Mechatronics 21 (2011) 876–885

Contents lists available at ScienceDirect

Mechatronics

journal homepage: www.elsevier.com/locate/mechatronics

Coverage path planning for UAVs based on enhanced exact cellular decomposition method

Yan Li *, Hai Chen, Meng Joo Er, Xinmin Wang

Northwestern Polytechnical University, West Youyi Road, #127, Xi'an, P.O. Box 183, Shaanxi 710072, PR China

ARTICLE INFO

Article history:
Available online 2 February 2011

Keywords:
Coverage Path Planning (CPP)
Unmanned Aerial Vehicles (UAVs)
Enhanced exact cellular decomposition
Width calculation
Convex decomposition
Subregion connection

ABSTRACT

In this paper, an enhanced exact cellular decomposition method to plan the coverage path of UAVs in a polygon area is proposed. To be more specific, the contributions of the paper are: firstly, the turning motion of UAVs is shown to be less efficient from the viewpoints of route length, duration and energy. Secondly, the problem of Coverage Path Planning (CPP) in a convex polygon area is transformed to width calculation of the convex polygon, and a novel algorithm to calculate the widths of convex polygons with time complexity of $O(n)$ is developed. The path of the least number of turns for an UAV based on the widths of convex polygons is devised. Thirdly, a convex decomposition algorithm for minimum width sum based on the greedy recursive method which revolves around decomposing the concave area into convex subregions is developed. It is proved that the algorithm is a polynomial time algorithm. To avoid unnecessary back and forth motion, some entirely adjacent subregions are combined. Finally, comparing different weights of two joint-points, a subregion connection algorithm based on minimum traversal of weighted undirected graph is proposed to connect the coverage paths of the subregions. Simulation results show that the proposed method is feasible and effective.

© 2010 Elsevier Ltd. All rights reserved.

1. Introduction

Over the past decade, there has been a great demand of Unmanned Aerial Vehicles (UAVs) in numerous industrial and military operations around the world. UAVs are often deployed for missions that are too "dull, dirty, or dangerous" for manned aircraft [1]. As part of mission planning, path planning technologies will play a pivotal role to improve viability and mission ability of UAVs. Usually, path planning refers to a "point-to-point" mission whose solution determines a path between start and goal points [2]. However, when the position of a reconnaissance target is uncertain or when the point of interest concerns the information of a certain area, the region Coverage Path Planning (CPP) should be carried out. The region coverage is defined as the sensor footprint covering all the points in a given area [3]. The CPP applications of UAVs mainly include security monitoring, battlefield surveillance, target search, terrain mapping, mineral exploration, etc.

In recent years, CPP researches have focused on ground-based vehicles. Although some techniques are effective for ground-based vehicles, they may not be applicable directly to UAVs since the coverage paths of ground-based vehicles allow more turns [4–7]. UAVs should reduce the number of turns because route length, duration and energy consumption will increase with increasing number of turns (This claim will be proved in Section 2).

The author of [8] has shown that several existing algorithms which are called exact cellular decomposition take the following basic approach in generating a coverage path. Firstly, the region to be covered is decomposed into subregions. Next, a traveling-salesman algorithm is applied to generate a sequence of subregions to visit and in turn a coverage path is generated from this sequence that covers each subregion. Finally, the subregions are individually covered using a back and forth motion which is called sweep method.

Most of the above algorithms employ the same sweep direction in all subregions which may not be able to obtain optimal results [2,9]. In addition, some of the algorithms are applicable for some specific regions only. When the region is changed, the algorithm is no longer feasible [8,10]. Moreover, to our knowledge, the problem of subregion connection, in particular, the joint-point differences of subregions, which will impact the weights of undirected graph, has not been addressed in the literatures.

Since the area to be covered is usually a polygon area, and most of the non-polygon areas can be transformed to polygon areas by polygonal approximation, this paper will focus on CPP in polygon area. Specifically, this paper presents an enhanced exact cellular decomposition method which can optimally solve the problem of CPP in the polygon area. The major contributions of this paper

* Corresponding author. Tel./fax: +86 029 88431303.
E-mail address: liyan@nwpu.edu.cn (Y. Li).

0957-4158/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.mechatronics.2010.10.009

are: firstly, the turning motion of UAVs is shown to be less efficient from the viewpoints of route length, duration and energy. Secondly, the problem of CPP in a convex polygon area is transformed to width calculation of the convex polygon, and a novel algorithm to calculate the widths of convex polygons with time complexity of $O(n)$ is developed. The path of the least number of turns for an UAV based on the widths of convex polygons is devised. Thirdly, a convex decomposition algorithm for minimum width sum based on the greedy recursive method which revolves around decomposing the concave area into convex subregions is developed. It is proved that the algorithm is a polynomial time algorithm. To avoid unnecessary back and forth motion, some entirely adjacent subregions are combined. Finally, comparing different weights of two joint-points, a subregion connection algorithm based on minimum traversal of weighted undirected graph is proposed to connect the coverage paths of the subregions. By virtue of the proposed methods, a complete coverage path can be devised.

