Deep Odometry Systems on Edge with EKF-LoRa Backend for Real-Time Indoor Positioning

Zhuangzhuang Dai, Muhamad Risqi U. Saputra, Chris Xiaoxuan Lu, Vu Tran, L. N. S. Wijayasingha, M. Arif Rahman, John A. Stankovic, Andrew Markham, and Niki Trigoni

***Abstract*—Ubiquitous positioning for pedestrians in adverse environments has been a long standing challenge. Despite dramatic progress made by Deep Learning, multi-sensor deep odometry systems still pose a high computational cost and suffer from cumulative drifting errors over time. Thanks to the increasing computational power of edge devices, we propose a novel ubiquitous positioning solution by integrating state-of-the-art deep odometry models on edge with an EKF (Extended Kalman Filter)-LoRa backend. We carefully select and compare three sensor modalities, i.e., an Inertial Measurement Unit (IMU), a millimetre-wave (mmWave) radar, and a thermal infrared camera, and implement their deep odometry inference engines to run in real-time. A pipeline for deploying deep odometry on edge platforms with different resource constraints is proposed. We design a LoRa link for positional data backhaul and project aggregated positions of deep odometry into the global frame. We find that a simple EKF backend is sufficient for generic odometry calibration with over 34% accuracy gains against any standalone deep odometry system. Extensive tests in different environments validate the efficiency and efficacy of our proposed positioning system.**

I. Introduction

POSITIONING systems play a key role in human-centric technologies. However, prevailing positioning systems, such as satellite navigation and radio positioning, depend on elaborate infrastructure as a prerequisite. Fingerprint-based localization requires expensive site-survey yet is restricted to a specific area. Recently, more research interests are drawn to ubiquitous positioning that is environment-agnostic. Search and rescue operation, underground inspection, and subterranean environment exploration will all benefit significantly from such positioning systems with limited or no access to preinstalled access points.

The use of sensors has witnessed an outburst in the past decade, such as the use of Inertial Measurement Units (IMU). The IMU is a particularly low-cost, low-power motion sensor which lends itself as a ubiquitous candidate in modern active devices. Nevertheless, owing to its ego-centric nature the IMU incurs a large amount of noise and intrinsic biases. For instance, state-of-the-art Inertial Navigation Systems (INS) which only rely on the IMU suffer from large drifting errors. Visual cameras, on the other hand, create rich visual data from the surroundings. Visual Odometry (VO) based on either feature points or direct tracking has demonstrated strengths in ubiquitous positioning [11]. With the aid of Deep Learning and visual-inertial data fusion, a sub-meter precision has been achieved. However, VO faces challenges posed by visual degradation such as smoke, glare, and darkness. For instance, VO performs poorly in smoky environments. Emerging sensors in conjunction with complex Deep Neural Networks (DNN) are recently proposed to tackle these problems, namely thermal infrared camera [16] and mmWave radar [18]. Thermal infrared cameras are able to detect radiation of objects so as to gain vision even in darkness. On the other hand, mmWave radar generates sparse point clouds that can penetrate smoke. Those sensor modalities make appealing alternatives for positioning in adverse environments.

Despite the capabilities to infer position in adverse environments, sensor-aided positioning systems may have to process the sensor data on the edge, especially when cloud or server off-loading is not possible [9]. Conducting complex positioning algorithms of DNN on resource-constrained devices remains non-trivial. Few studies have investigated the real-time performance of applying these deep odometry systems [10]. Specifically, on-device computing is often confronted by a lack of processing power, a lack of memory space, and a high power consumption.

Odometry systems often find severe drifting problems because positions are aggregated based on relative positional changes. To project a deep odometry positioning module to the global coordinate system, a reference positioning mechanism is needed. We propose to use a LoRa backend for not only reference positioning, but also wireless communications. LoRa, a low-power wide-area-network modulation scheme, stands out for its long range and easy deployment. Existing wireless technologies, such as Wi-Fi, LTE [1], Ultra-Wide Band (UWB) [12] etc., require a mammoth project to setup base stations. A LoRa access point (AP) can be a gateway or simply another LoRa node. Time of Flight (TOF) or Angle-of-Arrival (AOA) may be difficult to extract for positioning purposes as synchronisation and hardware setup can be costly [2]. We argue a device-free metric, the Received Signal Strength Indicator (RSSI), is all one needs for positioning calibration [15]. The RSSI reading which comes with every received packet can be used to reveal the range. Given excellent positioning accuracy delivered by deep odometry systems in a short term, we propose a generic EKF fusion of RSSIs to constrain the odometry drift. The basic hypothesis is that with sparsely

Z. Dai, V. Tran, A. Markham, and N. Trigoni are with University at Oxford, Oxford, England, United Kingdom, e-mail: firstname.surname@ox.ac.uk.

