

# Path Planning Method for Collision Avoidance of Multiple UAVs

Hyeok Kim<sup>1</sup>, Jeonghoon Kwak<sup>2</sup>, Guichang Sim<sup>1</sup>,  
and Yunsick Sung<sup>2</sup>(✉)

<sup>1</sup> Faculty of Computer Engineering, Keimyung University, Daegu, South Korea  
{hyeok, guichang}@kmu.ac.kr

<sup>2</sup> Department of Multimedia Engineering, Dongguk University,  
Seoul, South Korea  
{jeonghoon, sung}@dongguk.edu

**Abstract.** In recent years, the development of unmanned aerial vehicles (UAVs) has increased significantly and they are currently used in various fields and applications. In some applications, multiple UAVs need to be cooperated to accomplish tasks, because a single UAV is not sufficient. However, even when multiple UAVs are used, their autonomous control systems are not perfect, which leads to collisions between the UAVs. In this paper, we propose a path planning method for collision avoidance of UAVs, when multiple UAVs are controlled using a ground control system. Furthermore, using this method, the UAVs have less likelihood to be in a close encounter with obstacles, and collisions are avoided.

**Keywords:** UAV · Path planning · Collision avoidance · Multiple UAVs · Geometric approach

## 1 Introduction

As applications of unmanned aerial vehicles (UAVs) have been increasing, more fields are trying to use UAVs. A few examples are search and rescue operations, surveillance systems, crop dusting, and performances using multiple UAVs [1–4].

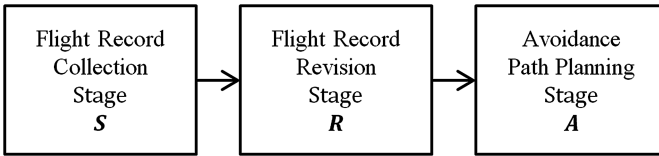
When a single UAV is used, it can take a long time to accomplish tasks, or even fail. This problem can be resolved using multiple UAVs. However, to operate multiple UAVs in a ground control system, it is necessary to apply a collision avoidance technique to the system.

In this paper, we propose a path planning method for collision avoidance of multiple UAVs based on recorded UAV flight data. A geometric arithmetic algorithm is used to plan the flight paths for multiple UAVs, predict possible collisions before operation, and prevent collisions [6]. While traditional geometric arithmetic algorithms use the velocity information of UAVs, the proposed method computes the differences in recorded UAV flight data. Because the method uses the recorded UAV flight data to plan the paths, it can consider the collision issues less than traditional algorithms.

The rest of this paper is organized as follows: In Sect. 2, we propose a path planning method for multiple UAVs. In Sect. 3, the implementation of the proposed method is described, the operations are verified, and the conclusions are provided.

## 2 Multiple UAV Control for Collision Avoidance

The method for generating collision avoidance paths is described in Fig. 1. In the first stage, i.e., the flight record correction stage, a user records UAV flight data while controlling the UAV with a controller. In the second stage, i.e., the flight record revision stage, a path revision algorithm is used to revise the paths recorded in the first stage. Finally, in the third stage, i.e., the avoidance path planning stage, the system generates the paths in which no collisions between UAVs would occur.



**Fig. 1.** This figure shows a diagram of the three stages to obtain results from the proposed algorithm. The first stage is flight record collection through the control of a UAV by a user, using a controller, the second stage is the path revision stage, and the last stage is the avoidance path planning stage.

### 2.1 UAV Flight Data Records

The hourly states of the  $n^{\text{th}}$  UAV record are defined as the collected state set,  $S_n$ , as shown in Eq. (1). The collected state,  $S_{n,i}$ , indicates the  $i^{\text{th}}$  collected state of the collected state set,  $S_n$ . Then, the collected state,  $S_{n,i}$ , is described in terms of four components,  $x_{n,i}$ ,  $y_{n,i}$ ,  $z_{n,i}$ , and  $t_{n,i}$ , which represent the GPS latitude, the GPS longitude, the absolute altitude, and the flight time from departure, respectively. Finally, the collected states,  $S_{n,i}$ , are added to the collected state set,  $S_n$ , to record.

$$S_n = [S_{n,1}, S_{n,2}, \dots, S_{n,i}, \dots] \quad (1)$$

$$S_{n,i} = [x_{n,i}, y_{n,i}, z_{n,i}, t_{n,i}] \quad (2)$$

### 2.2 Path Revision Computations

To revise the collected flight data, the Douglas–Peucker algorithm is applied [5]. This algorithm suggests a path revision method in two-dimensional space ( $f:p_{n,i} \rightarrow R_{n,j}$ ) and advances the method to a three-dimensional path revision method. This method is not included in this paper.

