Asynchronous Collaborative Localization by Integrating Spatiotemporal Graph Learning with Model-Based Estimation

Peng Gao[1], Brian Reily[3], Rui Guo[2], Hongsheng Lu[2], Qingzhao Zhu[1] and Hao Zhang[1]

***Abstract*— Collaborative localization is an essential capability for a team of robots such as connected vehicles to collaboratively estimate object locations from multiple perspectives with reliant cooperation. To enable collaborative localization, four key challenges must be addressed, including modeling complex relationships between observed objects, fusing observations from an arbitrary number of collaborating robots, quantifying localization uncertainty, and addressing latency of robot communications. In this paper, we introduce a novel approach that integrates uncertainty-aware spatiotemporal graph learning and model-based state estimation for a team of robots to collaboratively localize objects. Specifically, we introduce a new uncertainty-aware graph learning model that learns spatiotemporal graphs to represent historical motions of the objects observed by each robot over time and provides uncertainties in object localization. Moreover, we propose a novel method for integrated learning and model-based state estimation, which fuses asynchronous observations obtained from an arbitrary number of robots for collaborative localization. We evaluate our approach in two collaborative object localization scenarios in simulations and on real robots. Experimental results show that our approach outperforms previous methods and achieves state-of-the-art performance on asynchronous collaborative localization.**

I. Introduction

Object localization is an important area in robotics due to its necessity for improving situational awareness for robots. It aims to estimate the real-world locations of the objects moving in a dynamic environment using observations that are acquired by robot sensors such as cameras. Object localization is widely deployed in a variety of robotics applications, such as autonomous driving to perceive street objects and search and rescue to find victims. It is also applied as a critical component in robot capabilities such as scene reconstruction [1], [2] and simultaneous localization and mapping (SLAM) [3], [4]. Recently, collaborative object localization using a team of robots has attracted an increased interest because of its improved object localization accuracy and resilience to sensor failures [5], [6]. The goal of collaborative object localization is to estimate locations of observed objects by fusing observations obtained by multiple robots from different perspectives [7], [8], [9], [10]. For example, as shown in Figure 1, two connected vehicles are able to improve shared situational awareness and decrease blind spots at an intersection by combining their observations to collaboratively localize street objects.

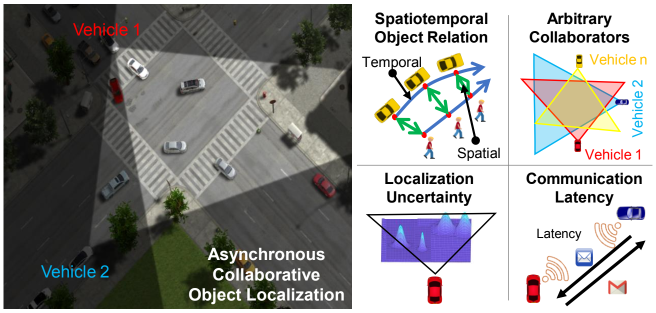


Fig. 1. A motivating scenario of asynchronous collaborative localization of street objects in connected driving. When some connected vehicles meet at an intersection, they can improve their shared situational awareness and decrease blind spots by incorporating their observations to collaboratively localize street objects.

Given its importance, several approaches have been developed to address collaborative object localization. Previous methods can be generally grouped into model-based and learning-based methods. Model-based methods are often based upon Bayesian filtering to localize objects, e.g., using Kalman filters [11] and other Bayesian filters [12]. These methods lack the ability to model the complex spatiotemporal relationship among objects, e.g., to model the impact of surrounding objects. The second category of methods use machine learning such as deep neural networks to localize objects [13], [14]. However, learning-based methods including deep networks assume a fixed number of observations, and cannot fuse observations from an arbitrary and dynamically changing number of robots. Moreover, for both categories of approaches, asynchronous observations caused by robot communication delays have not been well addressed.

In this paper, we introduce a novel asynchronous collaborative object localization approach, which integrates deep spatiotemporal graph learning and Bayesian modeling to perform multi-robot sensor fusion in an asynchronous fashion for collaborative object localization. We encode each observation obtained by each robot in a team as a graph, where the nodes denote detected objects in the robot's field of view and the edges denote their spatial relationships. We then encode a sequence of historical observations obtained by each robot as a spatiotemporal graph in order to encode temporal motions of the objects. Given the representation of spatiotemporal graphs, we formulate collaborative object localization as a multi-robot sensor fusion problem for state estimation. We

*This work was partially supported by NSF CAREER IIS-1942056, NSF IIS-1849359, and DARPA YFA D21AP10114-00.

[1] Peng Gao, Qingzhao Zhu and Hao Zhang are with the Human-Centered Robotics Laboratory at the Colorado School of Mines, Golden, CO 80401, USA. {gaopeng, zhuqingzhao, hzhang}@mines.edu.

[2] Rui Guo and Hongsheng Lu are with Toyota Motor North America, Mountain View, CA 94043, USA. endeavorui@gmail.com and hongsheng.lu@toyota.com.

[3] Brian Reily is with the DEVCOM Army Research Laboratory. brian.j.reily.civ@army.mil.

propose an uncertainty-aware graph learning method to estimate locations of objects and provide uncertainty quantification. Then, we introduce a method to integrate spatiotemporal graph learning and model-based estimation for asynchronous collaborative object localization.

The key contribution of the paper focuses on the proposal of an asynchronous collaborative object localization approach. The novelty of the proposed approach is twofold:

- We introduce a novel uncertainty-aware spatiotemporal graph learning approach, which is able to represent complex spatiotemporal relationships among observed objects and provide probabilistic estimations of object locations with uncertainty quantification.
- We propose a novel method for integrating deep spatiotemporal graph learning and model-based estimation, which fuses the asynchronous observations acquired by a team of robots. This approach explicitly addresses the latency of robot communications by asynchronous sensor fusion. It also offers the ability to fuse observations from an arbitrary and dynamically changing number of robots.