M. R. U. Saputra is with Monash University Indonesia, Tangerang, Indonesia, e-mail: risqi.saputra@monash.edu.

Chris Xiaoxuan Lu is with University of Edinburgh, Edinburgh, Scotland, United Kingdom, e-mail: xiaoxuan.lu@ed.ac.uk.

L. N. S. Wijayasingha, M. Arif Rahman, and John A. Stankovic are with University of Virginia, e-mails: lnw8px@virginia.edu; mir6zw@virginia.edu; stankovic@cs.virginia.edu.

deployed LoRa APs, the coarse range estimates will be able to calibrate the odometry in the long run.

In this work, we put forward a novel ubiquitous positioning solution, which integrates deep odometry on edge, LoRa, and an augmented EKF, to pedestrian localization in visually-degraded environments. We compare state-of-the-art deep odometry systems in various aspects. We evaluate their performance on edge platforms under resource constraints, and propose a universal pipeline for deployment. Thereafter, we propose a LoRa-EKF backend for positional data backhaul as well as drifting error calibration. Our contributions are: (1) We evaluate embedded multi-modality deep odometry models for real-time positioning and propose best practices for deployment under different resource constraints; (2) We propose an EKF-LoRa backend for wireless backhaul and positioning calibration projected in the global frame; and (3) We validate the performance of the proposed solution on various edge devices in different environments.

II. Deep Odometry on Edge for Ubiquitous Positioning

Deep learning-based odometry systems have demonstrated great potential in ubiquitous positioning [6]. However, their high computational costs draw concern for real-time, real-world feasibility. In this work, we encapsulate state-of-the-art deep odometry models using emerging sensors that work well in adverse conditions, namely IMU only, mmWave radar + IMU, and a thermal camera + IMU. We implement these DNNs on edge computing platforms with different resource constraints. In doing so, we design a pipeline of deploying these models. We develop a generic EKF-LoRa backend for positional data backhaul and projecting the ego-centric positioning of odometry into the global coordinate system. We find an augmented EKF fusion module, taking the odometry and LoRa RSSIs only, are sufficient for positioning calibration.

A. Sensors and Deep Odometry

Ubiquitous positioning in adverse environments requires sensors that persist in swift body motions, darkness, glare, smoke, and extreme temperatures. We shortlist emerging multi-sensor-aided deep odometry candidates with proven capabilities in adverse environments, including IMU only (IONet [3]), mmWave radar with IMU (milliEgo [14]), and thermal camera with IMU (DeepTIO [16]). We design a handheld device to hold these sensors as shown in Fig. 1. It coaxially aligns a TI AWR1843 mmWave radar, a Flir Boson 640 thermal camera, and an Xsens MTi 1-series IMU (and an RGB-D camera for reference). The IMUs are pervasive in modern active devices to provide ego-motion sensing for they are compact, low-power, and environment-agnostic. The IONet [3] is a particularly light-weight learning-based DNN to reconstruct 6-DoF translation and rotations, $\tilde{\mathbf{y}} = \left[\tilde{\mathbf{t}}, \tilde{\mathbf{r}}\right], \tilde{\mathbf{y}} \in \mathbb{R}$, from raw IMU data.

The mmWave radar uses Frequency Modulated Continuous Wave (FMCW) in which a linear chirp signal allows distance estimation from reflectors. State-of-the-art milliEgo [14] exhibits decent capability in estimating ego-motion, whilst immune to visual degradation.

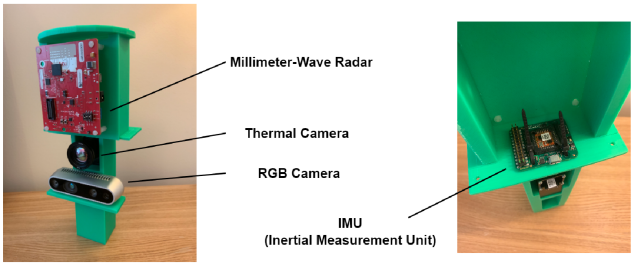

Fig. 1: A 3D printed handheld device (could be mounted on a helmet) to hold a mmWave radar, a thermal camera, an RGB-D camera, and an IMU.

