Alternating optimization for multimodal collaborating odometry estimation in CAVs

Nikos Piperigkos[1,2], Aris S. Lalos[2], Kostas Berberidis[1,2]
[1]Computer Engineering and Informatics Dept., University of Patras, Greece
[2]Industrial Systems Institute, Athena Research Center, Patras Science Park, Greece
Emails: piperigkos@ceid.upatras.gr, lalos@isi.gr, berberid@ceid.upatras.gr

***Abstract*—Cooperative, Connected and Automated Mobility will enable the close coordination of actions between vehicles, road users and traffic infrastructures, resulting in profound socioeconomic impacts. In this context, location and yaw angle of vehicles is considered vital for safe, secured and efficient driving. Motivated by this fact, we formulated a multimodal sensor fusion problem which provides more accurate localization and yaw information than the original sources. Simultaneously estimating location and yaw parameters of vehicles can be treated as the task of cooperative odometry or awareness. To do so, V2V communication as well as multimodal self and inter-vehicular measurements from various sensors are considered for the problem formulation. The solution strategy is based on the maximum likelihood criterion as well as a novel alternating gradient descent approach. To simulate realistic traffic conditions, CARLA autonomous driving simulator has been used. The detailed evaluation study has shown that each vehicle, relying only on its neighborhood, is able to accurately re-estimate both its own and neighboring states (comprised of locations and yaws), effectively realising the vision of 360° awareness.**

***Index Terms*—CCAM, Collaborating odometry, Multimodal sensor fusion, Alternating gradient descent**

I. Introduction

Cooperative Intelligent Transportation Systems [1] envision the integration of Cooperative, Connected and Automated Mobility (CCAM) technologies both in the public transportation sector and the mobility-as-a-service platforms. The goal is to introduce new mobility services for passengers and goods, fostering benefits for the road users and the transportation system as a whole. CCAM technology adoption from governments and industry aims to create user-centred, all-inclusive mobility, while increasing safety, reducing congestion and contributing to energy savings. Key role in the feasible operation of CCAM play the sensor rich Connected and Automated Vehicles (or simply CAVs), which through Vehicle-to-Vehicle (V2V) communication and sensor fusion are able to coordinate and achieve their driving actions. CAVs constantly broadcast cooperative awareness messages [2], [3] which include position, yaw angle, velocity, etc., to other vehicles with frequency $1-10$Hz and range $300-1000m$. Especially position and yaw angle are considered vital for efficient path and motion planning. Both parameters can be extracted by sensors like Global Positioning System (GPS) and Inertial Measurement Unit (IMU) or odometry solutions using raw visual and Light-Detection-and-Ranging (LIDAR) data, though degraded by measurement noise. Therefore, in this work collaborating multimodal fusion among the involved CAVs, which extract, exchange and process measurements of the surrounding environment using cameras, LIDARs, GPS, IMU, etc., aims explicitly to improve location and yaw angle estimation.

Distributed cooperative localization, in which the vehicle estimates its state though peer-to-peer V2V cooperation has attracted the attention of scientific community [4] due to its scalability, robustness to central node malfunctioning, communication and energy savings, etc. Different methods can be distinguished to either optimization-based or tracking. Optimization or in particular convex optimization-based ensure that the desired location will be attained in a "few" steps. Tracking, like the family of Kalman filters (KF), integrates the state prediction step via the motion model in the estimation approach. To improve ego vehicle position, local and global filters are optimally fused using KF in [5]. Self and relative measurements are considered by the ego vehicle. A cooperative odometry estimation algorithm for a group of robots is proposed in [6], which exploits raw visual and IMU data in the context of covariance intersection. Cooperation is realized through the transmission of visually detected features. Once again, each vehicle estimates only its own state. Localizing either vehicles or unmanned aerial vehicles in the presence of position outliers caused by cyberattacks or GPS-denied environments are discussed in [7] and [8]. The former formulates a centralized convex optimization problem in order to robustify position estimations, while the latter makes use of the renowned multi-dimensional scaling algorithm for centralized inter- and intra-cluster relative localization. Cooperative odometry in underwater environments is discussed in [9], where the state of autonomous underwater vehicle is augmented, apart from 2D position, with the yaw parameter, just like ours approach. To address the possible non-Gaussian measurement noise of yaw in this harsh environment, a robust two-stage Cubature KF is introduced.

To the best of our knowledge, the majority of localization approaches focuses only on ego position estimation. No published study has been found dealing with distributed multi 3 degrees-of-freedom (DOF) (x, y position and yaw angle) estimation. Our approach differentiates itself from state-of-the-art odometry or SLAM solutions [10]–[12], in

This paper was supported by the Hellenic Foundation for Research and Innovation (HFRI) under the 3rd Call for HFRI PhD Fellowships (Fellowship Number: 6372) and the EU's H2020 research and innovation programme CPSoSAware under grant agreement No 871738.

