Offline and Online Search: UAV Multiobjective Path Planning Under Dynamic Urban Environment

Chao Yin, Zhenyu Xiao, *Senior Member, IEEE*, Xianbin Cao, *Senior Member, IEEE*, Xing Xi, Peng Yang, and Dapeng Wu, *Fellow, IEEE*

***Abstract*—This paper is concerned with path planning for unmanned aerial vehicles (UAVs) flying through low altitude urban environment. Although many different path planning algorithms have been proposed to find optimal or near-optimal collision-free paths for UAVs, most of them either do not consider dynamic obstacle avoidance or do not incorporate multiple objectives. In this paper, we propose a multiobjective path planning (MOPP) framework to explore a suitable path for a UAV operating in a dynamic urban environment, where safety level is considered in the proposed framework to guarantee the safety of UAV in addition to travel time. To this aim, two types of safety index maps (SIMs) are developed first to capture static obstacles in the geography map and unexpected obstacles that are unavailable in the geography map. Then an MOPP method is proposed by jointly using offline and online search, where the offline search is based on the static SIM and helps shorten the travel time and avoid static obstacles, while the online search is based on the dynamic SIM of unexpected obstacles and helps bypass unexpected obstacles quickly. Extensive experimental results verify the effectiveness of the proposed framework under the dynamic urban environment.**

***Index Terms*—Low altitude urban environment, offline and online search, safety index map (SIM), unmanned aerial vehicle (UAV) path planning.**

I. Introduction

SINCE the federal aviation administration unveiled plans for domestic unmanned aerial vehicles (UAVs) to take off by 2015 [1], the usage of UAVs has received significant attention [2]–[4]. In the future smart cities, UAV itself would play an important role in but not limited to cargo delivery, emergency communications, reconnaissance, fire-fighting, aerial photo, remote sensing, disaster rescue, etc. Moreover, when using Internet of Things to improve smart cities, UAV can be used to collect sensing data all over the cities, as well as outside the cities, for the sensors that cannot access to a global network. To benefit these usages of a UAV, it should be first safely and timely designated to the target area, which is the so-called path planning problem.

Path planning is the process of determining a collision-free pathway between a UAV's current position and its destination. Researchers have been studying how to generate collision-free paths for vehicles in obstacle environments, which is critical for autonomous vehicles. So far, three types of path planning methods have been proposed, namely the geometry method, the velocity space method, and the graph search method. The geometry method takes geometry information of obstacles in the map as an input to search for a path. Typical geometry methods include the potential field method [5], the vector field histogram method [6], and the Voronoi diagram method [7]. The velocity space approach, which is a collision avoidance method considering the vehicle shape, kinematic and dynamic constraints, includes the curvature velocity method [8] and the dynamic window method [9]. The graph search approach aims to find the shortest path between two vertices in a certain graph, where the A* algorithm is known to be effective at finding the shortest path to a goal while avoiding obstacles [10].

However, these path planning approaches [5]–[9] cannot be directly exploited to perform the UAV path planning in the low altitude dynamic urban environment, because they rely on timely updated global and deterministic map information. In a low altitude dynamic urban environment, the timely update of the global map information is almost unavailable at a UAV, which usually has a limited communication bandwidth. Moreover, there are always unexpected obstacles that may not be indicated in a geography map. In such a case, a UAV would be possible to crash against the unexpected obstacles. Although replanning can be leveraged to mitigate the uncertainty during flying, there are at least two significant problems.

1) Generating dynamically feasible trajectories in real time with limited onboard computational resources.
2) Dealing with the unpredictable changing surrounding environments within limited sensing range as well as limited GPS position precision.

In this paper, we aim to realize an efficient path planning and meanwhile guarantee the safety of a UAV during flying

Manuscript received December 31, 2016; revised May 22, 2017; accepted June 14, 2017. Date of publication June 19, 2017; date of current version April 10, 2018. This work was supported in part by the National Key Research and Development Program under Grant 2016YFB1200100, in part by the National Natural Science Foundation of China under Grant 91538204, in part by the National Science Fund for Distinguished Young Scholars under Grant 61425014, and in part by the Foundation for Innovative Research Groups of the National Natural Science Foundation of China under Grant 61521091. *(Corresponding author: Zhenyu Xiao.)*

C. Yin, Z. Xiao, X. Cao, X. Xi, and P. Yang are with the School of Electronic and Information Engineering, Beihang University, Beijing 100083, China, also with the Beijing Key Laboratory for Network-Based Cooperative Air Traffic Management, and Beijing Laboratory for General Aviation Technology, Beijing 100191, China, and also with the Collaborative Innovation Center of Geospatial Technology, Wuhan 430079, China (e-mail: ycbuaa2011@buaa.edu.cn; xiaozy@buaa.edu.cn; xbcao@buaa.edu.cn; xixing@buaa.edu.cn; yangp09@buaa.edu.cn).

D. Wu is with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: dpwu@ieee.org).

Digital Object Identifier 10.1109/JIOT.2017.2717078

in a dynamic low altitude urban environment.[1] To this aim, we need to address the dynamic replanning problem as well as the static preplanning problem. The contributions of this paper can be summarized as follows.

1) We introduce a safety index concept with which a safety index map (SIM) of a low altitude urban environment is established. Specifically, we design two types of SIMs, i.e., a static SIM which incorporates the main obstacles in the corresponding geography map and a dynamic SIM which incorporates unexpected obstacles that are not available in the geography map during flying. The static SIM is offline constructed concerning the GPS position error, while the dynamic SIM is online constructed when detecting an unexpected obstacle concerning both the sensing range and the safety distance of the UAV.
2) A multiobjective path planning (MOPP) problem is formulated with both travel time and safety level considered, and a novel path planning method is proposed to solve the problem by jointly exploiting offline and online search based on the two types of SIMs. The offline search is to find a Pareto optimal path to avoid static obstacles based on the static SIM, while the online search is to find a Pareto optimal path to avoid unexpected obstacles based on the dynamic SIM with a low computational complexity.

The rest of this paper is as follows. In Section II, we introduce the related works. Section III builds a system model of the proposed framework and Section IV describes the framework and presents its detailed design and implementation. We show our experimental results in Section V and conclude this paper in Section VI.

II. Related Work

In recent years, researchers have developed various works to address the MOPP problem for UAVs [11]–[15]. For example, Roberge *et al.* [11] attempted to compute possible and quasi-optimal paths for UAVs in a complex 3-D environment by leveraging the genetic algorithm (GA) and the particle swarm optimization algorithm. Meanwhile, they adopted the "single-program, multiple-data" parallel programming paradigm technique to reduce the execution time so that the solution method was real-time. Hernández-Hernández *et al.* [12] proposed a multiobjective A* algorithm to find a UAV path from a start point to a destination point, where travel time, path angle, fuel consumption, and area deviation are considered. Tao *et al.* [13] presented a modified GA to solve the path planning problem with multiple objectives. In addition, Niendorf *et al.* [14] proposed a method to deal with the UAV mission planning in hostile environments, where a tradeoff between short mission execution time and low risk was made. Moreover, Liu *et al.* [15] put forward a synergistic self-adapted difference multiobjective evolution algorithm for path planning of UAV to fulfill the function of UAV's sensing and avoidance of obstacles. To accelerate the calculation speed the self-adapted adjustment strategy, crossing, and variation were involved. Although these MOPP algorithms [11]–[15] have been developed to find an achievable path for a UAV flying in a complex environment, none of them has addressed the problem of dynamic obstacle avoidance. Most of them attempted to construct risk or forbidden zones only for static threats so that a UAV can be guided to bypass these forbidden areas.

To avoid both static and dynamic threats, Wen *et al.* [16] proposed a novel method for UAV online path planning in low altitude dynamic environments. In this approach, the intuitionistic fuzzy set was used to model static threats, and a reachability set estimator is constructed to assess the dynamic threats. To reduce the cost of planning and rapidly obtain a low threat UAV path, a heuristic subgoal selector was proposed and integrated into the planning system. Although the framework proposed in [16] was certified to perform well in finding collision free paths for a UAV, it only focused on solving a single objective (i.e., safety) path planning problem.

To extend the solution to MOPP under dynamic environment, Wu *et al.* [10] presented a grid-based multistep A* (MSA*) search method to address the MOPP problem for UAVs operating in large, dynamic and uncertain 4-D environments. MSA* exploits a variable successor operator that enables variable track length, angle, and velocity trajectory segments to find least expense paths. This variable successor operator provides an inherent tolerance to uncertainty based on the minimum distance between the path and grid sequence boundaries. Furthermore, by leveraging variable successor operators to create a multiresolution, memory-efficient grid sampling structure, the search space was drastically reduced, and the computational efficiency was correspondingly improved. In addition, Peng *et al.* [17] developed an online path planning scheme to solve the dynamic multiobjective optimization problem, where the local environmental information collected by onboard sensors was continuously updated to replan the UAV trajectory. However, these methods [10], [17] only concentrate on online path planning. They cannot make use of offline information, i.e., the geography map, to reduce the travel time.