The collected state set,  $S_n$ , is revised to make the revised state set,  $R_n$ , as shown in Eq. (3). The revised state,  $R_{n,j}$ , is the  $j^{\text{th}}$  revised state in the revised state set,  $R_n$ , and is defined by Eq. (4). The element in the revised state set,  $R_n$ , uses the element of the collected state set,  $S_n$ , to define the collected state,  $S_{n,i}$ , which contains the  $[x_{n,i}, y_{n,i}, z_{n,i}]$  elements that are outside the error tolerance, as the revised state,  $R_{n,j}$ .

$$R_n = [R_{n,1}, R_{n,2}, \dots, R_{n,j}, \dots] \quad (3)$$

$$R_{n,j} = [x_{n,j}, y_{n,j}, z_{n,j}, t_{n,j}] \quad (4)$$

### 2.3 Collision Avoidance Path Planning

The avoidance state set,  $A_n$ , is defined by Eq. (5). The avoidance state,  $A_{n,k}$ , is defined as the  $k^{\text{th}}$  avoidance state, and is shown in Eq. (6). The state-difference set,  $D_n$ , is defined by Eq. (7). The state-difference,  $D_{n,j}$ , is the  $j^{\text{th}}$  element of the state-difference set,  $D_n$ , and is shown in Eq. (8). The avoidance point set,  $P_n$ , is defined by Eq. (9). The avoidance point,  $P_{n,k}$ , is the  $k^{\text{th}}$  element of the avoidance point set,  $P_n$ , and is shown in Eq. (10). Finally, the direction pointing to the avoidance point,  $P_{n,j}$ , is defined as  $U_{n,j} = [x_{n,j}, y_{n,j}, z_{n,j}]$ .

$$A_n = [A_{n,1}, A_{n,2}, \dots, A_{n,k}, \dots] \quad (5)$$

$$A_{n,k} = [x_{n,k}, y_{n,k}, z_{n,k}, t_{n,k}] \quad (6)$$

$$D_n = [D_{n,1}, D_{n,2}, \dots, D_{n,j}, \dots] \quad (7)$$

$$D_{n,j} = [x_{n,j}, y_{n,j}, z_{n,j}, t_{n,j}] \quad (8)$$

$$P_n = [P_{n,1}, P_{n,2}, \dots, P_{n,k}, \dots] \quad (9)$$

$$P_{n,k} = [x_{n,k}, y_{n,k}, z_{n,k}] \quad (10)$$

The computational time,  $t'_{n,m}$ , is defined as the  $m^{\text{th}}$  element of the computational time set,  $T'_n = \{0, 1, \dots, t_{n,|R|-1} \text{ of } R_n - t_{n,1} \text{ of } R_n - 1, t_{n,|R|} \text{ of } R_n - t_{n,1} \text{ of } R_n\}$ , where  $n! = \text{MAX}(n)$ .  $h(n1, n2, m)$  is  $\max_j(t_{n1,j} \text{ of } R_{n1} - t_{n1,1} \text{ of } R_{n1})$  where  $t_{n1,j} \text{ of } R_{n1} - t_{n1,1} \text{ of } R_{n1} < t'_{n2,m}$ .

There are nine steps in the generation procedure. First, the values of  $n1$  and  $n2$  that satisfy the condition  $n1 < n2$  need to be determined, and  $m$  is set to be 1. If the values of  $n1$  and  $n2$  cannot be determined, then the generation of the collision avoidance path is terminated.

Second, if  $A_n \neq \emptyset$ , then each element of the avoidance state set,  $A_n$ , is added to the revised state set,  $R_n$ , in order, and the avoidance state set,  $A_n$ , and the state-difference set,  $D_n$ , are set to be null sets.

Third, the state-difference set,  $D_n$ , is computed using the revised state set,  $R_n$ . Each element of the state-difference set,  $D_n$ , is calculated as shown in Eq. (11); it is the difference between the adjacent elements of the revised states in the revised state set,  $R_n$ .

$$\begin{aligned} x_{n,j} &= x_{n,j+1} - x_{n,j}, y_{n,j} = y_{n,j+1} - y_{n,j} \\ z_{n,j} &= z_{n,j+1} - z_{n,j}, t_{n,j} = t_{n,j+1} - t_{n,j} \end{aligned} \quad (11)$$

Fourth, the indices  $j1$  and  $j2$  are determined. Set  $j1 = h(n1, n1, m)$  and  $j2 = h(n2, n1, m)$ .