which the vehicle tries to estimate its own 6 DOF (3D rotation and translation) from raw visual or LIDAR data by finding the correspondences between key points (3D and/or pixels) of different keyframes. The state vector that needs to be estimated by the individual vehicle in our approach is comprised by the 2D positions and yaws of both ego vehicle and its neighbors. The novelty lies on the fact that the ego vehicle performs cooperative odometry estimation, i.e., multiple 3D pose estimation in absolute coordinates and over time horizon, so as to effectively be aware in a distributed manner of surrounding environment and not only of its own state. Our solution lends itself from our previous contribution in local cooperative awareness [13] related only to position estimation, as well as the cooperative gradient descent algorithm [14] which is extended by multimodal measurement models and unknown target parameters. The main contributions of this work can be summarized as follows:

- We formulate a cost function using maximum likelihood criterion, which is minimized by the ego vehicle and comprised of four multimodal measurement models and the unknown 3 DOF parameters of itself and closest neighbors.
- We derive a novel alternating gradient descent algorithm for effectively performing cooperative odometry.
- We verify that by treating yaw angles as parameters to be optimized, much more accurate localization can be achieved in the case of both self position and awareness.
- The extensive evaluation study performed in the realistic traffic conditions of CARLA simulator [15] has shown very promising results in terms of the accuracy of self and neighboring 3 DOF estimation.

Outline: Section II introduces the system model for the cooperative odometry problem; Section III presents the target cost function and a novel solution framework; Section IV is dedicated to experimental setup and evaluation, while Section V concludes our work.

II. System Model and Preliminaries

The localization module deployed in smart vehicles provides noisy self position and yaw measurements fusing GPS, IMU, Visual or LIDAR (frontend) odometry. Our framework should be seen as the backend of this module (Fig. 1), on top of the previous frontend solutions, aiming to provide much more accurate position and awareness information to the CAVs. For that reason, and instead of considering only ego 2D position and yaw, we define the state $\hat{\boldsymbol{x}}_{\boldsymbol{i}}^{(\boldsymbol{t})}$ of vehicle i as the collection of positions and yaws $\{\{x_j^{(t)}\},\{y_j^{(t)}\},\{\theta_j^{(t)}\}\}$ belonging to the connected neighbors j of i's neighborhood $\mathcal{N}_i^{(t)}$ (including i). Therefore, $\hat{\boldsymbol{x}}_{\boldsymbol{i}}^{(\boldsymbol{t})} \in \mathbb{R}^{3|\mathcal{N}_i^{(t)}|}$. This formulation enables i to be aware of itself as well as its nearby vehicles in terms of position and yaw angle so as to optimize its driving performance and enhance its safety.

Each vehicle utilizes a sensor-based perception system, which provides a detailed analysis of the traffic scene by detecting and classifying moving and static objects like vehicles, pedestrians, cyclists, etc. For this task, sensors including

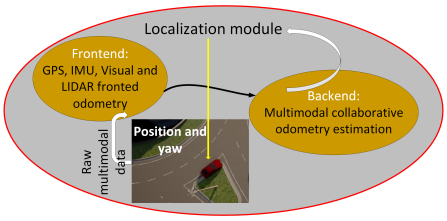


Fig. 1. Proposed backend architecture

(monocular or stereo) cameras, LIDARs, RADARs, etc., are used to combine and fuse the visual data so as to increase the quality of data analysis. Deep learning is nowadays a straightforward approach for the traffic scene analysis and understanding [16].

Assuming that the 2D ground truth position and yaw angle of ego vehicle i at time instant t are $\boldsymbol{x}_{\boldsymbol{i}}^{(\boldsymbol{t})} = \begin{bmatrix} x_i^{(t)} & y_i^{(t)} \end{bmatrix} \in \mathbb{R}^2$ and $\theta_i^{(t)}$ respectively, then its relative distance with respect to nearby vehicle j is equal to: $z_{d,ij}^{(t)} = \left\| \boldsymbol{x}_{\boldsymbol{i}}^{(\boldsymbol{t})} - \boldsymbol{x}_{\boldsymbol{j}}^{(\boldsymbol{t})} \right\|$. Additionally, the relative angle between i and j as measured by the former, is equal to: $z_{a,ij}^{(t)} = \arctan 2 \frac{y_j^{(t)} - y_i^{(t)}}{x_j^{(t)} - x_i^{(t)}} + \theta_i^{(t)}$, with $\phi_{ij}^{(t)} = \arctan 2 \frac{y_j^{(t)} - y_i^{(t)}}{x_j^{(t)} - x_i^{(t)}}$. Apart from the relative measurements with respect to other vehicles, ego vehicle extracts also self-measurements regarding its position and yaw. For these purposes, GPS, IMU or a low-cost odometry solution can be adopted. In any case, the extracted position information should be in the form of a common across the vehicles reference system, so as to the different positions correspond to the same map. Assuming that measurement noise follows the Gaussian distribution [4], [17], the two self-measurement models of vehicle i are defined as follows:

- Self position measurement model:

$$\tilde{\boldsymbol{z}}_{\boldsymbol{i}}^{(\boldsymbol{t})} = \boldsymbol{x}_{\boldsymbol{i}}^{(\boldsymbol{t})} + \boldsymbol{n}_{\boldsymbol{p}}^{(\boldsymbol{t})},\ \boldsymbol{n}_{\boldsymbol{p}}^{(\boldsymbol{t})} \sim \mathcal{G}(0, \boldsymbol{\Sigma}_{\boldsymbol{p}}) \tag{1}$$

