Ego motion estimation using optical flow equations from multi-camera setup

Ajith Sylvester, Kumar Selvakumaran, Parth Mahajan, Sakshi Bhatia, Utkarsh Rai

Northeastern University
{*sylvester.aj, selvakumaran.k, mahajan.parth, bhatia.saks, rai.ut*}*@northeastern.edu*

***Abstract*—Ego-motion estimation is a fundamental problem in robotics, which is essential for understanding the movement of a robot relative to its environment. Traditional correspondence-based methods for calculating ego motion rely on explicit feature matching, which can be challenging and prone to errors in scenarios with sparse or noisy feature sets. Optical flow, a technique commonly used in single-camera setups, offers a direct approach by analyzing pixel level intensity variations to infer motion estimates. However, it faces limitations in resolving scale ambiguity in motion estimates while handling pure rotational motion or restricted fields of view. This work aims to decipher an approach to estimating ego-motion from the flow-fields of multiple cameras placed in a static configuration on a moving rigid body robot.**

I. Introduction

Determining the velocity of the state space of a moving platform i.e. its ego-motion is critical for navigation in mobile robots of all kinds, yet achieving accurate and robust results remains challenging. Persistent issues include scale uncertainty, solution ambiguity, and limited precision. Conventional correspondence-based methods that rely on stable feature matches, becomes unreliable under sparse or erroneous point conditions. While optical flow directly encodes motion in image intensities, single-camera configurations often suffer under pure rotation or restricted fields of view. In response, this work investigates a multi-camera approach that combines flow-fields to enhance ego-motion estimation accuracy.

Our inspiration for this project [1] conducts it's experiments on idealized data which is not representative of real-world datasets. We implement and test various elements of the algorithm on the Nuscenes dataset [2]. The robot used to curate this dataset is an autonomous car with multiple cameras with unique orientations (extrinsics) and intrinsics which was driven in urban streets. Our key contributions in this project can be summarized as:

- **Derivation of scale-free motion equations**: We derive motion equations that eliminate depth dependence, allowing ego-motion estimation without explicit range or feature correspondences.
- **Geometric justification**: We provide a solid mathematical and geometric basis linking optical flow to rotational and translational velocities, ensuring a deeper theoretical understanding.
- **Real-world evaluation**: We test the theory on a real-world dataset (NuScenes), revealing challenges like noise sensitivity and degeneracies within our multi-camera setup.

II. Motivation

Earlier attempts at ego-motion estimation were mainly composed of single-camera methodologies. However, scale ambiguity from unknown depth is encountered with single-camera optical flow estimation. For example, for a vehicle moving forward, the optical flow fields generated by pure translation and pure rotation can appear quite similar in a single-camera setup [1]. Multiple cameras provide a more comprehensive view of the scene and precise 3D motion estimates. Using a multi-camera setup with different viewpoints can help us better characterize the translational and rotational motion of the ego vehicle.

III. Literature Survey

Traditional approaches to ego-motion estimation, such as Visual Odometry (VO) [6] and Simultaneous Localization and Mapping (SLAM) [7], rely heavily on feature matching, depth estimation, and optimization techniques. While effective, these methods are computationally intensive and often struggle in poor-texture or dynamic environments. As an efficient alternative, optical flow-based ego-motion estimation in multi-camera setups has gained traction. Early foundational methods by Horn and Schunck [8] and Lucas and Kanade [9] introduced dense and sparse optical flow techniques, later adapted for multi-camera configurations. Studies, such as those by Scaramuzza et al.[10], demonstrate that multi-camera systems capture a wider field of view, improving robustness in challenging environments. Recent work by Romera et al.[11] emphasizes the computational efficiency of optical flow on embedded platforms, making it ideal for resource-constrained, real-time applications.

Multi-camera setups excel in rapidly changing environments, isolating background motion effectively, and handling fast rotations (Fang et al)[12]. Furthermore, optical flow has been integrated with SLAM to provide rapid motion updates, highlighting its adaptability and utility in complex visual conditions. The paper "Ego-Motion Estimation Using Optical Flow Fields Observed from Multiple

Cameras" by Tsao et al[13]. was among the first to propose a multi-camera method for ego-motion estimation using optical flow without relying on point correspondence. This approach improves motion accuracy in multi camera setup, overcoming limitations of single-camera setups, and providing robustness in scenarios with dynamic scenes or limited visual features.

IV. Problem Definition

The precise mathematical formulation of the problem is defined here. Explicit notation in Sec. VIII-A is used for all mathematical formulations throughout this work. In a discrete time setting, let t be an arbitrary time step, $\Delta t = 1$, R be the robot-centric frame, K be the number of cameras contained by the robot.Given the following data:

- **Images**: 2 ordered sets of images with the same ordering: S_{t-1}, S_t, where all the images in a set S_t have been captured at the same single time step t. Each of the sets have K images $\{I_1, I_2, I_3 ... I_i ... I_K\}$ where i is the camera index.
- **Camera Extrinsics**: Poses of each camera k in the robot frame R given as a pair of rotation and translation: $({}_R R_k, {}_R b_k)$, where ${}_R R_K \in SO(3)$ and ${}_R b_k \in \mathbb{R}^3$
- **Camera Intrinsics**: The Camera matrices for each camera $K_k \subset \mathbb{R}^{3\times3}$

We aim to estimate:

- **Ego-motion**: The pair of instantaneous angular and translational velocities of the robot at the time t in the frame F which is aligned with the robot-frame R, but static w.r.t the world frame W: i,e $({}_F\omega_R, {}_F\dot{t}_R)$ where ${}_F\omega_R \in \mathbb{R}^3$ and ${}_F\dot{t}_R \in \mathbb{R}^3$

We specifically use the Nuscenes dataset which is a comprehensive collection of autonomous driving data. It comprises 1,000 driving scenes, captured in various urban environments. The dataset includes approximately 1.4 million camera images, providing a vast visual dataset. These images are captured from six cameras placed around the vehicle, offering a comprehensive 360-degree view of the surrounding environment. The NuScenes dataset provides the extrinsic and intrinsic camera parameters as well as the ground truth ego pose, and ego-motion data from the can bus module of the vehicle at 10 Hz. [2]