Fifth, the revised states,  $R_{n1,j1}$  and  $R_{n2,j2}$ , are added to the avoidance state sets,  $A_{n1}$  and  $A_{n2}$ , respectively, using Eq. (12).

$$A_n = A_n \cup \{R_{n,j}\} \quad (12)$$

Sixth, using the geometric arithmetic algorithm and the state-differences,  $D_{n1,j1}$  and  $D_{n2,j2}$ , the time for reaching the collision prediction point,  $t''$ , and the directions pointing to the avoidance point,  $U_{n1,j1}$  and  $U_{n2,j2}$ , are computed, where  $g : D_{n1,j1}, D_{n2,j2} \rightarrow [t'', U_{n1,j1}, U_{n2,j2}]$  [6]. If  $t'' \leq 0$  or  $j1 = |R_{n1}|$ , the system skips to the ninth step without performing the seventh and the eighth steps, because the collision cannot be predicted under those conditions. Otherwise, the system performs the remaining steps in order.

Seventh, the avoidance point is generated using the directions pointing to the avoidance point,  $U_{n1,j1}$  and  $U_{n2,j2}$ .  $U_{n1,j1}$  is always generated, while  $U_{n2,j2}$  is not. Therefore, when  $U_{n2,j2}$  is not generated,  $U_{n1,j1}$  is computed using Eq. (13) to generate the avoidance point,  $P_{n1,j1}$ .

$$\begin{aligned} x_{n,j} &= x_{n,j} \text{ of } R_n + t''_n \times x_{n,j} \text{ of } D_n + (x_{n,j+1} \text{ of } R_n - x_{n,j} \text{ of } R_n) \times t'' \times x_{n,j} \text{ of } U_{n,j}, \\ y_{n,j} &= y_{n,j} \text{ of } R_n + t''_n \times y_{n,j} \text{ of } D_n + (y_{n,j+1} \text{ of } R_n - y_{n,j} \text{ of } R_n) \times t'' \times y_{n,j} \text{ of } U_{n,j}, \\ z_{n,j} &= z_{n,j} \text{ of } R_n + t''_n \times z_{n,j} \text{ of } D_n + (z_{n,j+1} \text{ of } R_n - z_{n,j} \text{ of } R_n) \times t'' \times z_{n,j} \text{ of } U_{n,j} \end{aligned} \quad (13)$$

Eighth, the generated avoidance point,  $P_{n,j}$ , is added as the last element of the avoidance state set,  $A_n$ . For example, the possible flight time is predicted when the avoidance point,  $P_{n1,j1}$ , is generated, and added as the last element of the avoidance state,  $A_n$ , using Eq. (14).

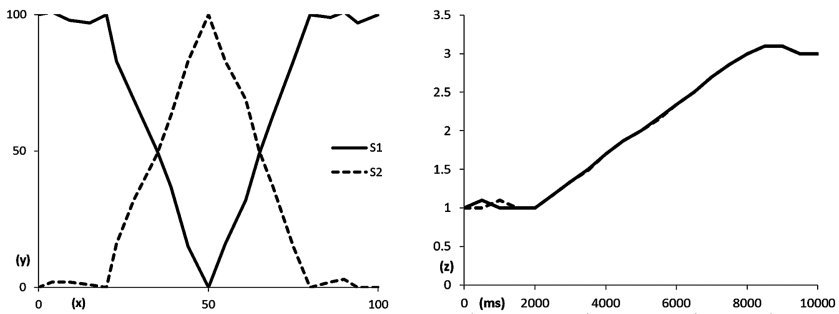
$$A_n = A_n \cup \{[P_{n,j}, t'_{n,m} + t'']\} \quad (14)$$

Ninth, if  $t'_{n,m}$  is not the last element of  $T'_n$ , then  $m$  is increased by 1 and the procedure is repeated starting from the fourth step. If  $t'_{n,m}$  is the last element of  $T'_n$ , then the procedure is repeated starting from the first step.

### 3 Experiments

#### 3.1 Three-Dimensional Experiments Scenario and UAV Flight Data Record

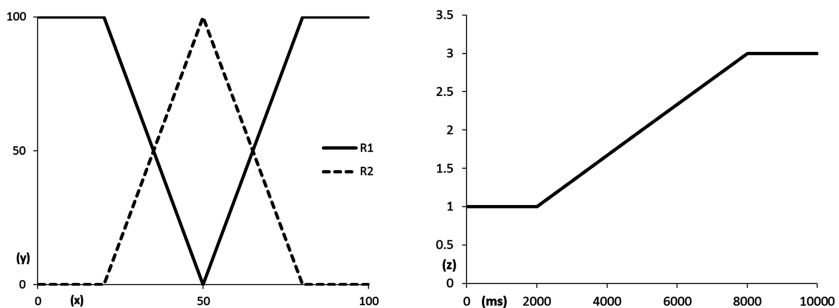
In this study, we conducted experiments to verify the proposed method for the path planning based on the collected UAV flight data and for avoiding collisions between the UAVs. The proposed method was verified using a scenario that we designed. The flight record in this scenario includes external disturbances, such as wind and an inexperienced pilot, and internal disturbances, such as GPS position error. The unstable records are shown in Fig. 2.