II. Related Work

Existing techniques for collaborative localization through sensor fusion can be grouped into two categories. *Model-based approaches* often apply Bayesian filtering to integrate observations obtained from multiple sensors, e.g., based on Kalman filters [11], particle filters [15], [16], covariance intersection [17] and Bayesian sequential filters [18], [19]. Previous model-based approaches are not able to represent the complex relationships among the objects or learn from data to improve the localization accuracy. *Learning-based approaches* apply machine learning methods to fuse multiple observations for object localization. For example, recurrent neural networks were used to encode object motions and localize the objects [13], graph neural networks were designed to represent spatial relationships of multiple objects [14], and convolutional neural networks were used to model visual-spatial relationships of the objects [20], [21], [22]. Learning-based methods are often not able to integrate observations that are obtained by an arbitrary and dynamically changing number of robots.

In addition, several learning-based methods were implemented to quantify the uncertainty in object localization, including Bayesian and non-Bayesian methods. Bayesian methods focus on modeling the distribution of model weights, such as Bayes by Backprop [23] and Monte Carlo Dropout [24]. However, Bayesian methods generally are computationally expensive compared with non-Bayesian methods [25]. A state-of-the-art non-Bayesian uncertainty quantification technique is deep ensemble, which averages predictions from multiple parallel networks to capture the deep network uncertainty [26], [27]. We propose a new multivariate loss function under this framework to quantify the uncertainty of estimated object locations in this paper.

Recently, several hybrid approaches were proposed for state estimation by integrating model-based and learning-based methods, e.g., by integrating convolutional neural networks and Bayesian filters for robot pose estimation [28], [29], and hybrid sensor fusion based on Gaussian process for location query [30]. However, existing hybrid approaches cannot well address asynchronous observations caused by the delay of robot communications. In this paper, we propose one of the first hybrid approaches for asynchronous collaborative object localization that is capable of quantifying localization uncertainty, addressing the latency of robot communications, and addressing the shortcomings of learning and model-based methods.

III. Approach

Notation. We use superscript t and n to denote the time step and the robot index, respectively. We use subscript i to denote the index of the object in an observation observed by a robot. For example, $\mathbf{v}_i^{t,n}$ denotes a feature vector of the i-th object observed by the n-th robot at time t.

A. Problem Formulation

Given M objects observed by a team of N robots, we represent each observation obtained by the n-th robot as a graph $\mathcal{G}^n = \{\mathcal{Z}^n, \mathcal{R}^n, \mathcal{E}^n\}, n = 1, 2, \dots, N$. The node set $\mathcal{Z}^n = \{\mathbf{z}_1^n, \mathbf{z}_2^n, \dots, \mathbf{z}_M^n\}$ denotes the observed locations of objects detected by the n-th robot, with $\mathbf{z}_i^n \in \mathbb{R}^3$ denoting the 3D central location of the i-th object. $\mathcal{R}^n = \{\mathbf{R}_1^n, \mathbf{R}_2^n, \dots, \mathbf{R}_M^n\}$ denotes the uncertainty in observations, where $\mathbf{R}_i^n \in \mathbb{R}^{3\times 3}$ is defined as the covariance of $\mathbf{z}_i^n$. The observation $\mathbf{z}_i^n$ and its uncertainty $\mathbf{R}_i^n$ can be assumed to follow a multivariate Gaussian distribution $\mathcal{N}(\mathbf{z}_i^n, \mathbf{R}_i^n)$ with $\mathbf{z}_i^n$ as the mean and $\mathbf{R}_i^n$ as the covariance. The edge set $\mathcal{E}^n = \{\mathbf{e}_{i,j}^n\}$ denotes the spatial relationships between a pair of detected objects, where $\mathbf{e}_{i,j}^{t,v} = 1$, if $\mathbf{z}_i^{t,v}$ and $\mathbf{z}_j^{t,v}$ are connected. We further encode a sequence of observations recorded from time 1 to time t by the n-th robot as a spatiotemporal graph $\mathcal{M}^n = \{\mathcal{G}^{1,n}, \mathcal{G}^{2,n}, \dots, \mathcal{G}^{t,n}\}$.

We represent the estimated locations of the objects observed by the n-th robot at time t as the states $\mathcal{X}^{t,n} = \{\mathbf{x}_i^{t,n}\}, i = 1, 2, \dots M$, where $\mathbf{x}_i^{t,n}$ denotes the state estimation (location) of the i-th object. In addition, we use $\mathcal{P}^{t,n} = \{\mathbf{P}_i^{t,n}\}, i = 1, 2, \dots, M$ to denote the uncertainties in $\mathcal{X}^{t,n}$, where $\mathbf{P}_i^{t,n} \in \mathbb{R}^{3\times 3}$ denotes the state estimation uncertainty in $\mathbf{x}_i^{t,n}$. Then, $\mathbf{x}_i^{t,n}$ and $\mathbf{P}_i^{t,n}$ can be assumed to follow the multivariate Gaussian distribution $\mathcal{N}(\mathbf{x}_i^{t,n}, \mathbf{P}_i^{t,n})$.

In this paper, we formulate collaborative object localization as a multi-robot state estimation problem, with the objective of estimating object states $\mathcal{X}^{t,n}$ (i.e., 3D locations of the objects) by fusing sequences of observations obtained from multiple robots $\{\mathcal{M}^n\}, n = 1, 2, \dots N$. The correspondences of objects in different observations can be obtained through the recent methods [31], [32].

B. Uncertainty-Aware Spatiotemporal Graph Learning

Based on the spatiotemporal graph representations, our approach is able to represent complex spatiotemporal relationships and estimates learning-based locations of observed objects based on their historical motions.

