Path Planning for Adaptive CSI Map Construction with A3C in Dynamic Environments

Xiaoqiang Zhu, Tie Qiu, *Senior Member, IEEE*, Wenyu Qu, Xiaobo Zhou, *Senior Member, IEEE*, Yifan Wang, and Dapeng Oliver Wu, *Fellow, IEEE.*

***Abstract*—The fingerprint localization based Channel State Information (CSI) plays a vital role given the popularity of Location-Based Service. Since its easy implementation, low device cost and CSI provides fine-grained information which can achieve adequate accuracy. However, the main drawback is that the approach has to construct the fingerprint map manually during the off-line stage, which is tedious and time-consuming. In this paper, we propose a novel data collection strategy for path planning based on reinforcement learning, namely Asynchronous Advantage Actor-Critic (A3C). Given the limited exploration step length, it needs to maximize the informative CSI data for reducing manual cost. We collect a small amount of real data in advance to predict the rewards of all sampling points by multivariate Gaussian process and mutual information. Then the optimization problem is transformed to a sequential decision process, which can exploit the informative path by A3C. We complete the proposed algorithm in two real-world dynamic environments and extensive experiments verify its performance. Compared to coverage path planning and several existing algorithms, our system not only can achieve similar indoor localization accuracy, but also reduce the CSI collection task.**

***Index Terms*—Path Planning, CSI, Reinforcement Learning, Fingerprint Localization.**

I. Introduction

As the demand for Location-Based Service grows in complex indoor environments [1], [2] and the development of mobile sensing technology, taking into account the availability of a large amount of spatial data, such as temperature, humidity, WiFi, bluetooth, localization technology has been gradually transformed from geometric method to fingerprint method [3]–[5]. The fingerprint localization method obtains the estimated location through matching features and Received Signal Strength (RSS), bluetooth, visible light, geomagnetic field, etc. can as the fingerprints [6]. Among them, CSI is physical layer information of WiFi which includes plentiful parameters and describes the transmission of wireless signals in detail. And CSI-based localization can achieve more accurate precision than the RSS-based method [7]. Moreover, it has low computational complexity and CSI data can be obtained by commodity off-the-shelf devices that are suitable for mobile terminals. Over the last decade, CSI-based fingerprint localization has attracted extensive attention and there are lots of relevant researches [8], [9]. And it has gradually become the mainstream fingerprint signal [10].

The fingerprint-based localization technology needs to build the fingerprint database during the off-line phase [11]. That is, professionals divide the experimental area into grids for sampling, measure and collect fingerprint data at each sampling location, and then store them in the location-fingerprint database. This process is called the site survey or fingerprint calibration. However, it is a time-consuming and laborious task if done manually. And the task quantity is positively correlated with the experimental environment size that is the main reason why the fingerprint method has not been widely utilized in practice [12]. Therefore, it is necessary to consider how to reduce the data collection task, a preferable way is to use robotic technologies. A robot can automatically collect the spatial data in environments with mobile sensing devices which also needs to avoid obstacles, and the robot is called an agent in this paper. Given the limited power of an agent, it is important to plan an effective path for the agent to collect data which brings infinite possibilities for adaptive fingerprint map construction in dynamic environments. The problem is also called path planning.

The path planning problem is formulated on graphs and transformed to the classic traveling salesman problem (TSP) generally [13], and uses heuristics strategies to find the optimal solution [14], [15]. Wei et al. [16] regarded fingerprint collection as the well-known orienteering problem and proposed a greedy algorithm and a genetic algorithm to solve the problem of edge-based non-additive rewards and revisits, which is NP-hard. They verified the proposed algorithms that have low computation complexity and localization errors in two indoor environments. However, heuristics strategies tend to fall into local optimal solutions and time-consuming. Piao et al. [17] used an Unmanned Aerial Vehicle (UAV) with a CSI measurement module for automating map construction, and applied a similar method as [16]. Especially, the UAV can be programmed to fly through obstacles (e.g., tables) at human height and obtain more CSI fingerprints. The proposed system can improve energy efficiency and achieve similar accurate localization over the manual collection. Furthermore, UAVs can not always occupy anywhere in complex indoor environ-

This work is supported in part by the National Key R&D Program of China (No. 2019YFB2102400 and No. 2019YFB1703601), the Joint Funds of the National Natural Science Foundation of China (No. U2001204), National Natural Science Foundation of China (No. 62072330), Tianjin Science Foundation for Distinguished Young Scholars (No. 20JCJQJC00250), Key R&D Program of Tianjin (No. 20YFZCGX01150) and the China Scholarship Council (No. 202006250078).

X. Zhu, T. Qiu, W. Qu and X. Zhou are with the School of Computer Science and Technology, College of Intelligence and Computing, Tianjin University, Tianjin 300350, China; and also with the Tianjin Key Laboratory of Advanced Networking, Tianjin 300350, China (e-mail: abc2611617@tju.edu.cn, qiutie@ieee.org, wenyu.qu@tju.edu.cn, xiaobo.zhou@tju.edu.cn).

Y. Wang and D. O. Wu are with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611-6130 USA (e-mail: fanwon018@ufl.edu, wu@ece.ufl.edu).

ments and it is more difficult to collect CSI data which needs to receive packets continuously. Faust et al. [18] completed the PRM-RL system to control an agent for navigation tasks by using the Deep Deterministic Policy Gradient (DDPG). It builds a roadmap to determine connectivity and makes dynamic decisions. However, the agent can be hard to train if rewards are sparse, and the inaccuracy of initial Q value causes low training efficiency and slow convergence speed.

In this paper, an adaptive CSI map construction algorithm for path planning is proposed to address the above issues based on the A3C algorithm. Specially, we still collect CSI data manually due to the limitation of devices, and the experimental area is formulated on a graph and transformed into a sequential decision process. Given the maximum exploration step length, and available action is decided by the A3C algorithm for obtaining the optimal strategy which is related to the current state (agent position). And we also design a novel exploration mechanism that includes a "step back" for saving computing resources. The total rewards are improved gradually and then we can get the most informative path. Our major contributions are summarized as follows:

- We utilize the A3C algorithm to guide the agent to move for automatically building the CSI fingerprint map in dynamic environments. The path planning problem is transformed into a sequential decision process that is different from the traditional TSP.
- We use very few pilot data instead of dense sampling to predict the global distribution of CSI data through a multivariate Gaussian model during the off-line stage, and design an exploration strategy to find the optimal path by mutual information which can greatly save computing resources and improve efficiency.
- We verify the performance of the proposed algorithm in two real-world environments, and it can generate the optimal strategy of actions to build an informative path. Importantly, our algorithm can reduce 72% collection tasks compared to coverage path planning and three existing algorithms, and achieve similar accuracy.

The remainder of this paper is organized as follows. Section II is the related work and Section III gives the preliminaries and shows the feasibility of the CSI map. The proposed algorithm is described in Section IV. Experiments and results analysis from two indoor environments are presented in Section V. Section VI gives the conclusions and future work.

II. Related Work

In this section, we review the literature and research of indoor localization techniques and path planning, including fingerprint-based approaches, geometry-based approaches and graph-based approaches.

A. Indoor Localization Techniques

With the rapid development of mobile devices, indoor localization technology is becoming intelligent. They provide various kinds of signals which can be utilized and combined with related methods for localization. And a large number of researches have emerged in recent years [19]. In this paper, we divide the indoor positioning techniques into the following two categories:

Fingerprint-based approaches: As mentioned earlier, it includes the off-line training stage and the on-line location stage. Thanks to the pervasive deployment of intelligent infrastructure and wearable sensor devices, several signals can be considered to build the fingerprint database for the position, such as RSS, CSI, RFID, bluetooth, and more. Guo et al. [20] utilize multiple fingerprints collected from RSS to localize, which is an advantageous strategy to overcome RSS sensitivity. The derivative fingerprint of RSS is fused with multiple classifiers, and it can achieve better robustness in the WiFi location. Gao et al. [21] consider that it only involves data frames of CSI which limits the actual deployment. They propose the CRISLoc which operates in a completely passive mode, overhearing packets during its own CSI acquisitions by smartphones. And combined with K Nearest Neighbors (KNN) and other methods, the accuracy reaches 0.29 m in the real environment test. Ma et al. [22] propose a multi-tag localization system using a weighted multidimensional scaling method based on passive ultrahigh-frequency RFID tags. The system further combines with the RSS method to estimate distance by the cooperation between tags. Faulkner et al. [23] design a passive position system based on visible light communication, which does not carry any active devices or tags. And they investigate the impact of localization performance with different distance metrics using weighted KNN.

