# A Hybrid Path Planning Method in Unmanned Air/Ground Vehicle (UAV/UGV) Cooperative Systems

Jianqiang Li, Genqiang Deng, Chengwen Luo, Qiuzhen Lin, Qiao Yan, and Zhong Ming

***Abstract*—In this paper, we study the automatic ground map building and efficient path planning in unmanned aerial/ground vehicles (UAV/UGV) cooperative systems. Using the UAV, a ground image can be obtained from the aerial vision, which is then processed with image denosing, image correction, and obstacle recognition to construct the ground map automatically. Image correction is used to help the UGV improve the recognition accuracy of obstacles. Based on the constructed ground map, a hybrid path planning algorithm is proposed to optimize the planned path. A genetic algorithm is used for global path planning, and a local rolling optimization (LRO) is used to constantly optimize the results of the genetic algorithm. Experiments are performed to evaluate the performance of the proposed schemes. The evaluation results show that our proposed approach can obtain much less costly path compared to the traditional path planning algorithms such as genetic algorithm and A star algorithm and can run in real-time to support the UAV/UGV systems.**

***Keywords*—Unmanned air vehicle, unmanned ground vehicle, obstacle identification, hybrid path planning.**

## I. Introduction

RECently, applications of unmanned aerial/ground vehicles (UAV/UGV) in civilian and military settings have gained great interests from both academia and industry [1], [2]. UAV/UGV systems can be used in a variety of applications, such as surveillance, navigation, crowd control, sensing and emergency rescue operations in the regions where it is difficult or dangerous for human beings to access. UAV and UGV have different characteristics. While a UAV can search a wide range of targets, the height and speed of the UAV limit its precise positioning of a target. By contrast, a UGV can have accurate locations of a target on the ground. Nevertheless, a UGV moves slowly and is easily obscured by obstacles. To solve the deficiencies of homogeneous unmanned vehicles, UAV and UGV should work cooperatively [2]–[4]. A typical UAV/UGV cooperative system is shown in Fig. 1, where UAV and UGV exchange information to achieve cooperations.

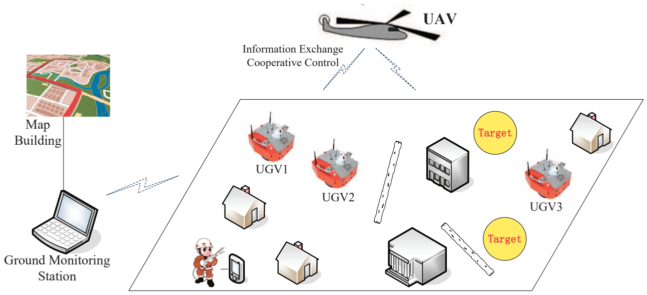


Fig. 1. A UAV/UGV cooperative system.

Despite the potential vision of UAV/UGV cooperative systems, the existing UAV/UGV systems are far from meeting the requirements, such as response speed, system stability, etc., due to the complexity of the practical environments and tasks [5], [6]. There are many issues that need to be addressed before UAV/UGV systems can be widely deployed in practical applications. First, the map building is not accurate in most existing UAV/UGV systems. This is a critical issue, because the first functional step in UAV/UGV systems is usually to build an accurate map in order to implement further operations. One of the reasons for inaccurate map building is that the aerial image taken by a UAV can be easily effected by the weather or other factors. In addition, obstacle recognition is difficult in complex environments. For example, the recognition of obstacles will be difficult in heavy fog surroundings. Second, the computation of recognizing obstacles is extremely complex, which will result in long computation time. Third, the cooperative control between UAV and UGV can be easily affected by unreliable wireless communications due to packet drops and delays. Based on the above reasons, UAV/UGV cooperative systems need further research in the areas of multi-tasking heterogeneous vehicles, environment map building and path planning, cooperative control, multi-sensor data fusion, intelligent operations, and practical applications.

In this paper, we study the environment map building and path planning issues in UAV/UGV cooperative systems. We consider a typical disaster-rescue application of UAV/UGV systems, where the main goal of UAV/UGV systems is to build a ground map and plan an efficient and feasible path for disaster rescue. The main contributions of this paper are as follows.

Jianqiang Li, Genqiang Deng, Chengwen Luo, Qiuzhen Lin, Qiao Yan and Zhong Ming are with College of Computer Science and Software Engineering, Shenzhen University. (E-mail: lijq@szu.edu.cn, denggenqiang@gmail.com, chengwen@szu.edu.cn, qiuzhlin@szu.edu.cn, yanq@szu.edu.cn, mingz@szu.edu.cn).

The corresponding author is Chengwen Luo.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2016.2623666, IEEE Transactions on Vehicular Technology

- Using the UAV, we can obtain a ground image from the aerial vision. Then, we process the ground image with image denosing, image correction, and obstacle recognition. Most existing works on UAV/UGV systems do not use image correction, which will affect obstacle recognition. In this paper, we add image correction to help the UGV improve the recognition accuracy of obstacles. Subsequently, the obstacles can be avoided and a more feasible path can be obtained.
- Based on the ground map information, we propose a hybrid path planning algorithm to optimize the distance of the path. A genetic algorithm is used for global path planning, and rolling optimization is used to constantly optimize the results of the genetic algorithm.
- Experiments are performed to evaluate the performance of the proposed schemes. Our proposed schemes can obtain much less costly path compared to traditional genetic algorithm and A star algorithm.

The rest of this paper is organized as follows: Section II discusses typical application of UAV/UGV, and the current related works of UAV/UGV. Section III introduces our hardware system of UAV/UGV, and introduces the whole process when UGV recieved the ground image. Section IV discusses the main steps of map building, such as classic denoising algorithms, method of image correction and obstacle identification. Section V formulates the path planning problem, and our hybrid path plan result are shown in Section VI. Section VII concludes this paper with future works.

## II. Related Works

In the field of UAV/UGV cooperative systems, most existing works focus on the target detection and switch control issues. The authors of [7] designed a switched UAV-UGV cooperation scheme for target detection, in which the simulation results show the effectiveness of the proposed scheme. A control-oriented agent framework has been designed [8]. In 2013, a research team used UGV onboard monocular vision to study autonomous takeoff, tracking and landing of a UAV [9]. The authors of [10] proposed UAV/UGV cooperation control for surveying operation in humanitarian demining. However, most of previous studies have not considered image correction in UAV/UGV cooperative systems, which will have significant impacts on the map building and practical applications of UAV/UGV cooperative systems.

In the field of map building, a variety of image algorithms have been developed in the literature. Median and morphological filters are classical image denoising algorithms [11]. Due to the simplicity and invariance to image rotation and translation, Harris corner detector [12] is one of the most popular interest point detectors. In 2009, Abdelkrim Nemra and Nabil Aouf proposed a new criterion to select the most stable features in order to improve the visual map building performance [13]. Moseley *et al.* at CMU have proposed a method in integrated long-range UAV/UGV collaborative target tracking systems [14]. In 2010, V. Ila *et al.* have proposed a method that is defined using interval arithmetic based on organizing the pose information in a balanced tree whose internal levels [15]. Compared to the monocular robot, the map building of heterogeneous robots could be more accurate.

In the field of path planning, it has become one of the key research areas in UAV/UGV cooperative systems. Traditional static path planning methods, such as Visibility Graph [16], are not efficient, and they are not suitable to be used in our project. To solve the problem of path planning, Y.G. Peng and W. Wei proposed a new path planning method based on an adaptive simulated annealing genetic algorithm in 2006 [17]. The authors of [18] proposed an improved hierarchical A-star algorithm to solve the parking path planning problem of a large park. In 2015, Ganeshmurthy and G.R.Suresh proposed a heuristic based method to search the feasible initial path efficiently to solve the problem of dynamic environments [19]. Z.X. Zhu and Q.Z. Lin proposed global path planning of wheeled robots using multi-objective algorithms [20], [21]. However, most existing methods have some limitations. For example, the computations of these methods are usually too complex, and the methods only have global path planning or local planning due to the use of monocular robots. Therefore, further study is needed to improve the performance of path planning.

In this paper, we study the environment map building and path planning issues in UAV/UGV cooperative systems. Firstly, we add image correction to improve recognition accuracy of obstacles. Secondly, we use various image process algorithm to model the map. Thirdly, we propose a hybrid path planning algorithm based on the basic architecture of the obstacles on the ground map information. This algorithm reduces the amount of obstacles in the final path. Moreover, it uses both global optimization and local optimization to plan a more feasible path, the path is shorter than the genetic algorithm.