Thermal infrared cameras capture the radiation emitted from objects in the Long-Wave Infrared (LWIR) portion of the spectrum. The strength of a thermal camera lies in its immunity to illumination conditions allowing perceiving the object's profile under poor visibility. The state-of-the-art DeepTIO [16] model incorporates motion, thermal, and RGB features to generate robust positioning estimation.

We take the above deep odometry models as front end candidates for our ubiquitous positioning system for their demonstrated capabilities in ubiquitous positioning. These deep odometry models, with proven robustness in different adverse conditions, have drastically different computational complexity. The next step is to deploy these deep odometry candidates on edge devices for real-time inference.

B. Edge Deployment

DNNs are computationally expensive. For instance, there are over 33 million parameters in milliEgo [14]. Its peak RAM usage exceeds $2GB$ running in Keras with a TensorFlow backend. Careful attention to the inference engine is vital to success. The core of this is to create a pipeline of sensor data fusion, synchronization, and the inference engine which should be systematically optimized with a stable throughput. Meanwhile, deep odometry models based on different sensors could be developed in any Machine Learning framework to be embedded in a variety of edge devices. Common deployment methods are yet limited to a one-off network minimization upon a specific piece of hardware, such as knowledge distillation [4]. Model compression methods, e.g., quantization, compromise accuracy which hinders their usability in positioning systems. To facilitate a universal and lossless deployment process, it is imperative to establish a pipeline that maximizes interoperability for the deep odometry models.

We explore an existing toolbox provided by Machine Learning frameworks to deploy the deep odometry models on the edge. We note the combination of Open Neural Network Exchange (ONNX) [13] and TensorRT [5] makes an ideal model compression pipeline. ONNX is a universal Deep Learning framework that represents DNN in a universal format initiated by Microsoft and Facebook. We integrate ONNX as

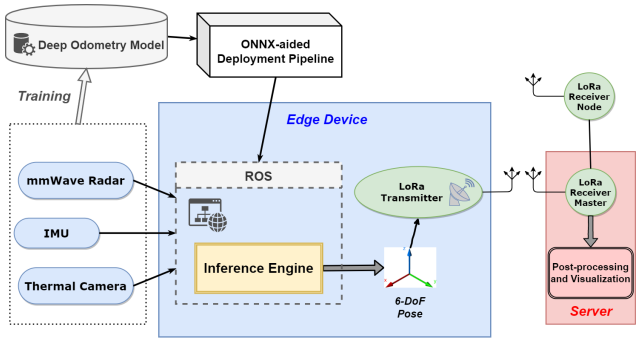

Fig. 2: A high-level architecture of the proposed positioning system. Sensor fusion and inference are performed on edge. A server conducts post-processing, data storage, and visualization.

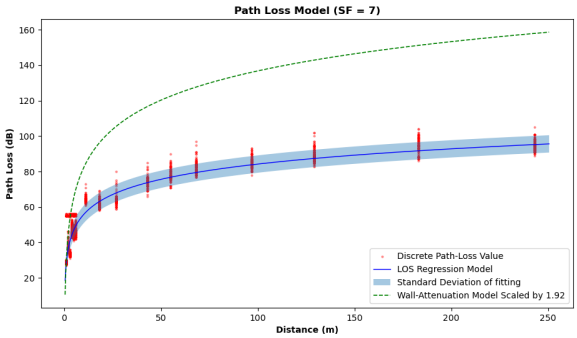

Fig. 3: LoRa path loss model at SF 7. Blue line shows the regression result based on measurement; Light blue area marks regression samples within the standard deviation; Dashed green line shows an empirically scaled path loss model for NLOS transmission.

an intermediate layer in our deployment pipeline to transfer deep odometry models originally developed in other frameworks such as Pytorch, TensorFlow etc. We use TensorRT, an NVIDIA proprietary frameword, as the inference engine vendor. TensorRT applies graph optimizations, layer fusion, and finding the fastest implementation of a given DNN model. This allows the deep odometry frontend to make full use of the computational power on the edge if a GPU or a Deep Learning Accelerator (DLA) is available.

Online sensor fusion and synchronization are accomplished via ROS, a *de-facto* robotic framework which allows simple system integration. A high-level diagram of the system integration via ROS is displayed in Fig. 2.