This paper is organized as follows: Section 2 presents the proof that the turning motion is less efficient. Section 3 proposes the method of CPP in convex polygon area. Section 4 studies convex decomposition for minimum width sum as well as combination of subregions. In Section 5, the subregion connection problem is solved. An illustrative example and simulation results are given in Section 6. Section 7 concludes the paper.

2. Analysis of turning motion

In this paper, we assume that UAVs with the camera are particles. In this section, we show that the turning motion is less efficient compared with the flat flying motion and this serves as a motivation for this work.

2.1. Route length and flight duration

Compared with ground-based vehicles, the coverage process of UAVs is executed in a three-dimensional space. The camera footprint of an UAV is shown in Fig. 1. As depicted in Fig. 1, the stereoscopic image is acquired via the front-mounted angle between the UAV and the camera.

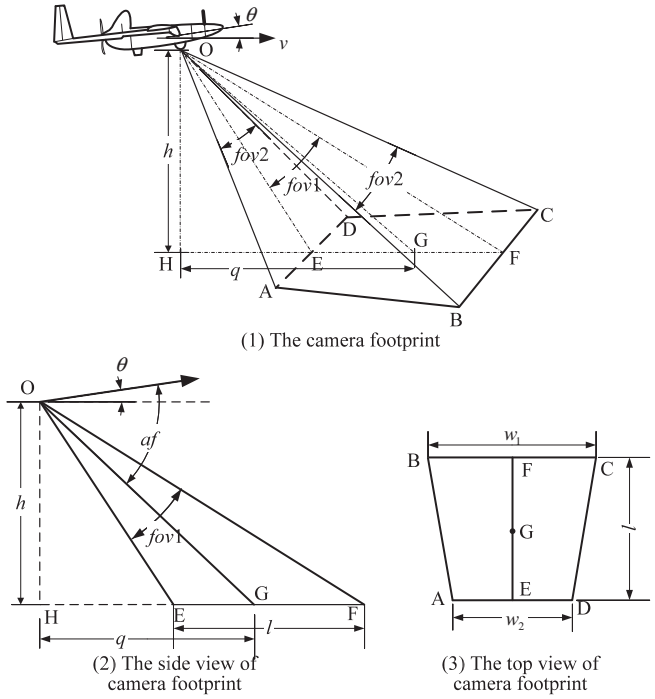

Fig. 1. The camera footprint of an UAV.

The trapezium ABCD in Fig. 1 is the camera footprint. Fig. 1-(2) is the side view of the camera footprint and Fig. 1-(3) is the top view. Point H is the projection of the camera in ground reference as in [11].

Variables in Fig. 1 are defined as follows:

h	Flight altitude.
$fov1$	Vertical angle of FOV (Field of View).
$fov2$	Horizontal angle of FOV.
w_1	Top width of camera footprint.
w_2	Bottom width of camera footprint.
l	Length of camera footprint.
θ	Pitch angle, $\theta \in (0, \frac{\pi}{2})$.
v	Flight velocity.
af	Front-mounted angle which is the included angle between longitudinal axis of the UAV and the bisector of $fov1$.
q	Distance between the projection of camera and the center of camera footprint and q is defined by

$$q = HG = HF - l/2 = \frac{h}{2}\left[\cot\left(af - \theta + \frac{fov1}{2}\right) + \cot\left(af - \theta - \frac{fov1}{2}\right)\right] \quad (1)$$

To avoid the blurry problem of image edges, we define the bottom width of camera footprint w_2 as the sweep breadth of sensor w:

$$w_2 = \frac{2h\tan(fov2/2)}{\sin(af - \theta + fov1/2)} = w \quad (2)$$

As shown in Fig. 1, the projection of the UAV and the center of camera footprint do not coincide with each other. The UAV has to fly beyond the boundary to make a U-turn in order to prevent missing the coverage area (shown in Fig. 2).

In Fig. 2, A_e, B_e, C_e and D_e are the center points of the camera footprint. Corresponding to the above points, A_a, B_a, C_a and D_a are the projections of the UAV on the ground respectively. The distance between a pair of the above points is q, i.e.

$$\overline{A_aA_e} = \overline{B_aB_e} = \overline{C_aC_e} = \overline{D_aD_e} = q \quad (3)$$

and q can be obtained by Eq. (1).