## III. System Description

We consider a UAV/UGV cooperative system, where a UAV and a group of UGVs work cooperatively in a typical disaster-rescue scenario. The architectures of UAV and UGV are shown in Fig. 2. The UAV has a control module, a camera module and a communication module. The control module is the core module of the UAV. It is mainly responsible for flight control and data processing. The camera module is mainly responsible for recording the ground image from aerial vision, and transmitting this image to the control module. The control module transmits this image to the communication module. The communication between the robots is an important part of this systems, L. Liu proposed a biology-inspired algorithm to solve the communication problem of muti-robots [22]. Then, the communication module transmits this image to the communication module of the UGV via wireless communications. After receiving the image, the control module of the UGV will process the image, and build a ground map. According to its location and the ground map, the control module will plan a feasible

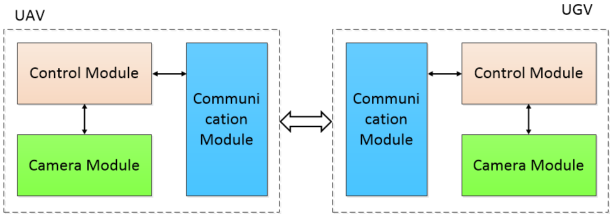


Fig. 2. System structure.

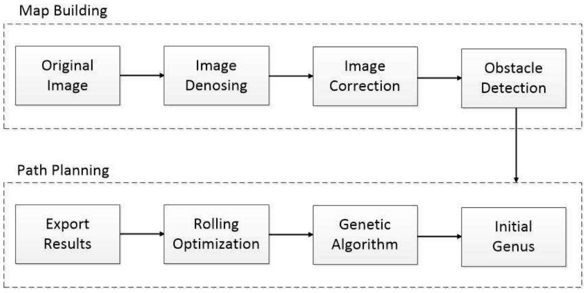


Fig. 3. Map building and path planning in the UAV/UGV cooperative system.

path to the destination point. The process is dynamic, which means that the planning path will be dynamically adjusted according to its location and the map.

Fig. 3 shows the map building and path planning processes in the UAV/UGV cooperative system. When the UAV transmits the ground image to the UGV, the first step is image denoising, because too much noise in the ground image will affect the performance of the following steps. When the UAV records the ground image in the air, the angle of a shot may not be perpendicular to the ground, so the image should be corrected. After image correction, the obstacles in the image will be detected, and a ground map is built. Based on the ground map, GPS signal and destination point, the UGV can plan a path. The first step of path planing is to use the genetic algorithm to make a feasible path. Then, local optimization is inserted in the global optimization for path planning based on the genetic algorithm. This method can update the perception of environmental variability by ground vehicles, and enhance the performance of route optimization. These processes will be detailed in the following sections.

## IV. Map Building with Aerial Vision

Map building is the foundation of path planning, which is critical for a robot to reach its target accurately. For a UGV, the map will be changed dynamically when the vehicle is moving. By contrast, for a UAV, the ground environmental information is relatively stationary. Therefore, after the UGV has collected the ground image, image processing is necessary in the UAV/UGV cooperative system. Fig. 4 shows the process of image processing, which will be described in details as follows.

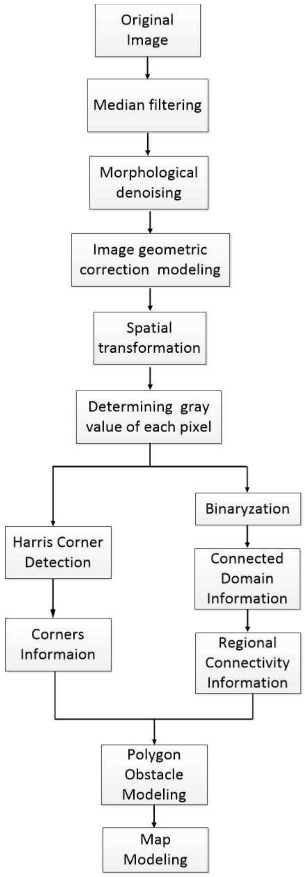

Original Image
Median filtering
Morphological denoising
Image geometric correction modeling
Spatial transformation
Determining gray value of each pixel
Harris Corner Detection
Corners Informaion
Binaryzation
Connected Domain Information
Regional Connectivity Information
Polygon Obstacle Modeling
Map Modeling

Fig. 4. Image processing in the UAV/UGV cooperative system.

### A. Image Denoising

Image noise is an inevitable problem in the process of image processing. In general, a image will be affected by various noise interference when we take an image, transmit an image or process an image. For example, in the atmospheric environment, because the light scatters in the air, the image will be affected by the grey effect, which means the value of the image contrast will be lower than the normal level, and it will influence the visual effect of the image. Therefore, before the image is further processed (such as pattern recognition, feature extraction, etc.), the first step of image processing is image denoising. In this paper, we use median filtering and morphological denoising, which are described in the following.

(1) Median filtering

Median filtering [11] is a nonlinear smoothing technique. The main idea of median filtering is to select the middle value from the sorted values. We call it the pattern of neighbors window. For 1D signals, the most obvious window is just the middle number, whereas for 2D (or higher-dimensional) signals such as images, more complex window patterns are possible (such as “box”, “circle” or “cross” patterns). Note that if the window has an odd number of

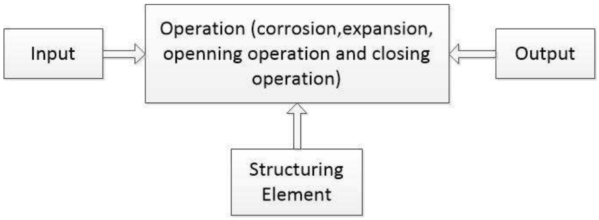

Input
Operation (corrosion,expansion, openning operation and closing operation)
Output
Structuring Element

Fig. 5. The process of morphological denoising.

entries, the median is just the middle value after all the entries in the window are sorted numerically.

(2) Morphological denoising

For an image, circular noise or block noise may exist around the target. Using median filtering may not filter the noise completely. In this case, we can use morphological denosing [23], [24] to filter out the circular noise or block noise. The method of image morphology can filter out the noise, and maintain their basic shape characteristics. The basic morphological operations are: corrosion, expansion, opening operation and closing operation. Usually, we can use one of the operations, or use a mixture of several operations in one image. The general process of the morphological denoising is shown in Fig. 5.

In this process, the structuring element is a shape, which is used to probe how this shape fits in a given image. The typical structuring elements are square, polygon, flat, circular, etc. Structuring elements can arbitrarily change the type and size of structural elements, but the size of structural elements, in general, is smaller than the size of the detected target.

Let $A(x)$ denote the structuring element, where $x$ means a point in the given image $T$. The definitions of corrosion and expansion are given as follows.

$$Corrosion : X = T \odot A = \{x : A(x) \subset T\} \tag{1}$$

$$Expansion : X = T \oplus A = \{y : A(y) \cap T \neq \emptyset\} \tag{2}$$

### B. Image Correction

In this paper, we locate the obstacles via image processing algorithms. Therefore, the requirements on the image are high. However, some common drawbacks of practical cameras usually exist. For example, the target on camera imaging can produce radial distortion [25]. If the optical axis of the camera is not vertical with the target, the target will have geometric distortion. Therefore, it will increase the difficulty to identify the obstacles. Thus, when using a digital camera to model the environment, image correction is a necessary step in image processing [26].

Geometric distortion is one of the common problems in image processing. The distortion will gradually increase from the central point to the edge of image, which means the actual location of the target will not match the image coordinates. A common strategy to the distortion correction is to apply the polynomial deformation method [27]. The main idea of this method is to use polynomial transformation between the coordinates to express the nonlinear distortion approximately, and then select some control points to eliminate distortion. If point $(x, y)$ is a point in the original image, point $(f, g)$ is the corresponding point in the corrected image. Using the method of polynomial deformation, their corresponding relationship is:

$$\begin{cases} x = \sum\limits_{i=0}^{n}\sum\limits_{j=0}^{n-i} a_{ij} f^i g^j \\ y = \sum\limits_{i=0}^{n}\sum\limits_{j=0}^{n-i} b_{ij} f^i g^j \end{cases} \tag{3}$$

In $(3)$, $a_{ij}$ and $b_{ij}$ are coefficients of polynomial, and $n$ means the times. Eq. $(3)$ can use the known coordinates of control points to solve the image distortion problem.