- Self yaw measurement model:

$$\tilde{\theta}_i^{(t)} = \theta_i^{(t)} + n_\theta^{(t)},\ n_\theta^{(t)} \sim \mathcal{G}(0, \sigma_\theta^2) \tag{2}$$

Gaussian measurement noise is used for the two relative measurement models (as [4] and [17] have adopted, too):

- Relative distance measurement model:

$$\tilde{z}_{d,ij}^{(t)} = z_{d,ij}^{(t)} + n_d^{(t)},\ n_d^{(t)} \sim \mathcal{G}(0, \sigma_d^2) \tag{3}$$

- Relative angle measurement model:

$$\tilde{z}_{a,ij}^{(t)} = z_{a,ij}^{(t)} + n_a^{(t)},\ n_a^{(t)} \sim \mathcal{G}(0, \sigma_a^2) \tag{4}$$

Notice that the yaw measurement model (2) is also integrated in (4). Model (2) is actually truncated Gaussian, so as to ensure that yaw is between $[0, 2\pi]$.

We made the assumption that i knows which range and angle measurement coming from LIDAR and camera correspond to its V2V neighbor j. Although this knowledge requires a data association pre-processing step, it is assumed that optimal data association has already been performed. Additionally, according to V2V communication standard of

[2], a communication range r_c (much lower than $300m$) indicating V2V connection is assigned to each vehicle, imposing that visually extracted measurements are discarded should relative distance exceeds r_c. In this way, computational load can be reduced, without compromising the accuracy. Furthermore, impact of communication delay can be considered negligible exactly due to restricted r_c and small amount of transmitted measurements between vehicles (shown in next Section), which are significantly lower than raw images or point clouds. These two practical challenges and their impacts on accuracy are of future investigation.

III. Distributed Multi 3 DoF estimation

This Section is dedicated to the derivation of the target cost function. Initially, we define the state vector of each ego vehicle i and formulate the optimization problem according to maximum likelihood criterion and the available multimodal measurements to i. Afterwards, a novel solution framework is presented based on gradient descent algorithm and alternating optimization. The state vector of ego vehicle comprises of $3|\mathcal{N}_i^{(t)}|$DOF, facilitating the awareness ability of vehicle.

A. Cost function derivation via maximum likelihood

The state $\hat{\boldsymbol{x}}_{\boldsymbol{i}}^{(\boldsymbol{t})} \in \mathbb{R}^{3|\mathcal{N}_i^{(t)}|}$ of vehicle i is equal to: $\hat{\boldsymbol{x}}_{\boldsymbol{i}}^{(\boldsymbol{t})} = \left\{\{x_j^{(t)}\},\{y_j^{(t)}\},\{\theta_j^{(t)}\}\right\}, \forall j \in \mathcal{N}_i^{(t)}$. As a first step, we define the following probability density functions (PDFs) for the relative measurement models of i with itself and its neighbor j treating the unknown parameters as deterministic variables:

$$\mathcal{P}_1 = \mathcal{P}(\tilde{\boldsymbol{z}}_{\boldsymbol{i}}^{(\boldsymbol{t})}|\boldsymbol{x}_{\boldsymbol{i}}^{(\boldsymbol{t})}) = \mathcal{G}(\boldsymbol{z}_{\boldsymbol{i}}^{(\boldsymbol{t})}, \boldsymbol{\Sigma_p})$$
$$\mathcal{P}_2 = \mathcal{P}(\tilde{\theta}_i^{(t)}|\theta_i^{(t)}) = \mathcal{G}(\theta_i^{(t)}, \sigma_\theta^2)$$
$$\mathcal{P}_3 = \mathcal{P}(\tilde{z}_{d,ij}^{(t)}|\boldsymbol{x}_{\boldsymbol{i}}^{(\boldsymbol{t})}, \boldsymbol{x}_{\boldsymbol{j}}^{(\boldsymbol{t})}) = \mathcal{G}(z_{d,ij}^{(t)}, \sigma_d^2)$$
$$\mathcal{P}_4 = \mathcal{P}(\tilde{z}_{a,ij}^{(t)}|\boldsymbol{x}_{\boldsymbol{i}}^{(\boldsymbol{k})}, \boldsymbol{x}_{\boldsymbol{j}}^{(\boldsymbol{t})}) = \mathcal{G}(z_{a,ij}^{(t)}, \sigma_a^2)$$