Our approach consists of three components. First, temporal motions of the objects are encoded using a long short-term memory (LSTM) encoder [33], which is able to process sequential data and mitigates the vanishing gradient problem. Second, the spatial relationships of objects are embedded by a graph attention neural network [34], which captures the spatial impacts of each object from an arbitrary number of neighbor objects (e.g., avoiding collision or changing movement direction). The spatial and temporal embeddings of each object are defined as follows:

$$\mathbf{m}_i^{t-1,n} = \phi\left(\mathbf{m}_i^{t-2,n}, \mathbf{z}_i^{t-1,n} - \mathbf{z}_i^{t-2,n}, \mathbf{W}^e\right) \tag{1}$$

$$\mathbf{s}_i^{t-1,n} = \mathrm{ReLu}\left(\sum_{e_{i,j}^{t-1,n}} \alpha_{i,j}^{t-1,n} \mathbf{W}^a \mathbf{m}_j^{t-1,n}\right) \tag{2}$$

where $\mathbf{z}_i^{t-1,n} - \mathbf{z}_i^{t-2,n}$ denotes the relative location (motion) of the i-th object observed by the n-th robot from time $t-2$ to $t-1$, ϕ denotes a one-layer LSTM encoder network with a trainable parameter matrix $\mathbf{W}^e$ that is shared among all objects, and $\mathbf{m}_i^{t-1,n}$ denotes the temporal motion embedding of the i-th object. By iteratively running the encoder, we capture the motion of the object from time 1 to $t-1$. In addition, $\mathbf{W}^a$ is the trainable parameter matrix for the attention network, $\alpha_{i,j}^{t-1,n}$ denotes the impact of the j-th neighborhood object on the i-th object, which is obtained through the graph attention neural network [34], $e_{i,j}^{t-1,n}$ denotes the connection between the i-th object and its j-th neighborhood object, $ReLu$ denotes the nonlinear activation function [35], and $\mathbf{s}_i^{t-1,n}$ denotes the spatial embedding of the i-th object, which captures the impacts on the i-th object from its neighbors. Given the temporal embedding $\mathbf{m}_i^{t-1,n}$ and spatial embedding $\mathbf{s}_i^{t-1,n}$, we predict the states of objects at current time t through a LSTM decoder [33], which is defined as

$$\mathbf{x}_i^{t,n} = \psi\left(\mathbf{m}_i^{t-1,n} || \mathbf{s}_i^{t-1,n}, \mathbf{z}_i^{t-1,n} - \mathbf{z}_i^{t-2,n}, \mathbf{W}^d\right) \tag{3}$$

where $||$ denotes the concatenation operator, ψ denotes a one-layer LSTM decoder network with a trainable parameter matrix $\mathbf{W}^d$, and $\mathbf{x}_i^{t,n}$ is the estimated states of the i-th object observed by the n-th robot at time t. By running Eq. (3) for multiple times, we can predict arbitrary steps in the future.

Due to the limited amount of training data and noisy observations, the learning-based state estimations obtained from the spatiotemporal graph neural network exhibit uncertainties caused by model bias and perception uncertainty [24]. Inspired by deep ensemble technique [36], we quantify the estimation uncertainties in two steps.

First, to quantify the data uncertainty, we modify the graph learning model to output the estimated location and the estimation uncertainty in the final layer. For simplification, we write $\mathbf{x}_i^{t,n}$ as $\mathbf{x}^t$ by ignoring the index of objects and robots. The output is assumed to follow the Gaussian distribution, which is defined as

$$q_\theta(\mathbf{x}^t | \mathcal{M}^{1:t-1}) \sim \mathcal{N}(\boldsymbol{\mu}_q^t, \boldsymbol{\Sigma}_q^t) \tag{4}$$

where the mean value $\boldsymbol{\mu}_q^t$ denotes the estimated location, the covariance $\boldsymbol{\Sigma}_q^t$ denotes the quantified uncertainty of $\boldsymbol{\mu}_q^t$, and θ denotes the network parameters. To train this model to provide uncertainty estimations, based on the score rule for deep ensemble [36], we design a multivariate negative log-likelihood loss (NLL) defined as follows:

$$\begin{aligned} L(\theta) &= -\log\left(q_\theta\left(\mathbf{x}^t | \mathcal{M}^{1:t-1}\right)\right) \\ &= -\log\left(\frac{1}{\sqrt{2\pi}^3 |\boldsymbol{\Sigma}_q^t|} e^{(\boldsymbol{\mu}_q^t - \mathbf{y}^t)^\top (\boldsymbol{\Sigma}_q^t)^{-1} (\boldsymbol{\mu}_q^t - \mathbf{y}^t)}\right) \\ &= \frac{1}{2}(\boldsymbol{\mu}_q^t - \mathbf{y}^t)^\top (\boldsymbol{\Sigma}_q^t)^{-1} (\boldsymbol{\mu}_q^t - \mathbf{y}^t) \\ &\quad + \frac{1}{2}\log(|\boldsymbol{\Sigma}_q^t|) + \frac{3}{2}\log(2\pi) \end{aligned} \tag{5}$$

where $|\cdot|$ denotes the determinant operator and $\mathbf{y}^t$ denotes the object's ground truth location at time t. $\boldsymbol{\Sigma}_q^t$ represents the covariance of the Gaussian distribution with respect to the estimations and the ground truth, which captures the data uncertainty [36].