Geometry-based approaches: It mainly includes Time of Arrival (ToA), Time Difference of Arrival (TDoA), and Angle of Arrival (AoA). ToA and TDoA achieve positioning based on the wireless signal transmission time between transmitters and receivers. Both of them have to know the position of at least three transmitters in the location stage, which requires high time synchronization [4], [19]. Zhang et al. [24] utilize ToA to design a localization scheme for tracking mobile devices and updating the inaccurate floor plan map. They combine a greedy partitioning scheme and a particle-Gaussian mixture filter for carrying information on mobile devices and map features, which can greatly improve the average localization and mapping accuracy compared with the existing filters. Du et al. [25] complete a visible positioning system combined with TDoA and cross-correlation. The improved TDoA algorithm uses a virtual local oscillator for cross-correlation which reduces the hardware complexity, and cubic spline interpolation is applied for improving the temporal resolution of cross-correlation. And the positioning system achieves an average accuracy of 9.2 cm. AoA involves measuring the incidence angle of a wireless signal, and the receiver must require a directional antenna array. AoA can also estimate the position by establishing an appropriate propagation model. Zheng et al. [26] consider that the array orientation has not been applied well in the existing researches, then they investigate the impact on the localization performance. They propose the OpArray localization system, it includes an array deployment scheme which can reduce the uncertainty in AoA estimation for improving accuracy. Besides, the system can be also applied to a commercial WiFi platform and achieve sub-meter accuracy.

To summarize, it is difficult for geometry-based approaches

to establish an accurate model between distance and signal strength in complex environments, and geometric positioning requires high cost or extra devices for localization. The fingerprint-based approach has gradually become the main trend of indoor location technology [4], [9], [10], [19]. The method is intelligent and easy to estimate locations combined with machine learning or probabilistic methods. And it will become more efficient if we can find a better way to reduce the cost of building the fingerprint database.

B. Path Planning

Path planning has important applications in positioning, navigation, visualization, and interior structure modeling for robotics [27]. And its aims to extend the working life cycle of robots and avoid obstacles [16], [17], [28]. The existing works take different approaches to intelligentized the fingerprint collection process. In general, the target region is divided into grids and transformed into graphs, and heuristic algorithms or machine learning are applied to find the optimal path.

Dai et al. [29] design the AuF which autonomously collects fingerprint data by a robot and saves time and energy. The system starts with an automatic initialization process and carries out the on-site investigation without retention. Then AuF recovers the abnormal data through the previously built fingerprint database based on a signal transmission model, which can save 61% collection works. Kolakowski [30] designs a low-effort method for collecting bluetooth and ultra-wideband fingerprint using crowdsourcing and interpolation approach. Especially, the parameters of a path loss model and a Gaussian process regressor are calculated by the recorded data and then interpolating the fingerprint map. And experiments verify its localization performance than the RSS-based positioning method. Jung et al. [31] use an unsupervised learning method to collect WiFi fingerprint via crowdsourcing, and propose a probabilistic localization algorithm, [32] is a similar work. Wei et al. [33] apply Q-learning to plan an effective path for the robot which makes the collected RSS fingerprint informative. A constrained exploration and exploitation strategy is proposed to automate sensing and navigation using a robot. They also utilize a Gaussian process regressor to predict the rewards for Q-learning as well as [30] during the off-line stage. And the method can be transfer learning when the input parameters have appropriate changes.

To summarize, graph-based approaches let each edge with a weight and the robot moving between two different vertices. The goal is to find an effective path with the most informative unity with some limitations (for example, before the robot runs out of power). The traditional methods prefer to autonomously collect the RSS data using a UAV; however, RSS is the coarse-grained information that fundamentally limits the accuracy of the fingerprint database, and the UAV can not always reach the sampling points in a complex indoor environment. At the same time, it is also considered whether the machine learning method falls into the local optimal solution.

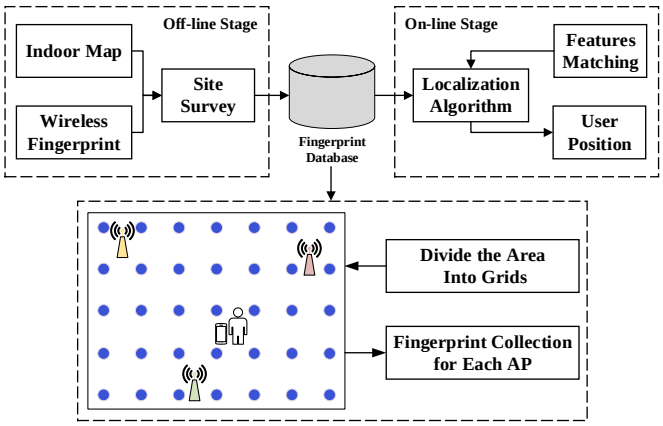


Fig. 1. Flowchart of fingerprint location method

III. Preliminaries

A. CSI-based Fingerprint Localization

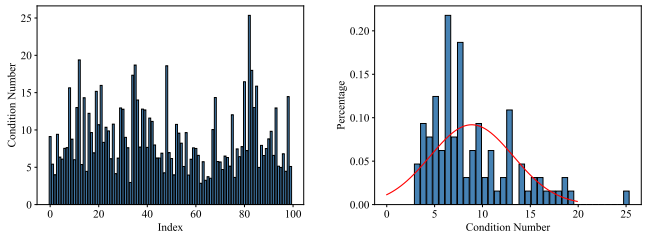
Thanks to the pervasive deployment of WiFi-enabled devices that we can get WiFi data easily in indoor environments, such as RSS and CSI. The fingerprint localization method with demonstrated ideal performance which has gradually replaced the geometric method [34], [35], includes two stages as shown in Fig. 1. The one is the off-line stage, it needs to build the fingerprint map that is relevant to the environment; another is to estimate the user's position through localization algorithms, such as machine learning or probabilistic method which is called the on-line stage. In this paper, we focus on the first stage of how to reduce the collection task manually, and multiple matching algorithms can be applied in the on-line stage which is not described in detail [4].

CSI can be collected for each packet by Linux 802.11n CSI tool with modified drivers as a fine-grained information [7], which includes 30 subcarriers and abundant channel characteristics. It can be represented by

$$H_i = |H_i| e^{j \sin(\angle H_i)}, \tag{1}$$

where $|H_i|$ is the amplitude and $\angle H_i$ is the phase. Theoretically, there is a linear relationship model between the fingerprint measurements at one location and the corresponding measurements at other locations according to the path loss model [36]. However, signals are affected by reflection, scattering, attenuation, etc in real environments, multicollinearity may exist between measurements at different locations. Therefore, it is necessary to validate the impact on CSI data for confirming the effectiveness of path planning.

Here, we use the condition number to evaluate the multicollinearity. It is a general metric and greater than 10 indicates multicollinearity, while greater than 30 indicates severe multicollinearity [37]. Especially, Fig. 2 shows the collinearity between CSI data at different locations, and all data are from the real-world environment. It is obvious that there is less multicollinearity which means we can utilize the classic multivariable linear regression models to predict the distribution of other CSI data. And CSI is more stable and accurate than RSS, there is reason to believe that better



(a) Condition number for each index (b) The distribution

Fig. 2. Multicollinearity diagnostics of CSI

localization performance can be achieved using path planning based CSI.

B. A3C Algorithm

Reinforcement learning is widely applied for fingerprint localization, Q-learning, Sarsa, Deep Q Network, etc [4], [38], [39]. The framework of reinforcement learning includes agent, environment, actor, state, and reward. The agent interacts with the environment to generate trajectories, and state transition occurs by performing actions; then the environment decides the action for the current agent by reward (positive or negative). More and more experience is accumulated through iteration to update the policy for making decisions. In this paper, we utilize the A3C algorithm for path planning which is effective in high-dimensional or continuous action spaces. Especially, A3C can learn stochastic policies with better convergence properties. Please note that A3C is not the contribution of this paper, we just introduce its basic background in this section and readers can refer to [40] for more details.

A3C asynchronously runs multiple agents that can remove the correlation between samples during the training process by experiencing different states and transitions. In addition, only a standard multi-core CPU is needed to implement the algorithm, which is superior to traditional methods in effect, time, and resource consumption. A3C has an Actor-Critic architecture, that is, optimizing a policy $\pi(a_t|s_t;\theta)$ with Actor to make it better and estimating the value function $V(s_t;\theta_v)$ with Critic to make it more accurate. The policy update can be performed by

$$\nabla_{\theta'} \log \pi(a_t|s_t;\theta')A(s_t,a_t;\theta,\theta_v) \quad (2)$$

and $A(s_t,a_t;\theta,\theta_v)$ is the advantage function, which can be represented by

$$A(s_t,a_t;\theta,\theta_v) = \sum_{i=0}^{k-1} \gamma^i r_{t+i} + \gamma^k V(s_{t+k};\theta_v) - V(s_t;\theta_v) \quad (3)$$

where θ is the parameter of the policy π and θ_v is the parameter of the value function. In next section, we will describe how to utilize A3C for obtaining optimal path.

IV. Proposed Solution

A. Overview

Fig. 3 shows the architecture of our proposed algorithm. It mainly includes three components, they are reward computation based multivariate Gaussian regression, strategy exploration based A3C and indoor localization. It works as follows:

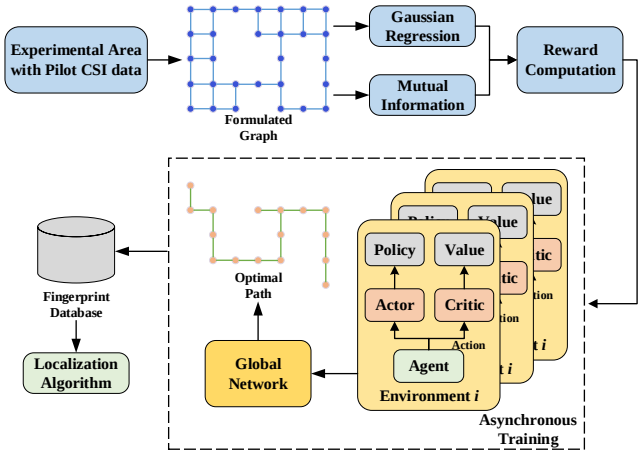


Fig. 3. Flowchart of path planning

First, we manually collect a small set of pilot CSI data at some grids in the experimental area, a desktop computer, and a laptop as the transmitter and receiver, respectively. A graph is constructed for calculating the reward of each edge, the pilot data fit a multivariate Gaussian regression model and estimates other data distribution. Then we utilize mutual information to obtain the reward for an agent.