V. Approach

The ego-estimation algorithm implemented and studied in this work was originally introduced by [1]. The approach hinges on using *perfect* optical flow fields computed from the image sets S_t, and S_{t+1}. More specifically, the algorithm assumes that all the assumptions required to compute optical flow are valid in our given system.

a) The key physical idea: In our system, the relative velocity ${}_k\dot{P}$ of some point ${}_kP$ in the field of view of camera k can be observed from the optical center of the moving camera frame (this frame is mentioned in explicit notation as k). To remove depth ambiguity, we normalize our point ${}_kP$ as $\frac{{}_kP}{{}_kP_z}$

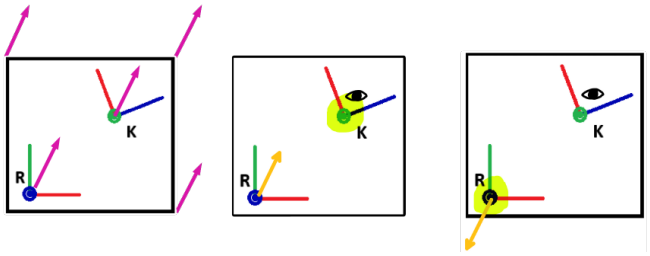

Fig. 1. The relation implied by the adjoint operation: The pink vectors describe ego motion ${}_F\dot{x}_R$, the eye logo indicates the observer's frame, the yellow vector indicates the observed velocity, and the yellow highlight indicates which frame is static. In the middle fig, the frame k is static (i.e. from C) and the motion ${}_F\dot{x}_R$ appears to be towards the frame k. In the third fig, the frame R is static (observing F from k) and the frame k is moving away from it, and hence the points are moving away from the camera frame k (with velocity $-{}_C\dot{x}_k * {}_C x_F$ refer to eq iii for the relation between $\dot{x}$ and x^{-1}.)

and the point velocity ${}_k\dot{P}$ as $\frac{{}_k\dot{P}}{{}_kP_z}$. Also, ${}_ku = \frac{{}_kP}{{}_kP_z}$ which is the image point projected on to the depth normalized plane (refer fig 2). Also, observing from the optical center of camera ${}_kO$, if we were to characterize the velocity $\frac{{}_k\dot{P}}{{}_kP_z}$ into translational component and rotational component about the optical center ${}_kO$ then the rotational component can be written down as:

$$ {}_k\hat{\omega}_{\dot{P}} = {}_ku \times \frac{{}_k\dot{P}}{P_z} $$

As shown in Figure 3, $\frac{{}_k\dot{P}}{{}_kP_z}$s at different depths along the ray from ${}_kO$ through u, have equal ${}_k\hat{\omega}_{\dot{P}}$ components. i.e. **${}_k\hat{\omega}_{\dot{P}}$ is invariant to the point velocities $\frac{{}_k\dot{P}}{{}_kP_z}$ that vary by translations of $\alpha * u$ where** $\alpha \in \mathbb{R}$. This relation can be seen in fig 3, where different instances of $\frac{\dot{P}}{P_z} + (\alpha * u)$ have the same component perpendicular to u.

Hence, this property is used to circumvent (abstract away) the effect of depth/scale of $\frac{{}_k\dot{P}}{{}_kP_z}$ for the estimation of ego-motion. A more detailed explanation of how this is done is given in the following sections.

b) Deriving the relations between ego motion, corresponding camera motion and their actions on points: We make some definitions to facilitate the derivation below.

- **The robot auxiliary frame** F : This coordinate frame is defined to be *aligned* with the robot frame R and *static* with respect to the world frame W. Sec.VIII-B.
- **The camera auxiliary frame** C: This coordinate frame is defined to be *aligned* with the camera frame k and *static* with respect to the world frame W. Sec.VIII-B

The action of the group defined on the product manifold $M \triangleq SO(3) \times \mathbb{R}^3$ representing poses as opted by the paper [1] is $R * P + b$ where $(R, b) \in M$ and $P \in \mathbb{R}^3$. The option of this particular action implies the $SE(3)$ characterization of poses, and hence we first define and derive ego motion with respect to this characterization.

Consider the coordinate frame transformation T_{RF} ($T_{RF} = I$) to denote it as a pose we use the notation ${}_R x_F$ which is read as the pose of the fixed frame in the robot frame R. It can act on a homogenous point $\bar{P} \in \mathbb{R}^4$ as follows:

$$ {}_R\bar{P} = {}_Rx_F * {}_F\bar{P} \tag{1} $$

The relative motion of the point P as observed from the origin of the robot centric frame is given by ${}_R\dot{\bar{P}}$, which can be interpreted as an action of ego motion on the point P, in the fixed robot frame ${}_F\bar{P}$ This can be given as: was derived [I] to be:

$$ {}_R\dot{\bar{P}} = -{}_F\dot{x}_R * {}_F\bar{P} \tag{2} $$

where ${}_F\dot{x}_R \in lie(SE(3))$ is the ego-motion of the robot with components ${}_F\omega_r \in skew(3)$ and ${}_F\dot{t}_R \in \mathbb{R}^3$ being the angular and translational velocities respectively. In non-homogeneous form eq 2 can be written as:

$$ {}_R\dot{P} = -{}_F\omega_R \times {}_FP - {}_F\dot{t}_R \tag{3} $$

Similarly, the homogeneous equation corresponding to camera motion action on points are derived in [II] to be

$$ {}_k\dot{\bar{P}} = -{}_kx_F * {}_F\dot{x}_R * {}_Rx_C * {}_C\bar{P} \tag{4} $$

From the above equation we obtain the instantaneous motion of the camera with respect to the camera auxiliary frame (denoted by ${}_k\dot{x}_C$) as given below:

$$ {}_k\dot{x}_C = {}_kx_F * {}_F\dot{x}_R * {}_Rx_C $$

Using the relation between $\dot{x}$ and $\dot{x^{-1}}$ as derived from eq iii (where $x \in G$, G is a multiplicative group)

$$ -{}_C\dot{x}_k * {}_Cx_F = {}_Cx_F * {}_F\dot{x}_R $$

The physical significance of the above equation is explained in fig 1. The non homogeneous equations are derived in III

$$ {}_k\dot{P} = -({}_RR_k^T * {}_F\omega_R) \times {}_CP - ({}_RR_k^T({}_F\omega_R \times {}_Fb_k + {}_F\dot{t}_R)) \tag{5} $$

$$ {}_k\dot{x}_C = ({}_k\omega_C, {}_k\dot{t}_C) $$

$$ {}_k\omega_C = {}_RR_k^T * {}_F\omega_R $$

$$ {}_k\dot{t}_C = {}_RR_k^T({}_F\omega_R \times {}_Fb_k + {}_F\dot{t}_R) $$

c) Optical flow in the ideal case: As shown in the fig2, we have three planes.