If the number of control points is equal to the number of unknown variables in $(3)$, we can use $(3)$ directly to solve the image distortion problem. However, in general correction processing of image distortion, in order to obtain high accuracy of correction, the number of control points is always more than the number of unknown variables in equations. It seems that such equations may be contradictory, but we can attain the optimal approximate solution, which is described as follows.

If the number of control points is $L$, the quadratic sum of polynomial fitting error is minimum, i.e.,

$$\varepsilon = \sum_{l=0}^{L} \left[ x - \sum_{i=0}^{n}\sum_{j=0}^{n-i} a_{ij} f_l^i g_l^j \right]^2 = min \tag{4}$$

and

$$\frac{\partial \varepsilon}{\partial \alpha_{st}} = 2\sum_{l=0}^{L} \left[ \sum_{i=0}^{n}\sum_{j=0}^{n-i} a_{ij} f_l^i g_l^j \right] f_l^s g_l^t = 0 \tag{5}$$

then,

$$\begin{aligned} &\sum_{l=0}^{L} \left[ \sum_{i=0}^{n}\sum_{j=0}^{n-i} a_{ij} f_l^i g_l^j \right] f_l^i g_l^j \\ &= \sum_{i=0}^{n}\sum_{j=0}^{n-i} a_{ij} \left[ \sum_{l=0}^{L} f_l^{i+s} g_l^{j+t} \right] f_l^s g_l^t \\ &= \sum_{l=0}^{L} x_l f_l^s g_l^t \end{aligned} \tag{6}$$

For the same reasons, we can obtain

$$\begin{aligned} &\sum_{l=0}^{L} \left[ \sum_{i=0}^{n}\sum_{j=0}^{n-i} b_{ij} f_l^i g_l^j \right] f_l^i g_l^j \\ &= \sum_{i=0}^{n}\sum_{j=0}^{n-i} b_{ij} \left[ \sum_{l=0}^{L} f_l^{i+s} g_l^{j+t} \right] f_l^s g_l^t \\ &= \sum_{l=0}^{L} y_l f_l^s g_l^t, \end{aligned} \tag{7}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2016.2623666, IEEE Transactions on Vehicular Technology

where $L$ means the number of control points, $s = 0, 1, 2, \ldots, n, t = 0, 1, \ldots, n-s$, and $s+t <= n$.

In (6) and (7), since both of them have $(n+1)(n+2)/2$, they can constitute linear equations to solve $a_{ij}$ and $b_{ij}$. Then, we can use the calculated results and (3) to solve the correct coordinates of the image point.

The accuracy of image correction is related to the polynomial's order $n$. The higher the order of polynomial, the smaller the fitting error. However, with the increase of the order $n$ and the coefficient of the polynomial, the amount of calculation will increase enormously. For general nonlinear distortion, we often use cubic polynomial fitting. This method is simple and effective, and the accuracy is high. Therefore, (3) can be written as:

$$\begin{cases} x = a_{00} + a_{01}f + a_{02}f^2 + a_{03}f^3 + a_{10}g + a_{11}fg \\ \quad + a_{12}f^2g + a_{20}g^2 + a_{21}fg^2 + a_{30}g^3 \\ y = b_{00} + b_{01}f + b_{02}f^2 + b_{03}f^3 + b_{10}g + b_{11}fg \\ \quad + b_{12}f^2g + b_{20}g^2 + b_{21}fg^2 + b_{30}g^3 \end{cases} \tag{8}$$

### C. Obstacle Identification

Mapping by obstacle identification constructing is an important technology in the field of vehicular studies, and it is the foundation for the vehicle to complete various tasks. Based on aerial vision, ground vehicles can get a global image information acquisition and build the global map.

In this paper, obstacles are approximated by a polygon model for environmental modeling, and the vehicle is regarded as a particle. So the polygon should be appropriately expanded to offset the volume of the vehicle. For the identification of the obstacles, there are threshold value of the color segmentation, connected domain and other methods. In order to extract the obstacle information positioning in the map building process well, hybrid recognition feature extraction methods based on color segmentation and connected domain detection are proposed to improve the recognition accuracy for the obstacles.

Based on aerial vehicle visual panoramic images, color segment and connected domain are detected at first, and the feature of corner points is extracted to get the information of the polygon. Identification and modeling of obstacles are shown as below. Image segmentation should be done before connected domain detection. As the obstacles are connected domains, and the pixel values difference is very small. So the target points are detected based on rough estimate in the range of colors of the obstacles. Each pixel of the image is detected, and the threshold condition is considered to be the target point: $R_{min} \leq R \leq R_{max}$, $R_{min}$ and $R_{max}$ are the minimum threshold and maximum threshold for the pixel intensity values, respectively. Then split image binarization processes and the target points are recorded as white. Fig. 4 shows the process, and the details are described as follows.

(1) Target points are detected by connected domains detection. Each connected domain is divided and marked. As small obstacles should not be regarded as obstacles, the threshold is set as a pixel. Then the value of the connected domain is not regarded as obstacles if it is less than this threshold.

i) After binarization, the next step is the progressive scan of the image. When encountered on a black point and it is made a note as this new segment point. Then it continues to scan until the next point is black, and the point is marked as the end of the segments. The entry and exit points of the line segment are deposited into line segment collection.

ii) Traverse the collection of the current scan line and the previous scan line segments, determine the relative position between the two segments. Incorporated into the same connectivity domain when the existence of the intersection, otherwise create a new connected domain or add to the tail connectivity domain.

iii) When the line is scanned completely, check the segment of the tail pointer to determine whether or not to add to the known connectivity of the domain or create a new connected domain.

(2) Extract the features within the obstacles connectivity and get feature information for obstacles. The main purpose of regional connectivity feature extraction is to identify corners and edges of obstacles, describe the area of obstacles, and build a global map that will be used in the following steps. The obstacle areas are described to build a global map. In this paper, Harris corner detection proposed by C. Harris and J. Stephens Harris is used to extract operator based on the signal point feature extraction operator [25]. This operator is inspired by the autocorrelation function of signal processing, and linked to the autocorrelation function matrix $M$. The eigenvalue of matrix $M$ is the first-order curvature of the autocorrelation function. The point is regarded as the feature point if the two curvature values are high. The specific process is shown as follows:

i) Each pixel of the image is filtered to obtain $I_x$ and $I_y$ using the horizontal and vertical difference operator. $I_x$ and $I_y$ are calculated by $3X3$ neighborhood calculation using the difference of the point and surrounding points on a single direction (horizontal & vertical axis), and the matrices are usually taken as:

$$I_X : \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, I_Y : \begin{bmatrix} -1 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \tag{9}$$

The four elements of matrix $M$ are calculated as

$$M = \begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix}, I_y^2 = I_yI_y, I_x^2 = I_xI_x \tag{10}$$

ii) The four elements of matrix $M$ are Gaussian smoothing filtered, then a new $M$ is obtained. The discrete two-dimensional zero-mean Gaussian function is:

$$Gauss = exp\left(-\frac{(X^2+Y^2)}{2\sigma^2}\right) \tag{11}$$

iii) Calculate each corner pixel of $R$ according to $M$:

$$R = \frac{I_x^2 * I_y^2 - (I_x I_y)^2}{I_x^2 + I_y^2} \tag{12}$$

iv) Finally, the points are considered as corners while $R$ is greater than the threshold thresh, and $R$ is the local maxima within a certain field.

(3) Combination of connected domain with corner points, the basic characteristics of the obstacles are obtained. Then, we can connect the corresponding corner in order, and build the map of the obstacles by polygon description on this basis.

## V. Path Planning

In the above section we described the automatic obstacle identification and map building in UAV/UGV systems. In this section, we utilize the constructed maps and introduce the path planning algorithm. We first formulate the path planning problem. Then, we propose a hybrid path planning algorithm in UAV/UGV cooperative systems.