Next, the path planning problem is transformed to the TSP by the formulated graph, A3C is the core for searching the optimal path. Especially, we use a convolutional neural network for sharing some parameters and multithreading process technology for improving training stability. And we propose a novel reward mechanism to get better performance.

Finally, we only collect CSI data on the optimal path and revise other CSI distribution for residual grids. Furthermore, the fingerprint database can be constructed through these pilot CSI data. It can be utilized for indoor localization, and many algorithms can match features in this step. As well as we all know, localization accuracy is the main measurement index for a positioning system, and this paper also pays attention to the accuracy. As the fine-grained physical layer information, CSI has richer data features than RSS. Similarly, many existing location algorithms based on CSI have reached cm-level accuracy [21], [41], [42]. Algorithm [16] also uses mutual information to construct the reward value of fingerprint data and utilizes professional positioning technology to achieve an accuracy of 2.7 m for RSS signal. We have reason to believe that utilizing CSI for path planning could achieve cm-level precision. Considering the length of the paper, and we do not focus on localization algorithms in this paper, therefore, we just choose the KNN algorithm for training the localization model. Then we describe each component in detail.

B. Reward Computation

We formulate the experimental environment into a graph that includes vertices and edges according to grids. The positions of samples are vertices and the line that can reach between two adjacent samples is an edge. Here, the rewards for each edge are calculated as similar to [16]. Theoretically, the distribution of fingerprint signals for a target area is normal;

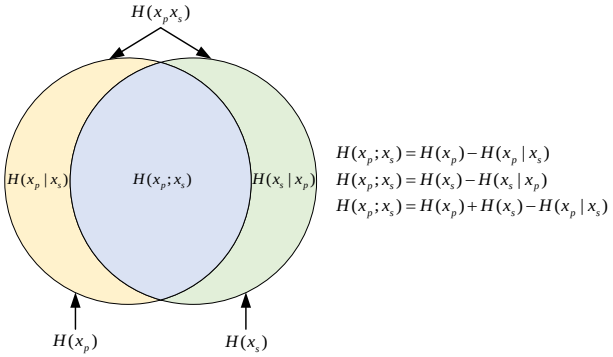


Fig. 4. Mutual Information with Pilot CSI Data

however, there are multipath effects, signal attenuation, or environmental changes which makes the model difficult. As mentioned previously, CSI is stable with less multicollinearity that we utilize a multivariate Gaussian process to model the collection data. For $\{f(x) : x \in \chi\}$, it can be formulated by

$$f(\cdot) \sim GP(m(\cdot), k(\cdot,\cdot)) \tag{4}$$

where $m(\cdot)$ is mean function and $k(\cdot,\cdot)$ is covariance function. Therefore, we have that $m(x) = E[x]$ and $k(x,x') = E[(x - m(x))(x' - m(x'))]$ for $\forall x, x' \in \chi$. For path planning problem, the positions are denoted as χ means that the distribution of CSI depends on the location. Especially, we choose the radial basis function as the kernel which is given by

$$k(x,x') = \exp\left(-\frac{1}{2\tau^2}\left\|x - x'\right\|^2\right), \tag{5}$$

where $\tau > 0$, and readers can refer to [43] for more details.

Then the mutual information is utilized to calculate the rewards for all edges with pilot CSI data as x_p. And we denote all the sample data as x_s, mutual information means the condition distribution of x_p given x_s is $p(x_p|x_s)$, the greater the average difference between $p(x_p|x_s)$ and x_p, the greater the information gain. We transform the condition distribution into the differential entropy H, and the relation of mutual information is shown in Fig. 4. A small set of pilot data can estimate other data distribution, and we just choose the first equation to calculate $H(x_p;x_s)$.

The differential entropy for a Gaussian distribution is given by

$$H(x) = -\int f(x)\log\phi(x)dx = \frac{n}{2}\log(2\pi e\sigma^2) \tag{6}$$

where σ depends on the covariance function $k(\cdot,\cdot)$ and other parameters are constants. We can obtain directly $H(x_p)$ through above equation, if the CSI measurements are denoted as y_s for all positions of samples, and the covariance matrix Σ can be represented by

$$\Sigma = k(x_p,x_p) - k(x_p,x_s)(k(x_s,x_s) + \hat{\sigma}_n^2 I)^{-1}k(x_s,x_p) \tag{7}$$

where $\hat{\sigma}_n$ is the noise variance. Then we can calculate $H(x_p|x_s)$ according to Eq. (6) which is given by

$$H(x_p|x_s) = \frac{n}{2}(\ln 2\pi + 1) + n\ln\hat{\sigma}. \tag{8}$$

Algorithm 1: Reward Computation based MI and GP

Input: Pilot CSI data, labels (physical positions);
Output: Reward R;
1 Formulate the experimental area into a graph;
2 Preprocess for pilot CSI data by multidimensional scaling algorithm;
3 Multivariate Gaussian regression model and optimization;
4 Calculate the noise variance $\hat{\sigma}$;
5 Calculate $H(x_p)$ according to Eq. (5) and (6);
6 Calculate $H(x_p|x_s)$ according to Eq. (7) and (8);
7 $R = H(x_p) - H(x_p|x_s)$;
8 Return reward R.

Finally, the reward R based mutual information is calculated by

$$R = H(x_p;x_s) = H(x_p) - H(x_p|x_s). \tag{9}$$

The pseudocode for reward computation is presented in Algorithm 1. Especially, pilot CSI data with multiple packets which need dimension reduction and we utilize multidimensional scaling algorithm to preprocess; the data are necessary for fitting hyperparameters and they make up only 10% of all sampling points. And differential entropy is related to covariance function $k(\cdot,\cdot)$, therefore, we can predict the CSI distribution and rewards in the off-line stage.

C. Exploration Strategy based A3C

In this section, we firstly give the definitions of concepts as follows and then describe the exploration strategy.

- Agent: It is a person who starts to research the optimal path at the start vertex S, and he will be reset at S if an iteration is completed. Given the maximum exploration step length, the agent tries to arrive at the goal vertex G within the limitation.
- Environment: It is a graph built from the layout of an environment. And we use two classic indoor environments, they are non-line-of-sight (NLOS) and line-of-sight (LOS), respectively.
- Actor: The agent chooses different actions varying from vertex to vertex and moves along the edges, the policy consists of a sequence of the chosen actions.
- State: It is the current position of the agent which is related to actions because there are obstacles in our experimental environment. Especially, A3C has a global network for sharing parameters which leads to obtaining a better state.
- Reward: The agent will obtain a value that may be positive or negative if it chooses an action, and the total reward increases as the agent moves toward G.

The major problems in utilizing reinforcement learning to search the optimal path are obstacles and efficiency. On the one hand, although the layout of environments is formulated into a graph, the two adjacent positions may not be directly reached due to the occlusion of obstacles; therefore, the agent needs to have the obstacle avoidance function. On the other

hand, the computation complexity of reinforcement learning is high and it violates the original intention of our motivation if the running time is too long; we design a novel strategy to make our system efficient by giving the maximum exploration step length. Please note that the optimal path is the most informative and has high total rewards, it does not mean the shortest path.

The path planning algorithm is offline and can be run on the local computer or server without consuming the agent's computational power to find the optimal path. In practical application, only a small amount of pilot data and sampling points on the optimal path are captured by the agent (it could be a UAV or a robotic vehicle). The agent simply acts as the receiver and does not need to run any algorithms to save a lot of battery power. The A3C algorithm used in this paper, needs a huge amount of computing power as a reinforcement learning method. At present, the existing A3C computing platform based on GPU can greatly save computing time [44].

1) Exploration Strategy: In the formulated graph, we set the start vertex S and the goal vertex G of the agent as diagonal positions. Theoretically, we can predict the distribution of fingerprint data for other sampling points by simply collecting fingerprint data along the diagonal path according to the Gaussian distribution. However, the action space is "up", "down", "left" and "right", and there are unpredictable interference factors in a complex indoor environment which make the diagonal path invalid. Therefore, we set the maximum exploration step length (denoted as max_step) to be far less than the number of all sample points in order to improve efficiency as much as possible. And the termination condition of one episode is that the agent arrives at the goal vertex G, that is, it finds a potentially available path; otherwise, the agent is reset to the start vertex S to explore again.