III. System Model

As mentioned in the previous section, our aim is to propose an MOPP framework to find a Pareto optimal path for a UAV flying through a low altitude dynamic urban space and meanwhile keep the UAV safe during flying. To this aim, two types of SIMs, i.e., a static SIM and a dynamic SIM, are created to capture static obstacles in the geography map, provided by Google Earth or Baidu Map, and unexpected obstacles that are unavailable in the geography map, respectively. Then a joint offline and online search-based method is developed to solve the MOPP problem considering two objectives, namely travel time and safety level. In this method, an offline search algorithm (OFSA) is exploited to plan a Pareto optimal path of avoiding static obstacles based on the static SIM, while an

[1]It is noteworthy that the proposed scheme in this paper also applies to the high attitude environment. In fact, the path planning of UAVs in a high attitude environment may be less challenging than that in a low altitude urban environment, because there are much fewer obstacles.

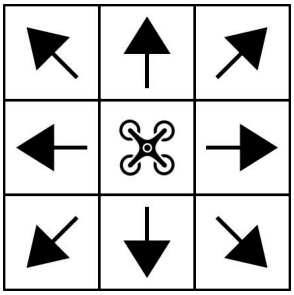


Fig. 1. Schematic of an eight-connectivity UAV motion model.

online search algorithm (ONSA) is explored to replan a Pareto optimal path of bypassing unexpected obstacles based on the dynamic SIM.

Since we need to deal with an MOPP problem in this paper, we first describe a general MOPP problem by means of space graph. Given a three-tuple $\langle G, S_{\text{start}}, S_{\text{target}}\rangle$, where $G = (N, A, c)$ represents a space graph, N is a set of points, A denotes an arc set, $c : A \to R^K$ represents a path cost function, K is the number of objective functions, the K-dimensional cost vector $c(m, n)$ denotes the cost of planning a path starting at point $\vec{s}_m$ and terminating at point $\vec{s}_n$. Although we focus on a 2-D space graph in this paper, this framework holds for space of any dimensions. When G represents undirected graph, $c(m, n) = c(n, m)$. In addition, $S_{\text{start}} \in N$ and $S_{\text{target}} \in N$ denote a start point set and a target point set, respectively.

Since we usually find a Pareto optimal solution for an MOPP problem, here we define a nondominant path, which is the so-called Pareto optimal path [18].

Definition 1: Given vectors $\vec{x}_1 \in R^K$ and $\vec{x}_2 \in R^K$. If and only if $\forall i$, $1 \le i \le K$, there exists $x_{1i} \le x_{2i}$ and $\vec{x}_1 \ne \vec{x}_2$, where x_{1i} and x_{2i} represent the ith element of $\vec{x}_1$ and $\vec{x}_2$, respectively, then we call $\vec{x}_1$ dominates $\vec{x}_2$ strongly, denoted by $\vec{x}_1 \prec \vec{x}_2$.

Definition 2: Given a vector set X. If and only if $\nexists \vec{y} \in X$ satisfies $\vec{y} \prec \vec{x}$, then $\vec{x} \in X$ is called nondominant.

Definition 3: Let P_{mn} be a set of all paths between two points $\vec{s}_m$ and $\vec{s}_n$, if and only if $\nexists p_2 \in P_{mn}$ satisfies $p_2 \prec p_1$, i.e., there is no path in P_{mn} that is better than p_1 at all objectives, then the path $p_1 \in P_{mn}$ can be considered as nondominant path.

To avoid an exhaustive construction of the graph G, a uniform Cartesian deterministic sampling scheme is utilized in this paper. The Cartesian sampling-based methods are widely used in mobile robotics including UAVs [19]. As a result, each point s is mapped uniquely to a grid or cell in the graph, and s refers simultaneously to both the cell and the point located in the center of each cell.

Furthermore, we leverage an eight-connectivity UAV motion model [19] shown in Fig. 1 to model the motion of a UAV in this paper. With this model, a UAV is assumed to have eight moving directions.

IV. Description of the Proposed MOPP

In this section, we explain how and why our MOPP framework works in details. MOPP introduces first a safety index concept with which a static SIM is offline constructed concerning the GPS position error and a dynamic SIM is online constructed when sensing an unexpected obstacle concerning both the sensing range and the safety distance of the UAV. On the basis of these two types of SIMs, we then formulate an MOPP problem and develop a novel path planning method to solve this problem by jointly leveraging offline and online search.

A. Static Safety Index Map

Urban obstacles in low altitude space threat the flying safety of UAVs. On the basis of the static obstacles we construct a static SIM of complex low altitude urban environment in this section to provide guidance for the safe flight of UAVs.

1) UAV Velocity Model: Given a trajectory traversal time n_t, the UAV cruise velocity (or airspeed of a UAV) can be derived from the trajectory length via the trajectory velocity (or ground speed of a UAV) $\vec{v}_t$

$$\vec{v}_t = \frac{||\vec{s}_1 - \vec{s}_2||_2}{n_t}. \tag{1}$$

This track velocity is itself a sum of the cruise velocity vectors

$$|\vec{v}_t|\begin{pmatrix}\sin\theta\\ \cos\theta\end{pmatrix} = |\vec{v}_c|\begin{pmatrix}\sin\alpha\\ \cos\alpha\end{pmatrix} \tag{2}$$

where θ is the horizontal track angle, $\vec{v}_c$ is the cruise velocity, α is the vehicle heading angle. All angles are measured from true north in navigational tasks [20].

Using the mentioned mobility model we can obtain the position of a UAV. However, both the UAV mobility model and onboard positioning and navigation system may determine the positions of UAVs. What is more, the positioning error inherent in the positioning and navigation system may lead to the position uncertainty. Next, we will discuss how to model the UAV position uncertainty.

2) Position Uncertainty Model: Uncertainty can be modeled using (approximated) probabilistic methods [21], [22]. In this paper, we only consider the case that the GPS uncertainty incurs the position uncertainty. In civil field, GPS systems have a horizontal accuracy of 5 to 10 meters of 95% confidence, and the vertical accuracy achieves approximately 1.4 times the horizontal accuracy [23]. Although the GPS error is small, the accumulated position uncertainty can be much greater.

For the purposes of simulation, a simple bivariate Gaussian model is utilized in this section. Given independent and isotropic GPS horizontal accuracy, the UAV position density function p can be expressed as

$$p(x_c, y_c, \sigma_x, \sigma_y) = \frac{1}{2\pi\sigma_x\sigma_y} e^{-\left(\frac{(x-x_c)^2}{2\sigma_x^2} + \frac{(y-y_c)^2}{2\sigma_y^2}\right)} \tag{3}$$

where σ_x and σ_y are the standard deviations, $\vec{s}_c = (x_c, y_c)$ is the coordinate of the center point of the position uncertainty model depicted in Fig. 2.

Given a normal distribution $X \sim (\mu, \sigma^2)$, it exists a famous 3σ principle, $P(\mu - 3\sigma < X \le \mu + 3\sigma) = 99.7\%$, i.e., if an event $X \notin (\mu - 3\sigma, \mu + 3\sigma]$, then its occurrence probability

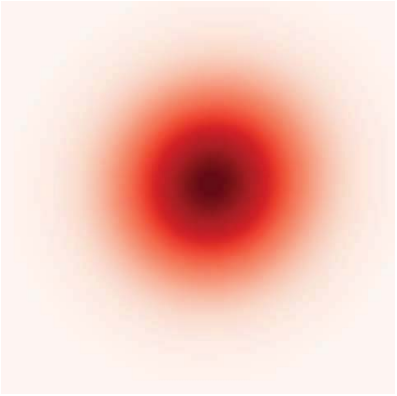


Fig. 2. Position uncertainty model.

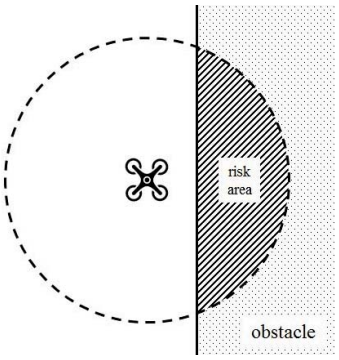


Fig. 3. UAV risk area.

is almost zeros. Similarly, we apply this 3σ principle in this paper to bind the above mentioned probability circle.

Summing up, the procedure of generating a static SIM can be as follows.

Definition 4 (Risk Probability): Given bivariate position uncertainty model $P(x_c, y_c)$, (x_c, y_c) represents the coordinate of a UAV located at the center point, the UAV risk probability refers to the probability that the UAV will hit obstacles due to the GPS positioning error, which can be denoted by

$$\Pr(x_c, y_c) = \int_{\phi \in \Phi} p\big(x_c, y_c, \sigma_x, \sigma_y\big) d\phi \tag{4}$$

where Φ represents the risk area shown in Fig. 3.

