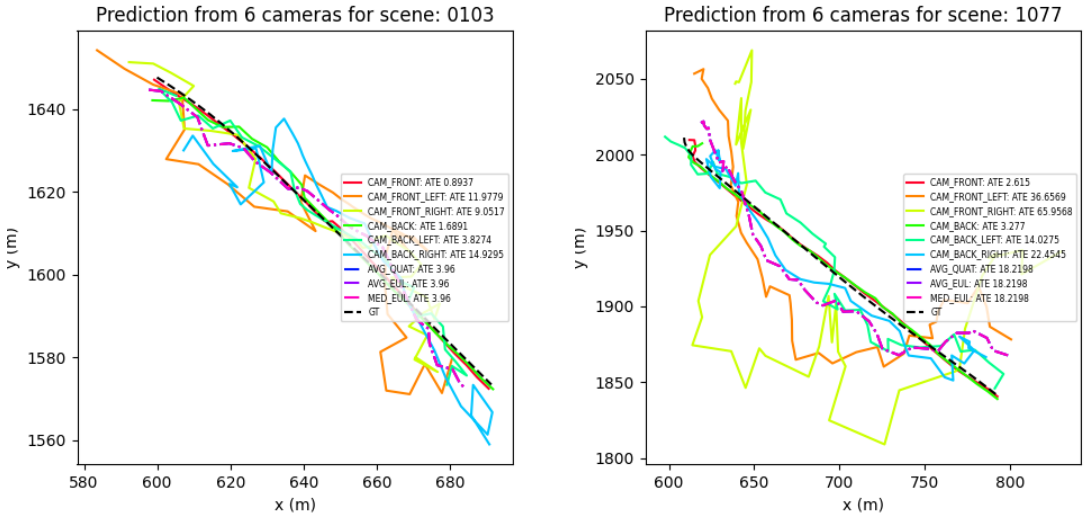

Figure 5: This figure shows the inference of TartanVO on **(a)** Sequence 103 from NuScenes dataset (left), **(c)** Sequence 1077 from NuScenes dataset (right) along with the Euler Mean, Euler Median, Quaternion Averaging and Ground Truth trajectories.