The solid line $A_aB_aC_aD_a$ represents the projection path of the UAV on the ground reference and the dot-dash line $A_eB_eC_eD_e$ represents the path of the camera footprint center. Note that the footprint path beyond the boundary is ignored and the line segment $\overline{B_eC_e}$ is a virtual path rather than the real path.

As shown in Fig. 2, the length of the flight path is longer than the length of the camera footprint center path in each turning motion. It is clear that the difference of the two lengths Δd is given by:

$$\Delta d = 2q + (\pi/2 - 1)w \quad (4)$$

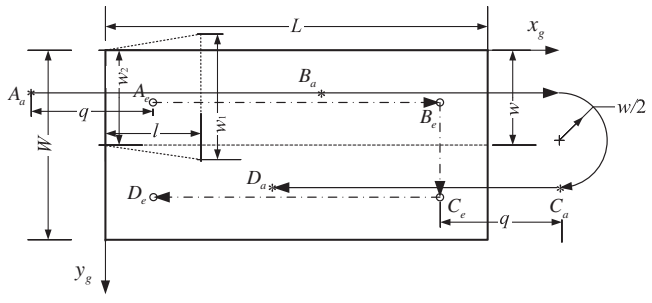

Fig. 2. Flight path of the UAV beyond the boundary in order to prevent missing the coverage area.

Therefore, the task to reduce the total length of the flight coverage path is to decrease the number of turns. Consequently, the total flight duration of coverage is reduced under the same flight velocity.

2.2. Energy consumption

We shall briefly review the steady turning motion in a horizontal plane here (see [12] for details) because it is the routine turning motion of the coverage process. We say that UAV is in the steady turning motion when it is at constant altitude h, velocity v and flight path angle γ. As discussed in [12], the following equations govern the steady turning motion:

Along the flight path:

$$T\cos\beta - D - G\sin\gamma = m\frac{dv}{dt} = 0 \tag{5}$$

Along the principal normal:

$$L\cos\mu - G\cos\gamma = mv\frac{d\gamma}{dt} = 0 \tag{6}$$

Along the binormal:

$$T\sin\beta + L\sin\mu - \frac{mv^2\cos\gamma}{r} = 0 \tag{7}$$

where m is the mass of UAV, $T$is the engine thrust, D is the drag, L is lift, G is the gravity of the UAV, μ is the bank angle, β is the sideslip angle and r is the radius of turn.

Consider a steady coordinated turning motion in the horizontal plane, i.e. $\gamma = 0$ and $\beta = 0$. For simplicity, let us ignore variations of the weight during the turn. It is not difficult to see that Eqs. (5)–(7) assume the following form:

$$D = T \tag{8}$$

$$G = L\cos\mu \tag{9}$$

$$m\frac{v^2}{r} = L\sin\mu \tag{10}$$

It follows from Eq. (9) that the load factor in the turning motion is governed by

$$n_z = \frac{L}{G} = \frac{1}{\cos\mu} \tag{11}$$

From above equation, it is apparent that the lift $L = G/\cos\mu$ is always larger than gravity G in the turning motion. The lift coefficient C_L can therefore be expressed as:

$$C_L = \frac{n_z G}{QS} = \frac{2n_z G}{\rho v^2 S} \tag{12}$$

where ρ is the air density, S is the reference wing area and $Q = \rho v^2/2$ is the dynamic pressure.

Thus, the drag of the UAV in the turning motion is given by

$$\begin{aligned} D &= \frac{1}{2}\rho v^2 C_D S = \frac{1}{2}\rho v^2 S(C_{D_0} + KC_L^2) \\ &= \frac{1}{2}\rho v^2 S\left[C_{D_0} + K\left(\frac{n_z G}{\frac{1}{2}\rho v^2 S}\right)^2\right] = D_0 + n_z^2 D_t \end{aligned} \tag{13}$$

where C_{D_0} is the zero-lift drag coefficient, K is the induced drag factor, D_0 and D_t is the zero-lift drag and induced drag under the same altitude and velocity during flat flying.

It can be observed from Eq. (13) that the drag of turning motion is larger than the drag of flat flying under the same altitude and velocity. To satisfy Eq. (8), the engine thrust should be increased and so does the fuel consumption. It is obvious that the fuel consumption can be reduced by decreasing the number of turns.

3. CPP in convex polygon area

In this paper, the convex polygon area is the basic region to be covered and the CPP in the convex polygon area is essential to the CPP in complex area which will be decomposed into some convex polygon subregions. Similar to most of the exact cellular decomposition methods, the back and forth motion is selected to cover the convex polygon area in this work.