Next, we can calculate the safety probability of the UAV located at the center point using the form

$$\overline{\Pr}(x_c, y_c) = 1 - \Pr(x_c, y_c). \tag{5}$$

Definition 5 (Static SIM): Given a space graph $G = (N, A, c)$, the static SIM describes the UAV flight safety index at an arbitrary point $\vec{s}_c$ in the space graph, which can be denoted by

$$I\left(\vec{s}_c\right) = -10 \log_{10}\left(\overline{\Pr}(x_c, y_c)\right). \tag{6}$$

It is worth to note that the safety index of a path is the sum of the safety indices of all points on the path, and a greater path safety index indicates a more dangerous path. In realization, the safety indices of the points far away from an obstacle are small, e.g., close to zero in general. Only those of the points near an obstacle contribute to the safety index of a path. Hence, in order to reduce the computational complexity, we may only calculate the probability that a UAV collides with each obstacle on the path instead and obtain the corresponding safety index, which is in fact the summation of the safety indices of all the points near the obstacle.

B. Dynamic Safety Index Map

Remarkably, the computational complexity of generating the above introduced static SIM is relatively high; thus, it may not be suitable for online methods. In this section, we construct a dynamic SIM to depict the safe and hazardous regions of unexpected obstacles.

1) Perception Range: UAVs need to detect surroundings in performing missions to perceive the surrounding information. Denote R by the sensing distance of onboard sensing sensors; thus, we can define the perceptual range of a UAV as a circle centered at the UAV, and R is the radius in 2-D space. In other words, the perceptual range of a UAV can take the form

$$(x_c - x)^2 + (y_c - y)^2 = R \tag{7}$$

where, (x, y) is an arbitrary point within the perceptual range.

If a UAV detects that obstacles appearing in its perception range are unknown, it will mark these obstacles as unexpected and call the following mechanism to construct a dynamic SIM for them.

2) Safety Margin: When a UAV finds an unexpected obstacle appear in its perception range, one of the obstacle avoidance measures that the UAV can take is emergency braking, which is associated with the speed of the UAV. We call the braking distance "safety margin" (denoted by d_{sm}) in this paper.

Next, we can utilize the follow formation to construct the dynamic SIM of unanticipated obstacles:

$$I\left(\vec{s}_c\right) = \begin{cases} +\infty, & d_c \le d_{\text{sm}} \\ 0, & \text{else} \end{cases} \tag{8}$$

where, d_c represents the vertical distance between the UAV and the unexpected obstacle.

Summing up, the main step of constructing the SIM can be described in Algorithm 1.

C. Multiobjective Path Planning Problem

This section discusses a set of relevant objectives and a formulation of an MOPP problem for a UAV flying in a low altitude urban environment. In addition, it describes the generation of the cost function for the proposed joint search algorithm.

The UAV path planning in complex low altitude urban environment is constrained by a set of internal constraints (e.g., arrival time, energy consumption, turning radius) and external constraints (e.g., other UAVs, highrise, birds, high power lines, and helikite). For the sake of mission scenario, two major decision objectives, the safety and travel time, are considered in this paper.

1) Safety: The safety objective is modeled with a criterion for collision avoidance. The collision avoidance criterion

Algorithm 1 Construction of an SIM

```
Input: origin map MAP, variance of bivariate normal distribution σ_x, σ_y
R = 3max{σ_x, σ_y}
cov = [σ_x^2  0; 0  σ_y^2]
for point in MAP do
    if point is not OBSTACLE then
        P[point]=1.0
        for d_s in circle_region(point,R) do
            p=bivariate_normal(d_s, point, cov)
            P[point]=P[point]-p
        end for
        SIM[point]=-10log_10(P[point])
    end if
end for
for node is in SIM do
    if node is OBSTACLE then
        tag point as FORBIDDEN
    end if
end for
Return SIM
```

needs a separation distance from the obstacles. Furthermore, there is uncertainty in the position, velocity, and direction of dynamic obstacles because of sensor error. Such uncertainty may increase the risk of collision, and the closer the cells are from the obstacles the greater the cross cost are, representing the danger of the cells.

2) Travel Time: The mission plan also needs to optimize for one crucial objective of the mission itself, i.e., the traveling time, for instance, the emergency delivery of a medical package to a remote area requires arriving at the target area in minimal time. The travel time is typically restricted by the cruise velocity, flight trajectory, and so on.

Here, we will formulate a multiobjective optimization problem with two objectives: 1) travel time and 2) safety. The solution of this problem is to find an optimal path p between the source point and target point in a graph $G(N, A, c)$. Each arc i in A has two non-negtive costs denoted by $c_{i,1}$ and $c_{i,2}$ representing the travel time and safety, respectively. Let $g_j(p)$ be the total cost of the whole arcs in a path p for the jth objective. Let $\vec{g}(p) = (g_1(p), g_2(p))$ be the cost vector for a path p. Therefore, when compared with any other path q the "optimal" path p should meet with the following conditions:

$$g_j(p) \le g_j(q) \quad \forall j \in \{1, 2\} \text{ and } \exists i \in \{1, 2\}, g_i(p) < g_i(q). \tag{9}$$

Solutions that are nondominated by any other solutions are the Pareto optimal solutions (denoted by a Pareto set P in this paper). Path p is a Pareto optimal path of the Pareto optimal set, which has the minimal total cost W

$$W = \min_{p \in P} \alpha g_1(p) + (1 - \alpha) g_2(p) \tag{10}$$

where the weight coefficient α can vary according to preferences of different users.

D. Proposed Offline and Online Search Method

In this section, we will propose a joint offline and online search-based path planning algorithm. The OFSA is applied to find a path of avoiding static obstacles while the ONSA is exploited to address the problem of bypassing dynamic threat zones caused by unexpected obstacles.

1) Offline Search: The OFSA carries the similar spirit as the level set method initially designed by Osher and Sethian [24].

Given a speed function $V(\vec{s})$, it denotes the speed of the wave propagation at point $\vec{s}$. For a path segment $l \in p$, let the cost w from one end of the arc to the other be $w = \alpha I(l) + (1 - \alpha)t$, where time $t = \int_l \frac{1}{V(\vec{s})} d\vec{s}$, $I(l) = \int_l I(\vec{s}) d\vec{s}$ represents the safety index of the path segment l. Next, we can define a new speed function $V^*(\vec{s})$ that can take the following form:

$$V^*\left(\vec{s}\right) = \frac{V\left(\vec{s}\right)}{\alpha I\left(\vec{s}\right) \cdot V\left(\vec{s}\right) + (1 - \alpha)}. \tag{11}$$

It indicates when the safety index value is large, i.e., the safety level is low, the wave propagation may slow down, which is reasonable.

The fundamental idea of the level set method is to match a single parameter family of moving wave fronts $\{\gamma_w\}_{w \ge 0}$, where γ_w represents the position of the wave front at cost w. To identify the front propagation, we then have to find and solve a partial differential equation for the movement of the evolving surface. Precisely, let γ_0 be an initial wave front in R^d ($d = 2$ in this paper), and assume that a level set function $\phi : R^d \times R_+ \to R$ is such that at cost $w \ge 0$ the zero level set of ϕ is the front γ_w. We further assume that $\phi(\vec{s}; 0) = \pm d(\vec{s})$; where $d(\vec{s})$ is the distance from $\vec{s}$ to the curve γ_0. We use plus sign if $\vec{s}$ is inside 0 and minus if $\vec{s}$ is outside. Let each level set of ϕ along its gradient field with speed V^*. Now consider the motion of, e.g., the level set

$$\left\{\vec{s} \in R^d : \phi\left(\vec{s}; w\right) = 0\right\}. \tag{12}$$

Let $\vec{s}(w)$ be trajectory of a particle located at this level set so that

$$\phi\left(\vec{s}(w); w\right) = 0. \tag{13}$$

The gain ratio $d\vec{s}/dw$ in the direction n normal to the level set is given by the speed function V^*, and hence,

$$\frac{d\vec{s}}{dw} \cdot n = V^* \tag{14}$$

where the normal vector n is given by

$$n = -\frac{\nabla \phi}{|\nabla \phi|}. \tag{15}$$

This is a vector pointing outwards, giving our initialization of n. By the chain rule

$$\frac{d\phi}{dw} + \frac{d\vec{s}}{dw} \cdot \nabla \phi = 0. \tag{16}$$

Therefore, $\phi(\vec{s}; w) = 0$ satisfies the partial differential equation (the level set equation)

$$\frac{\partial \phi}{\partial w} - V^* \cdot |\nabla \phi| = 0 \tag{17}$$

and the initial condition

$$\phi\left(\vec{s}; w=0\right) = \pm d\left(\vec{s}\right). \tag{18}$$

It is called a Eulerian formulation of the front propagation problem since it is written regarding a fixed coordinate system in a physical domain.

If the speed function V^* is either always positive or always negative, we can introduce a new variable $W(\vec{s})$ defined by $\phi(\vec{s}, W(\vec{s})) = 0$. In other words, $W(\vec{s})$ is the cost when $\phi(\vec{s}; w) = 0$. If the gain ratio $(d\vec{s}/dw) \neq 0$, then W will satisfy the stationary Eikonal equation

$$V^*|\nabla W| = 1 \tag{19}$$

coupled with the boundary condition

$$W|_{d\left(\vec{s}\right)=0} = 0. \tag{20}$$

Before discussing how to solve (19), we introduce an eight-degree-of-freedom coordinate system consisting of a Cartesian coordinate system and its counter-clockwise rotation of 45°.

