Towards Visual Ego-motion Learning in Robots

Sudeep Pillai
CSAIL, MIT
spillai@csail.mit.edu

John J. Leonard
MIT
jleonard@mit.edu

***Abstract*— Many model-based Visual Odometry (VO) algorithms have been proposed in the past decade, often restricted to the type of camera optics, or the underlying motion manifold observed. We envision robots to be able to learn and perform these tasks, in a minimally supervised setting, as they gain more experience. To this end, we propose a fully trainable solution to visual ego-motion estimation for varied camera optics. We propose a visual ego-motion learning architecture that maps observed optical flow vectors to an ego-motion density estimate via a Mixture Density Network (MDN). By modeling the architecture as a Conditional Variational Autoencoder (C-VAE), our model is able to provide introspective reasoning and prediction for ego-motion induced scene-flow. Additionally, our proposed model is especially amenable to *bootstrapped* ego-motion learning in robots where the supervision in ego-motion estimation for a particular camera sensor can be obtained from standard navigation-based sensor fusion strategies (GPS/INS and wheel-odometry fusion). Through experiments, we show the utility of our proposed approach in enabling the concept of self-supervised learning for visual ego-motion estimation in autonomous robots.**

I. Introduction

Visual odometry (VO) [1], commonly referred to as ego-motion estimation, is a fundamental capability that enables robots to reliably navigate its immediate environment. With the wide-spread adoption of cameras in various robotics applications, there has been an evolution in visual odometry algorithms with a wide set of variants including monocular VO [1], [2], stereo VO [3], [4] and even non-overlapping *n*-camera VO [5], [6]. Furthermore, each of these algorithms has been custom tailored for specific camera optics (pinhole, fisheye, catadioptric) and the range of motions observed by these cameras mounted on various platforms [7].

With increasing levels of model specification for each domain, we expect these algorithms to perform differently from others while maintaining lesser generality across various optics and camera configurations. Moreover, the strong dependence of these algorithms on their model specification limits the ability to actively monitor and optimize their intrinsic and extrinsic model parameters in an online fashion. In addition to these concerns, autonomous systems today use several sensors with varied intrinsic and extrinsic properties that make system characterization tedious. Furthermore, these algorithms and their parameters are fine-tuned on specific datasets while enforcing little guarantees on their generalization performance on new data.

Sudeep Pillai and John J. Leonard are with the Computer Science and Artificial Intelligence Lab (CSAIL), Massachusetts Institute of Technology (MIT), Cambridge MA 02139, USA. This work was partially supported by the Office of Naval Research under grants N00014-11-1-0688 and N00014-13-1-0588 and by the National Science Foundation under grant IIS-1318392, which we gratefully acknowledge. For more details, visit http://people.csail.mit.edu/spillai/learning-egomotion.

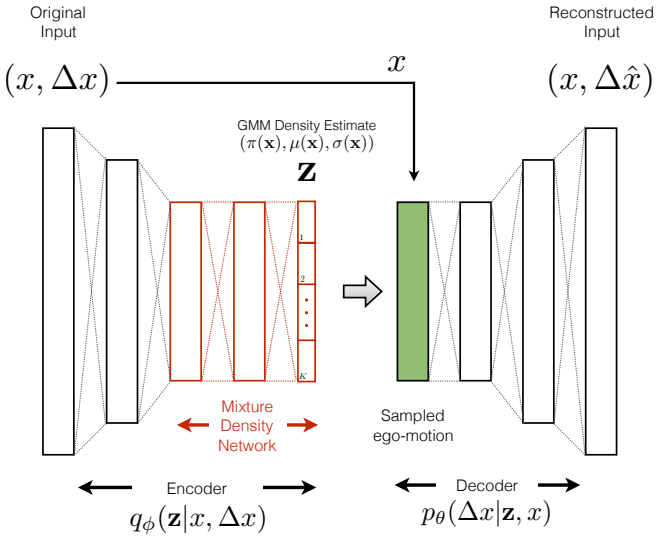


Fig. 1: **Visual Ego-motion Learning Architecture:** We propose a visual ego-motion learning architecture that maps optical flow vectors (derived from feature tracking in an image sequence) to an ego-motion density estimate via a Mixture Density Network (MDN). By modeling the architecture as a Conditional Variational Autoencoder (C-VAE), our model is able to provide introspective reasoning and prediction for scene-flow conditioned on the ego-motion estimate and input feature location.

To this end, we propose a fully trainable architecture for visual odometry estimation in generic cameras with varied camera optics (*pinhole*, *fisheye* and *catadioptric* lenses). In this work, we take a geometric approach by posing the regression task of ego-motion as a density estimation problem. By tracking salient features in the image induced by the ego-motion (via Kanade-Lucas-Tomasi/KLT feature tracking), we learn the mapping from these tracked flow features to a probability mass over the range of likely ego-motion. We make the following contributions:

- **A fully trainable ego-motion estimator**: We introduce a fully-differentiable density estimation model for visual ego-motion estimation that robustly captures the inherent ambiguity and uncertainty in relative camera pose estimation (See Figure 1).
- **Ego-motion for generic camera optics**: Without imposing any constraints on the type of camera optics, we propose an approach that is able to recover ego-motions for a variety of camera models including *pinhole*, *fisheye* and *catadioptric* lenses.
- **Bootstrapped ego-motion training and refinement**:

We propose a bootstrapping mechanism for autonomous systems whereby a robot self-supervises the ego-motion regression task. By fusing information from other sensor sources including GPS and INS (Inertial Navigation Systems), these indirectly inferred trajectory estimates serve as ground truth target poses/outputs for the aforementioned regression task. Any newly introduced camera sensor can now leverage this information to learn to provide visual ego-motion estimates without relying on an externally provided ground truth source.

- **Introspective reasoning via scene-flow predictions**: We develop a generative model for optical flow prediction that can be utilized to perform outlier-rejection and scene flow reasoning.

Through experiments, we provide a thorough analysis of ego-motion recovery from a variety of camera models including pinhole, fisheye and catadioptric cameras. We expect our general-purpose approach to be robust, and easily tunable for accuracy during operation. We illustrate the robustness and generality of our approach and provide our findings in Section IV.