Second, to quantify the model uncertainty, instead of using just one model, an ensemble of K models is used to form a Gaussian mixture model, which is defined as:

$$p(\mathbf{x}^t | \mathcal{M}^{1:t-1}) = \frac{1}{K}\sum_{k=1}^{K} q_{\theta_k}(\mathbf{x}^t | \mathcal{M}^{1:t-1}) \tag{6}$$

where $\{\theta_k\}^K$ denotes the set of parameters in K networks. Based on deep ensemble theory [36], the Gaussian mixture model can be approximated as the Gaussian distribution, defined as

$$\mathcal{N}(\boldsymbol{\mu}_p^t, \boldsymbol{\Sigma}_p^t) \approx \frac{1}{K}\sum_{k=1}^{K} \mathcal{N}(\boldsymbol{\mu}_{q_k}^t, \boldsymbol{\Sigma}_{q_k}^t) \tag{7}$$

Thus, the final estimation is $\mathbf{x}^t = \boldsymbol{\mu}_p^t$ and the estimation uncertainty is defined as:

$$\begin{aligned} \mathbf{Q}^t = &\overbrace{\frac{1}{K}\sum_{k}^{K}\left(\boldsymbol{\Sigma}_{q_k}^t\right)}^{\text{Data uncertainty}} \\ &+ \overbrace{\frac{1}{K}\sum_{k}^{K}\left(\mathrm{diag}(\boldsymbol{\mu}_{q_k}^t)\mathrm{diag}(\boldsymbol{\mu}_{q_k}^t)\right) - \mathrm{diag}(\boldsymbol{\mu}_p^t)\mathrm{diag}(\boldsymbol{\mu}_p^t)}^{\text{Model uncertainty}} \end{aligned} \tag{8}$$

where diag$(\cdot)$ is the diagonal operator, and $\mathbf{Q}_i^{t,n}$ is the process uncertainty, which captures the model and data uncertainties in the learning-based estimation. The data uncertainty is captured by each ensemble model trained with NLL, which describes the ambiguity in targets $\mathbf{x}^t$ given the inputs $\mathcal{M}^{1:t-1}$. The model uncertainty is captured by averaging a combination of ensemble models with consistent training data. If the training data is infinite, then $\boldsymbol{\mu}_p^t = \boldsymbol{\mu}_{q_k}^t$. In this case, the model uncertainty will be eliminated and the total uncertainty $\mathbf{Q}^t$ will be reduced to the data uncertainty.

C. Single-Robot State Estimation

Based on the quantified process uncertainties, we further integrate learning-based state estimations with single-robot measurements to obtain the single-robot state estimation. In conventional sensor fusion methods, the process uncertainties are generally identified through empirical settings [11], [29]. In this paper, we integrate the quantified process uncertainty into the sensor fusion process as:

$$\mathbf{P}_i^{t,n} = \mathbf{P}_i^{t-1,n} + \mathbf{Q}_i^{t,n} \tag{9}$$

where $\mathbf{P}_i^{t,n}$ denotes the uncertainty of the learning-based state estimation $\mathbf{x}_i^{t,n}$, which is computed by the sum of the process uncertainty $\mathbf{Q}_i^{t,n}$ and the state estimation uncertainty at the previous time step. Based on Kalman filter [37], we update the learning-based state estimation with the measurement $\mathbf{z}^{t,n}$ to obtain the single-robot state estimation, which is defined as follows:

$$\mathbf{K}^{t,n} = \mathbf{P}_i^{t,n}(\mathbf{P}_i^{t,n} + \mathbf{R}_i^{t,n})^{-1} \tag{10}$$

$$\hat{\mathbf{x}}_i^{t,n} = \mathbf{x}_i^{t,n} + \mathbf{K}^{t,n}(\mathbf{z}_i^{t,n} - \mathbf{x}_i^{t,n}) \tag{11}$$

$$\hat{\mathbf{P}}_i^{t,n} = (\mathbf{I} - \mathbf{K}_i^{t,n})\mathbf{P}_i^{t,n} \tag{12}$$

where $\mathbf{z}^t$ denotes the measurement at time t, $\mathbf{R}_i^{t,n}$ denotes the uncertainty in the measurement, $\mathbf{K}^{t,n}$ denotes the Kalman gain that encodes the relative weight of both the state estimation and the measurement given their uncertainties, $\hat{\mathbf{x}}_i^{t,n}$ denotes the updated state estimation and $\hat{\mathbf{P}}_i^{t,n}$ denotes the updated uncertainty of the single-robot state estimation.

D. Asynchronous Multi-robot State Estimation

Based on the single-robot state estimation and the state estimation uncertainty, our approach computes a final estimation for each robot by integrating multi-robot state estimations. We also address the challenges of asynchronous multi-robot state estimations caused by communication time delays and integrating arbitrary number of state estimations provided by multiple robots.

In real-world multi-robot collaborative object localization, asynchronous multi-robot state estimations caused by communication time delays usually significantly increase the error in object localization, as the time-delayed locations of objects can be far away from the current locations of the objects. In order to address asynchronous state estimations, we first compensate the state estimations with time delay by using a deep graph learning network as follows:

$$\tilde{\mathbf{x}}_i^{t,n}, \tilde{\mathbf{Q}}_i^{t,n} = \Phi\left(\hat{\mathbf{x}}_i^{1,n}, \hat{\mathbf{x}}_i^{2,n}, \dots, \hat{\mathbf{x}}_i^{t-\Delta T,n}\right) \tag{13}$$

where $\tilde{\mathbf{x}}_i^{t,n}$ and estimation uncertainty $\tilde{\mathbf{Q}}_i^{t,n}$ denotes the compensated state estimation and estimation uncertainty. Φ denotes the deep graph learning network, which has the same architecture as the the uncertainty-aware graph neural network. ΔT denotes the time delay in the state estimation provided by the n-th robot, which can be obtained through timestamps or learning models [38].