C. EKF-LoRa Backend

There is need to send pedestrian's positional data back to the server. We utilize LoRa for its long range in NLOS transmission and a highly configurable bandwidth [7]. The RSSI that comes free with each received LoRa packet provides a means to render deep odometry positions to the global frame. Thereby, it is of great importance to first derive a path loss model to determine range from RSSI. We measured LoRa RSSIs as a function of distance using SF from 7 to 11 in an open area. We recorded over 1.1k LoRa packets per SF in various LOS and NLOS scenarios. We find that SF 7 has a larger gradient, $\frac{\partial RSSI}{\partial d}$, implying a better range-indicative sensitivity compared to SF 11. We also measured their average air times in LOS condition when transmitting a 240-byte message as shown in Table I. Large SFs come with longer delays as well. As a result, we configure SF 7 for our LoRa link.

Since a ubiquitous positioning system does not make a-*priori* assumption of the environment, we adopt a generic path loss model with an attenuation factor, n. The path loss

TABLE I: LoRa Air Time Benchmark

SF	7	8	9	10	11
Air Speed (Kpbs)	62.5	19.2	9.6	4.8	1.2
Air Time (s)	0.447	0.551	0.674	0.86	1.708

model as a function of transmitter-receiver distance, d, can be expressed as;

$$PL(d) = PL(d_0) + 10n \cdot log\left(\frac{d}{d_0}\right) + C \tag{1}$$

where d_0 is the reference distance, $PL(d_0)$ is assumed the free space path loss at the reference distance in dB, and C is a constant to calibrate bias. We take $d_0 = 1m$ as reference and obtain a mean RSSI of $-8.483\,dBm$. Given a transmitting power of $22\,dBm$, it is determined $PL(d_0) = 30.483\,dB$. The attenuation factor and bias are derived through polynomial regression as shown in Fig. 3, in which the standard deviation equals $4.887dB$. To extract range from measurement, the RSSI can be directly expressed as a function of distance as below;

$$RSSI(d) = -28.5737 \times log_{10}(d) - 5.06 \tag{2}$$

It is fair to assume the RSSIs are log-normally distributed. The same applies to range. As a result, the measurement standard deviation may also be used as RSSI observation error in positioning fusion.

We argue that an EKF backend is sufficient in fusing the deep odometry poses and LoRa RSSIs. EKF is not only robust in handling non-linear motion modelling [8], but extremely lightweight, thus, suitable for real-time systems. A rigid transformation $\mathbf{T_{k,k-1}} \in \mathbb{R}^{4\times4}$ of two consecutive deep odometry poses from time $k-1$ to k can be expressed as:

$$\mathbf{T_{k,k-1}} = \begin{bmatrix} \mathbf{R_{k,k-1}} & \mathbf{t_{k,k-1}} \\ 0 & 1 \end{bmatrix} \tag{3}$$

where $\mathbf{R_{k,k-1}}$ is the rotation matrix at time k, and $\mathbf{t_{k,k-1}}$ is the translation matrix. Given a set of poses from the starting position $\mathbf{C_{0:k}} = \{\mathbf{C_0}, \mathbf{C_1}, \ldots, \mathbf{C_k}\}$, the current pose at time k can be written as;

$$\mathbf{C_k} = \mathbf{C_{k-1}}\mathbf{T_{k,k-1}} \tag{4}$$

Hence, the role of deep odometry is to calculate $\mathbf{T_k}$ according to the inertial and exteroceptive sensors in order to

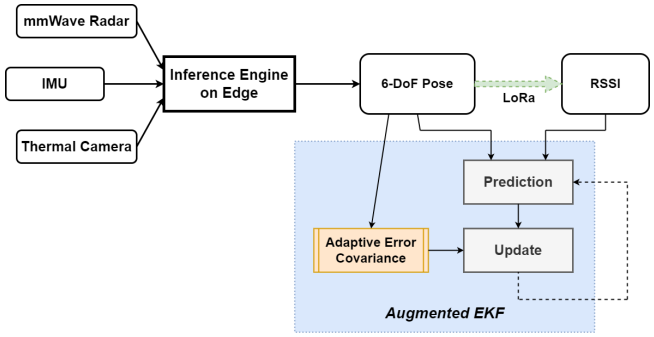

Fig. 4: An overview of online data flow and integration of the augmented EKF.

restore the path $\mathbf{C_{0:k}}$ [11]. To this end, the goal of the EKF-LoRa backend is to minimize the disparities, ε, of ranges from M anchors and the deep odometry pose by deriving a posterior $\hat{\mathbf{C}}$. For time k, this can be written as:

$$\varepsilon_k = \underset{\hat{\mathbf{C}}_\mathbf{k}}{\operatorname{argmin}} \sum_{m=1}^{M} |\|\mathbf{C_k} - \mathbf{p_m}\| - d_m| \tag{5}$$

where $\mathbf{p_m}$ is the pose of the m-th anchor, and d_m is the distance from the m-th anchor denoted in Eqn. 1.