II. Related Work

Recovering relative camera poses from a set of images is a well studied problem under the context of Structure-from-Motion (SfM) [8], [9]. SfM is usually treated as a non-linear optimization problem, where the camera poses (extrinsics), camera model parameters (intrinsics), and the 3D scene structure are jointly optimized via non-linear least-squares [8].

Unconstrained VO: Visual odometry, unlike incremental Structure-from-Motion, only focuses on determining the 3D camera pose from sequential images or video imagery observed by a monocular camera. Most of the early work in VO was done primarily to determine vehicle egomotion [10], [11], [12] in 6-DOF, especially in the Mars planetary rover. Over the years several variants of the VO algorithm were proposed, leading up to the work of Nister et al. [1], where the authors proposed the first real-time and scalable VO algorithm. In their work, they developed a 5-point minimal solver coupled with a RANSAC-based outlier rejection scheme [13] that is still extensively used today. Other researchers [14] have extended this work to various camera types including catadioptric and fisheye lenses.

Constrained VO: While the classical VO objective does not impose any constraints regarding the underlying motion manifold or camera model, it however contains several failure modes that make it especially difficult to ensure robust operation under arbitrary scene and lighting conditions. As a result, imposing egomotion constraints has been shown to considerably improve accuracy, robustness, and run-time performance. One particularly popular strategy for VO estimation in vehicles is to enforce planar homographies during matching features on the ground plane [15], [16], thereby being able to robustly recover both relative orientation and absolute scale. For example, Scaramuzza et al. [7], [17] introduced a novel 1-point solver by imposing the vehicle's non-holonomic motion constraints, thereby speeding up the VO estimation up to 400Hz.

Data-driven VO: While several model-based methods have been developed specifically for the VO problem, a few have attempted to solve it with a data-driven approach. Typical approaches have leveraged dimensionality reduction techniques by learning a reduced-dimensional subspace of the optical flow vectors induced by the egomotion [18]. In [19], Ciarfuglia et al. employ Support Vector Regression (SVR) to recover vehicle egomotion (3-DOF). The authors further build upon their previous result by swapping out the SVR module with an end-to-end trainable convolutional neural network [20] while showing improvements in the overall performance on the KITTI odometry benchmark [21]. Recently, Clarke et al. [22] introduced a visual-inertial odometry solution that takes advantage of a neural-network architecture to learn a mapping from raw inertial measurements and sequential imagery to 6-DOF pose estimates. By posing visual-inertial odometry (VIO) as a sequence-to-sequence learning problem, they developed a neural network architecture that combined convolutional neural networks with Long Short-Term Units (LSTMs) to fuse the independent sensor measurements into a reliable 6-DOF pose estimate for ego-motion. Our work closely relates to these data-driven approaches that have recently been developed. We provide a qualitative comparison of how our approach is positioned within the visual ego-motion estimation landscape in Table I.

Method Type	Varied Optics	Model Free	Robust	Self Supervised
Traditional VO [23]	✗	✗	✓	✗
End-to-end VO [20], [22]	✗	✓	✓	✗
This work	✓	✓	✓	✓

TABLE I: **Visual odometry landscape**: A qualitative comparison of how our approach is positioned amongst existing solutions to ego-motion estimation.

III. Ego-motion regression

As with most ego-motion estimation solutions, it is imperative to determine the minimal parameterization of the underlying motion manifold. In certain restricted scene structures or motion manifolds, several variants of ego-motion estimation are proposed [7], [15], [16], [17]. However, we consider the case of modeling cameras with varied optics and hence are interested in determining the full range of ego-motion, often restricted, that induces the pixel-level optical flow. This allows the freedom to model various unconstrained and partially constrained motions that typically affect the overall robustness of existing ego-motion algorithms. While model-based approaches have shown tremendous progress in accuracy, robustness, and run-time performance, a few recent data-driven approaches have been shown to produce equally compelling results [20], [22], [24]. An adaptive and trainable solution for relative pose estimation or ego-motion can be especially advantageous for several reasons: (i) a general-purpose end-to-end trainable model architecture that applies to a variety of camera optics including pinhole, fisheye, and

catadioptric lenses; (ii) simultaneous and continuous optimization over both ego-motion estimation and camera parameters (intrinsics and extrinsics that are implicitly modeled); and (iii) joint reasoning over resource-aware computation and accuracy within the same architecture is amenable. We envision that such an approach is especially beneficial in the context of bootstrapped (or weakly-supervised) learning in robots, where the supervision in ego-motion estimation for a particular camera can be obtained from the fusion of measurements from other robot sensors (GPS, wheel encoders etc.).

Our approach is motivated by previous minimally parameterized models [7], [17] that are able to recover ego-motion from a *single tracked feature*. We find this representation especially appealing due to the simplicity and flexibility in *pixel-level* computation. Despite the reduced complexity of the input space for the mapping problem, recovering the full 6-DOF ego-motion is ill-posed due to the inherently under-constrained system. However, it has been previously shown that under non-holonomic vehicle motion, camera ego-motion may be fully recoverable up to a sufficient degree of accuracy using a single point [7], [17].

We now focus on the specifics of the ego-motion regression objective. Due to the under-constrained nature of the prescribed regression problem, the pose estimation is modeled as a density estimation problem over the range of possible ego-motions[1], conditioned on the input flow features. It is important to note that the output of the proposed model is a density estimate $p(\hat{\mathbf{z}}_{t-1,t}|\mathbf{x}_{t-1,t})$ for every feature tracked between subsequent frames.

A. Density estimation for ego-motion

In typical associative mapping problems, the joint probability density $p(\mathbf{x}, \mathbf{z})$ is decomposed into the product of two terms: (i) $p(\mathbf{z}|\mathbf{x})$: the conditional density of the target pose $\mathbf{z} \in SE(3)$ conditioned on the input feature correspondence $\mathbf{x} = (\boldsymbol{x}, \Delta\boldsymbol{x})$ obtained from sparse optical flow (KLT) [25] (ii) $p(\mathbf{x})$: the unconditional density of the input data $\mathbf{x}$. While we are particularly interested in the first term $p(\mathbf{z}|\mathbf{x})$ that predicts the range of possible values for $\mathbf{z}$ given new values of $\mathbf{x}$, we can observe that the density $p(\mathbf{x}) = \sum_z p(\mathbf{x}, \mathbf{z}) d\mathbf{z}$ provides a measure of how well the prediction is captured by the trained model.