Since the compensation process also introduces uncertainty into the state estimation, we update the uncertainty of the compensated state estimation as follows:

$$\tilde{\mathbf{P}}_i^{t,n} = \hat{\mathbf{P}}_i^{t-\Delta T,n} + \tilde{\mathbf{Q}}_i^{t,n} \tag{14}$$

where $\tilde{\mathbf{P}}_i^{t,n}$ denotes the updated uncertainty of the compensated state estimation, which is obtained by the sum of the process uncertainties $\tilde{\mathbf{Q}}_i^{t,n}$ generated in the compensation process and the state estimation uncertainty $\hat{\mathbf{P}}_i^{t-\Delta T,n}$ computed ΔT time ago. To improve the robustness to noise in collaborative object localization, we propose a multi-robot fusion gain to integrate state estimations provided by an arbitrary number of collaborative robots, which is defined as follows:

$$\mathbf{E}_i^{t,n} = \left(\sum_{j=1}^{N}(\tilde{\mathbf{P}}_i^{t,j})^{-1}\right)^{-1} (\tilde{\mathbf{P}}_i^{t,n})^{-1} \tag{15}$$

where $\mathbf{E}_i^{t,n} \in \mathbb{R}^{3\times3}$ denotes the state fusion gain for each robot's state estimations. In addition, $\mathbf{E}_i^{t,n}$ follows the constraint $\sum_{n=1}^{N}\mathbf{E}_i^{t,n} = \mathbf{I}$, where $\mathbf{I} \in \mathbb{R}^{3\times3}$ denotes an identity matrix. The fusion gain for each robot represents the weight of each robot's state estimation in all the multi-robot state estimations given the normalized state estimation uncertainties. The final state is defined as follows:

$$\tilde{\mathbf{x}}_i^{\prime t,n} = \mathbf{E}_i^{t,n}\tilde{\mathbf{x}}_i^{t,n} + \sum_{j=1,j\neq n}^{N} \mathbf{E}_i^{t,j}\sigma(\tilde{\mathbf{x}}_i^{t,j}) \tag{16}$$

where σ denotes the transformation function that transforms the multi-robot state estimations to the n-th robot's coordinates based on camera extrinsic parameters [39]. The camera extrinsic parameters can be obtained through GPS [40] or deep learning algorithm [41]. $\tilde{\mathbf{x}}_i^{\prime t,n}$ denotes the final state estimation of the i-th object observed by the n-th robot at time t, which is computed by the sum of single-robot state estimations weighted by the fusion gains. If a robot's state estimation has large uncertainty (e.g., existing large communication delay), then its contribution will be heavily weakened during the fusion. The uncertainty of the final state estimation is defined as

$$\tilde{\mathbf{P}}_i^{\prime t,n} = \left(\sum_{n=1}^{N}(\tilde{\mathbf{P}}_i^{t,n})^{-1}\right)^{-1} \tag{17}$$

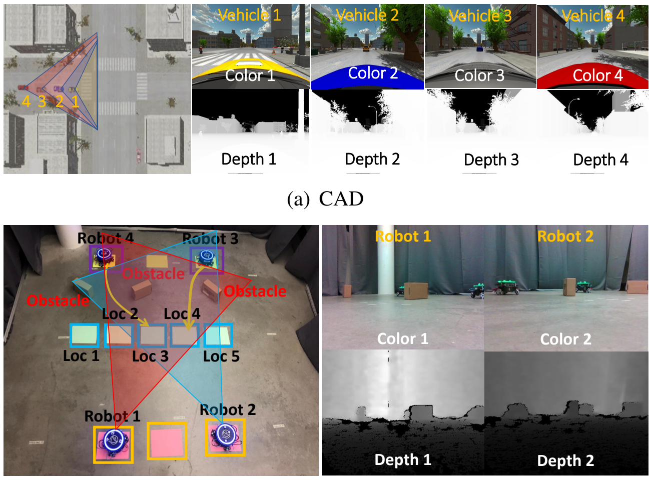
where $\tilde{\mathbf{P}}_i^{\prime t,n}$ denotes the uncertainty of the final state estimation $\tilde{\mathbf{x}}_i^{\prime t,n}$, which is obtained by integrating all the single-robot state estimation uncertainties.

The complexity of our algorithm is $O(NKG)$, where $O(G)$ denotes the complexity of the graph learning network. Since the time compensation for each robot can be run parallel, the complexity of our algorithm reduces to $O(KG)$. Since the ensemble models can also be run parallel, the complexity can further reduce to $O(G)$. When our algorithm is executed on a Linux machine with an i9 3.6 GHz CPU, 3 TB memory, and an Nvidia RXT 2080Ti GPU, the average runtime is around 10 Hz.

IV. Experiments

A. Experimental Setup

We use both high-fidelity robotic simulations and physical robots to evaluate our method for asynchronous collaborative object localization in multi-robot systems. Our evaluation consists of two scenarios, including simulated Connected Autonomous Driving (CAD) and real-world Multi-Robot Object



(a) CAD

(b) MROL

Fig. 2. Illustrations of the simulated Connected Autonomous Driving (CAD) and Multi-Robot Object Localization (MROL) scenarios that are used in our experiments.

Localization (MROL), as demonstrated in Figure 2. CAD includes 1500 data instances recorded at 10Hz by four RGB-D cameras mounted on the autonomous vehicles. The ground truth locations are obtained from the connected vehicle simulator provided by Toyota. MROL includes 800 data instances, which are recorded by two RGB-D cameras mounted on robots at 10Hz. The ground truth locations are obtained from an Optitrack motion tracking system.