3.1. Problem transformation

The total length of flight coverage path l_{all} in convex polygon area A can be rewritten as

$$l_{all} = l_{in} + l_{out} \tag{14}$$

where l_{in} is the length of the flight path inside A and l_{out} is the length of the flight path outside A.

Suppose that the UAV searches A unrepeatedly with a constant velocity v. Then, we have

$$l_{in} = \sum_{k=1}^{n_{turn}-1} l_{ink} = area(A)/w \tag{15}$$

where l_{ink} is the length of the flight path k inside A, $area(A)$ is the area of A and n_{turn} is the number of turns.

To obtain steady high-resolution images, the sweep breadth w must be a constant. Since the area of A is a constant, l_{in} must be a constant. Based on the above analysis, the total length of the flight path is therefore dependent on l_{out} only which is defined by:

$$l_{out} = \Delta d^* n_{turn} \tag{16}$$

where Δd can be considered as a constant which is determined by the characteristics of the UAV and the camera. Hence, the total length of the flight path is only dependent on n_{turn}, i.e. the total length will decrease with decreasing number of turns n_{turn}.

Before giving a formula to calculate n_{turn}, we shall define the concept of span and width in convex polygons.

Definition 1. The span (D) of a convex polygon (P) is the distance between a pair of support parallel lines (l_1, l_2) (see Fig. 3).

A line of support l is a line intersecting P and such that the interior of P lies to one side of l [13].

Definition 2. The width (W) of a convex polygon is the minimum span. Now, the support line can be represented as l_{width}.

From the Definitions 1 and 2, it is clear that n_{turn} of the UAV in a convex polygon is defined by

$$n_{turn} = \langle D/w \rangle \tag{17}$$

where D is the span of the convex polygon, and $\langle * \rangle$ refers to the smallest integer larger than or equal to $*$. n_{turn} will decrease with the decreasing of span D.

Thus, we can get the least number of turns in convex polygon area through the calculation of minimum span (width) of the convex polygon. If the UAV flies along the direction of support line l_{width}, the optimal coverage path can be obtained.

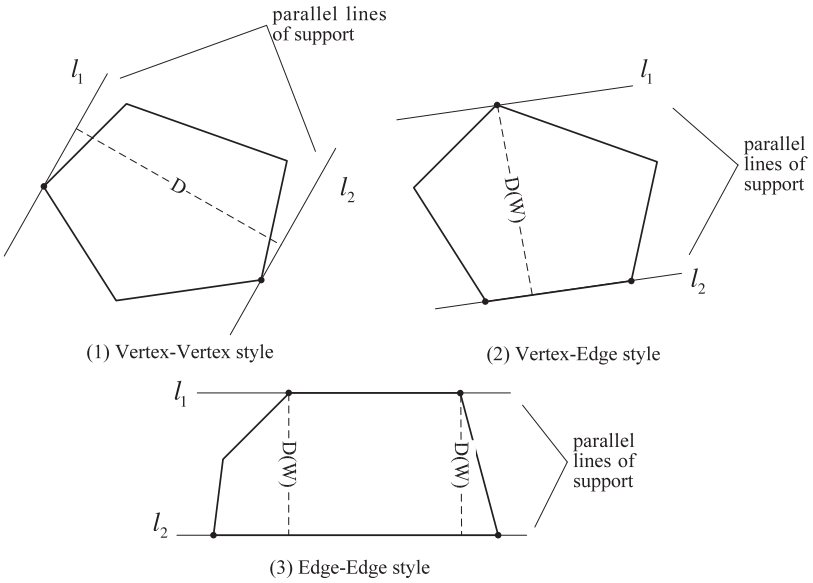

Fig. 3. The span and width of a convex polygon.

3.2. Acquisition of the width of a convex polygon using a linear time algorithm

A convex polygon admits parallel lines of support in any direction, and for each direction the span is (usually) different [14]. But the width is invariable for a convex polygon.

There are three cases for the span of a convex polygon: vertex–vertex (V–V) style (Fig. 3-(1)), vertex–edge (V–E) style (Fig. 3-(2)) and edge–edge (E–E) style (Fig. 3-(3)). The V–V style means that the lines of support intersect the polygon at two vertices only. The V–E style means that one line of support intersects the polygon at an edge while the other line of support is tangent at a vertex only. The E–E style means that the lines of support intersect the polygon at parallel edges [13]. The E–E style can be considered as a special case of the V–E style. Fig. 3-(2) and -(3) show that the minimum span is equal to the width of a convex polygon.