We denote a potential available path as $P = [v_S, ..., v_G]$ and its corresponding total reward is represented by $r(P)$, and try to find the optimal path which satisfies

$$P_{optimal} = \arg\max_{P \in \Psi} r(P) \tag{10}$$

where Ψ is the set of all valid paths from v_S to v_G. Let v_i represent the current position of the agent, and the set of available actions A is given by

$$A(v_i) = \{v_{i+1} \in V : (v_i, v_{i+1}) \in E\} \tag{11}$$

where V and E are the set of all the vertices and edges in the formulated graph, and it is obvious that A depends on the adjacent position v_{i+1}. For classic action selection methods, the agent randomly chooses an action from A at each episode even if the next state is at an obstacle, and then it is reset and runs a new loop.

We consider this method wastes an episode with high complexity and proposes a novel mechanism that includes "step back" and greedy strategy. That is, the agent returns to the previous state if it arrives at an obstacle after performing an action. Then we remove the action from A and give priority to selecting an action with probability α that will enable the agent to reach a position that has not existed, and assign a larger reward value, otherwise assign a smaller reward value. It can ensure the agent to explore more positions towards G and avoid obstacles in each episode. Although the novel method can make the agent approach G gradually and total rewards increase, the actions are randomly generated which may not make the path valid.

Algorithm 2: Exploration and Reward

Input: v_S, v_G, max_step, $max\_iteration$, env;
Output: $< s', a, r, done >$;

```
j = 0;
buffer_s, buffer_a, buffer_r, = [], [], [];
while j < max_iteration do
    s = env.reset();
    ep_r = 0;
    for n = 1 : max_step do
        a = choose_action(s);
        s', r, done = env.step(a);
        Check whether the current state s' exists;
        if n == max_step − 1 then
            done = True;
        else
            done = False;
        end
        ep_r += r;
        buffer_s.append(s);
        buffer_a.append(a);
        buffer_r.append(r);
        s = s';
        j += 1;
    end
end
Return < s', a, r, done > and ep_r
```

Here, we try to choose the best action according to the greedy strategy which is similar to Q-learning. Firstly, we determine whether the current state (observation) already exists, if not, we add it to a list; next, recording the Q-value of the action when the agent reaches the current state for each step; finally, the best action is randomly chosen with the maximum Q-value. The reward of each step is defined as

$$r(v_{i+1}) = r(v_i + A(v_i)) - r(v_i) \tag{12}$$

and the total reward also adds up as the last step is updated.

The pseudocode of exploration and reward in one step are shown in Algorithm 2. In each episode, the agent moves from state s to s' towards G by taking action a and obtains a reward r, and $done$ means whether update global and assign to local net which is the core of A3C algorithm. It returns the tuple $< s', a, r, done >$ which is stored in the buffer and ep_r.

2) Find the Optimal Path: To avoid local optimal solutions and accelerated convergence, A3C uses multithreading to find the optimal policy. And another optimization is to add the entropy for policy π with a coefficient β to the actor-critic strategy loss function, according to Eq. (2) it can be represented by

$$\theta = \theta + \alpha \nabla_\theta \log \pi_\theta(s_t, a_t) A + \beta \nabla_\theta H(\pi(s_t, \theta)). \tag{13}$$

Algorithm 3: The Optimal Path

Input: $< s', a, r, done >$, ep_r, max_step, $max\_iteration$;
Output: Optimal path;

```
Initialize neural network parameters;
Initialize gradients dθ ← 0;
Initialize the buffer;
for episode i ← 1 to max_iteration do
    Initialize s', r from the buffer;
    for step t ← 1 to max_step do
        if i mod I_global == 0 or done == True then
            r' = { r              if terminal s'
                 { r + γ × r'     else others s' ;
            Accumulate gradients according to Eq. (14);
            Update the global network;
        end
        Calculate total rewards by
          R_global = ω × R_global + (1 − ω) × ep_r;
    end
end
Calculate rewards of potential valid paths;
Return the Optimal path according to Eq. (10)
```

We take the derivative of the entropy and reduce it, then the probability of the output of each action is as unequal as possible.

There is a global network model of A3C as shown in Fig. 3, which is a neural network and includes the functions of the actor-network and critic network. In detail, it has n worker threads, each of which has the same network structure as the global network. Each thread independently interacts with the environment to obtain experiential data and they do not interfere with each other. A certain amount of experiential data will be obtained after each thread interacts with the environment, then we calculate the gradient of the neural network loss function in its own thread. The gradient of the local critic is updated by

$$d\theta \leftarrow d\theta + \frac{\partial (r - Q(s,a;\theta'))^2}{\partial \theta'}. \tag{14}$$

Especially, these gradients are used to update the global network instead of their in-thread ones. In other words, n threads will independently update the model parameters of the global network with the accumulated gradient. And the thread updates the parameters of its own neural network to the global network for guiding subsequent environment interactions at regular intervals.

Combined with the proposed exploration strategy, the agent may find several potential valid paths during the learning process. Although A3C uses the global neural network as the function approximator and the accumulated reward is gradually increased, it does not guarantee that the optimal solution will be found when the final learning is completed. Here, we utilize a greedy policy to confirm the optimal path that the reward of each potential path is calculated and let the maximum reward of a path as the final solution according to Eq. (10). And the corresponding pseudocode is shown in Algorithm 3.

D. Predict the Distribution of Fingerprints

The fingerprint data is collected according to the optimal path, and the dependency relationship between the fingerprint of adjacent locations is modeled by combining the pilot fingerprint data. Then we can build the whole fingerprint map based on the relationship model. As mentioned previously, we think the distribution of fingerprints depends on physical locations, and the real-time fingerprint along the optimal path is taken as the input for predicting the fingerprint of the rest locations. Especially, the invariance of the relationship model is the basic assumption of updating the whole fingerprint map.

The covariance function $k(\cdot,\cdot)$ is the core for predicting the distribution of fingerprint database. If there are n collection locations (includes pilot points and the optimal path) with corresponding CSI data CSI_i and m remaining locations, the predictive CSI data CSI_j can be calculated by

$$CSI_j = \sum_{j=1}^{m} \sum_{i=1}^{n} cov(i,j) \times CSI_i \tag{15}$$

where $cov(\cdot)$ is the covariance matrix. And the physical coordinates of all the positions are known (namely the label), we combine CSI_i and CSI_j as the final predictive fingerprint database.

V. Experiments and Performance Analysis

In this section, we give the experimental configurations and evaluations in detail. Especially, compared to several existing algorithms [16], [18], [45] which use heuristic algorithms and reinforcement learning; localization accuracy is verified through the KNN algorithm based on the optimal path.

A. Environments and Parameters Settings

We use CSI as the fingerprint data which is fine-grained information and more stable than RSS. For computing the hyperparameters of a GP model, it is necessary to collect a small set of pilot CSI data. Therefore, we utilize the monitoring mode of the Linux 802.11n CSI tool to transmit and receive data that can control the speed of transmission packets and save time in the off-line stage. We consider the three antennas of the transmitter/receiver with power draw less than 1W, using wide 40MHz as they are more energy-efficient, and transmit power is from -10 dBm to +15 dBm. The receiver (a laptop) is only responsible for collecting data, and all other algorithms run on the local server. The demand for energy and computing power of devices is low, which also meets the application of large-scale positioning scenes. And readers can refer to [46] for detailed power consumption. For a fair comparison, the NLOS and LOS environments are selected and formulated into graphs as shown in Fig. 5 and Fig. 6, respectively. Described as follows:

1) Area One: It is a $13.5 \times 11\text{m}^2$ computer laboratory with tables, chairs and obstacles, which can be considered as a NLOS scenario. In such a complex environment, people move

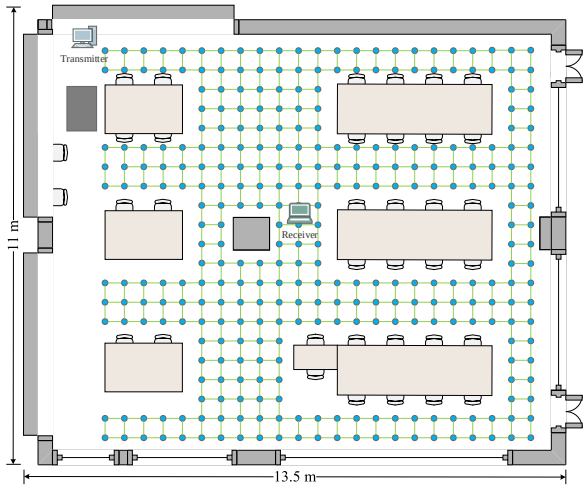


Fig. 5. Layout of Area One and corresponding sampling locations

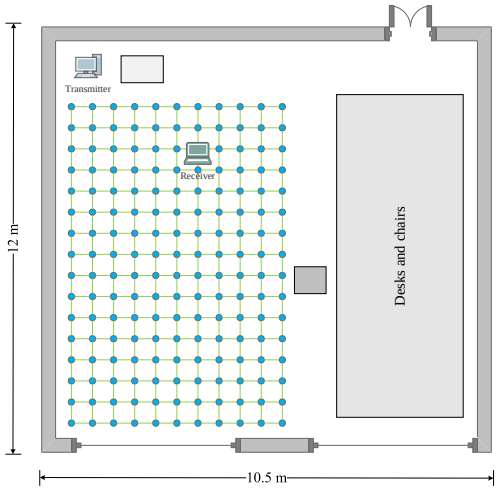


Fig. 6. Layout of Area Two and corresponding sampling locations

TABLE I
PARAMETER SETTINGS OF THE PROPOSED ALGORITHM

Parameters	Area One	Area Two
α	0.9	0.9
β	0.01	0.02
γ	0.95	0.8
ω	0.9	0.95
Learning rate for actor	0.25	0.15
Learning rate for critic	0.1	0.1
Maximum exploration step length	200	100
Maximum Iterations	100	100
Neurons of input layer for actor network	100	50
Neurons of input layer for critic network	50	30

TABLE II
LOCALIZATION ACCURACY

Data set	Area One		Area Two	
	Mean errors (m)	Std (m)	Mean errors (m)	Std (m)
Manual	4.22256	1.96362	3.19923	1.20988
Predictive	4.21950	2.15925	2.94827	1.56873

around during the collection of pilot data that creates more uncertainty. Therefore, the proposed algorithm needs to have robustness. There are 317 sampling locations and the lattice distance is 50 cm, we put the transmitter and receiver on the floor for continuously receiving packets.