The likelihood function $\mathcal{L}$ is equal to the product of the PDFs: $\mathcal{L} = \prod_{l=1}^{4} \mathcal{P}_l$. Maximum likelihood criterion states that to estimate the target parameters of 2D positions and yaws given the observed measurements, the likelihood function should be maximized, or equivalently, minus likelihood function be minimized. If in addition we take the natural logarithm of $-\mathcal{L}$, we end up to the target cost function that is assigned to vehicle i:

$$\underset{\hat{\boldsymbol{x}}_{\boldsymbol{i}}^{(\boldsymbol{t})}}{\operatorname{argmin}} f(\hat{\boldsymbol{x}}_{\boldsymbol{i}}^{(\boldsymbol{t})}), \tag{5}$$

with

$$f(\hat{\boldsymbol{x}}_{\boldsymbol{i}}^{(\boldsymbol{t})}) = \sum_{(k,j)\in\mathcal{N}_i^{(t)}} \frac{(\tilde{z}_{d,kj}^{(t)} - z_{d,kj}^{(t)})^2}{2{\sigma_d}^2} + \frac{(\tilde{z}_{a,kj}^{(t)} - z_{a,kj}^{(t)})^2}{2{\sigma_a}^2} + \sum_{j\in\mathcal{N}_i^{(t)}} \frac{(\tilde{z}_j^{(x,t)} - x_j^{(t)})^2}{2{\sigma_x}^2} + \frac{(\tilde{z}_j^{(y,t)} - y_j^{(t)})^2}{2{\sigma_y}^2} + \frac{(\tilde{\theta}_j^{(t)} - \theta_j^{(t)})^2}{2{\sigma_\theta}^2} \tag{6}$$

According to the definition of (6), each neighbor $j \in \mathcal{N}_i^{(t)}$ should transmit to i its noisy 2D position $\tilde{\boldsymbol{z}}_{\boldsymbol{j}}^{(\boldsymbol{t})}$ and yaw $\tilde{\theta}_j^{(t)}$, as well as the range and angle measurements $(\tilde{z}_{d,ji}^{(t)}, \tilde{z}_{a,ji}^{(t)})$ towards i. The latter, together with its own (self and relative) noisy measurements, formulates the target function, minimizes it using the framework of the following section, and re-estimates the positions and yaws of its neighborhood. As a final remark, the second term of relative angle model is reformulated so as to facilitate the solution:

$$\sum_{(k,j)\in\mathcal{N}_i^{(t)}} \frac{\left(\tan(\tilde{z}_{a,kj}^{(t)} - \theta_k^{(t)})(x_j^{(t)} - x_k^{(t)}) - (y_j^{(t)} - y_k^{(t)})\right)^2}{2{\sigma_a}^2}$$

A schematic diagram of the transmitted measurements to vehicle i is shown in Fig. 2.

Fig. 2. Transmitted and available measurements to i

B. Alternating Gradient Descent as solution strategy

Vehicle i has to address the minimization of (6) so as to be effectively aware of its surrounding environment. To do so, cooperative gradient descent algorithm of [14], [13] can be used. Directly minimizing (6) is non-beneficial due to the $tan(\cdot)$ term which connects position and yaw angle. Alternating minimization will be helpful in decoupling the estimation of 2D positions and yaws into two distinct optimization steps. Collecting all the positions and yaws that need to be re-estimated by i to vectors $\tilde{\boldsymbol{x}}_{\boldsymbol{i}}^{(\boldsymbol{t})} = \left\{\boldsymbol{x}_{\boldsymbol{j}}^{(\boldsymbol{t})}\right\}, \tilde{\boldsymbol{x}}_{\boldsymbol{i}}^{(\boldsymbol{t})} \in \mathbb{R}^{2|\mathcal{N}_i^{(t)}|}$ and $\boldsymbol{\theta}_{\boldsymbol{i}}^{(\boldsymbol{t})} = \left\{\boldsymbol{\theta}_{\boldsymbol{j}}^{(\boldsymbol{t})}\right\}, \boldsymbol{\theta}_{\boldsymbol{i}}^{(\boldsymbol{t})} \in \mathbb{R}^{|\mathcal{N}_i^{(t)}|}$, the alternating minimization problem can be casted as follows:

$$\underset{\tilde{\boldsymbol{x}}_{\boldsymbol{i}}^{(\boldsymbol{t},\boldsymbol{l})}}{\operatorname{argmin}} f(\tilde{\boldsymbol{x}}_{\boldsymbol{i}}^{(\boldsymbol{t},\boldsymbol{l})}, \boldsymbol{\theta}_{\boldsymbol{i}}^{(\boldsymbol{t},\boldsymbol{l-1})}) \tag{7}$$

$$\underset{\boldsymbol{\theta}_{\boldsymbol{i}}^{(\boldsymbol{t},\boldsymbol{l})}}{\operatorname{argmin}} f(\tilde{\boldsymbol{x}}_{\boldsymbol{i}}^{(\boldsymbol{t},\boldsymbol{l})}, \boldsymbol{\theta}_{\boldsymbol{i}}^{(\boldsymbol{t},\boldsymbol{l})}), \tag{8}$$

for $l = 1, \ldots L_{max}$ iterations. This formulation implies that during the first optimization step the vector of yaws, as determined in the previous iteration, is considered a known parameter in order to estimate the vector of positions. At the second step, where we are trying to re-estimate yaws, the positions are set equal to those determined in the first step. Gradient descent algorithm is used to minimize (7)-(8) by using the negative gradient of $f(\cdot)$ in order to identify the direction of steepest descent. More specifically, (7)-(8) can be minimized as follows:

$$\tilde{\boldsymbol{x}}_{\boldsymbol{i}}^{(\boldsymbol{t},\boldsymbol{l})} = \tilde{\boldsymbol{x}}_{\boldsymbol{i}}^{(\boldsymbol{t},\boldsymbol{l-1})} - \mu_1 \left.\frac{\partial f(\cdot)}{\partial \tilde{\boldsymbol{x}}_{\boldsymbol{i}}^{(\boldsymbol{t},\boldsymbol{l})}}\right|_{\tilde{\boldsymbol{x}}_{\boldsymbol{i}}^{(\boldsymbol{t},\boldsymbol{l-1})}} \tag{9}$$

$$\boldsymbol{\theta}_{\boldsymbol{i}}^{(\boldsymbol{t},\boldsymbol{l})} = \boldsymbol{\theta}_{\boldsymbol{i}}^{(\boldsymbol{t},\boldsymbol{l-1})} - \mu_2 \left.\frac{\partial f(\cdot)}{\partial \boldsymbol{\theta}_{\boldsymbol{i}}^{(\boldsymbol{t},\boldsymbol{l})}}\right|_{\boldsymbol{\theta}_{\boldsymbol{i}}^{(\boldsymbol{t},\boldsymbol{l-1})}}, \tag{10}$$

with small scalars $\mu_1, \mu_2 > 0$. To ensure that global minimum will be reached, we have to guarantee the convexity of $f(\cdot)$. Due to the term $\tan(\tilde{z}_{a,kj}^{(t)} - \theta_k^{(t)})(x_j^{(t)} - x_k^{(t)}) - (y_j^{(t)} - y_k^{(t)})$, the cost function of (10) is non-convex, since $\tan x$ is convex in $(0, \pi/2)$ and concave in $(-\pi/2, 0)$. Therefore, it is probable that the solution will be trapped in saddle points. Even with these limitations, our collaborative odometry solution achieves remarkable accuracy as indicated in Section IV. Imposing the convexity of cost function as a constraint in the framework of Alternating Direction Method of Multipliers is of future research. The main steps of the proposed approach **Collaborative Odometry using Gradient Descent (CO-GD)** are shown in **Algorithm 1**.

Algorithm 1: CO-GD

Input: simulation horizon T, max iterations L_{max}, scalars $\mu_1 = 10^{-3}, \mu_2 = 10^{-6}$

Output: $\hat{\boldsymbol{x}}_{\boldsymbol{i}}^{(\boldsymbol{t})} \in \mathbb{R}^{3|\mathcal{N}_i^{(t)}|}$

for $t = 1, 2, \dots T$ **do**
  Vehicles transmit to neighbors their 2D position and yaw, and the corresponding range and angle measurements (see Fig. 2) ;
  for *each vehicle* i **do**
    Define cost function $f(\cdot)$ from (6) ;
    for $l = 1, \dots L_{max}$ **do**
      $\tilde{\boldsymbol{x}}_{\boldsymbol{i}}^{(\boldsymbol{t},\boldsymbol{l})} = \tilde{\boldsymbol{x}}_{\boldsymbol{i}}^{(\boldsymbol{t},\boldsymbol{l-1})} - \mu_1 \frac{\partial f(\cdot)}{\partial \tilde{\boldsymbol{x}}_{\boldsymbol{i}}^{(\boldsymbol{t},\boldsymbol{l})}}\Big|_{\tilde{\boldsymbol{x}}_{\boldsymbol{i}}^{(\boldsymbol{t},\boldsymbol{l-1})}}$;
      $\boldsymbol{\theta}_{\boldsymbol{i}}^{(\boldsymbol{t},\boldsymbol{l})} = \boldsymbol{\theta}_{\boldsymbol{i}}^{(\boldsymbol{t},\boldsymbol{l-1})} - \mu_2 \frac{\partial f(\cdot)}{\partial \boldsymbol{\theta}_{\boldsymbol{i}}^{(\boldsymbol{t},\boldsymbol{l})}}\Big|_{\boldsymbol{\theta}_{\boldsymbol{i}}^{(\boldsymbol{t},\boldsymbol{l-1})}}$;
    end
    $\hat{\boldsymbol{x}}_{\boldsymbol{i}}^{(\boldsymbol{t})} = \{\tilde{\boldsymbol{x}}_{\boldsymbol{i}}^{(\boldsymbol{t},\boldsymbol{l})}, \boldsymbol{\theta}_{\boldsymbol{i}}^{(\boldsymbol{t},\boldsymbol{l})}\}$;
  end
end

IV. Numerical Results

In this section, we carry out simulations to verify the convergence and localization performance of the proposed scheme using Python and CARLA simulator. During the experiments a PC Laptop with 8GB RAM and Intel Core i7-1065G7 CPU @ 1.3 GHz was used.