### A. Problem Modeling and Formulation

In the UAV/UGV path planning, we consider a 2-dimensional rectangular area where the UGV can move around. In the area, there are several obstacles. The obstacles are abstracted as hexagons. Each hexagon has multiple vertexes. Suppose we have $N$ obstacles, which are represented by $O$, and $M$ vertexes, which are represented by $V$.

Based on the abstraction, a vehicle's path is defined as a sequence of selected vertexes in the area passed by the UGV. Assume that the first of the sequence is the source of the path, and the last is the destination. Between any two adjacent vertexes in the sequence, the vehicles move in a straight line. If the line connecting any two adjacent vertexes interconnect with obstacles, then the path is not valid. We define the length of a path as the total Euclid distance of the connective lines along the path. The path planning problem is to find a valid and shortest path from particular source to destination.

We can convert the path planning problem to the classic graph theory problem. In graph theory, vertex and edge are essential factors. In the last section, we have used image algorithm to detect the obstacle's vertexes, and have obtained a ground map. We assume the obstacle's vertexes are the vertexes in graph theory, any two vertexes can use a straight line to connect each other. If the line and any obstacles have no point of intersection, we can define this line as an edge. If this line and any obstacles have point of intersection, the line cannot become an edge.

Fig. 6 shows an example ground map. In this map, obstacles have been described as polygons, and every polygon has vertexes. We can transfer the image to a graph, which is shown in Fig. 7. In Fig. 7, the red point represents a vertex of an obstacle, and the black line represents an edge. Every edge is a path for the UGV.

The formulation of the problem is defined as follows.

**Given:**

1. Vertexes: $V = \{v_1, v_2 \cdots v_i \cdots v_t\}, 1 \le i \le t, t \in R$

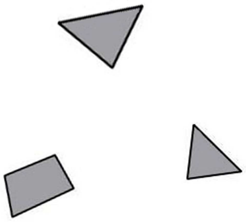


Fig. 6. The original map.

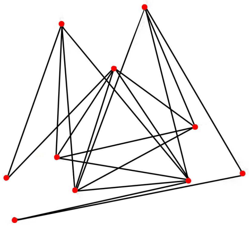


Fig. 7. The feasible path transferred from the original map.

2. Obstacles: $O = \{o_1, o_2 \cdots o_i \cdots o_n\}, 1 \le i \le n, n \in R$
3. Source point: $v_s$, destination point: $v_d$.
4. Path between vertex $i$ and vertex $j$: $\vec{s}(v_i, v_j)$
5. Path's length between vertex $i$ and vertex $j$: $|\vec{s}|(v_i, v_j)$.

**Objective:**

Our objective is to minimize path's length connecting source point and destination points. In this path, besides the source point and the destination point, we assume the final planned path has $k$ points, so the final sequence of vertexes of the path is : $v_s, v_1, v_2, v_3 \cdots v_i \cdots v_k, v_d$. So the objective of the path planning is:

$$min : f(\vec{s}(v_s, v_i)) = |\vec{s}|(v_s, v_i) + \sum_{i=1}^{k-1} |\vec{s}|(v_i, v_{i+1}) + |\vec{s}|(v_k, v_d) \tag{13}$$

**Constraints:**

In UAV/UGV systems, one critical issue is making sure the UGV will not collide with the obstacles in th path, therefore we have the following constraints in the formulation:

$$\begin{cases} \forall i, \vec{s}(v_i, v_{i+1}) \bigcap O = \emptyset, 1 \le i \le k-1; \\ \vec{s}(v_s, v_1) \bigcap O = \emptyset \\ \vec{s}(v_k, v_d) \bigcap O = \emptyset \end{cases} \tag{14}$$

### B. Hybrid Path Planning Algorithm

With the automatically constructed ground map from the image processing functionality in UAV/UGV systems, we propose a hybrid path planning algorithm. A genetic algorithm is used for global path planning, and the local rolling optimization is used to constantly optimize the results of the genetic algorithm. Before introducing the hybrid path planning algorithm, we first introduce the following definitions [28]:

*Definition 1: (Pareto-Dominance)*: A generated path (solution) $\vec{s}(v_i, v_j)$ connecting the vertex $v_i$ and $v_j$ is said *dominate* another path $\vec{s}'(v_i, v_j)$ (denote $\vec{s}(v_i, v_j) \succ \vec{s}'(v_i, v_j)$) iff $f(\vec{s}(v_s, v_i)) < f(\vec{s}'(v_s, v_i))$.

*Definition 2: (Pareto-Optimal)*: A generated path (solution) $\vec{s}(v_s, v_i)$ connecting the vertex $v_i$ and $v_j$ is said to be nondominated (Pareto-optimal) iff:

$$\neg \exists \vec{s}'(v_s, v_i) \in S(v_i, v_j) : \vec{s}'(v_i, v_j) \succ \vec{s}(v_i, v_j).$$

where $S(v_i, v_j)$ is the space of all possible paths connecting $v_i$ and $v_j$.

The ultimate goal of the path planning algorithms is to find the Pareto-Optimal path that minimizes the traveling time of the UGV. The main flow of the proposed hybrid path planning algorithm is shown in Algorithm 1. Several important operators are involved, including *initialization*, *evaluation*, *selection*, *crossover*, and *mutation*. In this section we provide a detailed description to each operator.

1) *Initialization*. For starting vertex $v_i$ and destination vertex $v_j$, generate the initial population $A(v_i, v_j) = \{\vec{s_1}(v_i, v_j), \vec{s_2}(v_i, v_j), ..., \vec{s_m}(v_i, v_j)\}$ with $m$ solutions generated randomly. The initial population becomes the parents for next generations in the evolutionary algorithm. Each path $\vec{s}(v_i, v_j) =< v_i, v_1, v_2, ..., v_k, v_j >$ contains the intermediate vertex covered by the path and is termed as the *individual* in the evolutionary process.

2) *Evaluation*. The fitness function $F(\vec{s}(v_i, v_j)$ represents the important evolutionary process using the hybrid path planning algorithm. The fitness function measures the quality of the individual and is defined as:

$$F(\vec{s}(v_i, v_j) = \begin{cases} \frac{1}{f(\vec{s}(v_s, v_i))}, & I(\vec{s}(v_i, v_j), O) = \emptyset \\ 0, & otherwise \end{cases} \tag{15}$$

where $f(\vec{s}(v_s, v_i))$ is the objective function defined in Equation (13), and function $I(\vec{s}(v_i, v_j), O)$ returns the empty set if the current path has no intersection with the obstacles and returns the intersection coordinates otherwise. $F(\vec{s}(v_i, v_j)$ defines the properties that the optimal path should have and eliminates those with poor fitness. In our case, the optimal path should have the shortest distance and has no intersection with obstacles detected by the UAV vision.

3) *Selection*. The individuals are selected based on the Tournament selection strategy. Each time a set of $k$ individuals are selected to form a selection pool: $\{\vec{s_p}(v_i, v_j), s_{\vec{p+1}}(v_i, v_j), ..., s_{\vec{p+k}}(v_i, v_j)\}$. The set is sorted in the descending order based on the fitness of each individual: $F(\vec{s_p}(v_i, v_j)), F(s_{\vec{p+1}}(v_i, v_j)), ..., F(s_{\vec{p+k}}(v_i, v_j))$. The sorted set of individuals in the selection pool with new indexes becomes $\{\vec{s'_0}(v_i, v_j), \vec{s'_1}(v_i, v_j), ..., \vec{s'_{k-1}}(v_i, v_j)\}$. For the $i$-th individual in the sorted set, the probability $p_s^{(i)}$ of being selected is:

$$p_s^{(i)} = p_s * (1 - p_s)^i \tag{16}$$

where $p_s$ is the selection rate parameter controlling the rate of decay in the selection process. The process continues until the number of selected individual reaches the predefined size.

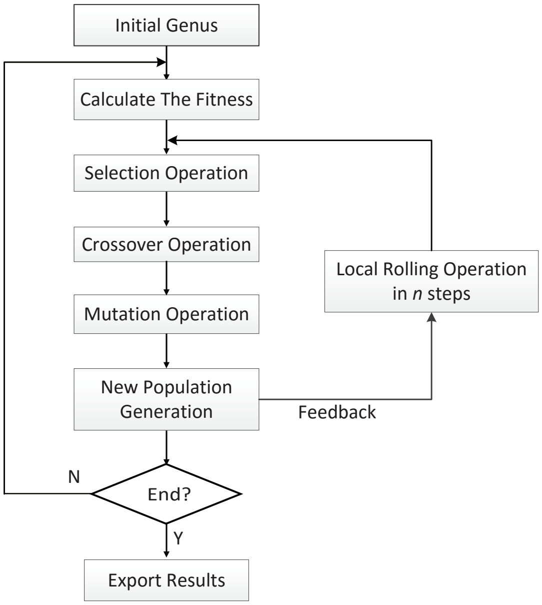


Fig. 8. The proposed path planning algorithm.