2) Area Two: It is a $7 \times 10\text{m}^2$ meeting room and almost empty. There is less signal shielding and no people move during the collection phase which can be considered as a LOS scenario. There are 176 sampling locations at a distance of 60 cm from each other. The pilot data are collected for several seconds in one location, and we utilize continuous 1000 packets to ensure data validity.

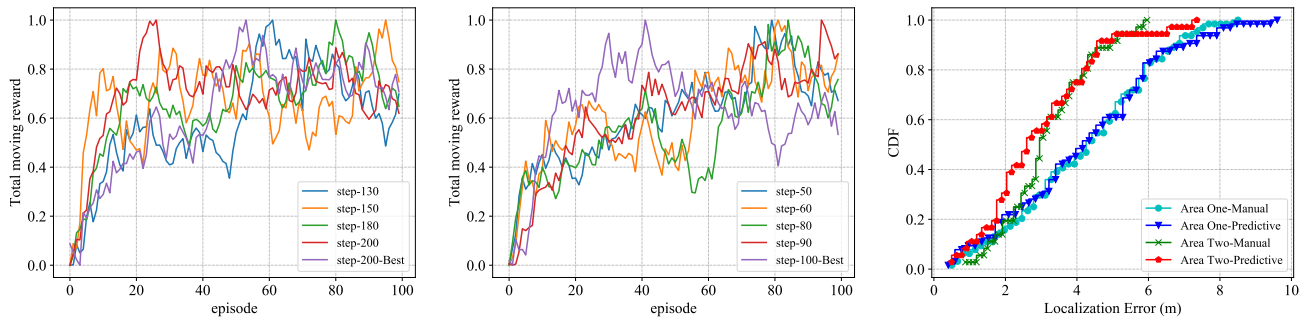
The proposed algorithm is programmed with Python and runs on a Dell laptop with a configuration of i7-8550U CPU and 16GB RAM. The implementation is similar to Gym which is a toolkit for reinforcement learning [47], including reset, step, render, and other functions. We also utilize Tensorflow and Tkinter for completing the main function. And the parameter settings are listed in Table I.

Considering the data collection process, we set the NLOS scenario which brings more interference to the raw CSI dataset and makes it closer to the practical application. If the agent is a UAV that will be hovering, jitter, and other problems, it is difficult to capture accurate CSI data and different from the "static" collection method. Here, the filtering algorithms can be utilized to eliminate noise interference, and thresholds can also be set using low/high-pass filtering [8]. [17] used an onboard processing unit to assist the UAV in capturing more accurate data to achieve meter level positioning accuracy. And most localization algorithms often include data preprocessing to improve the precision of the fingerprint database for training a better positioning model [19].

B. Training and Localization Using Optimal Path

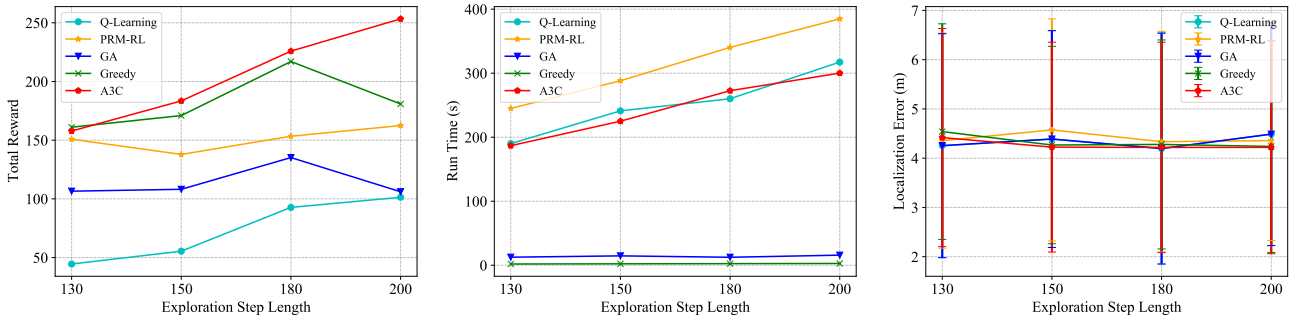
First, we compare the convergence performance with different maximum exploration step length and other parameters remain the same. The larger the step size in an episode, the bigger the total reward, we normalize the reward value with the training process for a fair comparison. The total moving reward in two scenarios are shown in Fig. 7 (a) and (b). The learning target is to get an effective path and make the data as informative as possible until the step length is used up, and run 100 episodes for different settings. The exploration strategy based A3C gradually approaches the stable reward value. And in the early stage of learning, although there are several inflection points which do not mean generated paths are valid, such as the agent does not arrive at G.

Furthermore, the convergence curve of area one fluctuates more than that of area two, because the first scenario has more complex environmental factors with a large scale formulated graph and people are walking around during the data collection process. It leads the agent to fall into the local optimal solution when searching valid paths, and cannot always make the total



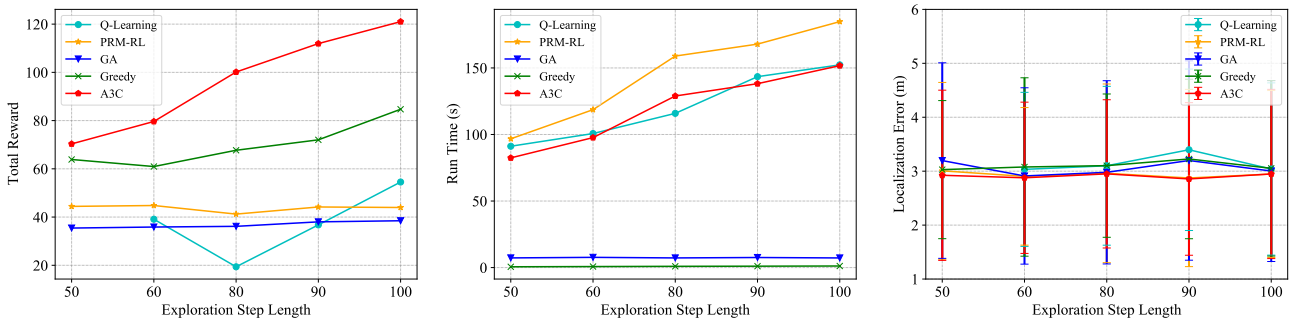
(a) Area One (b) Area Two (c) CDF of Localization Error

Fig. 7. Total moving reward with A3C and localization accuracy



(a) Total Reward (b) Run Time (c) Localization Error

Fig. 8. The performance comparison of different algorithms with different exploration step length in Area One



(a) Total Reward (b) Run Time (c) Localization Error

Fig. 9. The performance comparison of different algorithms with different exploration step length in Area Two

reward increase. We set the start and goal vertexes are diagonal with a novel action selection method, and several effective path combinations can be found within a limited number of iterations. And the agent can explore more positions with the increase of step size, therefore, we let such a path as the final result which has the maximum total reward under the maximum exploration step length.

Next, we compare the localization performance to verify the effectiveness of the predictive fingerprint database. To fully illustrating its effectiveness, we collected CSI data at all sampling positions for the two scenarios, namely the real fingerprint database. And then we compare the localization accuracy by utilizing the two sets of data sets. The core of this paper does not focus on how to implement the localization algorithm, therefore, we use the KNN algorithm for localization. More than 3000 CSI packets are collected at each location, and the sequential 1000 packets are utilized to train the model for ensuring the data validity. And we use the cross-validation method to split the two data sets with the same size and random sequence.