As the front can only expand ($V^* > 0$), the cost W is single valued. Osher and Sethian [24] proposed a discrete solution for the Eikonal equation. In 2-D, the area is discretized using a grid map.

We denote the row i and column j of the gridmap, which corresponds to a point $\vec{s} = (i, j)$ in the real world. Similar to [24], the discretization of the gradient ∇W can take the following form:

$$\max\left\{\begin{array}{l} \max\left(D_{ij}^{-x}W, 0\right)^2 + \min\left(D_{ij}^{+x}W, 0\right)^2 + \\ \max\left(D_{ij}^{-y}W, 0\right)^2 + \min\left(D_{ij}^{+y}W, 0\right)^2, \\ \max\left(D_{ij}^{-u}W, 0\right)^2 + \min\left(D_{ij}^{+u}W, 0\right)^2 + \\ \max\left(D_{ij}^{-v}W, 0\right)^2 + \min\left(D_{ij}^{+v}W, 0\right)^2 \end{array}\right\} = \frac{1}{V^{*2}_{ij}}. \tag{21}$$

According to [24], we can know that a simpler but less accurate solution for (21) can be expressed as follows:

$$\max\left\{\begin{array}{l} \max\left(D_{ij}^{-x}W, -D_{ij}^{+x}W, 0\right)^2 + \\ \max\left(D_{ij}^{-y}W, -D_{ij}^{+y}W, 0\right)^2, \\ \max\left(D_{ij}^{-u}W, -D_{ij}^{+u}W, 0\right)^2 + \\ \max\left(D_{ij}^{-v}W, -D_{ij}^{+v}W, 0\right)^2 \end{array}\right\} = \frac{1}{V^{*2}_{ij}} \tag{22}$$

where

$$\begin{aligned} D_{ij}^{-x}W &= \frac{W_{i,j} - W_{i-1,j}}{\Delta x} & D_{ij}^{+x}W &= \frac{W_{i+1,j} - W_{i,j}}{\Delta x} \\ D_{ij}^{-y}W &= \frac{W_{i,j} - W_{i,j-1}}{\Delta y} & D_{ij}^{+y}W &= \frac{W_{i,j+1} - W_{i,j}}{\Delta y} \\ D_{ij}^{-u}W &= \frac{W_{i,j} - W_{i-1,j-1}}{\Delta u} & D_{ij}^{+u}W &= \frac{W_{i+1,j+1} - W_{i,j}}{\Delta u} \\ D_{ij}^{-v}W &= \frac{W_{i,j} - W_{i+1,j-1}}{\Delta v} & D_{ij}^{+v}W &= \frac{W_{i-1,j+1} - W_{i,j}}{\Delta v} \end{aligned} \tag{23}$$

where Δx, Δy, Δu, and Δv are the grid spacing in x, y, u, and v directions of the x-u-y-v system. Substituting (23) in (22) and letting

$$\begin{aligned} W &= W_{i,j} \\ W_1 &= \min\left(W_{i-1,j}, W_{i+1,j}\right) \\ W_2 &= \min\left(W_{i,j-1}, W_{i,j+1}\right) \\ W_3 &= \min\left(W_{i-1,j-1}, W_{i+1,j+1}\right) \\ W_4 &= \min\left(W_{i+1,j-1}, W_{i-1,j+1}\right). \end{aligned} \tag{24}$$

We can rewrite the Eikonal equation for a discrete 2-D space as

$$\max\left\{\begin{array}{l} \max\left(\frac{W-W_1}{\Delta x}, 0\right)^2 + \max\left(\frac{W-W_2}{\Delta y}, 0\right)^2, \\ \max\left(\frac{W-W_3}{\Delta u}, 0\right)^2 + \max\left(\frac{W-W_4}{\Delta v}, 0\right)^2 \end{array}\right\} = \frac{1}{V^{*2}_{ij}}. \tag{25}$$

We use a simpler but less accurate solution for (25) the procedure of which can be described as follows.

First, we need decompose (25) into two subquestions and solve them, respectively, that is,

$$\max\left(\frac{W - W_1}{\Delta x}, 0\right)^2 + \max\left(\frac{W - W_2}{\Delta y}, 0\right)^2 = \frac{1}{V^{*2}_{ij}} \tag{26}$$

and

$$\max\left(\frac{W - W_3}{\Delta u}, 0\right)^2 + \max\left(\frac{W - W_4}{\Delta v}, 0\right)^2 = \frac{1}{V^{*2}_{ij}}. \tag{27}$$

Then, we can solve (26) in three steps. First, we solve the following quadratic:

$$\left(\frac{W - W_1}{\Delta x}\right)^2 + \left(\frac{W - W_2}{\Delta y}\right)^2 = \frac{1}{V^{*2}_{ij}}. \tag{28}$$

If $W > W_1$ and $W > W_2$ [taking the greater value of W when solving (28)] the obtained value is the correct solution for (26). Otherwise, if $W < W_1$ (or $W < W_2$), from (26) the corresponding member of $((W - W_1)/\Delta x, 0)$ [or $((W - W_2)/\Delta y, 0)$] is 0, and hence, (26) is reduced to

$$\begin{aligned} \frac{W - W_1}{\Delta x} &= \frac{1}{V^*} \\ \frac{W - W_2}{\Delta y} &= \frac{1}{V^*} \end{aligned} \tag{29}$$

depending on the final value of W.

We denote the solution of (26) as W'. Similar to solving (26), we obtained the solution of (27), and denote it as W''.

Finally, comparing W' and W'', let $W = \min(W', W'')$ denote the solution of (25).

Then we have the following Algorithms 2 and 3.

2) Online Search: Next, we will depict how to perform the ONSA. ONSA shares the similar inspiration as the A* algorithm [10]. However, ONSA may exist the following two differences.

1) It leverages the knowledge provided by the above OFSA to guide the search direction rather than adopt a strait-line distance in the A* algorithm.

Algorithm 2 Construction of a Weighted Cost Distribution Map (CWCDM)

```
Input: start point START, goal point GOAL, velocity of UAV VELOCITY, safety index map distribution SIM, weighted factor α
A = set containing GOAL
cost[START] = 0.0
NarrowBand = set containing neighbors of START
for node in NarrowBand do
    if node is not FORBIDDEN then
        V* = VELOCITY / (α * SIM[node] * VELOCITY + (1 - α))
        cost[node] = distance_between(node, GOAL) / V*
    end if
end for
FarAway = set containing all other points
for node in FarAway do
    cost[node] = inf
end for
while NarrowBand is not empty and FarAway is not empty do
    best_neighbor = pop smallest cost point from NarrowBand
    add best_neighbor to A
    for node in neighbor of best_neighbor do
        if node is not FORBIDDEN then
            V* = VELOCITY / (α * SIM[node] * VELOCITY + (1 - α))
            cost[node] = distance_between(node, best_neighbor) / V*
            if node in FarAway then
                remove point from FarAway
                add point to NarrowBand
            end if
        end if
    end for
end while
Return A
```

Algorithm 3 OFSA

```
Input: start point START, goal point GOAL, cost distribution map COST generated by Algorithm 2
current = START
path = queue containing START
while current has neighbors and current != GOAL do
    current = a point with the smallest cost value in COST from eight neighbors of current
    append current to path
end while
if current == GOAL then
    Return path
else
    Return failure
end if
```

2) A search space cut down mechanism is exploited in ONSA to accelerate the search efficiency.

The ONSA utilizes a cost heuristic function [denoted by $f(\vec{s})$] to identify the order in which the search visits points in a tree, which can take the form

$$f\left(\vec{s}\right) = k\left(\vec{s}\right) + h\left(\vec{s}\right) \tag{30}$$

where $k(\vec{s})$ is the path cost from the start point to the current point. $h(\vec{s})$ denotes an admissible heuristic estimation of the path cost from the current point to the target point. For the A* algorithm, it uses the straight-line cost to the target point to compute $h(\vec{s})$ because it is physically the smallest possible cost between any two points. However, we can find that OFSA can generate the Pareto optimal paths for all other points in the space graph moving toward the target point. Therefore, we straightforwardly assign this information to $h(\vec{s})$ in this paper.

Algorithm 4 ONSA

```
Input: start point START, goal point GOAL, cost distribution map COST
CLOSED = empty set
OPEN = priority queue containing START
came_from = empty map
g[start] = 0
f[start] = COST [START]
while OPEN is not empty do
    current = pop lowest rank item from OPEN
    if current == GOAL or line from current to GOAL does not intersect with area represented by points tagged as NEW_FORBIDDEN then
        Return reconstructed reverse path queue from map came_from
    end if
    add current to CLOSED
    for neighbor in neighbors of current do
        if node is not FORBIDDEN or NEW_FORBIDDEN then
            tentative_g = g[current] + cost(current, neighbor)
            if neighbor in OPEN and tentative_g < g[neighbor] then
                remove neighbor from OPEN
            else if neighbor in CLOSED and tentative_g < g[neighbor] then
                remove neighbor from CLOSED
            else if neighbor not in OPEN and neighbor not in CLOSED then
                g[neighbor] = tentative_g
                f[neighbor] = g[neighbor] + COST [neighbor]
                add neighbor to OPEN
                came_from[neighbor] = current
            end if
        end if
    end for
end while
Return failure
```

It is known that the computational complexity of the A* algorithm is $O(N\log(N))$. However, when the geography map is large enough, i.e., N is large enough, and the start point is far from the destination point, the A* algorithm may not be suitable as an online method. Furthermore, many works have been proposed to improve the A* algorithm such that it can be used as an online method [10], [25], [26].