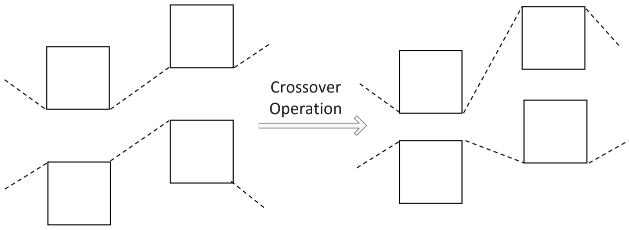


Fig. 9. Crossover operation.

4) *Crossover*. The uniform crossover operator [29] is used to produce new individual from a pair of parents. For each pair of selected parents $\vec{s_1}(v_i, v_j) =< v_i^{(1)}, v_1^{(1)}, v_2^{(1)}, ..., v_k^{(1)}, v_j^{(1)} >$, $\vec{s_2}(v_i, v_j) =< v_i^{(2)}, v_1^{(2)}, v_2^{(2)}, ..., v_k^{(2)}, v_j^{(2)} >$, each vertex of the generated path $\vec{s_c}(v_i, v_j) =< v_i^{(c)}, v_1^{(c)}, v_2^{(c)}, ..., v_k^{(c)}, v_j^{(c)} >$ after the crossover operation is randomly selected from the parents by tossing a fair coin:

$$v_k^{(c)} = \begin{cases} v_k^{(1)}, & if\ the\ coin\ comes\ up\ \texttt{HEADS} \\ v_k^{(2)}, & if\ the\ coin\ comes\ up\ \texttt{TAILS} \end{cases} \tag{17}$$

Figure 9 illustrates the crossover process. The crossover process repeats with the parents exchanged to produce the second new individual. Therefore, offspring contain a mixture of paths from each parent path.

5) *Mutation*. The mutation rate parameter $p_m$ controls the number of individuals that will have the mutations after

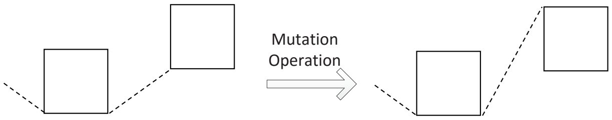


Fig. 10. Mutation operation.

the crossover process. For the generated path $\vec{s_c}(v_i, v_j) =< v_i^{(c)}, v_1^{(c)}, v_2^{(c)}, ..., v_k^{(c)}, v_j^{(c)} >$ after the crossover operation, each vertex $v_k^{(c)}$ in $\vec{s_c}(v_i, v_j)$ is randomly flipped to another vertex in $V$ to generate a new path $\vec{s_m}(v_i, v_j) =< v_i^{(m)}, v_1^{(m)}, v_2^{(m)}, ..., v_k^{(m)}, v_j^{(m)} >$. Therefore:

$$\forall v_k^{(m)} \in \vec{s_m}(v_i, v_j), v_k^{(m)} = \begin{cases} v_k^{(c)}, & p_r > p_m \\ V_r, & otherwise \end{cases} \tag{18}$$

where $r$ is a random number in the range $[1, t]$, $V_r$ is the $r$-th vertex in $V$, and $p_r$ is a random real number in $[0, 1]$. As a result, the path maintains the original vertex with probability $1 - p_m$ and flip the vertex to a random one with probability $p_m$. As shown in Figure 10, the mutation process produces a new path by randomly flipping vertexes in the path.

6) *Local rolling optimization (LRO).* The evolutionary process iteratively refines the initial population towards the optimal ones defined by the fitness function. However, the generated paths usually fall into the local optimal and there is no guarantee on the global optimal using the above evolutionary algorithm. In the hybrid path planning algorithm, we introduce an additional local rolling optimization to further improve the resulted path. The LRO takes the evolved population from the iteration process $\vec{s_m}(v_i, v_j)$ as input, and aims to find the path $\vec{s_{LRO}}(v_i, v_j)$ to solve the following optimization problem:

$$\begin{aligned} \underset{\vec{s_{LRO}}(v_i,v_j)}{\text{minimize}} \quad & f(\vec{s_{LRO}}(v_i, v_j)) \\ \text{subject to} \quad & I(\vec{s_{LRO}}(v_i, v_j), O) = \emptyset \end{aligned} \tag{19}$$

Though the objective is the same as the evolutionary process, in LRO, we divide the path $\vec{s_m}(v_i, v_j)$ into path segments $\vec{s_m}(v_i, v_{i+l})$, $\vec{s_m}(v_{i+l}, v_{i+2l})$, ..., ,$\vec{s_m}(v_{i+tl}, v_j)$ with each path segments having length $l$. A brute-force search is applied to each path segments to ensure the global optimal within each path segment. LRO efficiently improve the path quality while ensuring a real-time performance. The effectiveness of the LRO is shown in the evaluation section VI. Algorithm 1 shows the complete flow of the hybrid path planning algorithm.

### C. Benchmark Algorithms

In this subsection, we describe the benchmark algorithms for comparisons with the proposed algorithm.

**Algorithm 1:** Hybrid Path Planning Algorithm

```
Input: Vertexes $V$, obstacles $O$, starting point $v_i$,
ending point $v_d$, maximum generation iterations: $N$;
Output: Planned path $\vec{s_{LRO}}(v_i, v_j)$;
begin
    Generate the initial population
    $A(v_i, v_j) = \{\vec{s_1}(v_i, v_j), \vec{s_2}(v_i, v_j), ..., \vec{s_m}(v_i, v_j)\}$;
    for genaration<N do
        Calculate the fitness using Equation (15) for all
        individuals: $F(\vec{s_1}(v_i, v_j))$, $F(\vec{s_2}(v_i, v_j))$ ,...,
        $F(\vec{s_m}(v_i, v_j))$;
        Divide the population into selection pools with
        size $k$ and sort the selection pools in
        descending order based on the fitness;
        With probability $p_s^{(i)} = p_s * (1 - p_s)^i$ select
        each individual in the selection pools;
        For each pair of parents, perform the uniform
        crossover operator by tossing a fair coin using
        Equation (17) and generate the path
        $\vec{s_c}(v_i, v_j) =< v_i^{(c)}, v_1^{(c)}, v_2^{(c)}, ..., v_k^{(c)}, v_j^{(c)} >$;
        With mutation rate $p_m$, generate a new path
        $\vec{s_m}(v_i, v_j) =<$
        $v_i^{(m)}, v_1^{(m)}, v_2^{(m)}, ..., v_k^{(m)}, v_j^{(m)} >$ using
        Equation (18);
        The resulting $\vec{s_m}(v_i, v_j)$ becomes the new
        generation;
    Divide the paths $\vec{s_m}(v_i, v_j)$ into path segments
    $\vec{s_m}(v_i, v_{i+l})$, $\vec{s_m}(v_{i+l}, v_{i+2l})$, ..., ,$\vec{s_m}(v_{i+tl}, v_j)$;
    For each path segment, conduct brute-force search
    to find the global optimal connecting the starting
    point and ending point of each path segment and
    generate a generation of $\vec{s_{LRO}}(v_i, v_j)$;
    return the $\vec{s_{LRO}}(v_i, v_j)$ with the highest fitness
    value;
```

1) *Genetic Algorithms:* In the field of artificial intelligence, a genetic algorithm is a search heuristic that mimics the process of natural selection. This heuristic (also called a metaheuristic) is routinely used to generate useful solutions to optimization and search problems [30]. Genetic algorithms belong to the larger class of evolutionary algorithms (EA), which generate solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover. A typical genetic algorithm concludes initialization, selection, genetic operators, and termination.