In this work, we consider the positioning fusion in the 2D domain for simplicity. We specify the current pose as system state and RSSIs as observation with respect to the EKF. The system state is constantly updated by new observations. The schematic diagram is shown in Fig. 4. The system state, comprising 2D coordinates (x, y) and azimuth heading direction θ, can be represented by $\mathbf{x} = [x, y, \theta]^T$. The state transition and measurement equations can be expressed as:

$$\mathbf{x_k} = f(\mathbf{x_{k-1}}) + \omega_{\mathbf{k-1}} \tag{6}$$

$$\mathbf{z_k} = h(\mathbf{x_k}) + \nu_\mathbf{k} \tag{7}$$

where $\omega_{\mathbf{k-1}}$ is the process noise with covariance $\mathbf{Q}$, $\mathbf{z_k}$ is the measurement vector, and $\nu_\mathbf{k}$ is the measurement noise. Given the 6-DoF deep odometry pose represented in a 4-by-4 affine transformation matrix, $\mathbf{C_k}$, at time k, the initial pose can be defined as, $\mathbf{I}_{4\times4}$, with a starting position $(i_{1,4} = x_0, i_{2,4} = y_0)$. The aggregate pose follows Eqn. 4.

As a result, the prior state which can be extracted from $\mathbf{C_k}$ contains relative 2D position, current yaw angle, and the RSSIs from M anchors, denoted as ξ_x, ξ_y, ξ_θ, and γ_i, respectively:

We assign a large initial error covariance, $\mathbf{P_0} = \mathbf{diag}\,[10, 10, 1]$, to allow convergence from an unknown starting position. We specify the error covariance of observation as $\delta = 4.887$, derived from LoRa measurements, which holds for all observations.

We note that deep odometry errors are correlated with a high body turning speed [6]. Since σ_x and σ_y are associated with translation only, we assign small error covariances, $\sigma_x = \sigma_y = 0.15$, to minimize error from experiments. On the other hand, the error covariance of heading is correlated to turning velocity. We augment the EKF by associating the error covariance of heading, σ_θ^2, to current angular velocity at time k;

$$\omega_k = \frac{d\mathbf{R_{k,k-1}}}{d\triangle t} \tag{8}$$

where $\triangle t$ is the frame interval. Hence, we make the error covariance of the heading as a function of ω_k;

$$\sigma_\theta^2 = 10 \cdot \left(1 - e^{-0.2 \cdot \|\omega_k\|}\right) \tag{9}$$

In doing so, we realize stable accuracy gain in real-time tests with various turning speeds. Since RSSIs are noisy, we apply a weighted average filter to smooth the RSSIs so as to stabilize range estimates. Given a series of RSSIs observed at time k, γ, the filter can be written as:

$$\overline{\gamma_k} = \sum_{i=0}^{\lambda} \gamma_{\mathbf{k-i}} \cdot \frac{1}{\lambda} \tag{10}$$

where the window size $\lambda = 4$. An RSSI can be expressed as a function of range according to Eqn. 2;

$$\gamma_i = \alpha \cdot \log\left(\sqrt{(x - x_i)^2 + (y - y_i)^2 + (0 - z_i)^2}\right) + \beta \tag{11}$$

where $\alpha = -28.5737 \times C$, $\beta = -5.06$, and $\mathbf{p_i} = [x_i, y_i, z_i]$ are the i-th known anchor coordinates. Thereafter, the Jacobian matrix can be derived from $\mathbf{H} = \frac{\partial h}{\partial \mathbf{x}}\big|_{\hat{\mathbf{x}}_\mathbf{k}}$.

III. Evaluation

For a fair comparison, we collected 8 hours of synchronized thermal, mmWave radar, and IMU data to train the three candidate odometry models, respectively. A 10% split is used for validation, and another 20% is used for testing. We labelled the multi-sensor data based on a Vicon system and a Velodyne HDL-32E LiDAR which delivers sub-decimeter accuracy. We trained IONet, milliEgo, and DeepTIO, at 10 FPS for 200 epochs according to the authors' suggestions to get their optimal models. We setup six SX1262 LoRa modules with Raspberry Pi 4s as transceivers. Whenever receiving a packet, a LoRa receiver node predicts an RSSI, in dBm with $1dB$ resolution. We carried out a series of tests in uncharted environments to validate the proposed positioning system.