The critical component in estimating the ego-motion belief is the ability to accurately predict the conditional probability distribution $p(\mathbf{z}|\mathbf{x})$ of the pose estimates that is induced by the given input feature $\boldsymbol{x}$ and the flow $\Delta\boldsymbol{x}$. Due to its powerful and rich modeling capabilities, we use a *Mixture Density Network* (MDN) [26] to parametrize the conditional density estimate. MDNs are a class of end-to-end trainable (fully-differentiable) density estimation techniques that leverage conventional neural networks to regress the parameters of a generative model such as a finite Gaussian Mixture Model (GMM). The powerful representational capacity of neural networks coupled with rich probabilistic modeling that GMMs admit, allows us to model multi-valued or multi-modal beliefs that typically arise in inverse problems such as visual ego-motion.

For each of the F input flow features $\mathbf{x}_i$ extracted via KLT, the conditional probability density of the target pose data $\mathbf{z}_i$ (Eqn 1) is represented as a convex combination of K Gaussian components,

$$p(\mathbf{z}_i \mid \mathbf{x}_i) = \sum_{k=1}^{K} \pi_k(\mathbf{x}_i) \mathcal{N}(\mathbf{z} \mid \mu_k(\mathbf{x}_i), \sigma_k^2(\mathbf{x}_i)) \tag{1}$$

where $\pi_k(\mathbf{x})$ is the mixing coefficient for the k-th component as specified in a typical GMM. The Gaussian kernels are parameterized by their mean vector $\mu_k(\mathbf{x})$ and diagonal covariance $\sigma_k(\mathbf{x})$. It is important to note that the parameters $\pi_k(\mathbf{x}), \mu_k(\mathbf{x})$, and $\sigma_k(\mathbf{x})$ are general and continuous functions of $\mathbf{x}$. This allows us to model these parameters as the output (a^π, a^μ, a^σ) of a conventional neural network which takes $\mathbf{x}$ as its input. Following [26], the outputs of the neural network are constrained as follows: (i) The mixing coefficients must sum to 1, i.e. $\sum_K \pi_k(\mathbf{x}) = 1$ where $0 \leq \pi_k(\mathbf{x}) \leq 1$. This is accomplished via the *softmax* activation as seen in Eqn 2. (ii) Variances $\sigma_k(\mathbf{x})$ are strictly positive via the *exponential* activation (Eqn 3).

$$\pi_k(\mathbf{x}) = \frac{\exp(a_k^\pi)}{\sum_{l=1}^{K} \exp(a_l^\pi)} \tag{2}$$

$$\sigma_k(\mathbf{x}) = \exp(a_k^\sigma), \quad \mu_k(\mathbf{x}) = a_k^\mu \tag{3}$$

$$\mathcal{L}_{\mathcal{MDN}} = -\sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k(\mathbf{x}_n) \mathcal{N}(\mathbf{z} \mid \mu_k(\mathbf{x}_n), \sigma_k^2(\mathbf{x}_n)) \right\} \tag{4}$$

The proposed model is learned end-to-end by maximizing the data log-likelihood, or alternatively minimizing the negative log-likelihood (denoted as $\mathcal{L}_{\mathcal{MDN}}$ in Eqn 4), given the F input feature tracks ($\mathbf{x}_1 \ldots \mathbf{x}_F$) and expected ego-motion estimate $\mathbf{z}$. The resulting ego-motion density estimates $p(\mathbf{z}_i|\mathbf{x}_i)$ obtained from each individual flow vectors $\mathbf{x}_i$ are then fused by taking the product of their densities. However, to maintain tractability of density products, only the mean and covariance corresponding to the largest mixture coefficient (i.e. most likely mixture mode) for each feature is considered for subsequent trajectory optimization (See Eqn 5).

$$p(\mathbf{z}|\mathbf{x}) \simeq \prod_{i=1}^{F} \max_k \left\{ \pi_k(\mathbf{x}_i) \mathcal{N}(\mathbf{z}_i \mid \mu_k(\mathbf{x}_i), \sigma_k^2(\mathbf{x}_i)) \right\} \tag{5}$$

B. Trajectory optimization

While minimizing the MDN loss ($\mathcal{L}_{MDN}$) as described above provides a reasonable regressor for ego-motion estimation, it is evident that optimizing frame-to-frame measurements do not ensure long-term consistencies in the ego-motion trajectories obtained by integrating these regressed estimates. As one expects, the integrated trajectories are sensitive to even negligible biases in the ego-motion regressor.

[1] Although the parametrization is maintained as $SE(3)$, it is important to realize that the nature of most autonomous car datasets involve a lower-dimensional ($SE(2)$) motion manifold

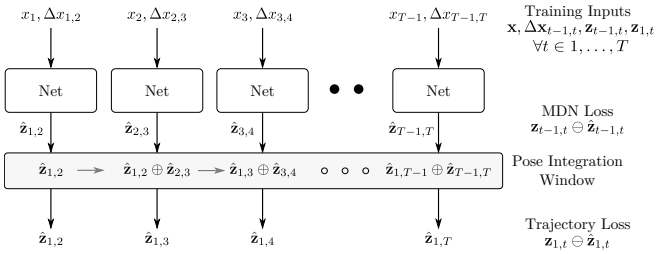


Fig. 2: **Windowed trajectory optimization**: An illustration of the losses introduced for training frame-to-frame ego-motion (*local*) and windowed ego-motion (*global*) by compounding the poses determined from each of the individual frame-to-frame measurements.