We develop a search space cut down mechanism in this paper. If the straight path between the current point and the target point does not appear in the dynamic threat zones generated by unexpected obstacles, then terminate the online search process. Thus, when the target point is far from the start point the search space is drastically reduced, and the execution efficiency of ONSA is significantly enhanced.

Summing up, we have the following Algorithms 4 and 5.

V. Experimental Results

A. Parameter Setting

We implement MOPP algorithm using the Python programming language on a single computer. The primary hardware

Algorithm 5 Proposed MOPP Algorithm

```
Input: start point START, goal point GOAL, origin map MAP, variance of bivariate normal distribution σx, σy, velocity of UAV VELOCITY, detected obstacles' safety margin MARGIN
SIM = SIM(MAP, σx, σy)
cost_distribution = CWCDM(GOAL, VELOCITY, SIM)
plan_path = OFSA(START, GOAL, cost_distribution)
replan_path = empty queue
actual_path = queue containing START
current = START
is_replanning = false
while not current == GOAL do
    if is_replanning is false then
        current = pop from plan_path
    else if replan_path is not empty then
        current = pop from replan_path
    else
        is_replanning = false
        plan_path = OFSA(current, GOAL, COST)
        continue
    end if
    append current to actual_path
    is_detected_new_obstacles = false
    new_obstacles = empty set
    for node in detected area of UAV do
        if node is UNDETECETED_OBSTACLE then
            tag point as DETECETED_OBSTACLE
            add point to new_obstacles
            is_detected_new_obstacles = true
        end if
    end for
    extended_new_obstacles = set containing points within distance of MARGIN of area represented by new_obstacles
    tag all points in extended_new_obstacles as NEW_FORBIDDEN
    if is_detected_new_obstacles is true then
        replan_path = ONSA(current, GOAL, COST)
        is_replanning = true
    end if
end while
if current == GOAL then
    return actual_path
else
    return failure
end if
```

configuration parameters of this computer are 4× Intel Core i5-3470 CPU @ 3.2 GHz and 3975 MB Memory. The Operation System installed at this computer is Ubuntu 16.04.1 LTS.

We design different experiments for various scenarios, i.e., synthetic scene and realistic urban scene.

The parameter setting for the synthetic scene is as follows: the standard deviation $\sigma_x = \sigma_y = 4.0$, the safety margin $d_s = 5$ m.

The realistic urban scene has the following parameter setting: the standard deviation $\sigma_x = \sigma_y = 5.0$, the UAV detecting range is 30 m, and the UAV speed is 8 m/s.

B. Performance Evaluation

In the following experiments, we evaluate the performance of MOPP method using five indicators including the average and maximum running time, the UAV trajectory, the travel time, and the total safety index. The average and maximum running time are defined as the average and maximum running time of calling the designed online search process, respectively. The UAV trajectory depicts the mission trajectory of a UAV. The definition of the travel time is the cost time of a UAV taking off at a starting location and arriving at the target location. The total safety index is defined as the summation of the safety index for all points on the obtained UAV trajectory. The larger the safety index value, the lower the safety level.

TABLE I
COORDINATES OF OBSTACLES

Known obstacle1	(0,40), (0,80), (60,80), (60,40), (55,40), (55,75), (5,75), (5,40)
Known obstacle2	(80,40), (80,80), (149,80), (149,40), (145,40), (145,75), (85,75), (85,40)
Known obstacle3	(30,60), (30,20), (115,20), (115,60), (110,60), (110,25), (35,25), (35,60)
Unexpected obstacle	(50,105), (50,135), (100,135), (100,105), (95,105),(95,130),(55,130),(55,105),(50,105)

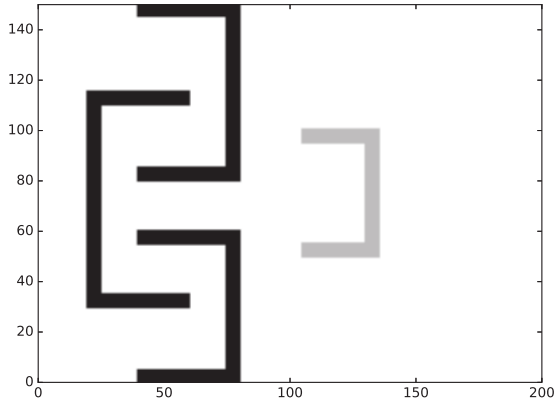


Fig. 4. Schematic of synthetic obstacles.

1) Effect of Weight Coefficient on Path Planning Through Synthetic Obstacles Scenario: This experiment is conducted under a synthetic obstacle scenario as illustrated in Fig. 4 to verify that MOPP method can be leveraged to provide path planning suggestions for users. With these recommendations, users can determine a Pareto optimal path according to their preferences.

In the setup, we generate a plan structure map of scale 150 lattices ×200 lattices; the lattice resolution is 1:1 meter. Fig. 4 shows a map of three known static obstacles with black mark and one unexpected obstacle with a gray mark (used in the following experiment). The coordinates of all vertexes of each obstacle in the map are shown in Table I. The start point and target point are marked with red "O" and red "X," respectively. The coordinate of the start point of the UAV is (5th lattice, 75th lattice) and the target point is located at (180th lattice, 60th lattice), and its speed is 1 m/s.

We call MOPP algorithm one hundred times with changed weight parameter α to obtain a travel time and safety index tradeoff curve shown in Fig. 5. Fig. 6 gives a UAV path result with the weight parameter $\alpha = 0.61$ that is a point in Fig. 5.

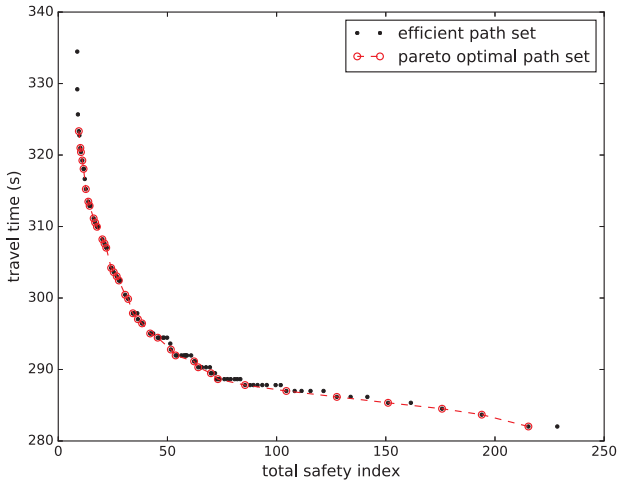


Fig. 5. Travel time and safety index tradeoff curve under a synthetic obstacles scenario.

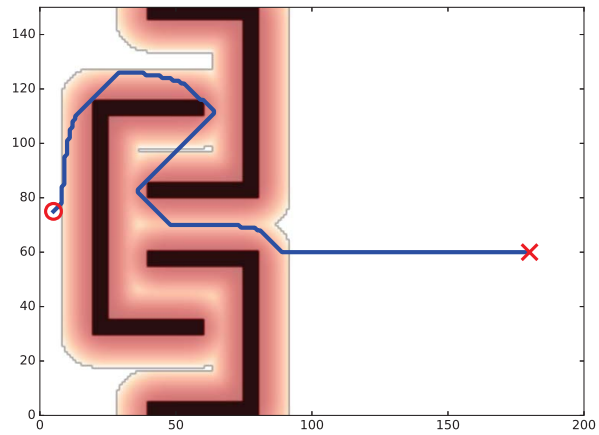


Fig. 6. Pareto path with weight parameter $\alpha = 0.61$.

In Fig. 6, the blue curve shows the actual trajectory of the UAV, and the safety index value is presented by the light red. The lighter the color, the smaller the safety index value. We can gain the following observations from these figures.

1) Shorter travel time leads to larger safety index value or higher level of danger. This is due to the contradiction between the two objectives.
2) A solution with optimal travel time and optimal safety cannot be found on the curve.
3) The travel time and safety tradeoff curve can provide references for users to make path planning decision. According to own preferences, users can select a Pareto optimal path having this curve in hand. A larger weight coefficient leads to a smaller total safety index. However, the obtained travel time alters slowly because known obstacles are in compact distribution and the distance between the start point and end point is short.
4) MOPP algorithm finds a trajectory of obstacle avoidance for the UAV. The total safety index of the trajectory is 38, and its travel time equals 296.5 s.