A. Accuracy and Robustness

To evaluate the performance, three experiment scenarios are chosen including a square-shape corridor, an office with no light on, and an outdoor car park in which the experimenter walked in and out of a ground-floor apartment at night. Four to five LoRa anchors were deployed surrounding the walking paths.

The Root-Mean-Square Errors (RMSE) of the aforementioned experiments are shown in Table II, Table III, and Table IV, respectively. Without EKF-LoRa backend, i.e., 'Anchor = 0', IONet incurs large drifting errors; MilliEgo and DeepTIO remain resilient in all three testing scenarios. It indicates that standalone milliEgo produce decent accuracy comparable to DeepTIO.

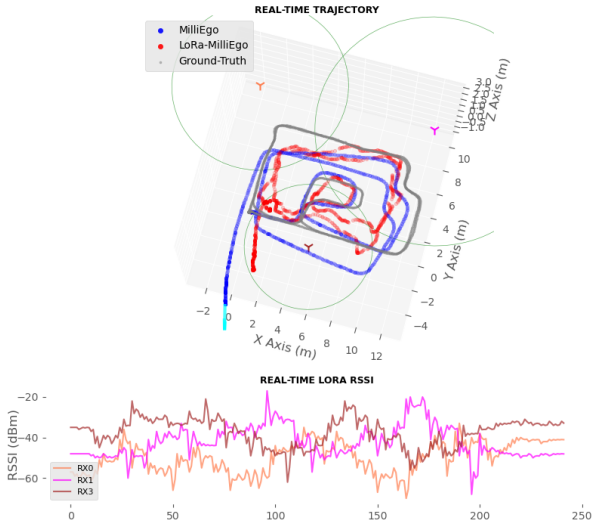

Fig. 5: Trajectory of a two-round search along corridors and entering and exiting a dark room. The ground-truth path is marked in grey; Blue path indicates using milliEgo only; Red path indicates fusion of milliEgo and LoRa RSSIs; Light green circles specify range estimates derived from the path loss model.

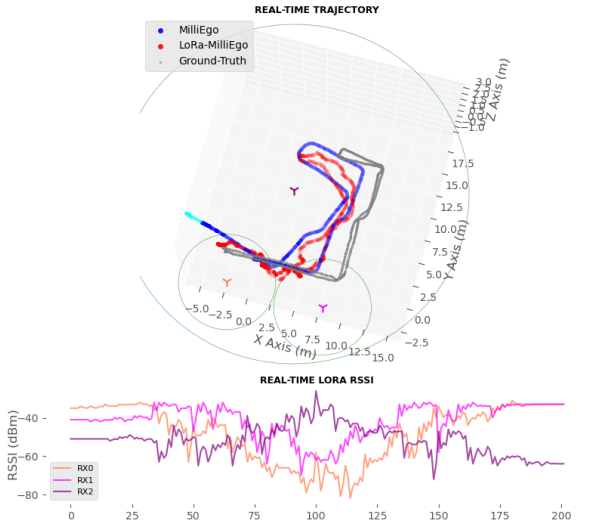

Fig. 6: Trajectory of a search in a ground-floor apartment and an outdoor car park; The ground-truth path is marked in grey; Blue path indicates using milliEgo only; Red path indicates fusion of milliEgo and LoRa RSSIs; Light green circles specify range estimates derived from the path loss model.

An EKF-LoRa calibrated trajectory in comparison to using milliEgo alone in the office test is shown in Fig. 5. The calibrated and original trajectories using DeepTIO in car park test are shown in Fig. 6. Notice the accuracy finds decent increment with even one anchor which verifies our hypothesis. As more RSSIs from different LoRa anchors are taken into positioning fusion, better accuracy is generally seen for all deep odometry candidates. Given three anchors IONet, milliEgo, and DeepTIO see an average of 69%, 34%, and 36% accuracy gains, respectively. This demonstrates the efficacy of our proposed positioning system in different environments based upon any deep odometry frontend. When there are more than three anchors, the benefit of having more LoRa anchors becomes more and more marginal.

B. Edge Platform and Latency

Common low-power edge platforms are single-board computers that are CPU-only and constrained in computational power. In contrast, latest higher-end edge devices have on-board GPU and deep learning acceleration with the capability of a desktop workstation. We take the selection of device into consideration and investigate runtime performance of three different edge devices: Raspberry Pi 4, NIVDIA Jetson TX2, and NVIDIA Jetson AGX Xavier. The latency benchmark results are shown in Table V. It can be seen the Jetson AGX Xavier delivers the lowest latency overall. It makes 10 FPS achievable for complex DNNs such as DeepTIO. The Jetson AGX Xavier is also the only platform currently supporting TensorRT framework. The Jetson TX2, with fewer GPU cores and no deep learning acceleration, poses moderate latency. Despite a resource-constrained device, Raspberry Pi 4 realizes possibilities to run IONet at 5 FPS which sheds light on low-cost ubiquitous positioning solutions.