Theorem 1. *The width of a convex polygon P only appears in the V–E style span (E–E style can be considered as a special case of V–E style) [13].*

Theorem 1 *indicates that the width is only obtained for V–E style and E–E style.*

By Theorem 1, *a basic algorithm for obtaining the width based on the V–E style is given below:*

(A) Let P is a convex polygon with n vertices in standard form, i.e., the vertices $V = \{v_1, v_2, \ldots, v_n\}$ are specified according to Cartesian coordinates in a counter-clockwise order and no three consecutive vertices are collinear.
(B) Calculate the distances $d_i (i = 1, 2, \ldots, n-2)$ between the first edge $\overline{v_1 v_2}$ and the vertices $(v_3, v_4, \ldots, v_n)$ which do not belong to $\overline{v_1 v_2}$.
(C) Find the maximum value of d_i to get the span of $\overline{v_1 v_2}$ (D_1).
(D) Calculate the span of the other edges $D_j (j = 2, 3, \ldots, n)$ by repeating Step 2 and Step 3 similarly.
(E) Find the minimum value of D_i to get the width of the convex polygon (W).

The time complexity of above algorithm is $O(n^2)$.

The above algorithm involves great calculation. In the following we present a way to reduce the calculation.

In fact we can get the span of $\overline{v_i v_{i+1}}$ by comparing the distance difference between two adjacent vertices except v_i and v_{i+1}.

The distance difference from the edge $\overline{v_i v_{i+1}}$ to two adjacent vertices $v_j(x_j, y_j)$ and $v_{j+1}(x_{j+1}, y_{j+1})$ is defined by

$$\Delta \mathrm{dis}_{ij} = \frac{\left|(y_{i+1} - y_i)x_{j+1} - (x_{i+1} - x_i)y_{j+1} + x_{i+1}y_i - x_i y_{i+1}\right|}{\sqrt{(y_{i+1} - y_i)^2 + (x_{i+1} - x_i)^2}} - \frac{\left|(y_{i+1} - y_i)x_j - (x_{i+1} - x_i)y_j + x_{i+1}y_i - x_i y_{i+1}\right|}{\sqrt{(y_{i+1} - y_i)^2 + (x_{i+1} - x_i)^2}} \tag{18}$$

The following theorem and corollary shows that the two numerators of fractions in Eq. (18) have the same sign:

Theorem 2. *Let $ax + by + c = 0$ be a line, and $A(x_1, y_1)$, $B(x_2, y_2)$ are two points in the plane. We can obtain*

(A) $ax_1 + by_1 + c = ax_2 + by_2 + c = 0 \Leftrightarrow A(x_1, y_1)$ *and* $B(x_2, y_2)$ *locate in the line* $ax + by + c = 0$.
(B) $(ax_1 + by_1 + c)(ax_2 + by_2 + c) > 0 \Leftrightarrow A(x_1, y_1)$ *and* $B(x_2, y_2)$ *locate on the same side of* $ax + by + c = 0$.
(C) $(ax_1 + by_1 + c)(ax_2 + by_2 + c) < 0 \Leftrightarrow A(x_1, y_1)$ *and* $B(x_2, y_2)$ *locate on the opposite side of* $ax + by + c = 0$.

The proof for Theorem 2 *is straightforward and omitted.*

Corollary 1. *Choose a random edge $\overline{v_i(x_i, y_i)\, v_{i+1}(x_{i+1}, y_{i+1})}$ $(i \in [1, n])$ of convex polygon P with n vertices. The $(n-2)$ polynomials $(y_{i+1} - y_i)x_j - (x_{i+1} - x_i)y_j + x_{i+1}\ y_i - x_i y_{i+1}$ $(j = 1, \ldots, i-1, i+2, \ldots, n)$ have the same signs.*

Proof. Extend the edge $\overline{v_i v_{i+1}}$ to a line whose equation is $(y_{i+1} - y_i)x - (x_{i+1} - x_i)y + x_{i+1}y_i - x_i y_{i+1} = 0$. It is obvious that the vertices of the convex polygon except two terminal points of $\overline{v_i v_{i+1}}$ locate on the same side of the line. By Theorem 2, Corollary 1 is achieved. □

Thus, Eq. (18) can be rewritten as

$$\Delta \mathrm{dis}_{ij} = \frac{(y_{i+1} - y_i)(x_{j+1} - x_j) - (x_{i+1} - x_i)(y_{j+1} - y_j)}{\sqrt{(y_{i+1} - y_i)^2 + (x_{i+1} - x_i)^2}} \tag{19}$$

Therefore, the span of edge $\overline{v_i v_{i+1}}$ can be obtained by the sign change of Δdis_{ij}.