Fig. 7 (c) shows the cumulative distribution function (CDF) of the localization errors using the predictive fingerprint database and the collected manually CSI data in the two experimental environments. We can find that the two data sets achieve similar localization performance, the mean and standard errors are listed in Table II. Moreover, there are 3% testing data within a mean error of 1.0 m by using the manual data set in area one, while for our predictive data set the percentage is 9%. And the mean localization errors are 4.22256 m and 4.21950 m, respectively. For area two, it achieves better accuracy because the scenario has fewer interference factors and no people walking around. The mean

TABLE III
THE PERFORMANCE COMPARISON IN TWO SCENARIOS

Methods	Area One				Area Two			
	Total Reward	Run Time (s)	Mean errors (m)	Std (m)	Total Reward	Run Time (s)	Mean errors (m)	Std (m)
	Exploration Step length = 200				**Exploration Step length = 100**			
A3C	253.35219	300.10682	4.21950	2.15925	121.03031	151.72604	2.94827	1.56873
Q-Learning [45]	101.35085	317.35193	4.37618	2.20117	54.53107	152.30881	3.04024	1.59705
PRM-RL [18]	162.43894	384.87586	4.35665	2.02906	43.96619	184.78901	2.94731	1.54847
GA [16]	106.06189	15.78796	4.48829	2.26239	38.46924	7.23833	3.00268	1.67747
Greedy [16]	180.82397	2.78396	4.23880	2.15458	84.69479	1.18099	3.05289	1.62637
	Exploration Step length = 180				**Exploration Step length = 90**			
A3C	225.86894	272.66450	4.21850	2.13576	111.92493	138.16815	2.85584	1.41478
Q-Learning [45]	92.73963	260.09768	4.30418	2.20426	36.75316	143.44650	3.39682	1.49680
PRM-RL [18]	153.35313	340.36793	4.33513	2.23905	44.17999	167.89782	2.87342	1.64312
GA [16]	135.17633	12.47385	4.19543	2.34389	38.01080	7.56436	3.19770	1.84855
Greedy [16]	217.01964	2.51556	4.27942	2.12121	72.02001	1.04741	3.22624	1.47884
	Exploration Step length = 150				**Exploration Step length = 80**			
A3C	183.45831	225.02004	4.22409	2.12986	100.14541	128.91621	2.95015	1.37449
Q-Learning [45]	55.41802	241.17544	4.51829	2.17337	19.38239	115.86989	3.10026	1.47255
PRM-RL [18]	137.86469	288.22238	4.57416	2.25421	41.22687	158.92232	2.96053	1.65651
GA [16]	108.14469	14.65390	4.38864	2.20088	36.15182	7.25264	2.97872	1.69871
Greedy [16]	170.95610	2.24030	4.26715	2.00359	67.69368	0.94330	3.10262	1.32866
	Exploration Step length = 130				**Exploration Step length = 60**			
A3C	157.77887	186.79154	4.41841	2.21395	79.62861	97.65190	2.87733	1.40243
Q-Learning [45]	44.49507	190.03929	4.37286	2.27029	39.10732	100.70341	3.03341	1.42829
PRM-RL [18]	150.81848	244.95809	4.35581	2.18276	44.78679	118.64098	2.90411	1.27569
GA [16]	106.5495	12.59183	4.25428	2.27332	35.86368	7.68108	2.91142	1.63512
Greedy [16]	161.0486	1.94028	4.53952	2.18896	60.94945	0.76994	3.07772	1.65411

errors are 3.19923 m with the manual database and 2.94827 m with the predictive database, respectively. The most important is that we can reduce 68% and 72% collection tasks for the two areas and achieve similar accuracy. Especially, we just show the localization accuracy by using KNN to verify the validation of our predictive fingerprint database, and the performance can be improved by data pre-processing techniques, such as Kalman filter and principal component analysis.

C. Comparison with Other Algorithms under Different Exploration Step Length

We compare the different performance based on the optimal path with different algorithms for two scenarios in Table III. The start and target vertex are still set to S and G, and the maximum iteration number is 100 episodes. In the previous section, the total rewards are normalized for the convenience of comparison; in fact, the total reward increases with the exploration step length because the agent can arrive at more new states. As shown in Fig. 8 and 9, it is obvious that our proposed algorithm can always find the optimal path with the best reward compared with other algorithms. Especially, the completed four algorithms do not get valid solutions when we make the exploration step length equal to 100 for the experimental area one, and we do not describe the situation in this section.

Q-learning is a kind of learning method which is also widely applied [45]. We keep the same exploration strategy and use Q values to evaluate the action list for obtaining the optimal path. The agent can gradually arrive at G by the exploration actions, however, the Q values cannot effectively let the agent select an effective action during the early stage of learning due to them not being accurate. Therefore, Q-learning performance is not as superior as A3C, and it can not always find an effective path when the step size is 50 for area two. Q-learning has the lowest total reward and tends to go up overall, however, maybe it gets a locally optimal solution is shown in Fig. 9 (a). And it has a run time similar to that of A3C but has a large localization error.

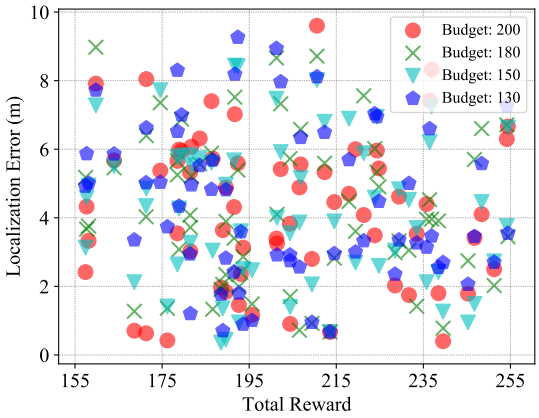


Fig. 10. Relation between reward and localization accuracy in Area One

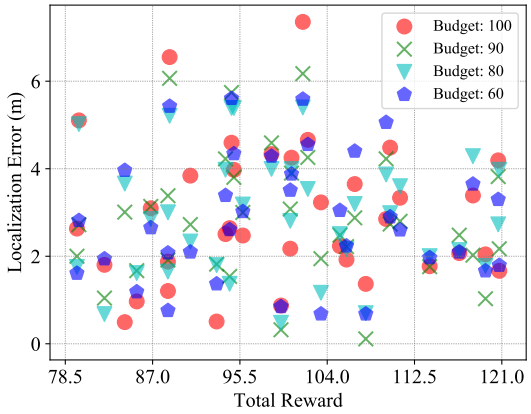


Fig. 11. Relation between reward and localization accuracy in Area Two

DDPG is a policy learning method for continuous actions which utilizes a convolutional neural network to simulate strategy function and Q Network for training based on the actor-critic framework. It is also suitable for path planning which is implemented following [18], namely PRM-RL. However, the inaccurate Q values lead to a slow convergence rate during the learning process as well as Q-learning. As shown in Table III, it is obvious that PRM-RL can also find the optimal paths in the both scenarios, but it has the longest run time and does not get the better total rewards using the same actor-critic framework as A3C. The better is that the localization performance of PRM-RL is relatively stable with less standard deviation under different exploration step lengths, it could be improved or using filters in our future research.

Genetic Algorithm (GA) is one of the classical solutions to the TSP problem, which includes crossover and mutation processes. We make the valid paths (which are obtained by A3C) as chromosomes, and let the path with the maximum value as the most optimal solution through constantly trying to select new actions which are adapted from [16]. Although the reward obtained by GA increases with step length, the slow growth may result in low learning efficiency due to the local optimal solution. GA has extremely short run times because we can directly calculate the final solution based on the results of A3C, but the large standard deviation of localization error means that the localization performance is unstable.

Greedy Algorithm preferentially selects with high reward value to form the effective path [48], which is implemented following [16]. Because of this greedy action selection strategy, the final effective path does not necessarily contain the target vertex G, and it tries to explore as many new positions as possible to increase the total reward. Compared with the other three algorithms, greedy algorithms can find valid solutions with high efficiency and better localization accuracy but not as good as A3C. However, the total reward does not always increase as the step size increases as shown in Fig. 8 (a), and its corresponding localization performance is not good.

The proposed algorithm in this paper is based on A3C, we utilize multiple agents combined with the novel exploration strategy to find the optimal path. For the two experimental scenarios, the total reward steadily increases and the localization precision gradually becomes more accurate with low standard deviation. The proposed algorithm has high computation complexity as well as Q-Learning and the maximum run time is about 5 minutes, the first scenario has more sampling positions and complex environmental factors, especially obstacles; the global network needs more time for training and optimization. Fortunately, the execution time of A3C can be significantly improved by GPU [44]. It is also obvious that the running time will increase with the increase of data scale as shown in Table III, but the optimal path can still be found within a considerable time. Therefore, our algorithm can also be utilized for large indoor scenes, such as airports and shopping malls.

D. Relation between Reward and Localization Accuracy

The previous section discusses the effect of the exploration step length on different algorithms in terms of total reward, etc. The reward is predicted by a Gaussian process regressor and the mutual information, the positioning errors will be lower if a path with a higher reward. Therefore, it is necessary to validate the relation between total reward and localization error. In detail, we collect the fingerprint data of some paths which are developed with the gradually increasing total reward by different budgets, namely the step length size. Then, building fingerprint databases and training different models for testing the performance of localization errors.

Fig. 10 and 11 show the total reward and the corresponding localization error in two scenarios with randomly selected test positions. We can find that the localization performance becomes better with the higher budget in both two experimental environments (although it is not always true), and area two performs better than area one because it is a LOS scenario that has less signal shielding. We conduct enough sampling point tests to ensure the effectiveness of the experiment under different budgets, and it reaches the optimal value on the whole when the budget is 200 or 100 in two scenarios. However, the errors of some test positions are even closer to 10 m which can not estimate the location of a target/user, it can be improved through the existing research [4], [21]. The localization algorithm is not the core of this paper, we do not discuss them in detail.