- The object plane: where the actual point and the velocity of the point lie.
- The depth normalized plane (DN): where the point $\frac{P}{P_z}$ lies.
- the Image Plane: Where the actual image and the optical flow lies.

We transform the image points and optical flows from image plane to depth normalized plane by $u = K^{-1}v$, and $\dot{u} = K^{-1}\dot{v}$.

The relationship between the "projected optical flow" ($\dot{u}$) and the point velocity($\dot{P}$) is derived in VIII-C0c given by:

$$ \dot{u} = -{}_k\omega_C \times u - \frac{{}_k\dot{t}_C}{P_z} - \frac{u * \dot{P}_z}{P_z} \tag{6} $$

Fig3 explains the geometrical relationship between $\dot{u}$ and the point velocity($\dot{P}$).

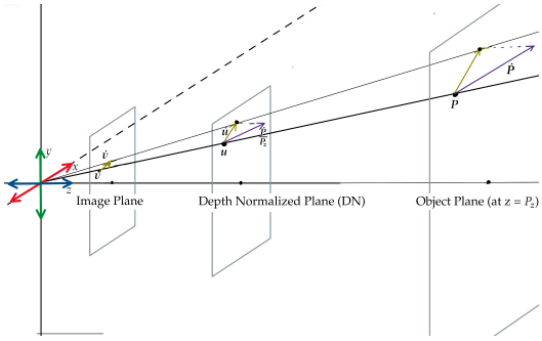

Fig. 2. Representation of the three planes: Image Plane, Depth Normalized Plane, and Object Plane present in our system.

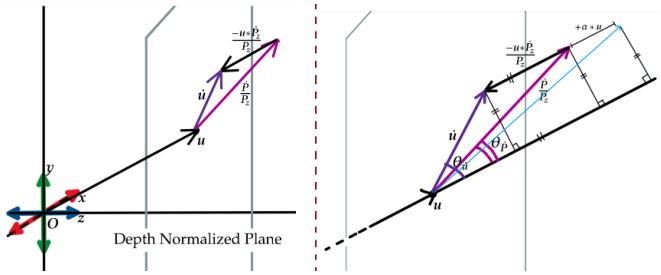

Fig. 3. $\dot{u}$ is the back projection of the optical flow, u is the position vector of the point ${}_kP$ on to the unit-depth plane (Depth normalized plane).

Eq 6 gives the relation between optical flow and the ego motion, in an ideal case(when optical flow is noiseless). there are 3 main terms that compose the ideal optical flow vector namely:

- $-{}_k\omega_C \times u$: the application of ego-motion's angular velocity component on the point u in the depth normalized plane. In figure 4, this term is represented by the orange vector out of the plane ('o' markers denote that vector is out of the *depth normalized plane*)
- $-{}_k\dot{t}_C$: the shift in the optical flow vector due to the translational velocity component of ${}_k\dot{x}_C$. This component is depicted the orange vector into the depth normalized plane in fig 4 (the 'x's indicate that the vector is into the plane.)
- $-\frac{u*\dot{P}_z}{P_z}$ This is the component that brings the relative point velocity vector resulting from ego motion *back to the depth normalized plane*.

The *in-plane* vector $\dot{u}$ is exactly the back projection of the optical flow vector observed in the image plane. This 'splatting' component is the yellow vector that points out of the depth normalized plane in fig 4.

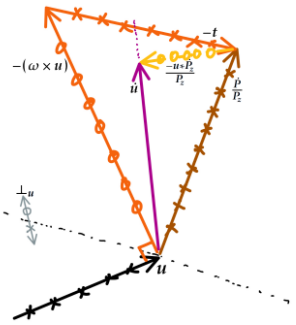

Fig. 4. The figure represents the decomposition of optical flow vector, projected to the DN plane(i.e. $\dot{u} = K^{-1}\dot{v}$) in terms of the components of ego motion (depicted by the orange vectors). The plane of reference here is the depth normalized plane. The 'x' and 'o' markers depict whether that particular vector is into or out of the plane of reference. The dotted line represents the intersection between the plane perpendicular to u, and the depth normalized plane.

d) Elements of the loss function: As mentioned in previous sections, instead of comparing the velocities, we can compare their corresponding $\hat{\omega}$s. the $\hat{\omega}$ of a point velocity vector $\dot{P}$ in frame k is it's angular velocity component about the origin ${}_kO$ (given by ${}_k\hat{\omega}_{\dot{P}}$). In the ideal case

$$ {}_k\hat{\omega}_{\dot{u}} = {}_k\hat{\omega}_{\frac{\dot{P}}{P_z}} $$

$$ {}_k\hat{\omega}_{\dot{u}} = u \times \dot{u} $$

$$ {}_k\hat{\omega}_{\frac{\dot{P}}{P_z}} = u \times \frac{\dot{P}}{P_z} $$

Using the ideal optical flow relation as mentioned in equation 6:

$$ u \times \dot{u} = -u \times ({}_C\omega_k \times u) - u \times \frac{{}_C\dot{t}_k}{{}_kP_z} - u \times \frac{u * {}_k\dot{P}_z}{{}_kP_z} $$

$$ u \times \dot{u} = -u \times ({}_C\omega_k \times u) - u \times \frac{{}_C\dot{t}_k}{{}_kP_z} $$

A dot product with ${}_k\dot{t}_C$ is done on both sides to remove the term involving the depth component of the point ${}_kP_z$. This finally gives us the relation that will be used in our loss function.