Since we only need to judge the sign of Δdis_{ij}, Eq. (19) can be simplified to

$$\Delta dis'_{ij} = (y_{i+1} - y_i)(x_{j+1} - x_j) - (x_{i+1} - x_i)(y_{j+1} - y_j) \tag{20}$$

For example, since the sign changes from $\Delta dis'_{16}$ to $\Delta dis'_{17}$ in Fig. 4, we only need to calculate $\Delta dis'_{13}, \Delta dis'_{14}, \Delta dis'_{15}, \Delta dis'_{16}$ and $\Delta dis'_{17}$ to get the span of edge $\overline{v_1 v_2}$. The span of $\overline{v_1 v_2}$ is the distance between the vertex v_7 and the edge $\overline{v_1 v_2}$ and the antipodal vertex of the edge $\overline{v_1 v_2}$ is v_7.

Theorem 3. *Consider a convex polygon P whose vertices $V = \{v_1, v_2, \ldots, v_n\}$ are specified according to Cartesian coordinates in a counter-clockwise order. If the antipodal vertex of the edge $\overline{v_i v_{i+1}}$ is v_v, the antipodal vertex of the edge $\overline{v_{i+1} v_{i+2}}$ is v_v or the subsequent vertices.*

Proof. As shown in Fig. 5, the antipodal vertex of the edge $\overline{v_i v_{i+1}}$ is v_v. From Definition 1, the whole vertices are located between the line l_3 and line $\overline{v_i v_{i+1}}$. From the properties of convex polygon [15], it is clear that the vertices from v_{i+2} to v_v are located in the shadow region.

Construct a line l_2 in parallel to l_1 through v_v. Since the shadow region is located between l_2 and l_1, the distances between vertices $(v_{i+2}, v_{i+3}, \ldots, v_v)$ and l_1 are less than the distance between v_v and l_1.

Therefore, the antipodal vertex of the edge $\overline{v_{i+1} v_{i+2}}$ is v_v or the subsequent vertices. □

Based on Theorem 3, an enhanced algorithm to obtain the width of a convex polygon is given as follows:

(A) Find the span D_1 and antipodal vertex v_v of edge $\overline{v_1 v_2}$ by comparing the distance differences between two adjacent vertices in turn.

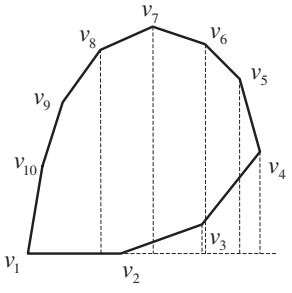

Fig. 4. Span calculation based on the judgement of distance.

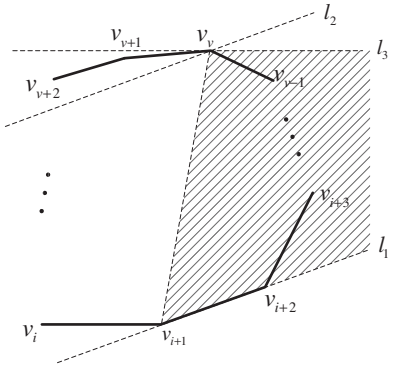

Fig. 5. The sketch map of Theorem 3.

(B) Continue to find the span and antipodal vertex of edge $\overline{v_2 v_3}$ by comparing the distance difference between two adjacent vertices from vertex v_v.
(C) Find the span of each edge in turn by repeating Step 2,
(D) Search the minimum value of all spans to obtain the width of the convex polygon.

Theorem 4. *The time complexity of the above algorithm is O(n), where n is the number of vertices.*

Proof. The above algorithm defines two pointers: one points to the edge of convex polygon while the other points to the vertex. The first pointer traverses all the edges of the convex polygon just once, and the second pointer traverses all the vertices no more than two times in the worst case. Each traversal only ends up calculating Eq. (20) once. Hence, the time complexity of the above algorithm is $O(n)$.

If an UAV flies along the vertical direction of width l_{width}, the coverage path can get the least number of turns. □

4. Convex decomposition of concave polygon

The strategy of flying along the vertical direction of width is not suitable to the concave polygon. Fig. 6 shows a simple example where assigning a different sweep direction to each subregion results in a better coverage path [8].

According to the definition of width, the width of polygon is equal to its altitude in Fig. 6. If an UAV flies along the vertical direction of width, there are 11 U-turns (Fig. 6-(1)). The optimal coverage path is shown in Fig. 6-(2). There are 8 U-turns and one right-angled turn.

To obtain the minimum number of turns in each subregion, the concave polygon area will be decomposed into convex subregions according to the principle of minimum sum of widths.