2) *A star Algorithm :* A star (A*) algorithm has been widely used in path finding and graph traversal problem in the robotic community. A star selects the path that minimizes:

$$f(n) = g(n) + h(n) \tag{20}$$

where $n$ is the last node on the path, $g(n)$ is the cost of the path from the start node to $n$, and $h(n)$ is a heuristic that estimates the cost of the cheapest path from $n$ to the

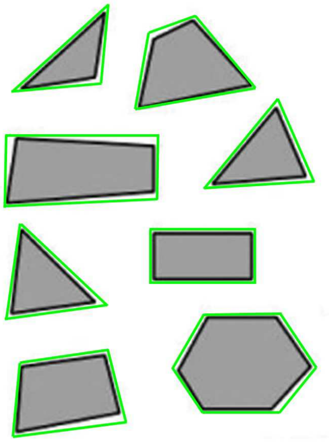


Fig. 11. Basic obstacle model.

TABLE I. PARAMETERS SETTING

| Population size | Population Generations | Crossover Probability | Mutation Probability |
|---|---|---|---|
| 50 | 100 | 0.75 | 0.2 |

destination node.

## VI. Experimental Results and Further study

In this section, we present the experimental results to show the effectiveness of the proposed algorithm. In our experiment, a ground image is recorded by a UAV from aerial vision. Then, the image is detected and segmented. According to the obstacle sampled pixel values and the color contrast of obstacles and background, automatic threshold segmentation is given based on our algorithm. Then the image is binarized. The processed image is detected in the connected domain, and the basic obstacle model is established, as shown in Fig. 11.

In order to improve the accuracy of the image description, especially for the obstacle sideline and corner, Harri corner detection method is used to get obstacles corners in the connected domain (as shown in Fig. 12). The detected obstacles are described by convex polyhedrons based on the corner points, and approximately expanded for the volume of the vehicle. Then the effective ground environment map is built.

After building the environment map, path planning is performed to reach the target. The parameters used in our experiments are shown in Table 1. As genetic algorithms are global optimization algorithms, local information and optimization are not considered completely in traditional genetic algorithms. For the change of ground vehicle vision and better optimization, local search optimization is inserted in global planning. For the UGV, the path with multiple steps is optimized (4 steps in this paper) in the current location.

Using our proposed algorithm, the local path has been optimized in the overall trajectory. We simulated a map

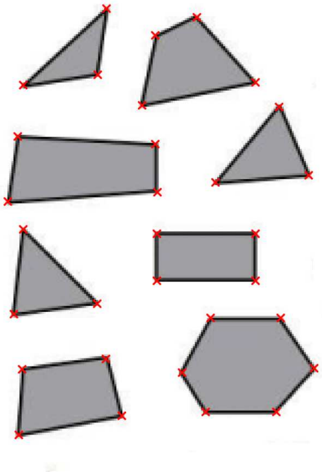


Fig. 12. Harri corner detection.

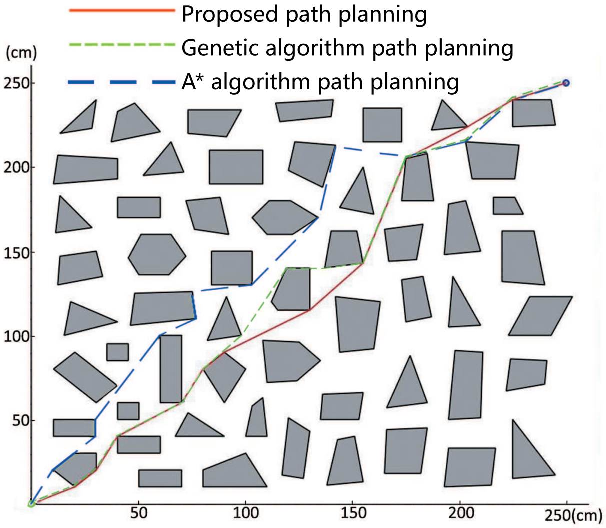


Fig. 13. Path planning using the proposed algorithm.

in Fig. 13, it shows the different results of path planning using the various algorithms. We can observe that the total number of vertexes is less, the overall distance is shorter, and the trajectory is smoother in the proposed algorithm compared to the original path planning algorithm and A* algorithm. The proposed algorithm uses the global search ability of the genetic algorithm and the local optimization of the rolling optimization algorithm. To further compare the algorithms, 20 experiments are performed. Table 2 shows the number of vertexes, overall path length and computing time of each individual experiment. Each experiment is planned by the genetic algorithms first, then optimized by scrolling optimization. Table 3 shows the average performance comparison. From Table 2 and Table 3, we can observe that, the proposed path planning algorithm, the number of vertexes

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2016.2623666, IEEE Transactions on Vehicular Technology

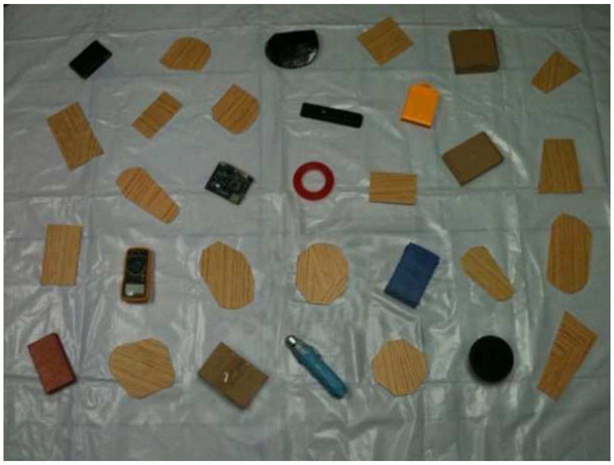


Fig. 14. Experimental environment.

through the number of obstacles is less than that in the genetic algorithm, and the path length is shorter, which show the effectiveness of the proposed algorithm.

Next, we increase the number of obstacles in the experiments to simulate a rescue scenario, as shown in Fig. 14. A number of different obstacles are placed on the floor. 10 experiments are made by changing the obstacles location and starting point. Ground image are shot by the UAV in the air, and then the image is sent to a computer. The experimental process is shown in Fig. 4. As the image is taken by the UAV in the air, there are some distortions, as shown in Fig. 14. Fig. 15 is the corrected image for Fig. 14. The result of edge detection map is shown in Fig. 16, where the edges of the obstacles are portrayed by canny operator. Fig. 17 is an image after contour detection and polygons. Each target is portrayed polygon according to its own objective external features.

Fig. 18 is the path planning maps corresponding to Fig. 17, which have been converted to a standard Cartesian coordinate system. The origin is the starting point, and the figure points (300, 260) is defined as the end point. The planning path is indicated by the red curve. We can observe from Fig. 18 that the planned path is smooth, and there is no twist and circuitous path. The overall trajectory is improved, which show the effectiveness of the proposed algorithm.

In order to further verify the effectiveness of the proposed hybrid path planning algorithm, more complicated experiment settings are required. In this experiment, we significantly increase the number of obstacles to simulate the complexity of the path planning task in practice.

As shown in Fig. 19, the generated path by the proposed algorithm is visually and statistically better than the original genetic algorithm with less vertexes, shorter distance, and the smoother trajectory. The evaluation result show that our proposed algorithm is able to work in complicated environments with large number of obstacles and achieve a better path planning performance in practice.

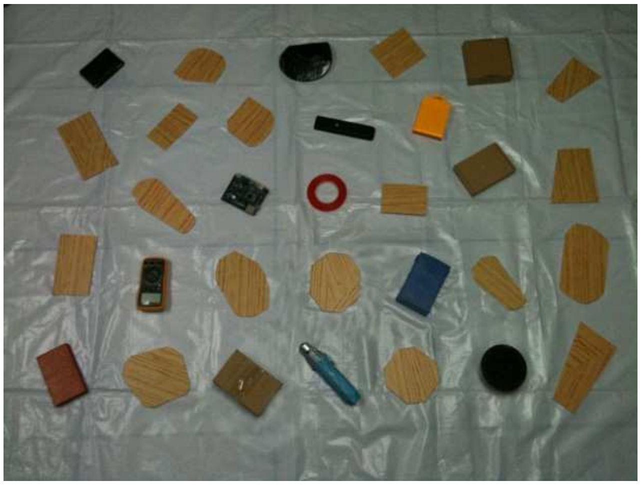


Fig. 15. Image correction.