In the experiments, we only perform our approach to the objects observed by more than one robot and the association of the same object in different observations is identified based on the recent work [32]. The node attributes are generated from 3D locations of objects and the edges are fully connected. The LSTM encoder in Eq. (1) and decoder in Eq. (3) are constructed by only one LSTM layer, with $\mathbf{W}^e$ setting to the dimension of 3×32 and $\mathbf{W}^d$ setting to 64×6. The graph attention network is constructed with two layers, with $\mathbf{W}^a$ having the dimension of 32×64 followed by dropout with probability 0.1. The number of ensemble models is $K = 5$. Initially, the state $\mathbf{x}$ is set to an all zero matrix and the state estimation uncertainty $\mathbf{P}$ is a diagonal matrix with the diagonal values set to 10. Observation uncertainty $\mathbf{R}$ is calculated based on the depth sensor model given the real depth values of objects [42]. ADMM is used as the optimization solver [43].

We implemented our full approach and two baseline methods for asynchronous collaborative object localization. The first baseline method named **Ours-U** that only quantifies the process uncertainty without addressing asynchronous observations caused by communication latency among multiple robots. The second baseline method named **Ours-D** that only addresses asynchronization and uses a fixed value as the process uncertainty. In addition, we compare with four previous methods, including **AOM** [44] that directly averages the locations of the same objects in different observations, **MMT** [11] that estimates the locations of objects based on the classical Kalman filter, **STTP** [13] that localizes objects based on spatiotemporal graph learning, **MCOL** [29] that integrates model-based and learning-based estimations for object localization, but without considering uncertainty quantification and asynchronous sensing data.

TABLE I
QUANTITATIVE RESULTS ON CAD AND MROL.THE UNIT IS METER.

Method	CAD		MROL	
	DE	Rel-DE	DE	Rel-DE
AOM [44]	2.2288	0.1197	0.0343	0.0283
STTP [13]	1.7859	0.0950	0.0357	0.0294
MMT [11]	1.6240	0.0866	0.0344	0.0283
MCOL [29]	1.4445	0.0771	0.0319	0.0263
Ours-D	1.4397	0.0768	0.0310	0.0256
Ours-U	1.3688	0.0730	0.0304	0.0251
Ours	**1.3533**	**0.0723**	**0.0288**	**0.0238**

We follow a widely used experimental setup [13], [45] to evaluate our approach. Displacement error (**DE**) is used to evaluate the localization accuracy, which is defined as the Euclidean distance between the estimated location and the ground truth location. The unit of DE is meter. Relative displacement error (**REL-DE**) is used to evaluate the localization accuracy relative to the measurement distance, which is defined as the ratio of the displacement error over the ground truth location.

B. Connected Autonomous Driving Simulation

Our approach is first evaluated on the CAD scenario, aiming to localize the dynamic pedestrians and vehicles given the observations from multiple connected vehicles. The number of connected vehicles is from 1 to 4, and communication delays between connected vehicles is from 0.1 to 0.7 sec.

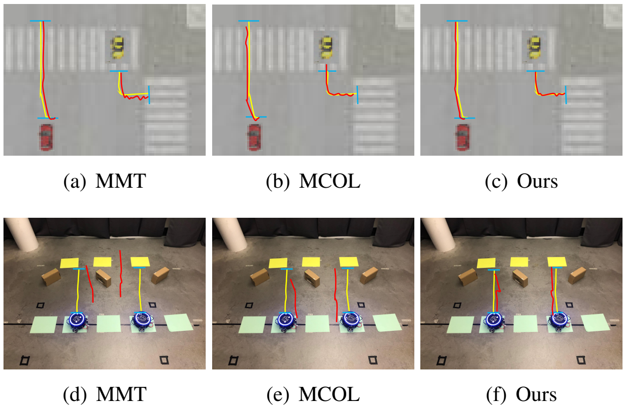
The qualitative results over CAD are presented in Figures 3(a)-3(c). We observe that the movement of the yellow vehicle has a sharp turn, which is challenging to accurately localize it. In this situation, our approach works much better compared with the other two methods. Moreover, we observe that MMT works poorly as shown in Figure 3(a), as the linear velocity assumption in MMT cannot estimate the non-linear movement of objects. MCOL improves the localization performance, however, its estimations contain a lagging effect (the estimated locations lag behind the ground truth), which is caused by the asynchronous state estimations provided by multiple connected vehicles. Our approach outperforms all these methods, due to its ability to model the complex spatiotemporal relationship among objects, and to address multi-robot asynchronous state estimations.

The quantitative results for the CAD scenario are shown in Table I. We observe that our baseline methods outperform the compared previous methods, which indicates the importance of quantifying estimation uncertainty and addressing asynchronous multi-robot state estimations caused by communication latency for collaborative object localization. In addition, AOM performs poorly due to the noisy observations. STTP and MMT obtain an improved performance by utilizing learning-based state estimation to encode historical estimations, and by using model-based sensor fusion for object localization. MCOL further improves the performance by integrating learning-based and model-based state estimations. Due to uncertainty quantification and addressing asynchronous state estimations resulting from communication latency, our approach can readily integrate multi-robot asynchronous state estimations and performs the best.

C. Real-world Multi-robot Object Localization

Our method is also evaluated in the MROL scenario. In this scenario, multiple robots collaboratively localize the objects to assist with collision avoidance. The object instances used in this scenario include different dynamic robots. The MROL scenario is challenging due to the large noise, large communication latency (ranging from 0.1 sec to 0.7 sec) and uncertainty existing in robot observations, which is caused by non-smooth movements of robots and occlusion.

The qualitative results obtained on MROL are presented in Figures 3(d)-3(f). We can see visually that our approach outperforms MMT and MCOL. Specifically, MCOL's estimations lag the ground truth caused by the communication latency in the multi-robot system. In addition, our approach has better localization accuracy than MCOL, as we use the quantified process uncertainty in the sensor fusion process, which can appropriately weight learning-based state estimations during sensor fusion, rather than using a constant value as in MCOL. Thus, the proposed approach achieves the best performance in this realistic multi-robot object localization scenario. The quantitative results obtained in the MROL scenario are presented in Table I. Similar to the other scenarios, our approach and its baseline methods also outperform MCOL by quantifying the uncertainty of the learning process and addressing asynchronous state estimations, resulting in the best performance on this real-world multi-robot scenario.