Let P is a concave polygon with n vertices and the concave vertices are $V_{cc} = \{v_1, v_2, \ldots, v_{ncc}\}$ where $ncc \in [0, n-2)$ is the number of concave vertices. After convex decomposition of P, we will get m convex polygons whose widths are $W_1, W_2, \ldots, W_m$ respectively. Thus, the convex decomposition of the concave polygon with the minimum sum of widths can be described by:

$$\begin{cases} \min & \sum_{i=1}^{m} W_i \\ \text{s.t.} & P_i \in \text{convex polygon} \end{cases} \tag{21}$$

Since the convex decomposition of the concave polygon with the minimum sum of widths is a NP-hard problem, we can only solve it by an approximate algorithm with polynomial time complexity.

4.1. Judgment of concave vertices

In this paper, the cross product of vectors is used to judge the concave vertices. Let $v_i(x_i, y_i)$ be a random vertex of polygon. v_{i+1} (x_{i+1}, y_{i+1}) and $v_{i-1}(x_{i-1}, y_{i-1})$ are adjacent vertices of v_i. The cross product of vectors is defined by

$$\boldsymbol{F} = v_{i-1} v_i \times v_i v_{i+1} \tag{22}$$

Accordingly, we can get the concave judgment matrix:

$$S(v_i) = \begin{vmatrix} x_{i-1} & y_{i-1} & 1 \\ x_i & y_i & 1 \\ x_{i+1} & y_{i+1} & 1 \end{vmatrix} = \begin{cases} > 0, & v_i \text{ is a convex vertex} \\ = 0, & v_{i-1}, v_i \text{ and } v_{i+1} \text{ are collinear} \\ < 0, & v_i \text{ is a concave vertex} \end{cases} \tag{23}$$

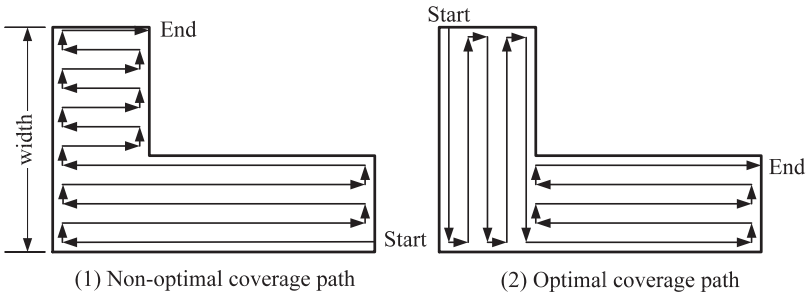

Fig. 6. The comparison of two CPP methods.

4.2. Type of convex decomposition

As shown in Fig. 7, there are four types of convex decomposition with minimum width sum in the concave polygon.

The four 'types' of convex decomposition with minimum width sum are:

(A) Extension from one edge of a concave vertex.
(B) Connection between two concave vertices.
(C) Make a line which passes a concave vertex and in parallel to one edge.
(D) Connection between a concave vertex and a convex vertex.

Fortunately, the common ground of the above type is the decomposition line in parallel to one edge of the polygon. Based on the above idea, an algorithm to obtain the minimum sum of widths of subregions is developed.

4.3. Computation of minimum width sum based on the greedy recursive method

The basic idea of the greedy recursive method is shown as follows:

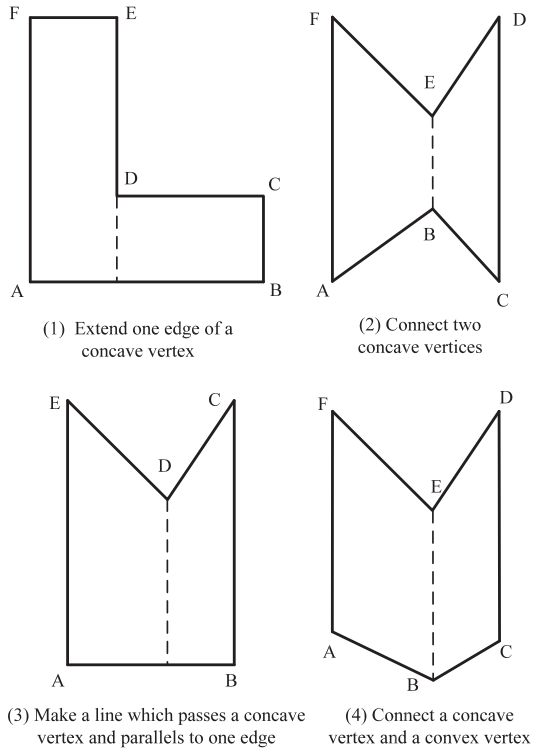

Fig. 7. The four types of concave decomposition with minimum width sum.

(A) Divide the problem of convex decomposition into some similar smaller sub-problems.
(B) Obtain an optimal solution of each sub-problem based on the recursive method.
(C) Obtain an approximate optimal solution of the whole problem by solutions of sub-problems.