Two-stage optimization: To circumvent the aforementioned issue, we introduce a second optimization stage that jointly minimizes the *local* objective ($\mathcal{L}_{MDN}$) with a *global* objective that minimizes the error incurred between the overall trajectory and the trajectory obtained by integrating the regressed pose estimates obtained via the *local* optimization. This allows the *global* optimization stage to have a warm-start with an almost correct initial guess for the network parameters.

As seen in Eqn 6, $\mathcal{L}_{TRAJ}$ pertains to the overall trajectory error incurred by integrating the individual regressed estimates over a batched window (we typically consider 200 to 1000 frames). This allows us to fine-tune the regressor to predict valid estimates that integrate towards accurate long-term ego-motion trajectories. As expected, the model is able to roughly learn the curved trajectory path, however, it is not able to make accurate predictions when integrated for longer time-windows (due to the lack of the *global* objective loss term in Stage 1). Figure 2 provides a high-level overview of the input-output relationships of the training procedure, including the various network losses incorporated in the ego-motion encoder/regressor. For illustrative purposes only, we refer the reader to Figure 3 where we validate this two-stage approach over a simulated dataset [27].

In Eqn 6, $\hat{\mathbf{z}}_{t-1,t}$ is the frame-to-frame ego-motion estimate and the regression target/output of the MDN function F, where $F : \mathbf{x} \mapsto \Big(\mu(\mathbf{x}_{t-1,t}), \sigma(\mathbf{x}_{t-1,t}), \pi(\mathbf{x}_{t-1,t})\Big)$. $\hat{\mathbf{z}}_{1,t}$ is the overall trajectory predicted by integrating the individually regressed frame-to-frame ego-motion estimates and is defined by $\hat{\mathbf{z}}_{1,t} = \hat{\mathbf{z}}_{1,2} \oplus \hat{\mathbf{z}}_{2,3} \oplus \cdots \oplus \hat{\mathbf{z}}_{t-1,t}$.

$$\mathcal{L}_{\text{ENC}} = \underbrace{\sum_t \mathcal{L}^t_{MDN}\Big(F(\mathbf{x}), \mathbf{z}_{t-1,t}\Big)}_{\text{MDN Loss}} + \underbrace{\sum_t \mathcal{L}^t_{TRAJ}(\mathbf{z}_{1,t} \ominus \hat{\mathbf{z}}_{1,t})}_{\text{Overall Trajectory Loss}} \tag{6}$$

C. Bootstrapped learning for ego-motion estimation

Typical robot navigation systems consider the fusion of visual odometry estimates with other modalities including estimates derived from wheel encoders, IMUs, GPS etc. Considering odometry estimates (for e.g. from wheel encoders) as-is, the uncertainties in open-loop chains grow in an unbounded manner. Furthermore, relative pose estimation may also be inherently biased due to calibration errors that eventually contribute to the overall error incurred. GPS, despite being noise-ridden, provides an absolute sensor

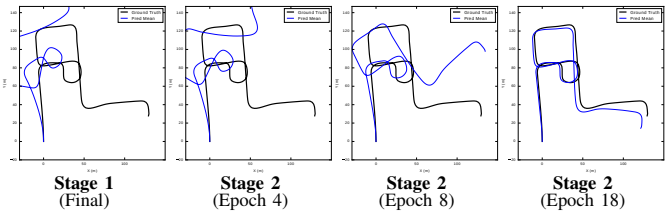


Fig. 3: **Two-stage Optimization**: An illustration of the two-stage optimization procedure. The *first* column shows the final solution after the first stage. Despite the minimization, the integrated trajectory is clearly biased and poorly matches the expected result. The *second*, *third* and *fourth* column shows the gradual improvement of the second stage (global minimization) and matches the expected ground truth trajectory better (i.e. estimates the regressor biases better).

reference measurement that is especially complementary to the open-loop odometry chain maintained with odometry estimates. The probabilistic fusion of these two relatively uncorrelated measurement modalities allows us to recover a sufficiently accurate trajectory estimate that can be directly used as ground truth data $\mathbf{z}$ (in Figure 4) for our supervised regression problem.

The indirect recovery of training data from the fusion of other sensor modalities in robots falls within the *self-supervised or bootstrapped* learning paradigm. We envision this capability to be especially beneficial in the context of life-long learning in future autonomous systems. Using the fused and optimized pose estimates $\mathbf{z}$ (recovered from GPS and odometry estimates), we are able to recover the required input-output relationships for training visual ego-motion for a completely new sensor (as illustrated in Figure 4). Figure 5 illustrates the realization of the learned model in a typical autonomous system where it is treated as an additional sensor

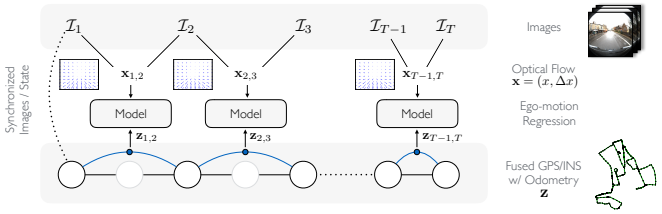


Fig. 4: **Bootstrapped Ego-motion Regression**: Illustration of the bootstrap mechanism whereby a robot self-supervises the proposed ego-motion regression task in a new camera sensor by fusing information from other sensor sources such as GPS and INS.

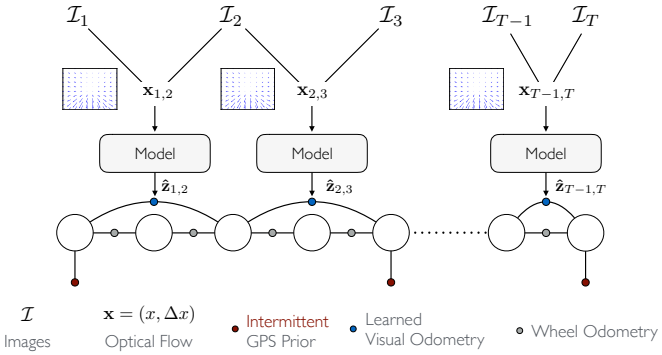


Fig. 5: **Learned Ego-motion Deployment**: During model deployment, the learned visual-egomotion model provides valuable relative pose constraints to augment the standard navigation-based sensor fusion (GPS/INS and wheel encoder odometry fusion).

source. Through experiments IV-C, we illustrate this concept with the recovery of ego-motion in a robot car equipped with a GPS/INS unit and a single camera.