C. Odometry Selection and Frame Rate

Lightweight deep odometry systems are more desirable for real-time applications on edge. Similarly, a low frame rate linearly reduces the computational burden. We investigate the effect of halving the original training and inference frame rate of the deep odometry candidates. The results are shown in Table VI. We repeat the corridor and car park tests using gapped sensor data stream recorded. It can be seen the accuracy of either deep odometry model deteriorates at 5 FPS. Nevertheless, milliEgo at 5 FPS pertains a decent tracking accuracy which lends itself a power-efficient choice under extreme resource constraints. Overall, we believe mmWave radar with IMU makes an optimal positioning solution in adverse environment in terms of both accuracy and efficiency. Note the DeepTIO results at 5 FPS are not presented since it does not converge well with a lower sample rate.

IV. Conclusion

This paper presents a novel edge-deployable ubiquitous positioning system based on deep odometry and an augmented EKF-LoRa backend. We reveal the feasibility of utilizing deep odometry to infer position in real-time under various resource constraints. We design a universal pipeline of deploying an arbitrary deep odometry model on the edge. To bring odometry positions to the global frame and mitigating drifting errors, we propose a generic EKF-LoRa fusion module. We find a consistent accuracy gain, over 34%, across all deep odometry tests conducted.

TABLE II: RMSE (m) among IONet, MilliEgo, and DeepTIO on three-round search along a square-shaped corridor

RMSE (m)	Anchor = 0	Anchor = 1		Anchor = 3				Anchor = 4		Anchor = 5
		Test i	Test ii	Test i	Test ii	Test iii	Test iv	Test i	Test ii	
IONet	17.603	12.87	13.975	6.741	5.679	9.559	9.04	5.731	6.024	**4.03**
MilliEgo	3.202	2.628	3.033	2.067	2.437	2.15	3.768	2.687	2.404	**1.959**
DeepTIO	3.517	3.461	2.977	2.024	2.858	1.95	2.146	1.713	2.052	**1.612**

TABLE III: RMSE (m) among IONet, MilliEgo, and DeepTIO on entering and exiting a dark office

RMSE (m)	Anchor = 0	Anchor = 1		Anchor = 3				Anchor = 4
		Test i	Test ii	Test Test i	Test ii	Test iii	Test iv	
IONet	15.237	12.674	12.835	8.979	9.956	8.33	10.031	**6.231**
MilliEgo	3.574	2.798	2.634	2.394	3.203	3.142	**2.312**	2.617
DeepTIO	3.122	3.223	2.841	**2.034**	2.676	3.055	2.971	2.629

TABLE IV: RMSE (m) among IONet, MilliEgo, and DeepTIO on wandering in a ground-floor apartment and a car park at night

RMSE (m)	Anchor = 0	Anchor = 1		Anchor = 3				Anchor = 4
		Test i	Test ii	Test Test i	Test ii	Test iii	Test iv	
IONet	20.609	15.882	13.09	13.214	16.348	14.653	14.484	**12.477**
MilliEgo	3.789	3.012	3.116	3.385	2.849	**2.826**	3.48	3.063
DeepTIO	3.682	3.598	3.746	2.411	2.586	2.71	2.542	**2.346**

TABLE V: Inference latency benchmark

Mean Latency (ms)		NIVDIA Jetson AGX Xavier	NVIDIA Jetson TX2	Raspberry Pi 4
IONet	Keras with TensorFlow backend	21.9	75.3	110
MilliEgo	Keras with TensorFlow backend	25.8	57.9	379
	ONNX-Runtime	10.4	141	N/A
	TensorRT serialized engine	3.5	N/A	N/A
DeepTIO	Keras with TensorFlow backend	71.9	237	10676
	TensorRT serialized engine	21.1	N/A	N/A

TABLE VI: Effect of Frame Rate to Deep Odometry

RMSE (m)		Corridor	Apartment and Cap Park
IONet	5 FPS	21.04	35.464
	10 FPS	17.603	20.609
milliEgo	5 FPS	6.277	4.483
	10 FPS	3.202	3.789

ACKNOWLEDGMENT

This research has been financially supported by the National Institute of Standards and Technology (NIST) via the grant Pervasive, Accurate, and Reliable Location-based Services for Emergency Responders (Federal Grant: 70NANB17H185).