E. Computational Complexity

The proposed scheme is mainly based on A3C and uses multithreading to complete, which can replace the experience replay to save the storage overhead and computing resources. The computational complexity of the proposed scheme is given by

$$\mathscr{O}\left(\left(N_u \cdot \frac{1}{N_u}\right) \cdot T \cdot E\right) = \mathscr{O}(TE) \tag{16}$$

where N_u is the number of CPU threads used to train the scheme, T is the training iterations, and E is the exploration step length. In this paper, the training iterations used in the policy gradient are set to 100 in experimental areas, making the convergence speed sufficient to meet the timeliness requirement.

VI. Conclusion

In this paper, we propose a novel CSI data collection strategy based on path planning for indoor localization using an A3C framework. We consider that the distribution of CSI data depends on physical positions and conforms to the multivariate Gaussian, and utilize the mutual information and differential entropy to predict the reward values for all sampling points during the off-line stage. Combined with the actor-critic framework of A3C, multiple agents have better efficiency and find a valid path based on the proposed action selection method. Furthermore, comparison experiments show that the proposed algorithm is superior to the other state-of-the-art algorithms. The most important is that our algorithm can save 72% collection tasks and is similar to the localization accuracy of the manual data set. Our data collection strategy applies to large scenarios and the localization accuracy can be further improved by preprocessing technology or other localization algorithms. In the future, we plan to utilize multiple robots for investigating the path planning problem and fuse other types of fingerprint data, such as bluetooth or RFID.

References

[1] H. Abdelnasser, R. Mohamed, A. Elgohary, M. F. Alzantot, H. Wang, S. Sen, R. R. Choudhury, and M. Youssef, "SemanticSLAM: Using environment landmarks for unsupervised indoor localization," *IEEE Transactions on Mobile Computing*, vol. 15, no. 7, pp. 1770–1782, 2015.

[2] M. Di Felice, C. Bocanegra, and K. R. Chowdhury, "WI-LO: Wireless indoor localization through multi-source radio fingerprinting," in *2018 10th International Conference on Communication Systems & Networks*, 2018, pp. 305–311.

[3] J. Dong, M. Noreikis, Y. Xiao, and A. Ylä-Jääski, "ViNav: A vision-based indoor navigation system for smartphones," *IEEE Transactions on Mobile Computing*, vol. 18, no. 6, pp. 1461–1475, 2018.

[4] F. Zafari, A. Gkelias, and K. K. Leung, "A survey of indoor localization systems and technologies," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2568–2599, 2019.

[5] X. Zhu, T. Qiu, W. Qu, X. Zhou, M. Atiquzzaman, and D. Wu, "BLS-Location: A Wireless Fingerprint Localization Algorithm Based on Broad Learning," *IEEE Transactions on Mobile Computing*, 2021.

[6] I. Bisio, F. Lavagetto, M. Marchese, and A. Sciarrone, "Energy efficient WiFi-based fingerprinting for indoor positioning with smartphones," in *2013 IEEE Global Communications Conference*, 2013, pp. 4639–4643.

[7] Y. Ma, G. Zhou, and S. Wang, "WiFi sensing with channel state information: A survey," *ACM Computing Surveys*, vol. 52, no. 3, pp. 1–36, 2019.

[8] F. Gu, X. Hu, M. Ramezani, D. Acharya, K. Khoshelham, S. Valaee, and J. Shang, "Indoor localization improved by spatial context - A survey," *ACM Computing Surveys*, vol. 52, no. 3, pp. 1–35, 2019.

[9] J. Rocamora, I. W.-H. Ho, M. W. Mak, and A. Lau, "Survey of CSI Fingerprinting-based indoor positioning and mobility tracking systems," *IET Signal Processing*, 2020.

[10] R. C. Shit, S. Sharma, D. Puthal, P. James, B. Pradhan, A. van Moorsel, A. Y. Zomaya, and R. Ranjan, "Ubiquitous localization (UbiLoc): a survey and taxonomy on device free localization for smart world," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3532–3564, 2019.

[11] Y. Zhuang, Y. Li, L. Qi, H. Lan, J. Yang, and N. El-Sheimy, "A two-filter integration of MEMS sensors and WiFi fingerprinting for indoor positioning," *IEEE Sensors Journal*, vol. 16, no. 13, pp. 5125–5126, 2016.

[12] H. Zheng, M. Gao, Z. Chen, X.-Y. Liu, and X. Feng, "An adaptive sampling scheme via approximate volume sampling for fingerprint-based indoor localization," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2338–2353, 2019.

[13] J. Blum, S. Funke, and S. Storandt, "Sublinear search spaces for shortest path planning in grid and road networks," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018, pp. 6119–6126.

[14] M. D. Redžić, C. Laoudias, and I. Kyriakides, "Image and WLAN bimodal integration for indoor user localization," *IEEE Transactions on Mobile Computing*, vol. 19, no. 5, pp. 1109–1122, 2019.

[15] A. Viseras, R. O. Losada, and L. Merino, "Planning with ants: Efficient path planning with rapidly exploring random trees and ant colony optimization," *International Journal of Advanced Robotic Systems*, vol. 13, no. 5, p. 1729881416664078, 2016.

[16] Y. Wei, C. Frincu, and R. Zheng, "Informative path planning for location fingerprint collection," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 3, pp. 1633–1644, 2019.

[17] S. Piao, Z. Ba, L. Su, D. Koutsonikolas, S. Li, and K. Ren, "Automating CSI Measurement with UAVs: from Problem Formulation to Energy-Optimal Solution," in *IEEE Conference on Computer Communications*, 2019, pp. 2404–2412.

[18] A. Faust, K. Oslund, O. Ramirez, A. Francis, L. Tapia, M. Fiser, and J. Davidson, "PRM-RL: Long-range robotic navigation tasks by combining reinforcement learning and sampling-based planning," in *2018 IEEE International Conference on Robotics and Automation*, 2018, pp. 5113–5120.

[19] X. Zhu, W. Qu, T. Qiu, L. Zhao, M. Atiquzzaman, and D. O. Wu, "Indoor Intelligent Fingerprint-based Localization: Principles, Approaches and Challenges," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 4, pp. 2634–2657, 2020.

[20] X. Guo, N. R. Elikplim, N. Ansari, L. Li, and L. Wang, "Robust WiFi localization by fusing derivative fingerprints of RSS and multiple classifiers," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 5, pp. 3177–3186, 2019.

[21] Z. Gao, Y. Gao, S. Wang, D. Li, and Y. Xu, "CRISLoc: Reconstructable CSI fingerprinting for indoor smartphone localization," *IEEE Internet of Things Journal*, 2020.

[22] Y. Ma, C. Tian, and Y. Jiang, "A multitag cooperative localization algorithm based on weighted multidimensional scaling for passive UHF RFID," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6548–6555, 2019.

[23] N. Faulkner, F. Alam, M. Legg, and S. Demidenko, "Watchers on the Wall: Passive Visible Light-Based Positioning and Tracking With Embedded Light-Sensors on the Wall," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 5, pp. 2522–2532, 2019.

[24] H. Zhang, S. Y. Tan, and C. K. Seow, "TOA-based indoor localization and tracking with inaccurate floor plan map via MRMSC-PHD filter," *IEEE Sensors Journal*, vol. 19, no. 21, pp. 9869–9882, 2019.

[25] P. Du, S. Zhang, C. Chen, A. Alphones, and W.-D. Zhong, "Demonstration of a low-complexity indoor visible light positioning system using an enhanced TDOA scheme," *IEEE Photonics Journal*, vol. 10, no. 4, pp. 1–10, 2018.

[26] Y. Zheng, M. Sheng, J. Liu, and J. Li, "OpArray: Exploiting array orientation for accurate indoor localization," *IEEE Transactions on Communications*, vol. 67, no. 1, pp. 847–858, 2018.

[27] T. T. Mac, C. Copot, D. T. Tran, and R. De Keyser, "Heuristic approaches in robot path planning: A survey," *Robotics and Autonomous Systems*, vol. 86, pp. 13–28, 2016.

[28] M. Sakr and N. El-Sheimy, "Efficient Wi-Fi signal strength maps using sparse Gaussian process models," in *2017 International Conference on Indoor Positioning and Indoor Navigation*. IEEE, 2017, pp. 1–8.

[29] S. Dai, L. He, and X. Zhang, "Autonomous WiFi Fingerprinting for Indoor Localization," in *2020 ACM/IEEE 11th International Conference on Cyber-Physical Systems*, 2020, pp. 141–150.

[30] M. Kolakowski, "Automatic radio map creation in a fingerprinting-based BLE/UWB localisation system," *IET Microwaves, Antennas & Propagation*, 2020.

[31] S.-H. Jung, G. Lee, and D. Han, "Methods and tools to construct a global indoor positioning system," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 6, pp. 906–919, 2017.

[32] X. Tian, W. Zhang, Y. Yang, X. Wu, Y. Peng, and X. Wang, "Toward a quality-aware online pricing mechanism for crowdsensed wireless fingerprints," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 7, pp. 5953–5964, 2018.

[33] Y. Wei and R. Zheng, "Informative Path Planning for Mobile Sensing with Reinforcement Learning," in *IEEE Conference on Computer Communications*, 2020, pp. 864–873.