D. Introspective Reasoning for Scene-Flow Prediction

Scene flow is a fundamental capability that provides directly measurable quantities for ego-motion analysis. The flow observed by sensors mounted on vehicles is a function of the inherent scene depth, the relative ego-motion undergone by the vehicle, and the intrinsic and extrinsic properties of the camera used to capture it. As with any measured quantity, one needs to deal with sensor-level noise propagated through the model in order to provide robust estimates. While the input flow features are an indication of ego-motion, some of the features may be corrupted due to lack of or ambiguous visual texture or due to flow induced by the dynamics of objects other than the ego-motion itself. Evidently, we observe that the dominant flow is generally induced by ego-motion itself, and it is this flow that we intend to fully recover via a conditional variational auto-encoder (C-VAE). By inverting the regression problem, we develop a generative model able to predict the most-likely flow $\hat{\Delta x}$ induced given an ego-motion estimate $\mathbf{z}$, and feature location x. We propose a scene-flow specific autoencoder that encodes the implicit egomotion observed by the sensor, while jointly reasoning over the latent depth of each of the individual tracked features.

$$\begin{aligned}\mathcal{L}_{\text{CVAE}} =& \mathbb{E}\big[\log p_\theta(\Delta x|\mathbf{z}, x)\big] \\ &- D_{KL}\big[q_\phi(\mathbf{z}|x, \Delta x)||p_\theta(\mathbf{z}|x)\big]\end{aligned} \tag{7}$$

Through the proposed denoising autoencoder model, we are also able to attain an introspection mechanism for the presence of outliers. We incorporate this additional module via an auxiliary loss as specified in Eqn 7. An illustration of these flow predictions are shown in Figure 8.

IV. Experiments

In this section, we provide detailed experiments on the performance, robustness and flexibility of our proposed approach on various datasets. Our approach differentiates itself from existing solutions on various fronts as shown in Table I. We evaluate the performance of our proposed approach on various publicly-available datasets including the KITTI dataset [21], the Multi-FOV synthetic dataset [27] (pinhole, fisheye, and catadioptric lenses), an omnidirectional-camera dataset [28], and on the Oxford Robotcar 1000km Dataset [29].

Navigation solutions in autonomous systems today typically fuse various modalities including GPS, odometry from wheel encoders and INS to provide robust trajectory estimates over extended periods of operation. We provide a similar solution by leveraging the learned ego-motion capability described in this work, and fuse it with intermittent GPS updates[2] (Secion IV-A). While maintaining similar performance capabilities (Table II), we re-emphasize the benefits of our approach over existing solutions:

- **Versatile**: With a fully trainable model, our approach is able to simultaneously reason over both ego-motion and implicitly modeled camera parameters (*intrinsics* and *extrinsics*). Furthermore, online calibration and parameter tuning is implicitly encoded within the same learning framework.
- **Model-free**: Without imposing any constraints on the type of camera optics, our approach is able to recover ego-motions for a variety of camera models including *pinhole*, *fisheye* and *catadioptric* lenses. (Section IV-B)
- **Bootstrapped training and refinement**: We illustrate a bootstrapped learning example whereby a robot self-supervises the proposed ego-motion regression task by fusing information from other sensor sources including GPS and INS (Section IV-C)
- **Introspective reasoning for scene-flow prediction**: Via the C-VAE generative model, we are able to reason/introspect over the predicted flow vectors in the image given an ego-motion estimate. This provides an obvious advantage in *robust* outlier detection and identifying dynamic objects whose flow vectors need to be disambiguated from the ego-motion scene flow (Figure 8)

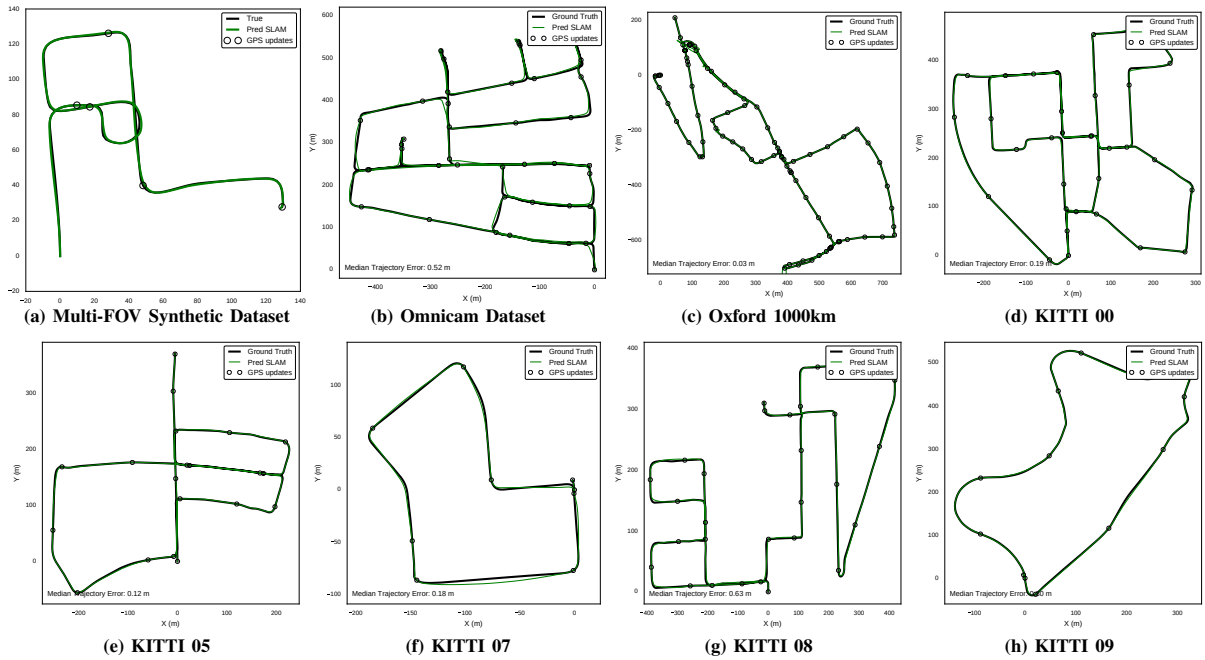
A. Evaluating ego-motion performance with sensor fusion