A. Evaluation metrics

In practice, vehicle i measures its relative angle with respect to j in the local reference system (*lrs*) of its LIDAR or camera. To realistically simulate this measurement, we have to notice that vehicle i has undergone both rotation and translation in the global reference system of CARLA. A more detailed illustration is provided in Fig. 3. As a matter of fact, $z_{a,ij}$ is actually equal to: $z_{a,ij} = \arctan 2\frac{y_j^{lrs} - y_i^{lrs}}{x_j^{lrs} - x_i^{lrs}}$, with time index omitted. The 3×3 homogeneous transformation matrix $\mathcal{T} \in SE(2)$ describes the relationship between global and local reference systems:

$$\boldsymbol{x}_{\boldsymbol{i}}^{(\boldsymbol{t},\boldsymbol{lrs})} = \mathcal{T} \begin{bmatrix} \boldsymbol{x}_{\boldsymbol{i}}^{(\boldsymbol{t})} & 1 \end{bmatrix}^T, \quad \boldsymbol{x}_{\boldsymbol{j}}^{(\boldsymbol{t},\boldsymbol{lrs})} = \mathcal{T} \begin{bmatrix} \boldsymbol{x}_{\boldsymbol{j}}^{(\boldsymbol{t})} & 1 \end{bmatrix}^T \quad (11)$$

Transformation matrix $\mathcal{T}$ comprises of the 2×2 rotation matrix $\boldsymbol{R} = \begin{bmatrix} \cos\theta_i^{(t)} & -\sin\theta_i^{(t)} \\ \sin\theta_i^{(t)} & \cos\theta_i^{(t)} \end{bmatrix}, \boldsymbol{R} \in SO(2)$ and the translation vector $\boldsymbol{p} \in \mathbb{R}^2$: $\mathcal{T} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{p} \\ \boldsymbol{0} & 1 \end{bmatrix}$. Note that the projection operation of (11) is only used for the purposes of realistic simulation, since CARLA can provide both the ground truth yaw and vector $\boldsymbol{p}$, while the latter coincides with the ground truth position determined by the simulator. In practice, visual sensors will "simply" return scalars $\tilde{z}_{d,ij}^{(t)}$, $\tilde{z}_{a,ij}^{(t)}$ using the centroid of the detected bounding box of target vehicle. In the same context, the estimated self yaw angle by vehicle i is used to define estimated rotation matrix $\boldsymbol{R_{est}}$. According to [18], (unit-less) yaw estimation error is equal to the following frobenius norm:

$$YE_i^{(t)} = \|\boldsymbol{P_{est}} - \boldsymbol{P}\|_F = \left\|\boldsymbol{P}^{-1}\boldsymbol{P_{est}} - \boldsymbol{I}\right\|_F,$$

which denotes the distance of estimated and ground truth rotation matrices.

For the case of yaw angle awareness of vehicle i, the root mean square error (RMSE) is used at each time instant:

$$YAE_i^{(t)} = \sqrt{\frac{1}{|\mathcal{N}_i^{(t)}|} \sum_{j \in \mathcal{N}_i^{(t)}} (YE_{j \leftarrow i}^{(t)})^2},$$

where $YE_{j \leftarrow i}^{(t)}$ is the yaw error of j as measured by i.

In terms of location evaluation, self localization error $LE_i^{(t)}$ is derived according to the euclidean distance between the actual and the estimated position. Location awareness error, $LAE_i^{(t)}$, is equal to:

$$LAE_i^{(t)} = \sqrt{\frac{1}{|\mathcal{N}_i^{(t)}|} \sum_{j \in \mathcal{N}_i^{(t)}} (LE_{j \leftarrow i}^{(t)})^2}$$

For the overall evaluation of vehicle's trajectory we measured the RMSE over simulation horizon for: i) rotation, i.e., Yaw RMSE over time (Y-RMSET) and Yaw Awareness RMSE over time (YA-RMSET) and ii) location, i.e., Location RMSE over time (L-RMSET) and Location Awareness RMSE over time (LA-RMSET).

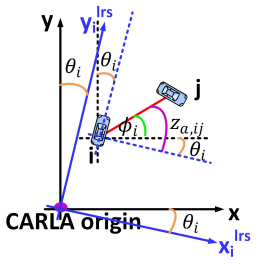


Fig. 3. Definition of yaw and relative angle measurement

B. Simulation setup

CARLA simulator has been used to extract the trajectories of 150 vehicles, for simulation horizon $T = 448$ time

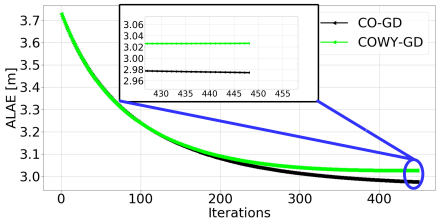
instances and time interval $dt = 0.4sec$. Ground truth positions and yaws have been degraded with noise variance $\sigma_x = 3.5m, \sigma_y = 2m$ and $\sigma_\theta = 10°$. Noise variances $\sigma_d = 1$ and $\sigma_a = 4°$ have been used for the relative (range and angle) measurements. The velocities of vehicles range between 0 (e.g. waiting at traffic lights) and $74km/h$. To validate the performance of the proposed collaborative odometry algorithm in terms of location and yaw accuracy, we have used the gradient descent based algorithm of [13] which doesn't consider yaw angle as a parameter to be estimated. We named it **CO without yaw using GD** or **(COWY-GD)**. The evaluation study has considered a random ego vehicle, along with its neighborhood of connected vehicles, demonstrating how *accurate* and how *quickly* can estimate 2D positions and yaws. Trajectory of vehicle in the CARLA environment is shown in Fig. 5.