2) Effect of Perception Range on Path Planning Through Unexpected Obstacles Scenario: This experiment is conducted under an unexpected emerging obstacle scenario as illustrated in Fig. 4 to verify that MOPP method can effectively bypass unexpected obstacles. The setup of this experiment is the same as the previous experiment except that an unexpected obstacle is involved. The coordinates of all vertexes of the unanticipated obstacle in the map are shown in Table I. In addition, we set the weight parameter $\alpha = 0.61$.

TABLE II
RUNNING TIME OF CALLING ONSA

detecting range	average runtime	maximum runtime
10	0.1588 us	0.5998 us
30	0.1333 us	0.6227 us

Figs. 7 and 8 depict intermediate results of avoiding known and unexpected obstacles obtained by MOPP method. In these figures, the black part indicates the existence of obstacles, including known obstacles and unanticipated obstacles detected during the flight. The perceived safety margin of the unanticipated obstacle is marked with brown. Moreover, the blue dot represents the current position of the UAV, and the transparent cyan circle indicates the detection range of the UAV; the blue curve shows the actual trajectory of the UAV, and the red dotted line denotes the trajectory of the replan. Table II shows the time consumption of path replanning by calling MOPP method.

From these experimental results, we can have the following observations.

1) Contrast the results shown in Fig. 6 and Figs. 7(a) and 8(a), we can find that MOPP method generates a path that can effectively bypass known and unexpected obstacles.
2) The changing of the detection range of a UAV has no effect on the UAV path obtained by MOPP method through the static SIM. The changing range results in various paths bypass unexpected obstacles. This is because OFSA uses the complete prior knowledge of known obstacles to discover a global optimal UAV path. Once detecting unanticipated obstacles, ONSA will be called to utilize the perceived limited obstacle information to probe a local optimal UAV path. Moreover, different detection ranges incur various perceptual information.
3) When probing unexpected obstacles on its optimal path, a UAV will replan the path by executing ONSA. The red dotted line is just the replanning result. Due to the usage of the search space cut down scheme, the endpoint of the red dotted line is altering and far away from the target point.
4) Although the UAV path obtained by OFSA is changed, the direction of the path remains to point to the destination point. This is because ONSA employs the information produced by OFSA to guide the search.
5) Thanks to a larger perception range, ONSA will help a UAV escape from unanticipated obstacles earlier.
6) The running time of the replanning process is less than one microsecond. This is because the search space of ONSA is dramatically cut down.

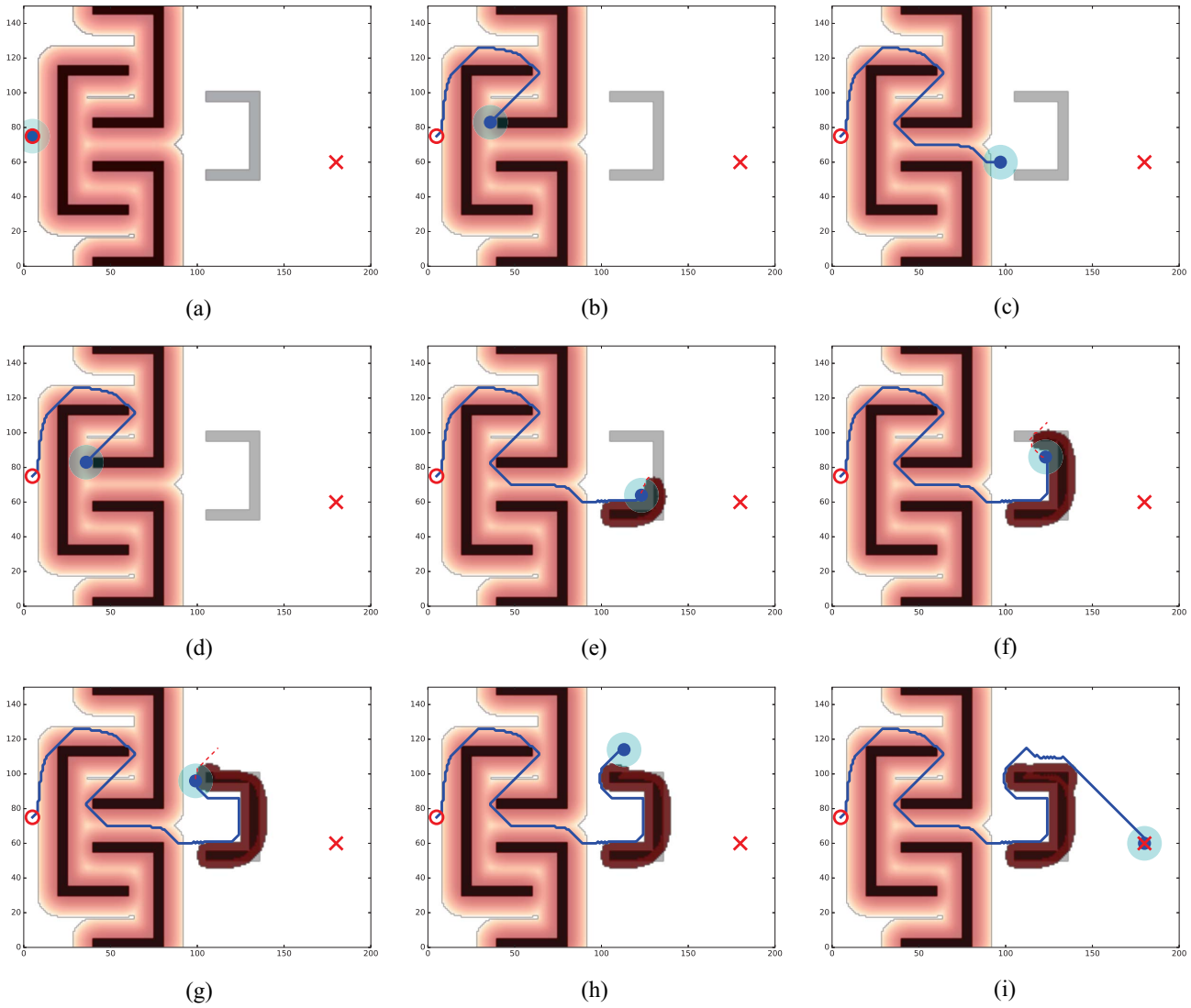


Fig. 7. Process of exploring a path with perception range equaling ten lattices. (a) Starting point. (b) Avoid the static obstacle. (c) Discover the unanticipated obstacle. (d)–(g) Search for the unanticipated obstacle. (h) Escape from the unanticipated obstacle. (i) Arrive at the end point.

7) Should we choose a large perception range for a UAV or a small one? It may depend on the realistic situation as a larger perception range may result in a more expensive UAV.

3) Effect of Weight Coefficient on Path Planning Through Realistic Obstacles Scenario: This following experiment is conducted under a realistic scenario as illustrated in Fig. 9 to further certify that the MOPP framework can be considered by users as a path selection tool. With this framework, users can choose a Pareto optimal path according to their preferences. This scenario is a region on the Satellite Map of Tiantong-zhongyuan west, Changping District, Beijing (N40° 04', E116° 25'). In the setup, we divide the map into disjointed lattices the resolution of which is 1:1.13 m and the outlines of all buildings are extracted and marked with white lines. We set the coordinate of the starting point of the UAV be (10th lattice, 320th lattice) and the ending point be located at (317th lattice, 179th lattice).

We plot the travel time and safety index tradeoff curve of the UAV path planning problem in Fig. 10, and give the obtained UAV paths of two cases with various objective preferences in Fig. 11. From these figures we can have the following observations.

1) MOPP algorithm achieves a set of Pareto optimal paths, and a path with higher safety requirement signifies a longer path travel time.
2) MOPP algorithm provides a path planning tool for users to make route decision regarding their preferences. For example, an Amazon staff decides to delivery commodities using a UAV. When the freight traffic needed to be delivered is light he may prefer a safe delivery path but ensure that the time is less than a certain constant. Specifically, he can set an acceptable time constraint, for instance, the travel time is less than 80 s; thus, our method can provide him with a path of optimal safety that can be seen in Fig. 11(a), where the total safety index equals 13 and travel time equals 79.9 s. However, when facing with a huge amount of freight traffic he may want to finish the delivery task as soon as possible while guaranteeing the safety index is less than a constant,