$$ \{u \times [\dot{u} + ({}_C\omega_k \times u)]\} \cdot {}_C\dot{t}_k = 0 \tag{7} $$

To bring the equation into the robot frame R and to express in terms of ego motion ${}_F\dot{x}_R$, we can rewrite eq 7 as

$$ {}_RR_k \cdot \{{}_ku \times [{}_k\dot{u} + ({}_RR_k^T\,{}_F\omega_r \times {}_ku)]\} \cdot ({}_F\omega_R \times {}_Rb_k + {}_F\dot{t}_R) = 0 \tag{8} $$

We now characterize the key terms of the loss function as follows:

$$ {}_Rm_{ki} \triangleq {}_RR_k \cdot \{{}_ku \times [{}_k\dot{u} + ({}_RR_k^T\,{}_F\omega_r \times {}_ku)\} \tag{9} $$

$$ {}_Rh_k \triangleq {}_F\omega_R \times {}_Rb_k \tag{10} $$

- $\boldsymbol{m_{ki}}$: xix shows that m_{ki} is an estimate of angular velocity of point u acted on by instantaneous camera translational velocity ${}_C\dot{t}_k$ about the origin ${}_kO$ (this is denoted as ${}_k\hat{\omega}_{{}_k\dot{t}_C}$).
- $\boldsymbol{h_k}$ is the application of the angular velocity component of ego motion on the position the camera ${}_Rb_k$.

e) Formulation of the loss: We have seen that the optimal estimates for ego-motion will abide by the equations mentioned in xx and xxi (in the ideal case, where the optical flow is perfect, and all the assumptions of optical flow is upheld by the system.). Hence the loss function is formulated based on this intention. In the ideal case as per 7 and 8,

$$ {}_Rm_{ki}^T(h_k + t) = 0 $$

, The loss function is formulated as the sum of the inner products of individual ${}_Rm_{ki}$ s and corresponding camera's $(h_k + t)$'s over all cameras k and points i:

$$ J'({}_F\omega_R, {}_F\dot{t}_R) \triangleq \sum_{k=1}^{K}\sum_{i=1}^{N_k} ||{}_Rm_{ki}^T(h_k + t)||^2 \tag{11} $$

The optimization can be formulated as :

$$ \omega^*, \dot{t}^* \triangleq \underset{{}_F\omega_R, {}_W\dot{t}_R}{argmin}(J') $$

The search can be narrowed down to stationary points where $\frac{\partial J'}{\partial {}_F\dot{t}_R} = 0$. By doing so, we obtain a refined loss function in terms of only ${}_F\omega_R$:

$$ J({}_F\omega_R) = -c^TM^{-1}c + \sum_{k=1}^{K}\sum_{i=1}^{N_k}(m_{ki}^Th_k)^2 \tag{12} $$

$$ M \triangleq \sum_{k=1}^{K}\sum_{i=1}^{N_k} m_{ki}m_{ki}^T \tag{13} $$

$$ c \triangleq -\sum_{k=1}^{K}\sum_{i=1}^{N_k} m_{ki}m_{ki}^Th_k \tag{14} $$

$$ {}_F\dot{t}_R \triangleq M^{-1}c \tag{15} $$

f) Physical interpretation of M, c:

- $\boldsymbol{M}$: Consider the matrix D which consists m_{ki}s stacked as rows with shape $[(K * N_k), 3]$, then $M = D^TD$ which can be thought of an empirical covariance matrix but without mean-centering. Physically, this is the (empirical un-centered) "covariance" between the different angular velocities about the origin in the robot frame R. Degeneracy will occur, when all the data points (m_{ki}s) lie on a plane / line containing the origin. This situation corresponds to cases when the robot observes redundant angular velocities that do not give you information about 1 or more of the 3 degrees of freedom.

- $\boldsymbol{c}$ is the sum of the angular velocities m_{ki} weighted by the scalar $m_{ki}^T h_k$ which corresponds degree of alignment between the estimate of the (frame-transformed) angular velocities due to translational motion of the camera, and the application of angular velocity from ego motion of the robot.

  Physically, The sum of the angular velocities of all the points of a rigid body (the world) about a fixed point ${}_R O$ (origin of the robot frame) is the angular velocities of the rigid body about ${}_R O$. $\boldsymbol{c}$ specifically can be interpreted as the angular velocities of the world about ${}_R O$ conditioned by the action of ${}_W\omega_R$ on each point in the world.

To avoid degenerate cases, we use the Moore-Penrose inverse for M^{-1} in $J({}_F\omega_R)$. The algorithm used by this work is given below:

Algorithm 1 Solve Ego-Motion from multi-cam flow-fields

Require: $imgs_{t0}$, $imgs_{t1}$, $Extrinsics$, $Instrinsics$, w_{init}, $max\_iterations$, lr

Step 1: Preprocess Data
Compute `multical_optical_flow(imgs`$_{t0}, imgs_{t1}$`)`
`Filter optical fields`
`Back Project optical flow on Depth`
`Normalization plane`
`Initialize` $\omega \leftarrow \omega_{init}$
Step 2: Set Up Optimizer
$optimizer \leftarrow$ `Adagrad(`w`, `lr`, weight decay)`
Step 3: Define Loss Function
function ComputeLoss(w)
  `Compute` mi, c, M, hk, t `via eqs[9 - 15]`
  `Compute` J_1 `via eq 12`
  return `Loss` J_1
end function
Step 4: Optimize w
for $i \leftarrow 1$ `to` $max\_iterations$ **do**
  `Reset gradients,`
  `Compute` $LossJ1 \leftarrow$ `ComputeLoss(`w`)`
  `Compute gradients`
  `Update` w `via gradients`
end for
$\omega_{optimal} \leftarrow \omega$
Step 5: Compute Translation Vector
$t_{optimal} \leftarrow$ `Pseudoinverse solution`
Step 6: Return $\omega_{optimal}$, $t_{optimal}$

VI. Experiments

This section presents the results of our experiments conducted using a subset of the NuScenes dataset called NuScenes-Mini.v1 as the primary source of data. The NuScenes-Mini.v1 dataset consists of 10 diverse scenes capturing a moving car navigating various urban environments. It provides ground-truth measurements of rotation and translation rates recorded by the car's CAN-bus module, along with imagery from six cameras arranged in a circular configuration, offering a full 360-degree field of view.