In this section, we evaluate our approach against a few state-of-the-art algorithms for monocular visual odometry [4]. On the KITTI dataset [21], the pre-trained estimator is used to robustly and accurately predict ego-motion from KLT features tracked over the dataset image sequence. The frame-to-frame ego-motion estimates are integrated for each session to recover the full trajectory estimate and simultaneously fused with intermittent GPS updates (incorporated every 150 frames). In Figure 6, we show the qualitative performance in the overall trajectory obtained with our method. The entire pose-optimized trajectory is compared against the ground truth trajectory. The translational errors are computed for each of the ground truth and prediction pose pairs, and their median value is reported in Table II for a variety of datasets with varied camera optics.

B. Varied camera optics

Most of the existing implementations of VO estimation are restricted to a class of camera optics, and generally avoid implementing a general-purpose VO estimator for varied camera optics. Our approach on the other hand, has shown the ability to provide accurate VO with intermittent GPS trajectory estimation while simultaneously being applicable to a varied range of camera models. In Figure 7, we compare with intermittent GPS trajectory estimates for all three camera models, and verify their performance accuracy compared to ground truth. In our experiments, we found that while our proposed solution was sufficiently powerful to model different camera optics, it was significantly better at modeling pinhole lenses as compared to fisheye and

[2]For evaluation purposes only, the absolute ground truth locations were added as weak priors on datasets without GPS measurements



(a) Multi-FOV Synthetic Dataset **(b) Omnicam Dataset** **(c) Oxford 1000km** **(d) KITTI 00**

(e) KITTI 05 **(f) KITTI 07** **(g) KITTI 08** **(h) KITTI 09**

Fig. 6: **Sensor fusion with learned ego-motion**: On fusing our proposed VO method with intermittent GPS updates (every 150 frames, black circles), the pose-graph optimized ego-motion solution (in green) achieves sufficiently high accuracy relative to ground truth. We test on a variety of publicly-available datasets including **(a)** Multi-FOV synthetic dataset [27] (*pinhole* shown above), **(b)** an omnidirectional-camera dataset [28], **(c)** Oxford Robotcar 1000km Dataset [29] (2015-11-13-10-28-08) **(d-h)** KITTI dataset [21]. *Weak supervision* such as GPS measurements can be especially advantageous in recovering improved estimates for localization, while simultaneously minimizing uncertainties associated with pure VO-based approaches.

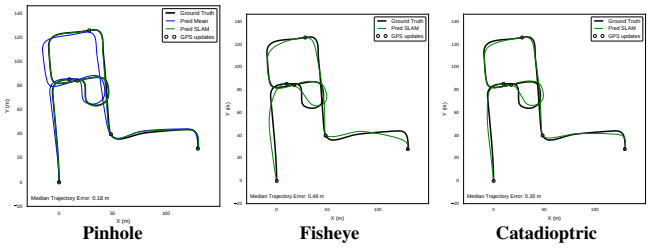
Dataset	Camera Optics	Median Trajectory Error
KITTI-00	Pinhole	0.19 m
KITTI-02	Pinhole	0.30 m
KITTI-05	Pinhole	0.12 m
KITTI-07	Pinhole	0.18 m
KITTI-08	Pinhole	0.63 m
KITTI-09	Pinhole	0.30 m
Multi-FOV [27]	Pinhole	0.18 m
Multi-FOV [27]	Fisheye	0.48 m
Multi-FOV [27]	Catadioptric	0.36 m
Omnidirectional [28]	Catadioptric	0.52 m
Oxford 1000km† [29]	Pinhole	0.03 m

TABLE II: **Trajectory prediction performance**: An illustration of the trajectory prediction performance of our proposed ego-motion approach when fused with intermittent GPS updates (every 150 frames). The errors are computed across the entire length of the optimized trajectory and ground truth. For Oxford 1000km dataset, we only evaluate on a single session (2015-11-13-10-28-08 [80GB]: †Stereo Centre)

catadioptric cameras (See Table II). In future work, we would like to investigate further extensions that improve the accuracy for both fisheye and catadioptric lenses.

C. Self-supervised Visual Ego-motion Learning in Robots

We envision the capability of robots to self-supervise tasks such as visual ego-motion estimation to be especially beneficial in the context of life-long learning and autonomy. We experiment and validate this concept through a concrete example using the 1000km Oxford Robot Car dataset [29]. We train the task of visual ego-motion on a new camera sensor by leveraging the fused GPS and INS information collected on the robot car as ground truth trajectories (6-DOF), and extracting feature trajectories (via KLT) from image sequences obtained from the new camera sensor. The timestamps from the cameras are synchronized with respect



Pinhole **Fisheye** **Catadioptric**

Fig. 7: **Varied camera optics:** An illustration of the performance of our general-purpose approach for varied camera optics (pinhole, fisheye, and catadioptric lenses) on the Multi-FOV synthetic dataset [27]. Without any prior knowledge on the camera optics, or the mounting configuration (extrinsics), we are able to robustly and accurately recover the full trajectory of the vehicle (with intermittent GPS updates every 500 frames).

to the timestamps of the fused GPS and INS information, in order to obtain a one-to-one mapping for training purposes. We train on the `stereo_centre` *(pinhole)* camera dataset and present our results in Table II. As seen in Figure 6, we are able to achieve considerably accurate long-term state estimates by fusing our proposed visual ego-motion estimates with even sparser GPS updates (every 2-3 seconds, instead of 50Hz GPS/INS readings). This allows the robot to reduce its reliance on GPS/INS alone to perform robust, long-term trajectory estimation.