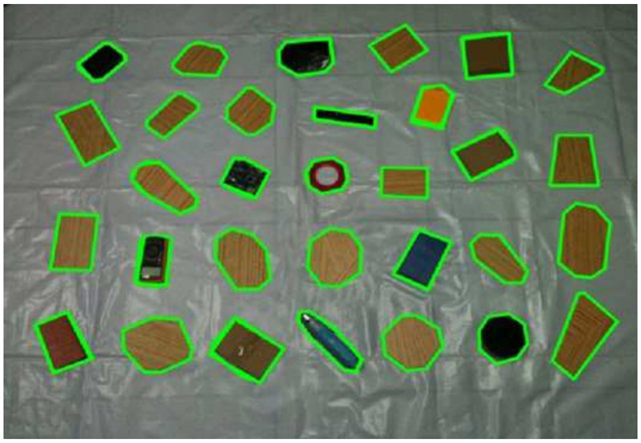


Fig. 16. Edge detection.

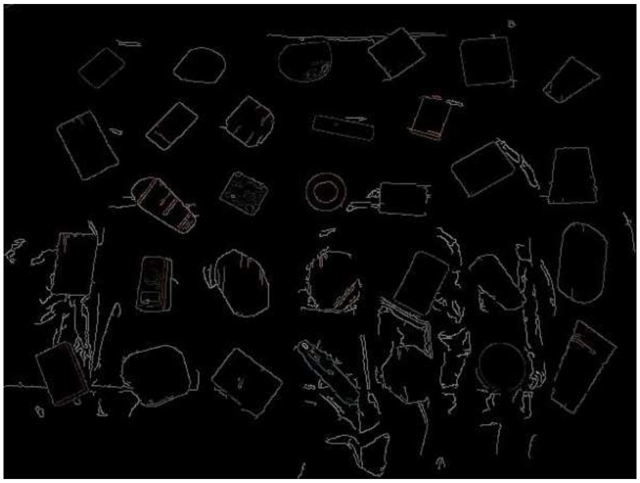


Fig. 17. Polygonal approximation.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2016.2623666, IEEE Transactions on Vehicular Technology

TABLE II. Performance comparison

| | Genetic algorithm | | | Improved genetic algorithm | | |
|---|---|---|---|---|---|---|
| group | vertices | overall distance (pixel points) | computing time (second) | vertices | overall (pixel points) | computing (second) |
| 1 | 17 | 376.036 | 1.835 | 12 | 367.224 | 1.881 |
| 2 | 16 | 385.939 | 1.810 | 14 | 365.562 | 1.861 |
| 3 | 16 | 389.197 | 1.787 | 12 | 367.224 | 1.809 |
| 4 | 16 | 388.208 | 1.833 | 13 | 372.551 | 1.884 |
| 5 | 15 | 387.999 | 1.838 | 11 | 372.925 | 1.888 |
| 6 | 16 | 397.454 | 1.786 | 12 | 377.582 | 1.836 |
| 7 | 15 | 381.879 | 1.749 | 14 | 381.297 | 1.799 |
| 8 | 15 | 396.489 | 1.876 | 14 | 371.248 | 1.975 |
| 9 | 16 | 385.258 | 2.057 | 13 | 383.679 | 2.080 |
| 10 | 15 | 392.872 | 1.866 | 12 | 372.798 | 1.918 |
| 11 | 17 | 393.711 | 1.801 | 13 | 369.856 | 1.812 |
| 12 | 15 | 371.291 | 1.845 | 14 | 370.844 | 1.896 |
| 13 | 17 | 375.125 | 1.823 | 13 | 371.234 | 1.879 |
| 14 | 16 | 461.535 | 1.796 | 11 | 375.562 | 1.855 |
| 15 | 17 | 395.472 | 1.758 | 14 | 366.158 | 1.811 |
| 16 | 16 | 397.766 | 1.835 | 13 | 364.143 | 1.891 |
| 17 | 15 | 386.567 | 1.862 | 12 | 362.459 | 1.910 |
| 18 | 17 | 399.857 | 1.789 | 14 | 372.462 | 1.812 |
| 19 | 16 | 398.578 | 1.818 | 13 | 375.867 | 1.875 |
| 20 | 17 | 396.487 | 1.862 | 14 | 369.578 | 1.899 |

TABLE III. Average performance comparison

| parameter<br>Algorithm | Average vertexes | Overall distance | Computing time (second) | Simulation group |
|---|---|---|---|---|
| Improved genetic algorithm | 12.9 | 371.512 | 1.878 | 20 |
| Genetic algorithm | 16 | 392.886 | 1.831 | 20 |

## VII. Conclusions and Future Work

Unmanned aerial/ground vehicles (UAV/UGV) are very useful to complete dangerous tasks where it is difficult for human beings to access. In this paper, we studied the environment map building and path planning issues in UAV/UGV cooperative systems. With the aerial vision by the UAV, a ground image can be obtained. Then, we used hybrid recognition feature extraction method based on color segmentation and connected domain detection to improve the recognition accuracy for the obstacles. Harris corner detection was used to extract the features of corner points. In addition, we built the map by polygon description of the obstacles. Based on the ground map information, we have proposed a hybrid path planning algorithm to optimize the distance of the path. Moreover, we presented a genetic algorithm for global path planning and a rolling optimization algorithm to constantly optimize the results of the genetic algorithm. Experimental results were presented to show the effectiveness of the proposed schemes. Future work is in progress to consider the effects of unreliable wireless communications in the proposed framework.

## Acknowledgment

The authors gratefully acknowledge the contribution of the National Science Foundation of China [61572330][61472258][61602319], the Natural Science foundation of Guangdong Province [2014A030313554], and the Technology Planning Project (Grant No. 2014B010118005) from Guangdong Province.

## References

[1] J. Casper and R. Murphy, "Human-robot interactions during the robot-assisted urban search and rescue response at the world trade center," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 33, no. 3, pp. 367–385, 2003.

[2] S. Minaeian, J. Liu, and Y. J. Son, "Vision-based target detection and localization via a team of cooperative uav and ugvs," *IEEE Trans. Systems, Man, and Cybernetics: Systems*, no. 99, pp. 1–12, 2015.

[3] B. Grocholsky, J. Keller, V. Kumar, and G. Pappas, "Cooperative air and ground surveillance," *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, pp. 16–25, 2006.

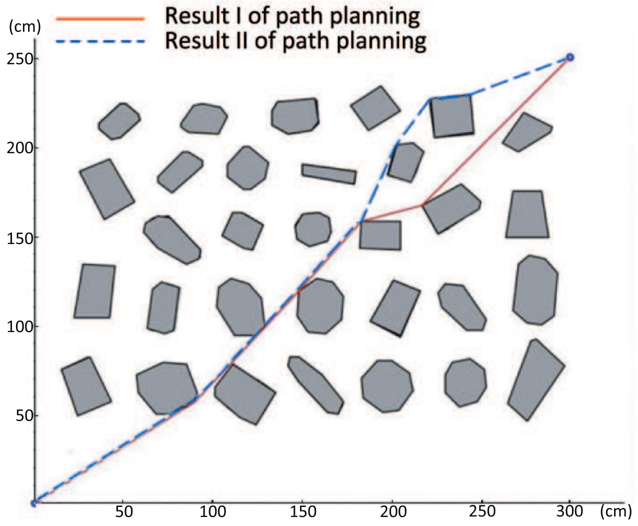


Fig. 18. Path planning of experiment.

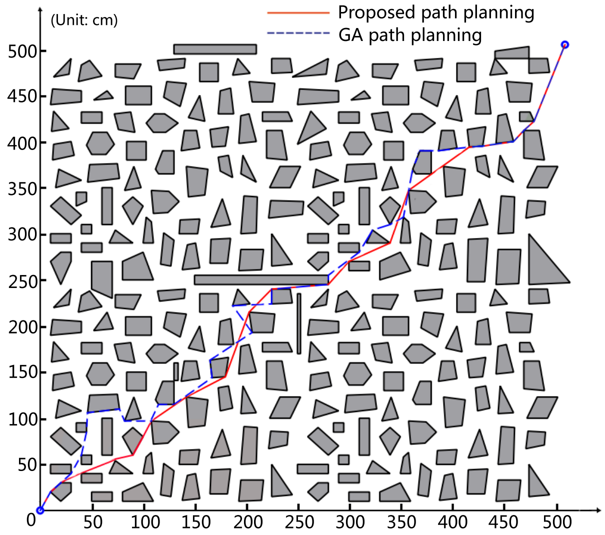


Fig. 19. A complex map using the proposed algorithm.