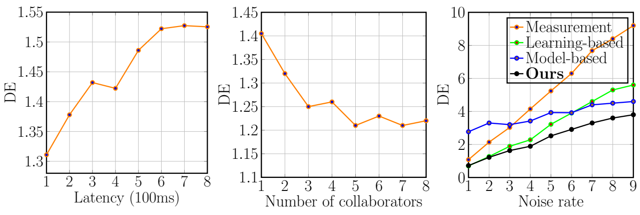
(a) MMT (b) MCOL (c) Ours

(d) MMT (e) MCOL (f) Ours

Fig. 3. Qualitative experimental results of our approach and comparisons with the previous and baseline methods on CAD and MROL. Ground truth paths are shown in yellow, the estimated paths are shown in red, and the blue lines denotes the starting and ending points of trajectories (indicating the lagging effects in Figure 3(b) and 3(e)).

D. Discussion

Using the CAD simulations, we further investigate our approach's characteristics, including the effect of communication latency, the number of collaborating robots, and the robustness to noise.



(a) Latency (b) # Collab. (c) Robustness

Fig. 4. Analysis of our method's characteristics. The unit of DE is meter.

Figure 4(a) depicts how our approach is affected by increasing communication time delays among vehicles, which causes asynchronous state estimations. We observe that the performance of our approach gradually decreases as the communication latency among robots increases. When the communication latency increases beyond 600 ms, the performance becomes stable with small fluctuations. In this case, each robot uses only its own observations to localize objects without using any collaboration from other robots. The figure generally shows that our approach can effectively fuse asynchronous multi-robot state estimations with large communication delays.

Figure 4(b) illustrates the influence of the number of collaborative robots on CAD. We can see that the performance of our approach gradually improves as the number of collaborators increases. When the number of collaborators is larger than 5, the performance becomes stable with small fluctuations. The best performance is around $[1.2145, 1.2373]$ meter. Thus, integrating multi-robot observations can effectively improve the performance of collaborative object localization.

Figure 4(c) illustrates the effect of varying noise rates. It compares the performance of measurement-only localization, only the learning-based state estimation, only the model-based state estimation, and on our complete method. We observe that the area under using measurements directly is 0.5068, the area under only using model-based state estimation is 0.3744, the area under only using learning-based state estimations is 0.3217 and our complete approach is 0.2561. Thus, integrating model and learning-based estimations effectively improves the robustness to noise.

V. Conclusion

In this paper, we propose a novel asynchronous collaborative object localization approach that integrates uncertainty-aware spatiotemporal graph learning and model-based state estimation in a principled way to perform asynchronous collaborative object localization. Our approach learns spatiotemporal relationships among objects to provide probabilistic estimations of object locations. Then, we propose a new method to integrate learning and model-based state estimations, which is able to fuse asynchronous observations obtained from an arbitrary number of robots to collaboratively localize objects. Extensive experiments are conducted to evaluate our approach and the experimental results show that our approach outperforms existing methods and achieves state-of-the-art performance on asynchronous collaborative object localization.

References

[1] J. Delmerico, S. Isler, R. Sabzevari, and D. Scaramuzza, “A comparison of volumetric information gain metrics for active 3D object reconstruction,” *AuRo*, vol. 42, no. 2, pp. 197–208, 2018.

[2] J. I. Vasquez-Gomez, L. E. Sucar, and R. Murrieta-Cid, “View planning for 3D object reconstruction with a mobile manipulator robot,” in *IROS*, 2014.

[3] S. L. Bowman, N. Atanasov, K. Daniilidis, and G. J. Pappas, “Probabilistic data association for semantic SLAM,” in *ICRA*, 2017.

[4] S. Sharma, J. A. Ansari, J. K. Murthy, and K. M. Krishna, “Beyond pixels: Leveraging geometry and shape cues for online multi-object tracking,” in *ICRA*, 2018.

[5] S. Wei, D. Yu, C. L. Guo, L. Dan, and W. W. Shu, “Survey of connected automated vehicle perception mode: from autonomy to interaction,” *ITS*, vol. 13, no. 3, pp. 495–505, 2018.

[6] B. Reily, C. Reardon, and H. Zhang, “Multi-modal sensor fusion and selection for enhanced situational awareness,” in *Virtual, Augmented, and Mixed Reality (XR) Technology for Multi-Domain Operations*. International Society for Optics and Photonics, 2021.

[7] J. J. Acevedo, J. Messias, J. Capitán, R. Ventura, L. Merino, and P. U. Lima, “A dynamic weighted area assignment based on a particle filter for active cooperative perception,” *RAL*, vol. 5, no. 2, pp. 736–743, 2020.

[8] R. Guo, H. Lu, P. Gao, Z. Zhang, and H. Zhang, “Collaborative localization for occluded objects in connected vehicular platform,” in *VTC*, 2019.

[9] E. E. Marvasti, A. Raftari, A. E. Marvasti, Y. P. Fallah, R. Guo, and H. Lu, “Cooperative Lidar object detection via feature sharing in deep networks,” *ArXiv*, 2020.

[10] H. Wang, C. Zhang, Y. Song, and B. Pang, “Master-followed multiple robots cooperation SLAM adapted to search and rescue environment,” *Int J Control Autom Syst*, 2018.

[11] X. Weng, J. Wang, D. Held, and K. Kitani, “3D multi-object tracking: A baseline and new evaluation metrics,” *IROS*, 2020.