D. Implementation Details

In this section we describe the details of our proposed model, training methodology and parameters used. The input $\mathbf{x} = (\boldsymbol{x}, \Delta\boldsymbol{x})$ to the density-based ego-motion estimator are feature tracks extracted via (Kanade-Lucas-Tomasi) KLT feature tracking over the raw camera image sequences. The input feature positions and flow vectors are normalized to

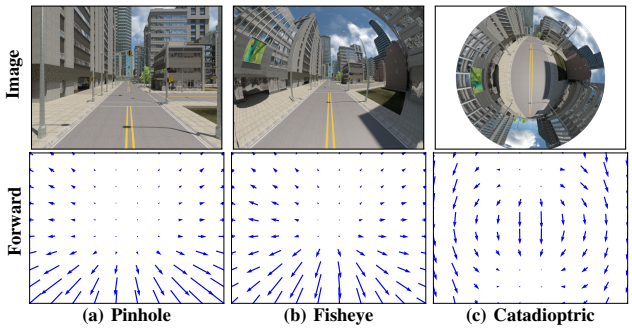


Fig. 8: **Introspective reasoning for scene-flow prediction**: Illustrated above are the dominant flow vectors corresponding to scene-flow given the corresponding ego-motion. While this module is not currently used in the ego-motion estimation, we expect it to be critical in outlier rejection. **Row 1**: Sample image from camera, **Row 2**: Flow induced by forward motion

be the in range of $[-1, 1]$ using the dimensions of the input image. We evaluate sparse LK (Lucas-Kanade) optical flow over 7 pyramidal scales with a scale factor of $\sqrt{2}$. As the features are extracted, the corresponding robot pose (either available via GPS or GPS/INS/wheel odometry sensor fusion) is synchronized and recorded in $SE(3)$ for training purposes. The input KLT features, and the corresponding relative pose estimates used for training are parameterized as $\mathbf{z} = (\mathbf{t}, \mathbf{r}) \in \mathbb{R}^6$, with a Euclidean translation vector $\mathbf{t} \in \mathbb{R}^3$ and an Euler rotation vector $\mathbf{r} \in \mathbb{R}^3$.

Network and training: The proposed architecture consists of a set of fully-connected stacked layers (with 1024, 128 and 32 units) followed by a Mixture Density Network with 32 hidden units and 5 mixture components (K). Each of the initial fully-connected layers implement *tanh* activation after it, followed by a dropout layer with a dropout rate of 0.1. The final output layer of the MDN (a^π, a^μ, a^σ) consists of $(O + 2) * K$ outputs where O is the desired number of states estimated.

The network is trained (in Stage 1) with loss weights of 10, 0.1, 1 corresponding to the losses $\mathcal{L}_{MDN}, \mathcal{L}_{TRAJ}, \mathcal{L}_{CVAE}$ described in previous sections. The training data is provided in batches of 100 frame-to-frame subsequent image pairs, each consisting of approximately 50 randomly sampled feature matches via KLT. The learning rate is set to $1\mathrm{e}{-3}$ with Adam as the optimizer. On the synthetic Multi-FOV dataset and the KITTI dataset, training for most models took roughly an hour and a half (3000 epochs) independent of the KLT feature extraction step.

Two-stage optimization: We found the one-shot joint optimization of the *local* ego-motion estimation and *global* trajectory optimization to have sufficiently low convergence rates during training. One possible explanation is the high sensitivity of the loss weight parameters that is used for tuning the local and global losses into a single objective. As previously addressed in Section III-B, we separate the training into two stages thereby alleviating the aforementioned issues, and maintaining fast convergence rates in Stage 1. Furthermore, we note that during the second stage, it only requires a few tens of iterations for sufficiently accurate ego-motion trajectories. In order to optimize over a larger time-window in stage 2, we set the batch size to 1000 frame-to-frame image matches, again randomly sampled from the training set as before. Due to the large integration window and memory limitations, we train this stage purely on the CPU for only 100 epochs each taking roughly 30s per epoch. Additionally, in stage 2, the loss weights for $\mathcal{L}_{TRAJ}$ are increased to 100 in order to have faster convergence to the *global* trajectory. The remaining loss weights are left unchanged.

Trajectory fusion: We use GTSAM[3] to construct the underlying factor graph for pose-graph optimization. Odometry constraints obtained from the frame-to-frame ego-motion are incorporated as a 6-DOF constraint parameterized in $SE(3)$ with $1 * 10^{-3}$ rad rotational noise and $5 * 10^{-2}$ m translation noise. As with typical autonomous navigation solutions, we expect measurement updates in the form of GPS (absolute reference updates) in order to correct for the long-term drift incurred in open-loop odometry chains. We incorporate absolute prior updates only every 150 frames, with a weak translation prior of 0.01 m. The constraints are incrementally added and solved using iSAM2 [30] as the measurements are streamed in, with updates performed every 10 frames.

While the proposed MDN is parametrized in Euler angles, the *trajectory integration module* parameterizes the rotation vectors in quaternions for robust and unambiguous long-term trajectory estimation. All the rigid body transformations are implemented directly in Tensorflow for pure-GPU training support.

Run-time performance: We are particularly interested in the run-time / test-time performance of our approach on CPU architectures for mostly resource-constrained settings. Independent of the KLT feature tracking run-time, we are able to recover ego-motion estimates at roughly 3ms on a consumer-grade Intel(R) Core(TM) i7-3920XM CPU @ 2.90GHz.

Source code and Pre-trained weights: We implemented the MDN-based ego-motion estimator with Keras and Tensorflow, and trained our models using a combination of CPUs and GPUs (NVIDIA Titan X). All the code was trained on an server-grade Intel(R) Xeon(R) CPU E5-2630 v3 @ 2.40GHz and tested on the same consumer-grade machine as mentioned above to emulate potential real-world use-cases. The source code and pre-trained models used will be made available shortly[4].