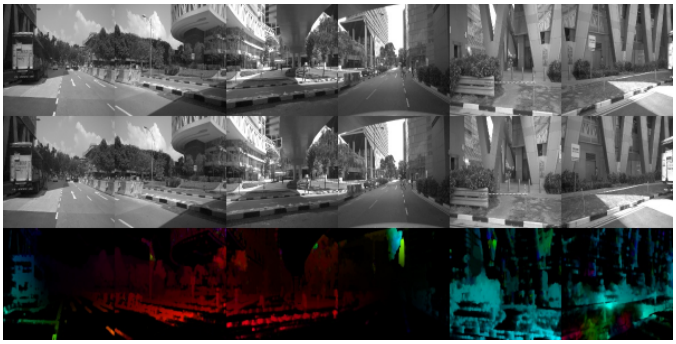

Fig. 5. Noisy optical flow on Nuscenes dataset

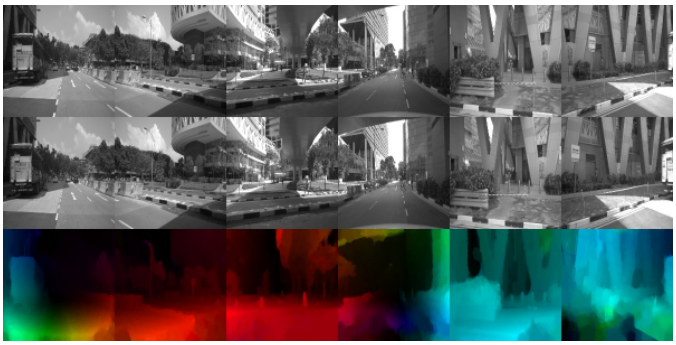

Fig. 6. Smoothened-dense optical flow on Nuscenes dataset

The algorithm for minimizing above state loss for the optimization process was implemented as a computational graph in PyTorch, enabling efficient computation of gradient estimates through automatic differentiation. To minimize the loss function J 11 described earlier, we evaluated the performance of multiple optimizers, including Adam, SGD, RMSprop, and Adagrad. Each experiment utilized a fixed learning rate and limited the gradient descent process to a maximum of 5 seconds for a single time step estimate of rotation $({}_F\omega_R)$ and translation $({}_F t_R)$ between 2 frames.

Among the optimizers tested, Adagrad demonstrated reliable convergence, irrespective of the initial learning rate. Consequently, Adagrad was chosen as the default optimizer, with an initial learning rate of 0.01 and a weight decay of 0.001.

The primary experiment involved minimizing the loss J1 using optical flow computed from all six camera configurations [1,2,3,4,5,6], as well as specific camera pairs: [1,4][Front,Back] and [2,6][Front left,Front Right]. The optimization process was performed using the extracted flow fields.

During the analysis, it became evident that many scenes in the NuScenes dataset exhibited degenerate cases. In these scenarios, the optical flow could only recover the rotation and translation directions but not the translation magnitudes. This limitation likely arises from ambiguous motion patterns

captured by the cameras, which hinder the reconstruction of complete motion parameters for certain scenes.

TABLE I
EXPERIMENTAL RESULTS ACROSS SCENES

Metric → Translations(m/s) and Rotations(rad/s)	**Camera [1,2,3,4,5,6]**	**Camera [1,4]**	**Camera [2,6]**
Trajectory Samples	200	200	200
Degenerate Cases	**164**	182	177
Non-Degenerate Cases	**36**	18	23
RMSE (Translation Rate)	**5.86**	6.50	6.10
RMSE (Translation Rate tx)	**9.61**	11.0	10.56
RMSE (Translation Rate ty)	**0.46**	2.0	1.45
RMSE (Translation Rate tz)	0.12	0.022	**0.0165**
RMSE (Rotation Rate)	**0.042**	0.103	0.067
RMSE (Rotation Rate Rx)	0.013	**0.012**	0.015
RMSE (Rotation Rate Ry)	0.015	**0.016**	0.017
RMSE (Rotation Rate Rz)	**0.070**	0.17	0.114

On careful visualization of the entire dataset. We selected a representative scene, that upheld the assumptions of optical flow calculations. We can infer the following from our experiment table I

- Having the 6-camera configuration setup is better than [1,4] and [2,6] configurations as it reduces the RMS translation and rotation error for our ego vehicle.
- The 6-camera configuration, also encounters a lesser number of degenerate cases.
- Even with a 6-camera configuration, the translation error is high.

For this scene(constituting 200 trajectory points) the issues we faced during our experiments, which might explain the high translation error, are summarized as follows:

- We found that the optical flow itself was noisy, which violates our initial proposition that the optical flows obtained are perfect. This was verified by computing the loss with ground truth values and we still obtained high error.
- Given the naive loss function, $\omega = 0$ is a valid optimal solution in every time step, which is a trivial solution.

VII. CONCLUSION

In this work, we revisited the ego-motion estimation approach proposed by Tsao et al. [1] and provided a thorough mathematical derivation and physical interpretation of its underlying principles. By carefully analyzing the relationships between ego-motion, point velocities, and optical flow, we showed how angular velocity components derived from the flow fields of multiple cameras can, in principle, circumvent depth ambiguities. Our derivations clarified why using multi-camera optical flow data should—under ideal conditions—enable the recovery of true translational and rotational velocities of a moving platform.

However, when tested on real-world data, such as the NuScenes dataset, several challenges arose. High sensitivity to noise in the computed optical flow and frequent degeneracies limited the successful recovery of translation magnitudes. These discrepancies highlight that the ideal assumptions for perfect optical flow are not always met in complex urban environments. Future refinements—such as improved loss functions, better noise handling, and more robust optimization strategies—are essential to bridge the gap between theory and practical deployment. Additionally, we can validate the algorithm using other camera combinations and an increased number of data points to ensure its robustness and generalizability.

VIII. APPENDIX

Here, the notation, each equation used, along with it's geometric interpretation is detailed.