[12] P. Gao, R. Guo, H. Lu, and H. Zhang, “Correspondence identification for collaborative multi-robot perception under uncertainty,” *Autonomous Robots*, pp. 1–16, 2021.

[13] Y. Huang, H. Bi, Z. Li, T. Mao, and Z. Wang, “STGAT: Modeling spatial-temporal interactions for human trajectory prediction,” in *ICCV*, 2019.

[14] B. Ivanovic and M. Pavone, “The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs,” in *ICCV*, 2019.

[15] X. Deng, A. Mousavian, Y. Xiang, F. Xia, T. Bretl, and D. Fox, “PoserBPF: A rao-blackwellized particle filter for 6D object pose tracking,” *RSS*, 2019.

[16] F. Qin, Y. Li, Y.-H. Su, D. Xu, and B. Hannaford, “Surgical instrument segmentation for endoscopic vision with data fusion of rediction and kinematic pose,” in *ICRA*, 2019.

[17] S. J. Julier and J. K. Uhlmann, “Using covariance intersection for slam,” *Robotics and Autonomous Systems*, vol. 55, no. 1, pp. 3–20, 2007.

[18] B. Stenger, A. Thayananthan, P. H. Torr, and R. Cipolla, “Model-based hand tracking using a hierarchical bayesian filter,” *PAMI*, vol. 28, no. 9, pp. 1372–1384, 2006.

[19] M. Ullah, A. K. Mohammed, F. A. Cheikh, and Z. Wang, “A hierarchical feature model for multi-target tracking,” in *ICIP*, 2017.

[20] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, “Multi-view 3D object detection network for autonomous driving,” in *CVPR*, 2017.

[21] Y. Meng, Y. Lu, A. Raj, S. Sunarjo, R. Guo, T. Javidi, G. Bansal, and D. Bharadia, “SIGNET: Semantic instance aided unsupervised 3D geometry perception,” in *CVPR*, 2019.

[22] Z. Yin and J. Shi, “GeoNet: Unsupervised learning of dense depth, optical flow and camera pose,” in *CVPR*, 2018.

[23] W. Bao, Q. Yu, and Y. Kong, “Uncertainty-based traffic accident anticipation with spatio-temporal relational learning,” in *ACM-MM.*, 2020.

[24] Y. Gal and Z. Ghahramani, “Dropout as a Bayesian approximation: Representing model uncertainty in deep learning,” in *ICML*, 2016.

[25] P. Gao and H. Zhang, “Bayesian deep graph matching for correspondence identification in collaborative perception,” in *RSS*, 2021.

[26] S. Fort, H. Hu, and B. Lakshminarayanan, “Deep ensembles: A loss landscape perspective,” *ArXiv*, 2019.

[27] B. Lakshminarayanan, A. Pritzel, and C. Blundell, “Simple and scalable predictive uncertainty estimation using deep ensembles,” in *NIPS*, 2017.

[28] N. Akai, T. Hirayama, and H. Murase, “Hybrid localization using model- and learning-based methods: Fusion of Monte Carlo and E2E localizations via importance sampling,” in *ICRA*, 2020.

[29] P. Gao, R. Guo, H. Lu, and H. Zhang, “Multi-view sensor fusion by integrating model-based estimation and graph learning for collaborative object localization,” *ICRA*, 2021.

[30] T. D. Barfoot, C. H. Tong, and S. Särkkä, “Batch continuous-time trajectory estimation as exactly sparse gaussian process regression.” in *RSS*, vol. 10, 2014.

[31] P. Gao, Z. Zhang, R. Guo, H. Lu, and H. Zhang, “Correspondence identification in collaborative robot perception through maximin hypergraph matching,” in *ICRA*, 2020, pp. 3488–3494.

[32] P. Gao, R. Guo, H. Lu, and H. Zhang, “Regularized graph matching for correspondence identification under uncertainty in collaborative perception,” in *RSS*, 2020.

[33] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, “LSTM: A search space odyssey,” *IEEE transactions on neural networks and learning systems*, vol. 28, no. 10, pp. 2222–2232, 2016.

[34] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, “Graph attention networks,” *ICLR*, 2018.

[35] P. Ramachandran, B. Zoph, and Q. V. Le, “Searching for activation functions,” *ICLR Workshop*, 2018.

[36] B. Lakshminarayanan, A. Pritzel, and C. Blundell, “Simple and scalable predictive uncertainty estimation using deep ensembles,” *NIPS*, 2017.

[37] Q. Li, R. Li, K. Ji, and W. Dai, “Kalman filter and its application,” in *IINS*, 2015, pp. 74–77.

[38] Q. Zhang and A. B. Chan, “Single-frame based deep view synchronization for unsynchronized multi-camera surveillance,” *ArXiv*, 2020.

[39] Q. Zhang and R. Pless, “Extrinsic calibration of a camera and laser range finder,” in *IROS*, 2004.

[40] S. Brahmbhatt, J. Gu, K. Kim, J. Hays, and J. Kautz, “Geometry-aware learning of maps for camera localization,” in *CVPR*, 2018.

[41] A. Kendall, M. Grimes, and R. Cipolla, “PoseNet: A convolutional network for real-time 6-DOF camera relocalization,” in *ICCV*, 2015.

[42] K. Khoshelham, “Accuracy analysis of Kinect depth data,” in *ISPRS workshop laser scanning*, 2011.

[43] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, *et al.*, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

[44] M. Ji, J. Gall, H. Zheng, Y. Liu, and L. Fang, “SurfaceNet: An end-to-end 3D neural network for multiview stereopsis,” in *ICCV*, 2017.

[45] C. Godard, O. Mac Aodha, and G. J. Brostow, “Unsupervised monocular depth estimation with left-right consistency,” in *CVPR*, 2017.