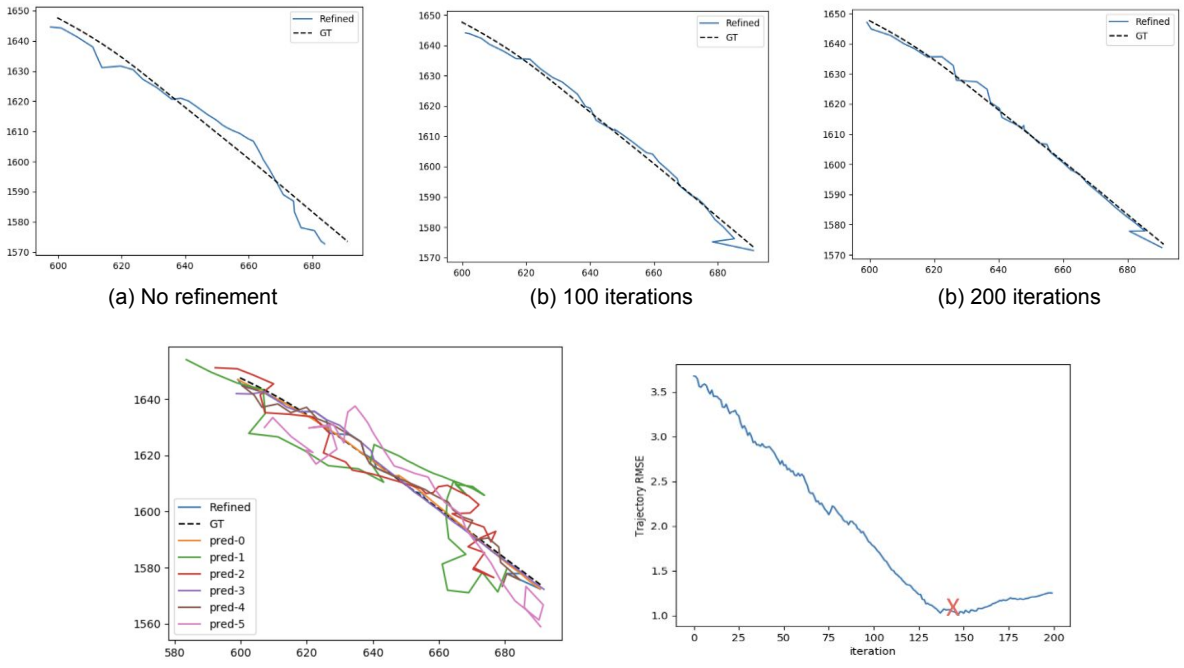

Figure 6: This figure shows the inference of TartanVO on **(a)** Sequence 103 from NuScenes dataset with no refined mean strategy applied (left-most), **(b)** Refined Mean run for 100 iterations **(c)** Refined Mean run for 200 iterations **(d)** Refined Mean run for 100 iterations with GT and Individual Camera Estimates **(e)** RMSE Vs Refinement iterations.

4.2.3 Learning-based Late Fusion

We aim to aggregate features across multiple views to predict a fused estimate which is more robust, using per-camera predictions from TartanVO through simple feature concatenation. Given d-dimensional features from each image, we construct a resultant feature vector of $C \times d$, where C denotes the number of cameras ($C = 6$ in our case). Now,

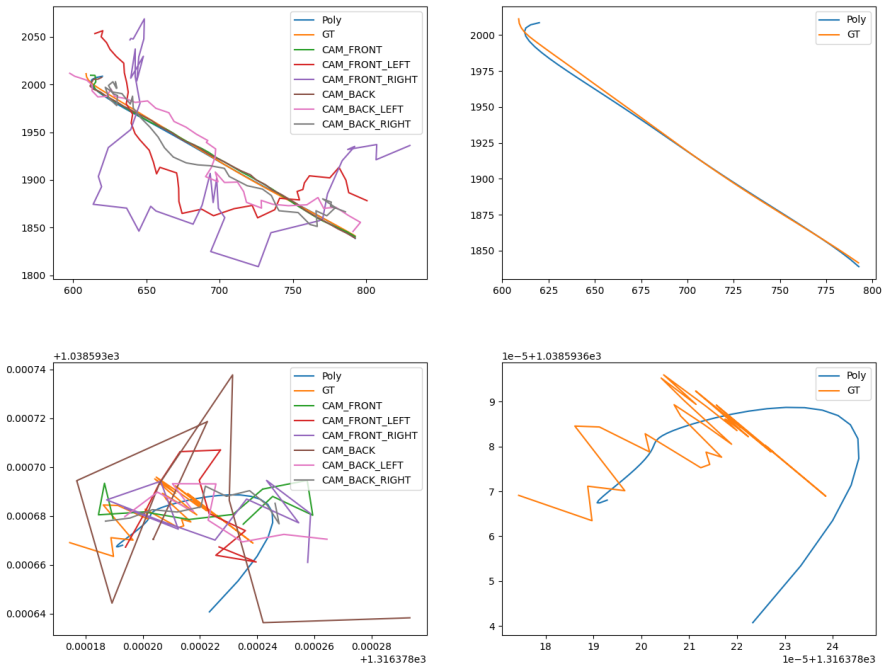

Figure 7: Trajectory Fusion via polynomial fitting with RANSAC applied on two Nuscenes sequences. The images on the left include the ground-truth trajectory (orange) along with the individual MVO predictions. The corresponding images on the right only visualize the ground truth and fused trajectory for clarity.