A. Explicit notation

Throughout this work *explicit notation* is used in the format ${}_AX_{m,n}$ which is read as the quantity X of n with respect to m in coordinate frame A. Further, the presence of a single symbol in the right subscript such as ${}_AX_n$ refers to the quantity X of n with respect to the origin at frame A. Transformations are expressed in the form T_{AB} which takes entities from frame B to frame A.

B. Definitions

Throughout this work we use the following frames:

- The world coordinate frame : W
- The robot coordinate frame : R
- The coordinate frame of each camera k: This frame is referred to as k in explicit notation from here on. This frame is defined by its origin ${}_Rb_k \in \mathbb{R}^3$ and the axes being the column vectors of its orientation ${}_RR_k \in SO(3)$. Alternatively, the pose of the camera k (i.e ${}_Rx_k \in SE(3)$) can be expressed in the robot frame R as follows:

$$ {}_Rx_k = \begin{bmatrix} {}_RR_k & {}_Rb_k \\ 0 & 1 \end{bmatrix} $$

- The robot auxiliary frame F : This coordinate frame is defined to be *aligned* with the robot frame R and *static* with respect to the world frame W. $Aligned$ here indicates that the origin and the axes of the two frames are identical VIII-B.
- The camera auxiliary frame C : This coordinate frame is defined to be *aligned* with the camera frame k and *static* with respect to the world frame W.
- $[\cdot]_\times : \mathbb{R}^3 \rightarrow skew(3)$: This operator is used to represent any vector $a \in \mathbb{R}^3$ as a skew symmetric matrix $(skew(3))$.
- $(\bar{})$: This operator is used to homogenize a vector. eg: if $a \in \mathbb{R}^3$, then $\bar{a} \in \mathbb{R}^4$ is the homogeneous form of the vector a
- $(\dot{})$: This notation refers to the time-derivative of some entity.
- x: The alphabet x consistently refers to poses further specified using subscripts throughout this work.

- Ego motion of the robot ${}_F\dot{x}_R$: It is the velocity of the robot expressed in the auxiliary frame F. Since F is aligned with R, we have ${}_Fx_R = I$, implying:

$$ {}_F\dot{x}_R \in T_{{}_Fx_R}(SE(3)) = lie(SE(3)) $$

We also refer to ego-motion as the pair $({}_F\omega_R, {}_F\dot{t}_R)$, where ${}_F\omega_R \in \mathbb{R}^3$ is the angular velocity vector of the robot in frame F and ${}_F\dot{t}_r$ is the translational velocity vector of the robot in frame.

C. derivations

a) Deriving the relative velocity imposed on a point in the robot auxiliary frame ${}_F\bar{P}$ by the ego-motion of the robot ${}_F\dot{x}_R$ (i.e. ${}_R\dot{\bar{P}}$):

$$ {}_R\bar{P} = {}_Rx_F * {}_F\bar{P} \tag{I} $$

Note that ${}_F\bar{P}$ does not vary with time since F is static with respect to the world frame F. Taking the time derivative of the above equation gives us the following:

$$ {}_R\dot{\bar{P}} = {}_R\dot{x}_F * {}_F\bar{P} \tag{i} $$

To get the above expression in terms of the ego-motion ${}_F\dot{x}_R$, we need the relation between ${}_F\dot{x}_R$ and ${}_R\dot{x}_F$. We know that:

$$ {}_Rx_F = {}_Fx_R^{-1} \tag{ii} $$

To find the relation between the time derivative a group element x, and it's inverse, For any multiplicative group $(G, *)$, if $x, I \in G$, and $x * I = x$ we have the following property:

$$ x * x^{-1} = x^{-1} * x = I $$

Taking the time derivative of the above expression :

$$ \dot{x} * x^{-1} + x * \dot{x^{-1}} = \dot{x^{-1}} * x + x^{-1} * \dot{x} = 0 $$

$$ \Rightarrow \dot{x} = -x * \dot{x^{-1}} * x \tag{iii} $$

using $x = {}_Rx_F$ in eq iii gives:

$$ {}_R\dot{x}_F = -{}_Rx_F * {}_F\dot{x}_R * {}_Rx_F $$

$$ \Rightarrow {}_R\dot{x}_F = -{}_F\dot{x}_R \tag{iv} $$

Using iv in i, we get:

$$ {}_R\dot{\bar{P}} = -{}_F\dot{x}_R * {}_F\bar{P} \tag{v} $$

The de-homogenized expression for v is:

$$ {}_R\dot{P} = -{}_F\omega_R \times {}_FP - {}_F\dot{t}_R \tag{vi} $$

b) deriving the relative velocity imposed on a point ${}_kP$ in the camera frame k (i.e. ${}_k\dot{P}$) due to the motion of the camera in terms of ego-motion of the robot:

Since we want the expression in terms of the ego-motion, we project the point ${}_k\bar{P}$ to and from the robot frame R. Note that, since R and F is aligned, ${}_kx_R = {}_kx_F$.

$$ {}_k\bar{P} = {}_kx_R * {}_R\bar{P} \tag{II} $$

Note, ${}_Rx_k$ does not vary with time. Taking the time derivative of the above expression:

$$ {}_k\dot{\bar{P}} = {}_kx_R * {}_R\dot{\bar{P}} $$

Using v in the above expression:

$$ {}_k\dot{\bar{P}} = -{}_kx_R * {}_F\dot{x}_R * {}_F\bar{P} $$

$\because {}_kx_R = {}_kx_F$, we can rewrite the above expression as:

$$ {}_k\dot{\bar{P}} = -{}_kx_F * {}_F\dot{x}_R * {}_F\bar{P} $$

Using ${}_F\bar{P} = {}_Fx_C * {}_C\bar{P} = {}_Rx_C * {}_C\bar{P}$ in the above expression:

$$ {}_k\dot{\bar{P}} = -{}_kx_F * {}_F\dot{x}_R * {}_Rx_C * {}_C\bar{P} \tag{vii} $$