C. Evaluation study

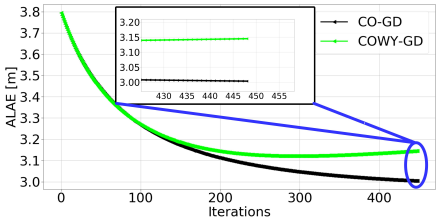
1) Convergence ability: The convergence performance of our algorithm is shown in Fig. 4, under three different choices of communication range r_c. The latter parameter dictates the number of connected neighbors to each ego vehicle i. More specifically, number of neighbors ranges between 1-7, 1-9 and 1-11, for $r_c = 20m, 30m, 40m$, respectively. Fig. 4 presents the averaged over time horizon LAE (ALAE) of vehicle i, so as to depict the convergence behavior in the mean sense. For $L_{max} = 450$ iterations, the proposed **CO-GD** succeeds in reducing the location awareness error at every iteration, reaching almost $3m$ at the end of optimization. In all three cases, it is apparent that our scheme successively minimizes location error due to the convexity of (9). On the contrary, **COWY-GD** which doesn't consider yaw as an unknown parameter, behaves much more worse when the number of neighbors grows larger. Actually, during the optimization procedure the solution of **COWY-GD** gets away from the minimum, especially in the third case. This is due to the fact that the noise of angle measurement model caused by noisy yaw and relative measurement, which isn't addressed by the algorithm, seriously impacts on the accumulation of position estimation error. Therefore, we can conclude that larger number of neighbors don't influence negatively our scheme (instead of **COWY-GD**) in terms of location awareness, while it is expected to be more accurate should the L_{max} increases. Scalability to the size of neighborhood is attained in the case of self position and yaw estimation, as pointed out next.

2) Time complexity: Averaged (over L_{max} and T) timing results for every iteration of both schemes are summarized in **TABLE I**, for different communication ranges. Clearly, our method is more time consuming than **COWY-GD**, reaching 2.6 $msec$ when $r_c = 40$, due to the two optimization problems which have to be solved. Choice of L_{max} in practical considerations is related to these timing outcomes, since optimization procedure has to be completed within the time interval dt.

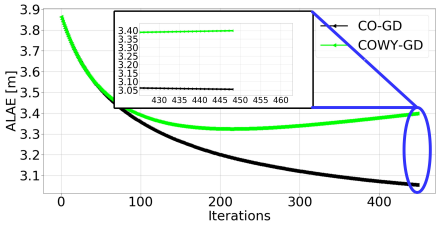
3) Position accuracy evaluation: Additionally, **Table II** and **III** present the self L-RMSET and neighborhood's LA-



(a) ALAE with $r_c = 20m$



(b) ALAE with $r_c = 30m$



(c) ALAE with $r_c = 40m$

Fig. 4. Convergence with different communication ranges

TABLE I
AVERAGED TIME RESULTS FOR EVERY ITERATION

Comm. range	*CO-GD (msec)*	*COWY-GD (msec)*
$r_c = 20m$	**1.6**	0.3
$r_c = 30m$	**2**	0.5
$r_c = 40m$	**2.6**	0.6

RMSET of ego vehicle. The second column of **Table II** dictates the benefits of **CO-GD** in terms of scalability. This fact means that as the number of neighbor increases, so does the accuracy of self position estimation, since higher amount of relative measurements fused by ego vehicle result on reducing the noise of original position source. Our scheme succeeds in reducing self position error with respect to noisy source by **30%**, **34%** and **42%**. As it was also pointed out before, the performance of **COWY-GD** shown in the third column, is significantly degraded when r_c increases. In terms of location awareness performance, we conclude from **Table III** that our scheme once again outperforms **COWY-GD**, achieving almost the same error of $3.1m$, i.e., **20%** reduction of original source's error. Therefore, in terms of 2D position, the proposed backend module enabled ego vehicle to be more aware of both itself and the surrounding environment than the original position source. Furthermore, greater amount of information with respect to nearby vehicles, seem to be more

beneficial for self position.

TABLE II
L-RMSET

Comm. range	*CO-GD (m)*	*COWY-GD (m)*	*Noisy source (m)*
$r_c = 20m$	**2.76**	2.85	3.92
$r_c = 30m$	**2.58**	2.92	3.93
$r_c = 40m$	**2.33**	3.07	3.99

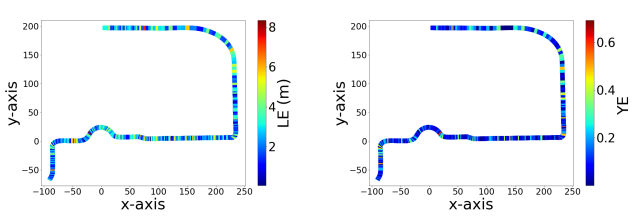
TABLE III
LA-RMSET

Comm. range	*CO-GD (m)*	*COWY-GD (m)*	*Noisy source (m)*
$r_c = 20m$	**3.13**	3.18	3.89
$r_c = 30m$	**3.1**	3.27	3.87
$r_c = 40m$	**3.15**	3.47	3.93

4) Yaw angle accuracy evaluation: In terms of yaw angle evaluation, **Table IV** imply performance analogous to the previous case. More specifically, self yaw estimation becomes more accurate when the number of neighbors increases. That means that the property of scalability is also ensured for the yaw angle, apart from position. The proposed **CO-GD** reduced the noise of original yaw source by **13%**, **30%** and **40%**. In terms of yaw awareness (two last columns of **Table IV**), our method achieved the same error of 0.2 in all three cases. As a last remark, the estimated location and yaw error of ego vehicle with respect to its trajectory using $r_c = 40m$, have been depicted in Fig. 5 in the form of a heatmap. The "randomness" of error is due to the fact that measurement noise has been modelled in a probabilistic manner through the Gaussian distribution.