[34] Q. Liang and M. Liu, "An automatic site survey approach for indoor localization using a smartphone," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 1, pp. 191–206, 2019.

[35] M. Sattarian, J. Rezazadeh, R. Farahbakhsh, and A. Bagheri, "Indoor navigation systems based on data mining techniques in internet of things: a survey," *Wireless Networks*, vol. 25, no. 3, pp. 1385–1402, 2019.

[36] H. Shiri, J. Park, and M. Bennis, "Massive autonomous UAV path planning: A neural network based mean-field game theoretic approach," in *IEEE Global Communications Conference*, 2019, pp. 1–6.

[37] A. Spanos, "Near-collinearity in linear regression revisited: The numerical vs. the statistical perspective," *Communications in Statistics-Theory and Methods*, vol. 48, no. 22, pp. 5492–5516, 2019.

[38] C. Luo, L. Cheng, M. C. Chan, Y. Gu, J. Li, and Z. Ming, "Pallas: Self-bootstrapping fine-grained passive indoor localization using WiFi monitors," *IEEE Transactions on Mobile Computing*, vol. 16, no. 2, pp. 466–481, 2016.

[39] X. Tong, K. Liu, X. Tian, L. Fu, and X. Wang, "Fineloc: A fine-grained self-calibrating wireless indoor localization system," *IEEE Transactions on Mobile Computing*, vol. 18, no. 9, pp. 2077–2090, 2018.

[40] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. P. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International Conference on Machine Learning*, 2016, pp. 1928–1937.

[41] A. Sobehy, E. Renault, and P. Mühlethaler, "CSI based indoor localization using Ensemble Neural Networks," in *International Conference on Machine Learning for Networking*, 2019, pp. 367–378.

[42] M. T. Hoang, B. Yuen, K. Ren, X. Dong, T. Lu, R. Westendorp, and K. Reddy, "A CNN-LSTM Quantifier for Single Access Point CSI Indoor Localization," *arXiv preprint arXiv:2005.06394*, 2020.

[43] GPy, "GPy: A gaussian process framework in python," http://github.com/SheffieldML/GPy, 2012.

[44] M. Babaeizadeh, I. Frosio, S. Tyree, J. Clemons, and J. Kautz, "GA3C: GPU-based A3C for deep reinforcement learning," *CoRR abs/1611.06256*, 2016.

[45] E. S. Low, P. Ong, and K. C. Cheah, "Solving the optimal path planning of a mobile robot using improved Q-learning," *Robotics and Autonomous Systems*, vol. 115, pp. 143–161, 2019.

[46] D. Halperin, B. Greenstein, A. Sheth, and D. Wetherall, "Demystifying 802.11 n power consumption," in *Proceedings of the 2010 International Conference on Power Aware Computing and Systems*, 2010, pp. 1–5.

[47] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "OpenAI Gym," arXiv:1606.01540, 2016.

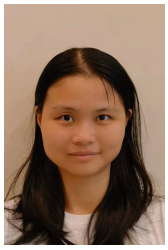
[48] V. Magnago, L. Palopoli, R. Passerone, D. Fontanelli, and D. Macii, "Effective landmark placement for robot indoor localization with position uncertainty constraints," *IEEE Transactions on Instrumentation and Measurement*, vol. 68, no. 11, pp. 4443–4455, 2019.

Xiaoqiang Zhu received the M.S. degree in computer technology from Dalian University of Technology (DUT), Dalian, China, in 2018. He is current working toward the Ph.D. degree with the College of Intelligence and Computing, Tianjin University, Tianjin, China. He would serve as a joint Ph.D. in ETH Zurich supported by China Scholarship Council (CSC) in 2021. He is an excellent graduate student of Liaoning province and DUT, and has been awarded several scholarships in academic excellence, such as National Inspirational Scholarship. His research interests include Internet of Things, indoor localization, and machine learning.

Tie Qiu (M'12-SM'16) received Ph.D degree in computer science from Dalian University of Technology in 2012. He is currently a Full Professor at School of Computer Science and Technology, Tianjin University, China. Prior to this position, he held assistant professor in 2008 and associate professor in 2013 at School of Software, Dalian University of Technology. He was a visiting professor at electrical and computer engineering at Iowa State University in U.S. (2014-2015). He serves as an associate editor of IEEE Transactions on SMC: Systems, area editor of Ad Hoc Networks (Elsevier), associate editor of IEEE Access Journal, Computers and Electrical Engineering (Elsevier), Human-centric Computing and Information Sciences (Springer), a guest editor of Future Generation Computer Systems. He serves as General Chair, Program Chair, Workshop Chair, Publicity Chair, Publication Chair or TPC Member of a number of international conferences. He has authored/co-authored 9 books, over 100 scientific papers in international journals and conference proceedings, such as IEEE/ACM ToN, IEEE TMC, TKDE TII, TIP, TCY, TITS, TVT, IEEE Communications Surveys & Tutorials, IEEE Communications, INFOCOM, GLOBECOM etc. There are 10 papers listed as ESI highly cited papers. He has contributed to the development of 3 copyrighted software systems and invented 14 patents. He is a senior member of China Computer Federation (CCF) and a Senior Member of IEEE and ACM.

Wenyu Qu received the B.S. degree in information science and M.S. degree in engineering mechanics from Dalian University of Technology, Dalian, China, in 1994 and 1997, respectively, and the Ph.D. degree in applied mathematics from the Japan Advanced Institute of Science and Technology, Nomi, Japan, in 2006. She is currently a Professor with the School of Computer Software, Tianjin University, Tianjin, China. From 2007 to 2015, she was a Professor with Dalian Maritime University, Dalian, China. From 1997 to 2003, she was an Assistant Professor with Dalian University of Technology. She has authored more than 80 technical papers in international journals and conferences. She is on the committee board for a couple of international conferences. Her research interests include cloud computing, computer networks, and information retrieval.

Xiaobo Zhou (S'11-M'13-SM'19) received the B.Sc. in electronic information science and technology from the University of Science and Technology of China (USTC), Hefei, China, the M.E. in computer application technology from Graduate University of Chinese Academy of Science (GUCAS), Beijing, China, and the Ph.D. degree in information science from School of Information Science, Japan Advanced Institute of Science and Technology (JAIST), Ishikawa, Japan, in 2007, 2010, and 2013, respectively. From April 2014 to March 2015, he was a Researcher with the Department of Communications Engineering, University of Oulu, Oulu, Finland. Currently, he is an Associate Professor with the School of Computer Science and Technology, Tianjin University, Tianjin, China. His research interests include cooperative wireless communications, wireless networks, data center networks, vehicular networks and mobile edge computing.



Yifan Wang received a B.S. degree in computer science from Fudan University, Shanghai, China in 2009, and a Ph.D. degree in computer science from University of Florida in 2015. She is now a postdoctoral researcher at the University of Florida. Her research interests are communications, artificial intelligence, and security.

Dapeng Oliver Wu (S'98-M'04-SM'06-F'13) received a B.E. degree in electrical engineering from Huazhong University of Science and Technology, Wuhan, China, in 1990, an M.E. degree in electrical engineering from Beijing University of Posts and Telecommunications, Beijing, China, in 1997, and a Ph.D. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, in 2003.

He is a professor at the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL. His research interests are in the areas of networking, communications, signal processing, computer vision, machine learning, smart grid, and information and network security. He received University of Florida Term Professorship Award in 2017, University of Florida Research Foundation Professorship Award in 2009, AFOSR Young Investigator Program (YIP) Award in 2009, ONR Young Investigator Program (YIP) Award in 2008, NSF CAREER award in 2007, the IEEE Circuits and Systems for Video Technology (CSVT) Transactions Best Paper Award for Year 2001, and the Best Paper Awards in IEEE GLOBECOM 2011 and International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks (QShine) 2006.

He has served as Editor in Chief of IEEE Transactions on Network Science and Engineering, Editor-at-Large for IEEE Open Journal of the Communications Society, founding Editor-in-Chief of Journal of Advances in Multimedia, and Associate Editor for IEEE Transactions on Communications, IEEE Transactions on Signal and Information Processing over Networks, IEEE Signal Processing Magazine, IEEE Transactions on Circuits and Systems for Video Technology, IEEE Transactions on Wireless Communications and IEEE Transactions on Vehicular Technology. He is also a guest-editor for IEEE Journal on Selected Areas in Communications (JSAC), Special Issue on Cross-layer Optimized Wireless Multimedia Communications and Special Issue on Airborne Communication Networks. He has served as Technical Program Committee (TPC) Chair for IEEE INFOCOM 2012, and TPC chair for IEEE International Conference on Communications (ICC 2008), Signal Processing for Communications Symposium, and as a member of executive committee and/or technical program committee of over 100 conferences. He was elected as a Distinguished Lecturer by IEEE Vehicular Technology Society in 2016. He has served as Chair for the Award Committee, and Chair of Mobile and wireless multimedia Interest Group (MobIG), Technical Committee on Multimedia Communications, IEEE Communications Society. He was an elected member of Multimedia Signal Processing Technical Committee, IEEE Signal Processing Society from Jan. 1, 2009 to Dec. 31, 2012. He is an IEEE Fellow.