The obtained result can be interpreted as the relative velocity imposed on a point in the auxiliary camera frame ${}_CP$, by the motion of the camera frame (k) in the auxiliary camera frame C , i.e ${}_k\dot{x}_C = {}_kx_F * {}_F\dot{x}_R * {}_Rx_C = Ad_{{}_kx_R}{}_F\dot{x}_R$, where Ad is the adjoint operation [3]. It follows that :

$$ {}_k\dot{x}_C * {}_Cx_R = {}_kx_F * {}_F\dot{x}_R $$

$$ -{}_k\dot{x}_C * {}_Cx_F = {}_Cx_F * {}_F\dot{x}_R \tag{viii} $$

Below, each factor is expanded into matrix form and with algebraic operations, the non-homogeneous expression for vii is derived:

$$ {}_k\dot{\bar{P}} = -\begin{bmatrix} {}_RR_k^T & -{}_RR_k^T * {}_Rb_k \\ 0 & 1 \end{bmatrix} * \begin{bmatrix} [{}_F\omega_R]_\times & {}_F\dot{t}_R \\ 0 & 0 \end{bmatrix} * {}_Rx_C * {}_C\bar{P} \tag{III} $$

$$ {}_k\dot{\bar{P}} = -\begin{bmatrix} {}_RR_k^T * [{}_F\omega_R]_\times & {}_RR_k^T * {}_F\dot{t}_R \\ 0 & 0 \end{bmatrix} * \begin{bmatrix} {}_RR_k & {}_Rb_k \\ 0 & 1 \end{bmatrix} * {}_C\bar{P} $$

On simplifying and de-homogenizing:

$$ {}_k\dot{P} = -({}_RR_k^T * {}_F\omega_R) \times {}_CP - ({}_RR_k^T({}_F\omega_R \times {}_Fb_k + {}_F\dot{t}_R)) \tag{ix} $$

$$ {}_k\dot{x}_C = ({}_k\omega_C, {}_k\dot{t}_C) $$

$$ {}_k\omega_C = {}_RR_k^T * {}_F\omega_R $$

$$ {}_k\dot{t}_C = {}_RR_k^T({}_F\omega_R \times {}_Fb_k + {}_F\dot{t}_R) $$

c) The relation between ego-motion and optical flow in the ideal case :

The optical flow observed given all the assumptions of optical flow are met, is simply a function of the relative velocity of 3D points in the world, as observed by the camera. To derive this relation, consider the following:

- The image plane ι: The plane that is $+f$ away from the optical center. This is the plane where optical flow data is obtained.
- The Depth normalized plane DN: The plane that contains the unit depth projections of 3D points in the world.
- The world point plane ρ: for a given point $P = (P_x, P_y, P_z)^T$ The plane that perpendicular to the optical axis and is P_z away from the optical center.

For this derivation, the left subscript requirement of explicit notation is relaxed since all quantities are in the camera frame. Given some point P in the camera frame k, its depth-normalized projection $u = \frac{P}{P_z}$, and its image plane projection $v = K * \frac{P}{P_z} = K * u$, we begin with the equations:

$$v = K * u$$

$$u = \frac{P}{P_z}$$

To obtain a relation between optical flow $\dot{v}$, and point velocity $\dot{P}$, the above equations are differentiated w.r.t time. Since K does not change with time:

$$\dot{v} = K * \dot{u} \tag{x}$$

$$\dot{u} = \frac{\dot{P}}{P_z} - \frac{P * \dot{P}_z}{P_z^2}$$

$$\dot{u} = \frac{\dot{P}}{P_z} - \frac{u * \dot{P}_z}{P_z} \tag{xi}$$

$$\dot{u} = \frac{1}{P_z}\begin{bmatrix}\dot{P}_x \\ \dot{P}_y \\ \dot{P}_z\end{bmatrix} - \begin{bmatrix}u_x * \dot{P}_z \\ u_y * \dot{P}_z \\ 1 * \dot{P}_z\end{bmatrix}$$

$$\dot{u} = \frac{1}{P_z}\begin{bmatrix}\dot{P}_x - u_x * \dot{P}_z \\ \dot{P}_y - u_y * \dot{P}_z \\ 0\end{bmatrix} \tag{xii}$$

From xi, and x, we can write down the relation between optical flow $\dot{v}$, and the point velocity $\dot{P}$ (which is a relative velocity imposed by ego-motion):

$$\dot{v} = K(\frac{\dot{P}}{P_z} - \frac{u * \dot{P}_z}{P_z})$$

d) Deriving elements of the loss function:

The objective is to derive a loss function that quantifies the error between our estimates of ego-motion ${}_F\dot{x}_R = ({}_F\omega_R, {}_F\dot{t}_R)$ as a function of the optical flow observed. From xi, we know that:

$$\dot{u} = \frac{{}_k\dot{P}}{{}_kP_z} - \frac{u * {}_k\dot{P}_z}{{}_kP_z}$$

As $\dot{P} = {}_k\dot{P}$, using ix

$${}_k\dot{P} = -({}_C\omega_k \times \frac{{}_CP}{{}_kP_z}) - \frac{{}_C\dot{t}_k}{{}_kP_z}$$

$\because u = \frac{{}_kP}{{}_kP_Z} = \frac{{}_CP}{{}_CP_z}$

$$\dot{u} = -({}_C\omega_k \times u) - \frac{{}_C\dot{t}_k}{{}_kP_z} - \frac{u * {}_k\dot{P}_z}{{}_kP_z} \tag{xiii}$$

$$\dot{u} + ({}_C\omega_k \times u) = -\frac{{}_C\dot{t}_k}{{}_kP_z} - \frac{u * {}_k\dot{P}_z}{{}_kP_z}$$

Applying a left cross product with u to get rid of the ${}_kP_z$ scaled u term in the RHS.

$$u \times [\dot{u} + ({}_C\omega_k \times u)] = -u \times \frac{{}_C\dot{t}_k}{{}_kP_z} \tag{xix}$$

Applying $\cdot {}_C\dot{t}_k$ to get rid of another ${}_kp_z$ scaled component.

$$\{u \times [\dot{u} + ({}_C\omega_k \times u)]\} \cdot {}_C\dot{t}_k = 0 \tag{xx}$$