REFERENCES

[1] A. A. Abdallah, K. Shamaei, and Z. M. Kassas. *Indoor Positioning Based on LTE Carrier Phase Measurements and an Inertial Measurement Unit*. ION GNSS Conference, 2018.

[2] M. Aernouts, N. BniLam, R. Berkvens, and M. Weynn. *TDAoA: A combination of TDoA and AoA localization with LoRaWAN*. Internet of Things Journal, Vol. 11, 2020.

[3] C. Chen, S. Rosa, Y. Miao, C. X. Lu, W. Wu, A. Markham, and N. Trigoni. *Selective Sensor Fusion for Neural Visual-Inertial Odometry*. Computer Vision and Pattern Recognition, 2019.

[4] J. Chen and X. Ran. *Deep Learning With Edge Computing: A Review*. Proceedings of The IEEE, Vol. 107, No. 8, 2019.

[5] Y. Choi, M. El-Khamy, and J. Lee. *Universal Deep Neural Network Compression*. 32nd Conference on Neural Information Processing Systems (NIPS), 2018.

[6] C. Debeunne and D. Vivet. *A Review of Visual-LiDAR Fusion based Simultaneous Localization and Mapping*. Sensors, 2020.

[7] S. S. Dhanjal, M. Ghaffari, and R. M. Eustice. *DeepLocNet: Deep Observation Classification and Ranging Bias Regression for Radio Positioning Systems*. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2019.

[8] F. Espinosa, C. Santos, M. Marron-Romera, D. Pizarro, F. Valdes, and J. Dongil. *Odometry and Laser Scanner Fusion Based on a Discrete Extended Kalman Filter for Robotic Platooning Guidance*. Sensors, 11, 2011.

[9] T. Guo. *Cloud-based or On-device: An Empirical Study of Mobile Deep Inference*. IEEE International Conference on Cloud Engineering, 2018.

[10] G. Gupta, N. Kejriwal, P. Pallav, E. Hassan, S. Kumar, and R. Hebbalaguppe. *Indoor Localisation and Navigation on Augmented Reality Devices*. IEEE International Symposium on Mixed and Augmented Reality Adjunct Proceedings, 2016.

[11] M. He, C. Zhu, Q. Huang, B. Ren, and J. Liu. *A review of monocular visual odometry*. The Visual Computer, Jun., 2019.

[12] M. Li, H. Zhu, S. You, and C. Tang. *UWB-Based Localization System Aided With Inertial Sensor for Underground Coal Mine Applications*. IEEE SENSORS JOURNAL, VOL. 20, NO. 12, June, 2020.

[13] W. Lin, D. Tsai, L. Tang, C. Hsieh, C. Chou, P. Chang, and L. Hsu. *ONNC: A Compilation Framework Connecting ONNX to Proprietary Deep Learning Accelerators*. IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS), 2019.

[14] C. X. Lu, M. R. U. Saputra, P. Zhao, Y. Almalioglu, P. P. B. de Gusmao, C. Chen, K. Sun, N. Trigoni, and A. Markham. *milliEgo: Single-chip mmWave Radar Aided Egomotion Estimation via Deep Sensor Fusion*. ACM Conference on Embedded Networked Sensor Systems (SenSys), 2020.

[15] H. Sallouha, A. Chiumento, and S. Pollin. *Localization in Long-range Ultra Narrow Band IoT Networks using RSSI*. IEEE International Conference on Communications (ICC), 2017.

[16] M. R. U. Saputra, P. P. B. de Gusmao, C. X. Lu, Y. Almalioglu, S. Rosa, C. Chen, J. Wahlstrom, W. Wang, A. Markham, and N. Trigoni. *DeepTIO: A Deep Thermal-Inertial Odometry with Visual Hallucination*. IEEE Robotics and Automation Letters, Vol. 5, Issue 2, 2020.

[17] W. Xu, J. Y. Kim, W. Huang, S. Kanhere, S. Jha, and W. Hu. *Measurement, Characterization and Modeling of LoRa Technology in Multi-floor Buildings*. IEEE Internet of Things Journal, Vol. 7, Issue: 1. 2020.

[18] P. Zhao, C. X. Lu, J. Wang, C. Chen, W. Wang, N. Trigoni, and A. Markham. *mID: Tracking and Identifying People with Millimeter Wave Radar*. 15th International Conference on Distributed Computing in Sensor Systems (DCOSS), 2019.