[4] H. Duan and S. Liu, "Unmanned air/ground vehicles heterogeneous cooperative techniques: Current status and prospects," *Science China Technological Sciences*, vol. 53, no. 5, pp. 1349–1355, 2010.

[5] A. Kim and R. Eustice, "Real-time visual slam for autonomous underwater hull inspection using visual saliency," *IEEE Transactions on Robotics*, vol. 29, no. 3, pp. 719–733, 2013.

[6] W. Zeng, T. Zhang, and D. Jiang, "Analysis of data association methods of slam," *Systems Engineering and Electronics*, vol. 32, no. 4, pp. 860–864, 2010.

[7] H. G. Tanner, "Switched uav-ugv cooperation scheme for target detection," in *Proc. IEEE International Conference on Robotics and Automation*, 2007, pp. 3457–3462.

[8] K. Fergene, D. Kennedy, and D. Wang, "Toward a systems-and control-oriented agent framework," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 35, no. 5, pp. 999–1012, 2005.

[9] H. Cheng, Y. Chen, X. Li, and W. Wong, "Autonomous takeoff, tracking and landing of a uav on a moving ugv using onboard monocular vision," in *Proc. IEEE Control Conference (CCC), 32nd Chinese*, 2013, pp. 5895–5901.

[10] L. Cantelli, M. Mangiameli, C. Melita, and G. Muscato, "Uav/ugv cooperation for surveying operations in humanitarian demining," in *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2013, pp. 1–6.

[11] J. Bangham, C. Pye, and S. Impey, "Multiscale median and morphological filters used for 2d pattern recognition," in *Morphological and Nonlinear Image Processing Techniques, IEE Colloquium on*, 1993, pp. 5–1.

[12] C. Harris and M. Stephens, "A combined corner and edge detector." in *Proc. Alvey vision*, vol. 15. Citeseer, 1988, pp. 50–51.

[13] A. Nemra and N. Aouf, "Robust feature extraction and correspondence for uav map building," in *Control and Automation, 17th Mediterranean Conference on*. IEEE, 2009, pp. 922–927.

[14] M. Moseley, B. Grocholsky, C. Cheung, and S. Singh, "Integrated long-range uav/ugv collaborative target tracking," *SPIE defense, security, and sensing*, pp. 733 204–733 204, 2009.

[15] V. Ila, J. Porta, , and J. Andrade-Cetto, "Information-based compact pose slam," *IEEE Transactions on Robotics*, vol. 26, no. 1, pp. 78–93, 2010.

[16] H. Mitchell, "A retraction method for planning the motion of a disc," *Journal of Algorithms*, vol. 6, no. 1, pp. 104–111, 1985.

[17] Y. Peng and W. Wei, "A new trajectory planning method of redundant manipulator based on adaptive simulated annealing genetic algorithm (asaga)," in *Proc. 2006 International Conference on Computational Intelligence and Security*, vol. 1, 2006, pp. 262–265.

[18] L. Cheng, C. Liu, , and B. Yan, "Improved hierarchical a-star algorithm for optimal parking path planning of the large parking lot," in *Proc. Information and Automation (ICIA), 2014 IEEE International Conference on*, 2014, pp. 695–698.

[19] Ganeshmurthy and G. Suresh, "Path planning algorithm for autonomous mobile robot in dynamic environment," in *Proc. 3rd Int'l Conf. Signal Processing, Communication and Networking*, 2015, pp. 1–6.

[20] Z. X. Zhu, J. Xiao, J. Q. Li, F. X. Wang, and Q. F. Zhang, "Global path planning of wheeled robots using multi-objective memetic algorithms," *Integrated Computer-Aided Engineering*, vol. 22, no. 4, pp. 387–404, 2015.

[21] Q. Z. Lin, J. Q. Li, Z. H. Du, J. Y. Chen, and Z. Ming, "A novel multi-objective particle swarm optimization with multiple search strategies," *European Journal of Operational Research*, vol. 247, no. 3, pp. 732–744, 2015.

[22] L. Liu, Y. N. Song, H. Y. Zhang, H. D. Ma, and V. A. V, "Physarum optimization: A biology-inspired algorithm for the steiner tree problem in networks," *IEEE Transactions on Computers*, vol. 64, no. 3, pp. 818–831, 2015.

[23] M. Sedaaghi, "Morphological operators," *Electronics Letters*, vol. 38, no. 22, pp. 1333–1335, 2002.

[24] Y. Zhu, C. Huang, and Z. Xu, "Image denoising algorithm based on the median morphological filter," in *Proc. 7th World Congress on Intelligent Control and Automation*, 2008, pp. 3985–3989.

[25] Y. Nomurm, M. Sagara, and H. Naruse, "Simple calibration algorithm for high-distortion lens camera," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 11, pp. 1095–1099, 1992.

[26] S. Shah and J. Aggarwa, "A simple calibration procedure for fisheye (high-distortion) lens camera," in *Proc. IEEE International Conference on Robotics & Automation*, 1994, pp. 3422–3427.

[27] J. Wu, "The research of robot path planning based on improved genetic algorithm," *Master thesis, Northwest University, China*, 2012.

[28] K. Deb, *Multi-objective optimization using evolutionary algorithms*. NY, USA: John Wiley & Sons Press, 2001.

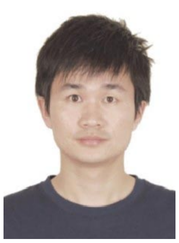
[29] S. Gilbert, “Uniform crossover in genetic algorithms,” in *International Conference on Genetic Algorithms*, 1989, pp. 2–9.

[30] M. Mitchell, *An Introduction to Genetic Algorithms*. Cambridge, MA: MIT Press, 1996.

**Jianqiang Li** received his B.S and Ph.D. degree from South China University of Technology in 2003 and 2008. He is an associated professor at the College of Computer and Software Engineering of Shenzhen University. He led a project of the National Natural Science Foundation, and a project of the Natural Science Foundation of Guangdong province, China. His major research interests include vehicleic, hybrid systems, internet of thing and embedded systems.

**Genqiang Deng** received his B.E. in Computer Science and Technology, College of Computer Science, Zhuhai Campus, Beijing Institute of Technology in 2014 and he is the master student of College of Computer Science and Software Engineering of Shenzhen University. His major research interests includes robotic, internet of things and embedded systems.

**Chengwen Luo** currently is an assistant professor in College of Computer Science and Software Engineering, Shenzhen University, China. Before joining SZU, he was a postdoctoral researcher in CSE, The University of New South Wales (UNSW), Australia. He received his Ph.D. degree from School of Computing, National University of Singapore (NUS), Singapore. Dr. Luo is the author and co-author of several research papers in top venues of mobile computing and WSN such as IEEE TMC, IEEE/ACM TON, ACM SenSys, ACM/IEEE IPSN, etc. His research interests include mobile and pervasive computing, indoor localization, wireless sensor networks, and security aspects of Internet of Things.



**Qiuzhen Lin** received the B.S. degree from Zhaoqing University and the M.S. degree from Shenzhen University, China, in 2007 and 2010, respectively. He received the Ph.D. degree from Department o Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong, in 2014. He is currently a Lecturer in College of Computer Science and Software Engineering, Shenzhen University. He has published over 20 research papers since 2008. His current research interests include artificial immune system, multi-objective optimization and dynamic system.

**Qiao Yan** is a professor at the College of Computer Science and Software Engineering at Shenzhen University, Shenzhen, China. She received her Ph.D. degree in information and communication engineering from Xidian University, Xian, China, in 2003. From 2004 to 2005 she worked at Tsinghua University, Beijing, China as a postdoctor. From 2013 to 2014 she worked at Carleton University, Ottawa, Canada, as a visiting scholar. Her research interests are in network security, cloud computing, and software-defined networking. Her current focus is research and development of security of software defined networking.

**Zhong Ming** is a professor at the College of Computer and Software Engineering of Shenzhen University. He is a senior member of the Chinese Computer Federation. He led three projects of the National Natural Science Foundation, and two projects of the Natural Science Foundation of Guangdong province, China. His major research interests include home networks, internet of thing and cloud computing.