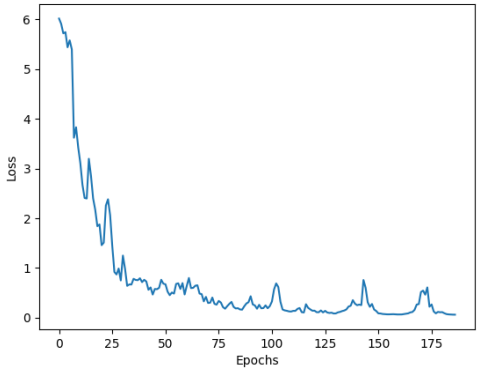

Figure 8: Training loss of our proposed learning-based fusion method, plotted across epochs.

using a multi-layer perceptron with learned weights, we reduce this concatenated feature vector to predict the relative transformation. We supervise the network using the ground-truth relative poses. Specifically, for the translation component, we follow TartanVO and apply a cosine embedding loss. For the rotation component, we apply an L2-loss over the quaternions. See [10] and our code implementation for more details.

Results: To validate whether this problem is indeed tractable to solve, we first visualize the training loss across epochs in Fig 8. We observe that the loss decreases, which indicates the network is indeed learning. However, due to unanticipated setup difficulties on fusing TartanVO features on NuScenes, we are unable to evaluate the performance of this method. We note that future work could precompute features for evaluation, which we are unable to explore due to resource constraints.

5 Conclusion & Future Work

In this project, we aim to develop various simple baselines for multi-view visual odometry using MVO methods. In particular, we rely on TartanVO and explore a variety of late fusion methods to generate a more robust pose prediction. Preliminary results show promise in a few of these approaches, however, the results are still very early to compare against the existing state-of-the-art.

In the future, we aim to explore a few directions. Firstly, we wish to incorporate the fact that the relative configuration of the cameras remains fixed during the motion of the agent. To this end, we wish to explore optimization-based methods which can directly solve for the noise in each pose prediction. Secondly, it is important to account for the uncertainty while aggregating the per-camera predictions. For instance, in our current setup, the mean trajectory estimates can be improved by using weights that indicate the uncertainty in pose predictions by each camera. Thirdly, more analysis is required on how to fuse the predictions when each camera provides incorrect estimates such that their central tendency deviates away from the ground truth trajectory. We believe that these are some promising directions and our project provides an implementation-level groundwork to explore these ideas in the future.

References

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

[2] W. Wang, D. Zhu, X. Wang, Y. Hu, Y. Qiu, C. Wang, Y. Hu, A. Kapoor, and S. Scherer, "Tartanair: A dataset to push the limits of visual slam," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4909–4916, IEEE, 2020.

[3] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The euroc micro aerial vehicle datasets," *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1157–1163, 2016.

[4] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.

[5] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11621–11631, 2020.

[6] N. Kaygusuz, O. Mendez, and R. Bowden, "Aft-vo: Asynchronous fusion transformers for multi-view visual odometry estimation," *arXiv preprint arXiv:2206.12946*, 2022.

[7] A. O. Françani and M. R. O. A. Maximo, "Dense prediction transformer for scale estimation in monocular visual odometry," in *2022 Latin American Robotics Symposium (LARS), 2022 Brazilian Symposium on Robotics (SBR), and 2022 Workshop on Robotics in Education (WRE)*, oct 2022.

[8] H. Zhao, J. Zhang, S. Zhang, and D. Tao, "Jperceiver: Joint perception network for depth, pose and layout estimation in driving scenes," in *European Conference on Computer Vision*, pp. 708–726, Springer, 2022.

[9] S. Shen, Y. Cai, W. Wang, and S. Scherer, "Dytanvo: Joint refinement of visual odometry and motion segmentation in dynamic environments," *arXiv preprint arXiv:2209.08430*, 2022.

[10] W. Wang, Y. Hu, and S. Scherer, "Tartanvo: A generalizable learning-based vo," *arXiv preprint arXiv:2011.00359*, 2020.

[11] Z. Teed, L. Lipson, and J. Deng, "Deep patch visual odometry," *arXiv preprint arXiv:2208.04726*, 2022.

[12] F. L. Markley, Y. Cheng, J. L. Crassidis, and Y. Oshman, "Averaging quaternions," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 4, pp. 1193–1197, 2007.