**Fig. 2.** UAV flight data collected from two different UAVs. The graphs are linear but noisy and unstable.

#### 3.2 Path Revision Computation

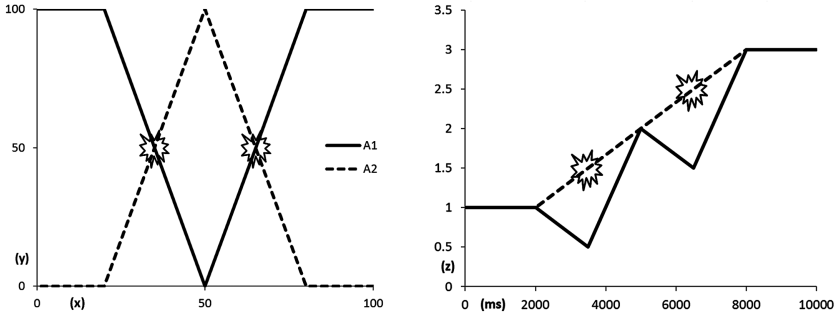
In this study, the path was revised in three-dimensional space. The results are shown in Fig. 3. By correcting the collected flight data, the graphs became considerably stable and therefore, it is expected that the UAVs would not stop during autonomous flight.



**Fig. 3.** The corrected path of the collected flight data, S. Dogulas–Peucker algorithm was applied in three-dimensional space.

### 3.3 Experiments for the Generation of the Collision Avoidance Path

In this study, a three-dimensional collision avoidance algorithm was applied [6]. The path generated through collision avoidance is shown in Fig. 4.



**Fig. 4.** A generated path, A, by predicting collisions at different time intervals, and generating and adding the avoidance points.

In the experiments, two collision points were predicted and these were reflected in the system such that the UAV could change its path along the Z-axis to avoid collision. Finally, the two collision points, (65.00504, 49.98345, 0.50015, 3.5) and (34.98355, 50.02151, 1.500367, 6.5), were added to the avoidance state set,  $A_1$ .

### 3.4 Conclusion

In this paper, we propose a method to collect and revise UAV flight data through a user with a UAV controller, and path adjusting and planning techniques for collision avoidance of UAVs. When multiple UAVs are controlled by a ground control system using the proposed method, the UAVs have less likelihood to be in a close encounter with obstacles and other UAVs. Therefore, the system would have less burden to consider the collision issues than traditional algorithms.

**Acknowledgments.** This work was supported by the Korea Foundation for the Advancement of Science & Creativity (KOFAC), and funded by the Korean Government (MOE).

### References

1. Fahlstrom, P.G., Gleason, T.J.: Introduction to UAV Systems, 4th edn. Wiley, Chichester (2012)
2. Bellingham, J.S., Tillerson, M., Alighanbari, M., How, J.P.: Cooperative path planning for multiple UAVs in dynamic and uncertain environments. In: 41st IEEE Conference on Decision and Control, Las Vegas, Nevada USA, 10–13 December, vol. 3, pp. 2816–2822 (2002)

3. Cacace, J., Finzi, A., Lippiello, V.: Multimodal interaction with co-located drones for search and rescue. In: 2nd Workshop on Artificial Intelligence and Robotics, Ferrara, Italy, 22 September, vol. 1544, pp. 54–67 (2015)
4. Sung, Y., Kwak, J., Yang, D., Park, Y.: Ground station design for the control of multi heterogeneous UAVs. In: Korea Multimedia Society Spring Conference, 28–30 May, vol. 18, no. 1, pp. 829–830. Andong Culture and Art Center, Andong (2015)
5. Douglas, D.H., Peucker, T.: Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Can. Cartographer* **10**(2), 112–122 (1973)
6. Park, J., Oh, H., Tahk, M.: UAV collision avoidance based on geometric approach. In: The Society of Instrument and Control Engineers (SICE) Annual Conference, 20–22 August 2008, pp. 2122–2126. The University of Electro-Communications, Tokyo (2008)