TABLE IV
Y-RMSET AND YA-RMSET

Comm. range	**Y-RMSET**		**YA-RMSET**	
	CO-GD	*Noisy source*	*CO-GD*	*Noisy source*
$r_c = 20m$	**0.19**	0.22	**0.2**	0.22
$r_c = 30m$	**0.14**	0.20	**0.2**	0.22
$r_c = 40m$	**0.12**	0.2	**0.19**	0.22



(a) Heatmap of LE (b) Heatmap of YE

Fig. 5. Heatmaps of self location and yaw angle error on top of trajectory

V. Conclusion

In this paper, a collaborative odometry estimation algorithm has been developed, aiming to offer to the vehicles increased awareness of the surrounding environment. Each vehicle through V2V cooperation formulates the target cost function via multimodal sensor fusion and maximum likelihood criterion. Gradient descent is then applied so as to minimize the cost function. Our approach succeeds in reducing both the position and yaw angle error introduced by a noisy original source, like GPS, Visual or LIDAR odometry. We have verified that each vehicle is in place to accurately re-estimate both self as well as neighboring 2D positions and yaws in a distributed manner. Additionally, we have validated that treating the noisy yaw as a parameter to be optimized, positively affects the position estimation.

References

[1] M. Aloqaily, H. Elayan, and M. Guizani, "C-healthier: A cooperative health intelligent emergency response system for c-its," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–11, 2022.

[2] *ETSI EN 302 637-2 V1.4.0 (2018-08)*, European Telecommunications Standards Institute Std., Nov. 2018.

[3] M. A. Javed and E. B. Hamida, "Measuring safety awareness in cooperative ITS applications," in *2016 IEEE Wireless Communications and Networking Conference*. IEEE, 2016.

[4] R. M. Buehrer, H. Wymeersch, and R. M. Vaghefi, "Collaborative sensor network localization: Algorithms and practical issues," *Proceedings of the IEEE*, vol. 106, no. 6, pp. 1089–1114, 2018.

[5] P. Yang *et al.*, "Multi-sensor multi-vehicle (msmv) localization and mobility tracking for autonomous driving," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 14 355–14 364, 2020.

[6] P. Zhu, Y. Yang, W. Ren, and G. Huang, "Cooperative visual-inertial odometry," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 13 135–13 141.

[7] C. Vitale *et al.*, "CARAMEL: results on a secure architecture for connected and autonomous vehicles detecting GPS spoofing attacks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2021, no. 1, may 2021.

[8] L. Ruan *et al.*, "Cooperative relative localization for uav swarm in gnss-denied environment: A coalition formation game approach," *IEEE Internet of Things Journal*, pp. 1–1, 2021.

[9] B. Xu, X. Wang, J. Zhang, Y. Guo, and A. A. Razzaqi, "A novel adaptive filtering for cooperative localization under compass failure and non-gaussian noise," *IEEE Transactions on Vehicular Technology*, pp. 1–1, 2022.

[10] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 3, pp. 611–625, 2018.

[11] T. Shan and B. Englot, "Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 4758–4765.

[12] F. Dellaert and M. Kaess, "Factor graphs for robot perception," *Foundations and Trends in Robotics*, vol. 6, no. 1-2, pp. 1–139, 2017.

[13] N. Piperigkos, A. S. Lalos, and K. Berberidis, "Multi-modal cooperative awareness of connected and automated vehicles in smart cities," in *2021 IEEE International Conference on Smart Internet of Things (SmartIoT)*, 2021, pp. 377–382.

[14] H. Wymeersch, J. Lien, and M. Z. Win, "Cooperative localization in wireless networks," *Proceedings of the IEEE*, vol. 97, no. 2, pp. 427–450, 2009.

[15] A. Dosovitskiy *et al.*, "CARLA: An open urban driving simulator," in *Conference on Robot Learning*, 2017, pp. 1–16.

[16] S. Nousias, E.-V. Pikoulis, C. Mavrokefalidis, A. S. Lalos, and K. Moustakas, "Accelerating 3d scene analysis for autonomous driving on embedded ai computing platforms," in *2021 IFIP/IEEE 29th International Conference on Very Large Scale Integration (VLSI-SoC)*, 2021, pp. 1–6.

[17] T. Liang *et al.*, "Uav aided positioning systems for ground devices: Fundamental limits and algorithms," *IEEE Internet of Things Journal*, pp. 1–1, 2022.

[18] D. Q. Huynh, "Metrics for 3d rotations: Comparison and analysis," *J. Math. Imaging Vis.*, vol. 35, no. 2, p. 155–164, oct 2009.