The above condition holds in the ideal case where all the assumptions of optical flow are upheld. Since this condition is what we desire, it can be used in constucting a loss function, as it is in terms of known/estimated quantities. A factor from xix turns out to be the expression for the data points that compose a data matrix that will facilitate the estimation of ego-motion. The geometric and physical break-down of the different components of the xix is detailed below.

xx can be expressed in terms of ego motion in the robot frame, as defined by the transformations in ix

$${}_RR_k \cdot \{{}_ku \times [{}_k\dot{u} + ({}_RR_k^T\,{}_F\omega_r \times {}_ku)]\} \cdot ({}_F\omega_R \times {}_Rb_k + {}_F\dot{t}_R) = 0 \tag{xxi}$$

for notational convenience, we will split the above expression into two terms as defined below, for each point index i:

$${}_km_i = u \times [\dot{u} + ({}_C\omega_k \times u)] \tag{xxii}$$

$$\Rightarrow {}_Rm_{ki} = {}_RR_k \cdot \{{}_ku \times [{}_k\dot{u} + ({}_RR_k^T\,{}_F\omega_r \times {}_ku)\}$$

$${}_Rh_k = {}_F\omega_R \times {}_Rb_k$$

e) Insights for the geometric and physical interpretation of m_{ki}, and h_k:

- **${}_k\dot{\boldsymbol{P}}, {}_k\boldsymbol{u}, {}_k\dot{\boldsymbol{u}}$ lie on / span the same plane** : Recall xi:

$$\dot{u} = \frac{\dot{P}}{P_z} - \frac{u * \dot{P}_z}{P_z}$$

  Since $P_z, \dot{P}_z$ are scalars, $\dot{u}$ is a linear combination of $\dot{P}$ and u, which proves the premise. Physically this makes sense because, the higher the point in the plane, the more veloc

- $\boldsymbol{u \times \frac{\dot{P}}{P_z} = u \times \dot{u}}$

$$u \times \dot{u} = (u \times \frac{\dot{P}}{P_z}) - (u \times \frac{u * \dot{P}_z}{P_z}) = (u \times \frac{\dot{P}}{P_z}) \quad \text{(xxiii)}$$

  Since, ${}_k\dot{P}, {}_k u, {}_k\dot{u}$ lie on / span the same plane, the cross products of any pair of these lie on the same line. equation xxiii shows that the $u \times \dot{u}$ is invariant to the *splatting component* $\frac{-u * \dot{P}_z}{P_z}$ because it is parallel to u. $u \times \dot{u}$ is physically significant because using the optical flow estimate *here* (in the cross product) is equivalent to using the $\frac{1}{P_z}$-scaled 3D point-velocity in the cross product instead.

- $\boldsymbol{m_{ki}}$: xix shows that m_{ki} is an estimate of angular velocity of point mass u acted on by instantaneous camera translational velocity about the origin ${}_kO$ (notationally this is ${}_C\dot{t}_k$). We know that ${}_k\hat{\omega}_{k}\dot{t}_C$

REFERENCES

[1] Tsao, A.-T., Hung, Y.-P., Fuh, C.-S., & Chen, Y.-S. (1997). Ego-motion estimation using optical flow fields observed from multiple cameras. In Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (pp. 457–462). link

[2] Caesar, H., Bankiti, V., Lang, A. H., Vora, S., Liong, V. E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., & Beijbom, O. (2020). nuScenes: A multimodal dataset for autonomous driving[Data set]. link

[3] J. Solà, J. Deray, and D. Atchuthan, "A micro Lie theory for state estimation in robotics," arXiv:1812.01537 [cs.RO], 2018.

[4] E. Gallo, "The SO(3) and SE(3) Lie Algebras of Rigid Body Rotations and Motions and their Application to Discrete Integration, Gradient Descent Optimization, and State Estimation," arXiv:2205.12572 [cs.RO], 2023.

[5] Nazari, N., et al. (2022). A.I. Solutions NASA Project: Final Report. NASA Technical Reports Server link

[6] Nistér, D., Naroditsky, O., and Bergen, J. "Visual Odometry." Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004, CVPR 2004, Washington, DC, USA, 2004, pp. I-I. doi:10.1109/CVPR.2004.1315094.

[7] Durrant-Whyte, H., & Bailey, T. "Simultaneous localization and mapping: part I," IEEE Robotics & Automation Magazine, vol. 13, no. 2, pp. 99-110, June 2006. doi: 10.1109/MRA.2006.1638022.

[8] Horn, B., & Schunck, B. (1981). Determining Optical Flow. Artificial Intelligence, 17, 185-203. doi: 10.1016/0004-3702(81)90024-2.

[9] Bruce, L., & Kanade, T. (1981). An Iterative Image Registration Technique with an Application to Stereo Vision. Proceedings of the 7th International Joint Conference on Artificial Intelligence, IJCAI 1981.

[10] Fraundorfer, F., & Scaramuzza, D. "Visual Odometry: Part II: Matching, Robustness, Optimization, and Applications," IEEE Robotics & Automation Magazine, vol. 19, no. 2, pp. 78-90, June 2012. doi: 10.1109/MRA.2012.2182810.

[11] Romera, T., Petreto, A., Lemaitre, F., Bouyer, M., & Meunier, Q. L. (2023). Optical Flow Algorithms Optimized for Speed, Energy, and Accuracy on Embedded GPUs. Journal of Real-Time Image Processing, 20(2), pp.32. doi: 10.1007/s11554-023-01288-6.

[12] Fang, N., & Zhan, Z. (2022). High-resolution optical flow and frame-recurrent network for video super-resolution and deblurring. Neurocomputing, 489, pp. 128-138. doi: 10.1016/j.neucom.2022.02.067.

[13] Tsao, A.-T., Fuh, C.-S., Hung, Y.-P., & Chen, Y.-S. (1997). Ego-Motion Estimation Using Optical Flow Fields Observed from Multiple Cameras. Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 457-462. doi:10.1109/CVPR.1997.609365.

[14] Gallo, E. "The SO(3) and SE(3) Lie Algebras of Rigid Body Rotations and Motions and their Application to Discrete Integration, Gradient Descent Optimization, and State Estimation," arXiv:2205.12572 [cs.RO], 2023.link