V. Discussion

The initial results in bootstrapped learning for visual ego-motion has motivated new directions towards life-long learning in autonomous robots. While our visual ego-motion model architecture is shown to be sufficiently powerful to recover ego-motions for non-linear camera optics such as fisheye and catadioptric lenses, we continue to investigate further improvements to match existing state-of-the-art models for these lens types. Our current model does not capture distortion effects yet, however, this is very much a

[3] http://collab.cc.gatech.edu/borg/gtsam

[4] See http://people.csail.mit.edu/spillai/learning-egomotion and https://github.com/spillai/learning-egomotion

future direction we would like to take. Another consideration is the resource-constrained setting, where the optimization objective incorporates an additional regularization term on the number of parameters used, and the computation load consumed. We hope for this resource-aware capability to transfer to real-world limited-resource robots and to have a significant impact on the adaptability of robots for long-term autonomy.

VI. Conclusion

While many visual ego-motion algorithm variants have been proposed in the past decade, we envision that a fully end-to-end trainable algorithm for generic camera ego-motion estimation shall have far-reaching implications in several domains, especially autonomous systems. Furthermore, we expect our method to seamlessly operate under resource-constrained situations in the near future by leveraging existing solutions in model reduction and dynamic model architecture tuning. With the availability of multiple sensors on these autonomous systems, we also foresee our approach to bootstrapped task (visual ego-motion) learning to potentially enable robots to learn from experience, and use the new models learned from these experiences to encode redundancy and fault-tolerance all within the same framework.

References

[1] David Nistér, Oleg Naroditsky, and James Bergen. Visual odometry. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–652. IEEE, 2004. 1, 2

[2] Kurt Konolige, Motilal Agrawal, and Joan Sola. Large-scale visual odometry for rough terrain. In *Robotics research*, pages 201–212. Springer, 2010. 1

[3] 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. 1

[4] Bernd Kitt, Andreas Geiger, and Henning Lategahn. Visual odometry based on stereo image sequences with RANSAC-based outlier rejection scheme. In *Intelligent Vehicles Symposium*, pages 486–492, 2010. 1, 5

[5] Gim Hee Lee, Friedrich Faundorfer, and Marc Pollefeys. Motion estimation for self-driving cars with a generalized camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2746–2753, 2013. 1

[6] Laurent Kneip, Paul Furgale, and Roland Siegwart. Using multi-camera systems in robotics: Efficient solutions to the n-PnP problem. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 3770–3776. IEEE, 2013. 1

[7] Davide Scaramuzza. 1-point-RANSAC structure from motion for vehicle-mounted cameras by exploiting non-holonomic constraints. *Int'l J. of Computer Vision*, 95(1):74–85, 2011. 1, 2, 3

[8] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustment A modern synthesis. In *International workshop on vision algorithms*, pages 298–372. Springer, 1999. 2

[9] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 2

[10] Hans P Moravec. Obstacle avoidance and navigation in the real world by a seeing robot rover. Technical report, DTIC Document, 1980. 2

[11] Larry Henry Matthies. Dynamic stereo vision. 1989. 2

[12] Clark F Olson, Larry H Matthies, H Schoppers, and Mark W Maimone. Robust stereo ego-motion for long distance navigation. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 453–458. IEEE, 2000. 2

[13] Martin A Fischler and Robert C Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 2

[14] Peter Corke, Dennis Strelow, and Sanjiv Singh. Omnidirectional visual odometry for a planetary rover. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 4, pages 4007–4012. IEEE, 2004. 2

[15] Bojian Liang and Nick Pears. Visual navigation using planar homographies. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 1, pages 205–210. IEEE, 2002. 2

[16] Qifa Ke and Takeo Kanade. Transforming camera geometry to a virtual downward-looking camera: Robust ego-motion estimation and ground-layer detection. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1, pages I–390. IEEE, 2003. 2

[17] Davide Scaramuzza, Friedrich Fraundorfer, and Roland Siegwart. Real-time monocular visual odometry for on-road vehicles with 1-point RANSAC. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 4293–4299. IEEE, 2009. 2, 3

[18] Richard Roberts, Christian Potthast, and Frank Dellaert. Learning general optical flow subspaces for egomotion estimation and detection of motion anomalies. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 57–64. IEEE, 2009. 2

[19] Thomas A Ciarfuglia, Gabriele Costante, Paolo Valigi, and Elisa Ricci. Evaluation of non-geometric methods for visual odometry. *Robotics and Autonomous Systems*, 62(12):1717–1730, 2014. 2

[20] Gabriele Costante, Michele Mancini, Paolo Valigi, and Thomas A Ciarfuglia. Exploring Representation Learning With CNNs for Frame-to-Frame Ego-Motion Estimation. *IEEE Robotics and Automation Letters*, 1(1):18–25, 2016. 2

[21] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2012. 2, 5, 6

[22] Ronald Clark, Sen Wang, Hongkai Wen, Andrew Markham, and Niki Trigoni. VINet: Visual-Inertial odometry as a sequence-to-sequence learning problem. AAAI, 2016. 2

[23] Davide Scaramuzza and Friedrich Fraundorfer. Visual odometry [tutorial]. *IEEE robotics & automation magazine*, 18(4):80–92, 2011. 2

[24] Kishore Konda and Roland Memisevic. Learning visual odometry with a convolutional network. In *International Conference on Computer Vision Theory and Applications*, 2015. 2

[25] Stan Birchfield. KLT: An implementation of the Kanade-Lucas-Tomasi feature tracker, 2007. 3

[26] Christopher M Bishop. Mixture Density Networks. 1994. 3

[27] Zichao Zhang, Henri Rebecq, Christian Forster, and Davide Scaramuzza. Benefit of large field-of-view cameras for visual odometry. In *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016. 4, 5, 6

[28] Miriam Schönbein and Andreas Geiger. Omnidirectional 3d reconstruction in augmented manhattan worlds. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 716–723. IEEE, 2014. 5, 6

[29] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000 km: The Oxford RobotCar dataset. *Int'l J. of Robotics Research*, page 0278364916679498, 2016. 5, 6

[30] Michael Kaess, Hordur Johannsson, Richard Roberts, Viorela Ila, John J Leonard, and Frank Dellaert. iSAM2: Incremental smoothing and mapping using the bayes tree. *Int'l J. of Robotics Research*, 31(2):216–235, 2012. 7