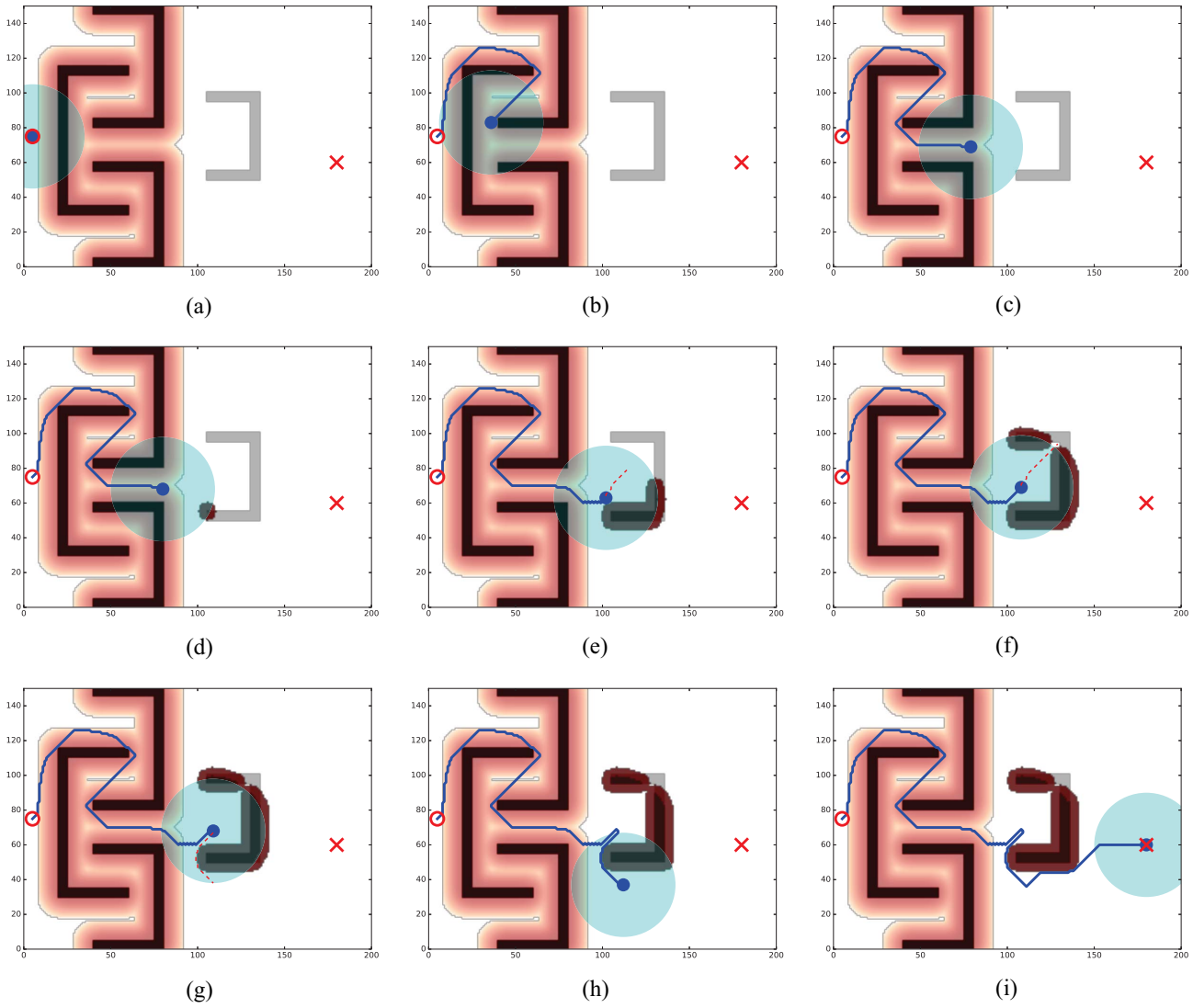


Fig. 8. Process of exploring a path with perception range equaling 30 lattices. (a) Starting point. (b) Avoid the static obstacle. (c) Escape from the static obstacle. (d) Discover the unanticipated obstacle. (e)–(g) Search for the unanticipated obstacle. (h) Escape from the unanticipated obstacle. (i) Arrive at the end point.

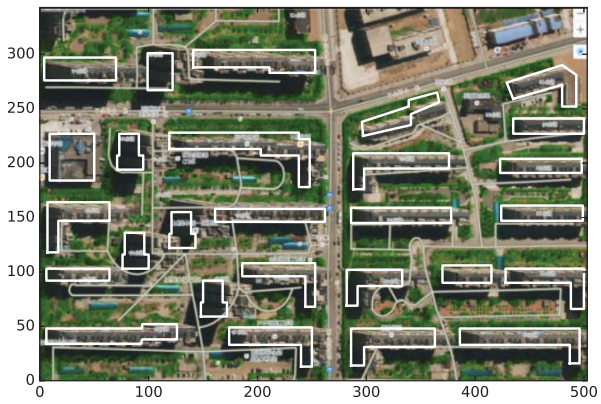


Fig. 9. Schematic of realistic obstacles.

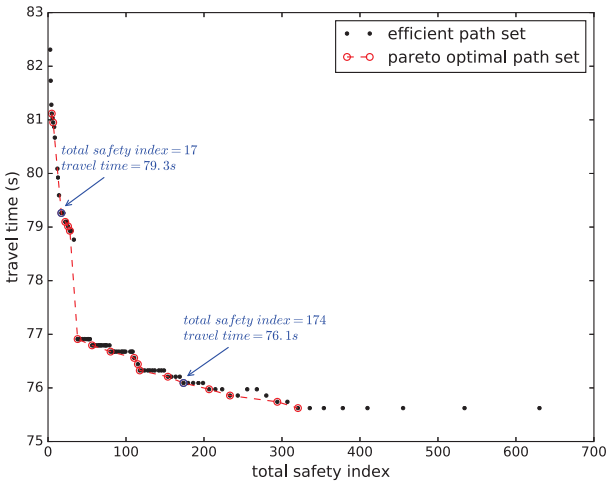


Fig. 10. Travel time and safety index tradeoff curve under a realistic scenario.

e.g., 80. Our method can also satisfy his requirement and provides him with a Pareto optimal path that can be found in Fig. 11(b), where the total safety index is 74 and travel time is 76.6 s.

3) Fig. 11 reflects that when users prefer a high safety level, our method can supply a set of Pareto paths that are far away from surfaces of obstacles and narrow

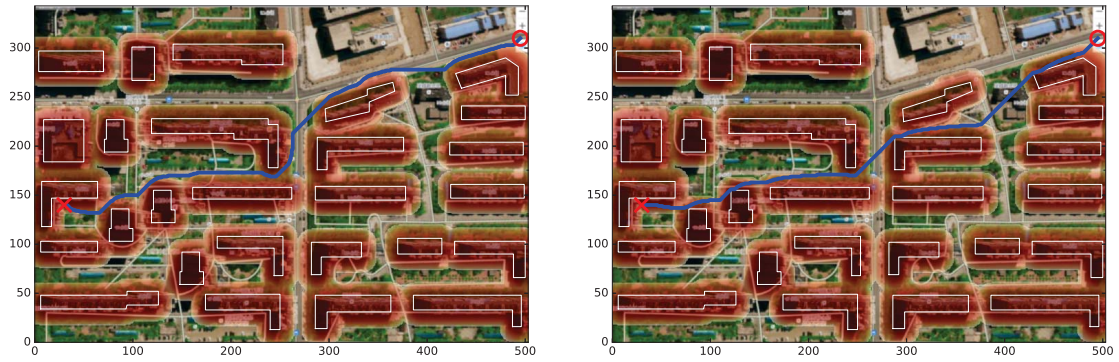


Fig. 11. Obtained paths versus different weight coefficient values. Path with weight coefficient (a) $\alpha = 0.89$ and (b) $\alpha = 0.22$.

passageways. However, when users pursuit short travel time, our method can also offer a set of Pareto optimal paths that pass through narrow passageways and turn close to obstacles.

In summary, all the experimental results demonstrate that the MOPP framework can always supply effective path planning suggestions for UAVs under different cases, and can effectively solve the problem of static and unanticipated obstacle avoidance, and the MOPP framework is a valid selection for a solution of UAV path planning flying through low altitude dynamic urban space.

VI. Conclusion

This paper is concerned with the path planning problem for UAV under dynamic urban environment. An MOPP framework concerning two objectives, namely travel time and safety level, has been proposed. In particular, a static SIM is offline established to indicate the main static obstacles in the geography map, and a dynamic SIM is online constructed to capture unexpected obstacles that are not available in the geography map during flying. Then a joint offline and online search method has been developed to address the MOPP problem, where the offline search is exploited to find the shortest path with static obstacle avoidance based on the static SIM, and the online search is exploited to quickly bypass unexpected obstacles according to the temporarily constructed dynamic SIM. The results of synthetic and realistic experiments show that an effective path, which meets the safety requirement and meanwhile has a short travel time, can be always suggested by the proposed framework under different cases.

References

[1] FAA, "FAA unveils plan for domestic drone takeoff by 2015," *Time*, 2013. http://swampland.time.com/2013/11/07/faa-clears-drones-for-domestic-take-off

[2] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Unmanned aerial vehicle with underlaid device-to-device communications: Performance and tradeoffs," *IEEE Trans. Wireless Commun.*, vol. 15, no. 6, pp. 3949–3963, Jun. 2015.

[3] A. Al-Hourani, S. Kandeepan, and S. Lardner, "Optimal LAP altitude for maximum coverage," *IEEE Wireless Commun. Lett.*, vol. 3, no. 6, pp. 569–572, Dec. 2014.

[4] R. I. B. Yaliniz, A. El-Keyi, and H. Yanikomeroglu, "Efficient 3-D placement of an aerial base station in next generation cellular networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kuala Lumpur, Malaysia, 2016, pp. 1–5.

[5] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. J. Robot. Res.*, vol. 5, no. 1, pp. 90–98, 1986.

[6] I. Ulrich and J. Borenstein, "Vfh/sup */: Local obstacle avoidance with look-ahead verification," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, vol. 3. San Francisco, CA, USA, 2000, pp. 2505–2511.

[7] N. S. V. Rao, N. Stoltzfus, and S. S. Iyengar, "A 'retraction' method for learned navigation in unknown terrains for a circular robot," *IEEE Trans. Robot. Autom.*, vol. 7, no. 5, pp. 699–707, Oct. 1991.

[8] J. Minguez and L. Montano, "Extending collision avoidance methods to consider the vehicle shape, kinematics, and dynamics of a mobile robot," *IEEE Trans. Robot.*, vol. 25, no. 2, pp. 367–381, Apr. 2009.

[9] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robot. Autom. Mag.*, vol. 4, no. 1, pp. 23–33, Mar. 2002.

[10] P. P.-Y. Wu, D. Campbell, and T. Merz, "Multi-objective four-dimensional vehicle motion planning in large dynamic environments," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 41, no. 3, pp. 621–634, Jun. 2011.

[11] V. Roberge, M. Tarbouchi, and G. Labonte, "Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning," *IEEE Trans. Ind. Informat.*, vol. 9, no. 1, pp. 132–141, Feb. 2013.

[12] L. Hernández-Hernández, A. Tsourdos, H.-S. Shin, and A. Waldock, "Multi-objective UAV routing," in *Proc. Int. Conf. Unmanned Aircraft Syst.*, Orlando, FL, USA, 2014, pp. 534–542.

[13] H. Tao, Z. Wang, and J. Li, "Three-dimensional path planning for unmanned aerial vehicles based on multi-objective genetic algorithm," in *Proc. Chin. Control Conf.*, Nanjing, China, 2014, pp. 8617–8621.

[14] M. Niendorf, P. T. Kabamba, and A. R. Girard, "Stability analysis of multi-objective planning problems for unmanned aircraft," in *Proc. IEEE Conf. Decis. Control*, Osaka, Japan, 2015, pp. 7238–7243.

[15] Y. Liu, R. Lv, X. Guan, and J. Zeng, "Path planning for unmanned aerial vehicle under geo-fencing and minimum safe separation constraints," in *Proc. IEEE 12th World Congr. Intell. Control Autom. (WCICA)*, Guilin, China, 2016, pp. 28–31.

[16] N. Wen, L. Zhao, X. Su, and P. Ma, "UAV online path planning algorithm in a low altitude dangerous environment," *IEEE/CAA J. Autom. Sinica*, vol. 2, no. 2, pp. 173–185, Apr. 2015.

[17] X. Peng, D. Xu, and F. Zhang, "UAV online path planning based on dynamic multiobjective evolutionary algorithm," in *Proc. 30th Chin. Control Conf. (CCC)*, Yantai, China, 2011, pp. 5424–5429.

[18] L. Mandow and J. L. P. D. L. Cruz, "Multicriteria heuristic search," *Eur. J. Oper. Res.*, vol. 150, no. 2, pp. 253–280, 2003.

[19] C. Petres *et al.*, "Path planning for autonomous underwater vehicles," *IEEE Trans. Robot.*, vol. 23, no. 2, pp. 331–341, Apr. 2007.

[20] W. F. Phillips, *Mechanics of Flight*, vol. 10. Hoboken, NJ, USA: Wiley, 2009, pp. 181–190.

[21] S. Thrun, *Probabilistic Robotics*. Cambridge, MA, USA: MIT Press, 2005.

[22] D. Rathbun, S. Kragelund, A. Pongpunwattana, and B. Capozzi, "An evolution based path planning algorithm for autonomous motion of a UAV through uncertain environments," in *Proc. Digit. Avionics Syst. Conf.*, vol. 2. Irvine, CA, USA, 2002, pp. 8D2-1–8D2-12.
[23] K. D. McDonald *et al.*, "The modernization of GPS: Plans, new capabilities and the future relationship to Galileo," *Positioning*, vol. 1, no. 3, pp. 1–17, 2002.
[24] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton–Jacobi formulations," *J. Comput. Phys.*, vol. 79, no. 1, pp. 12–49, 1988.
[25] P. E. Hart, N. J. Nilsson, and B. Raphael, "Correction to 'a formal basis for the heuristic determination of minimum cost paths,"' *ACM SIGART Bull.*, vol. 4, no. 37, pp. 28–29, 1972.
[26] A. Kushleyev and M. Likhachev, "Time-bounded lattice for efficient planning in dynamic environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, Kobe, Japan, 2009, pp. 1662–1668.

Chao Yin received the B.S. degree from Beihang University, Beijing, China, in 2015, where he is currently pursuing the M.S. degree.

His current research interest includes unmanned aerial vehicle networks.

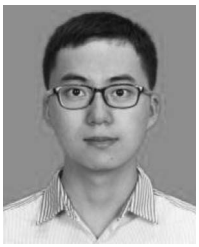
Zhenyu Xiao (M'11–SM'17) received the B.E. degree from the Department of Electronics and Information Engineering, Huazhong University of Science and Technology, Wuhan, China, in 2006, and the Ph.D. degree from the Department of Electronic Engineering, Tsinghua University, Beijing, China, in 2011.

From 2011 to 2013, he held a Post-Doctoral position with the Electronic Engineering Department, Tsinghua University. From 2013 to 2016, he was a Lecturer with the Department of Electronic and Information Engineering, Beihang University, Beijing, where he is currently an Associate Professor. He has authored or co-authored over 50 papers and served as a Reviewer for the IEEE TRANSACTIONS ON SIGNAL PROCESSING, the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, and IEEE COMMUNICATIONS LETTERS. His current research interests include communication signal processing and practical system implementation for wideband communication systems, which covers synchronization, multipath signal processing, diversity, and multiple antenna technology. He is currently dedicated to millimeter-wave communications and UAV networks.

Dr. Xiao has been a TPC member of IEEE GLOBECOM'12, IEEE WCSP'12, and IEEE ICC'15.

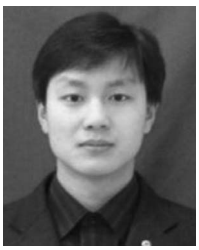
Xianbin Cao (M'08–SM'10) received the B.S. and M.S. degrees in computer applications and information science from Anhui University, Hefei, China, in 1990 and 1993, respectively, and the Ph.D. degree in information science from the University of Science and Technology of China, Hefei, China, in 1996.

He is currently a Professor with the School of Electronic and Information Engineering, Beihang University, Beijing, China. He is also the Director of the Laboratory of Intelligent Transportation System, Cambridge, MA, USA. His current research interests include intelligent transportation systems, airspace transportation management, and intelligent computation.



Xing Xi received the B.S. degree from Beihang University, Beijing, China, in 2016, where he is currently pursuing the Ph.D. degree.

His current research interests include intelligent transportation systems and next-generation mobile cellular systems.



Peng Yang is currently pursuing the Ph.D. degree in information and communication engineering at Beihang University, Beijing, China.

His current research interests include intelligent transportation systems, Internet of Things and sensor networks, next-generation mobile cellular systems, and unmanned aerial vehicle networks.

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

He is a Professor with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL, USA. His current research interests include networking, communications, signal processing, computer vision, machine learning, smart grids, and information and network security.

Dr. Wu was a recipient of the University of Florida Term Professorship Award in 2017, the University of Florida Research Foundation Professorship Award in 2009, the AFOSR Young Investigator Program (YIP) Award in 2009, the ONR YIP Award in 2008, the NSF CAREER Award in 2007, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY Best Paper Award in 2001, and the Best Paper Awards of IEEE GLOBECOM 2011 and International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks (QShine) 2006. He currently serves as the Editor-in-Chief of the IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING and an Associate Editor of the IEEE TRANSACTIONS ON COMMUNICATIONS, the IEEE TRANSACTIONS ON SIGNAL AND INFORMATION PROCESSING OVER NETWORKS, and *IEEE Signal Processing Magazine*. He was the Founding Editor-in-Chief of the *Journal of Advances in Multimedia* from 2006 to 2008 and an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, and the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY. He is also a Guest-Editor for the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS "Special Issue on Cross-Layer Optimized Wireless Multimedia Communications." He has served as a Technical Program Committee Chair for IEEE INFOCOM 2012 and the TPC Chair for IEEE International Conference on Communications (ICC 2008), Signal Processing for Communications Symposium, and a member of the Executive Committee and/or Technical Program Committee of over 80 conferences. He was elected as a Distinguished Lecturer by the IEEE Vehicular Technology Society in 2016. He has served as the Chair for the Award Committee and the Chair of the Mobile and Wireless Multimedia Interest Group, Technical Committee on Multimedia Communications, and the IEEE Communications Society. He was an elected member of the Multimedia Signal Processing Technical Committee and IEEE Signal Processing Society from 2009 to 2012.