The flowchart of the algorithm is shown in Fig. 8.

4.4. Time complexity analysis of proposed algorithm

Time complexity of the convex decomposition algorithm depends on two variables, namely the number of total vertices *n*

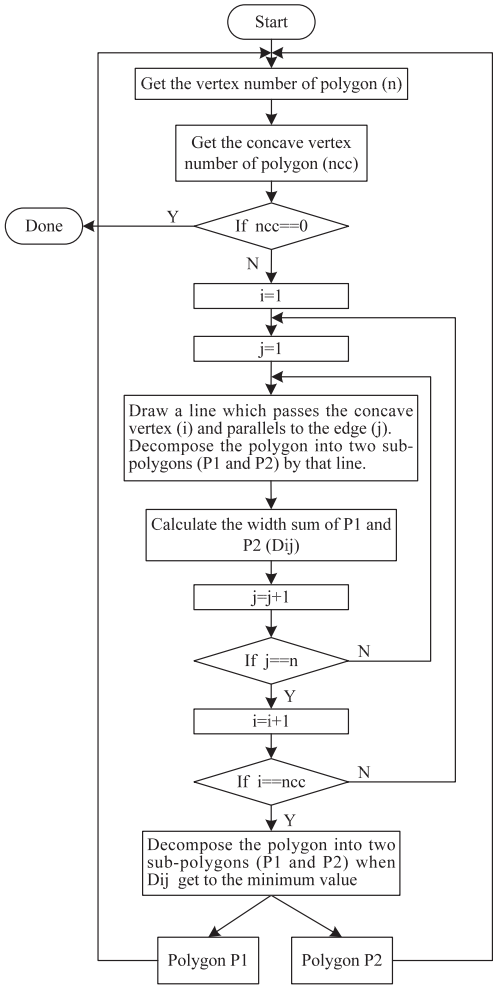

Fig. 8. Flowchart of the convex decomposition algorithm.

and the number of concave vertices *ncc*. The function of time complexity $T(n,ncc)$ is defined by

$$T(n,ncc) \leqslant \begin{cases} C_1 n, & ncc = 0 \\ T(n_1,ncc_1) + T(n_2,ncc_2) + C_2 \cdot ncc \cdot n^2, & ncc > 0 \end{cases} \tag{24}$$

If $ncc = 0$, the input polygon P is a convex polygon and does not require the decomposition. Since the time complexity of the concave vertex judgment is $O(n)$, the first formula of Eq. (24) is achieved.

If $ncc > 0$, the input polygon P is a concave polygon and requires the decomposition. The time complexity of the operation to decompose P into two sub-polygons is $O(n)$. In order to find the decomposition with minimum width sum, there are $n \cdot ncc$ decomposing operations. Thus, the complexity of an optimal decomposition to decompose P into two sub-polygons with minimum width sum is $O(ncc \cdot n^2)$. Hence, the last part $C_2 \cdot ncc \cdot n^2$ in the second formula of Eq. (24) is established.

An operation of decomposition can reduce a concave vertex so that

$$ncc_1 + ncc_2 = ncc - 1 \tag{25}$$

However, the total number of vertices of two sub-polygons $(n_1 + n_2)$ is probably larger than the number of vertices of original polygon (n). As shown in Fig. 9, there are four kinds of relationships between $(n_1 + n_2)$ and n.

Hence, $n_1$and n_2 of Eq. (24) satisfy

$$n \leqslant n_1 + n_2 \leqslant n + 3 \tag{26}$$

Since the minimum number of vertices of a convex polygon is 3, we have

$$0 \leqslant n_1, n_2 \leqslant n \tag{27}$$

Accordingly, Eq. (24) can be rewritten as follows:

$$T(n,ncc) \leqslant T(n,ncc_1) + T(n,ncc_2) + C_2 \cdot ncc \cdot n^2 \tag{28}$$

The worst case of the greedy recursive algorithm is shown in Fig. 10, where the polygon is decomposed into a convex polygon and a concave polygon in every operation of decomposition.

As a consequence, we have

$$\begin{aligned} T(n,ncc) &\leqslant T(n,ncc_1) + T(n,ncc_2) + C_2 \cdot ncc \cdot n^2 \quad \mathrm{ncc}_1 + \mathrm{ncc}_2 = ncc - 1 \\ &\leqslant T(n,0) + T(n,ncc-1) + C_2 \cdot ncc \cdot n^2 \\ &\leqslant T(n,0) + [T(n,0) + T(n,ncc-2) + C_2 \cdot (ncc-1) \cdot n^2] \\ &\quad + C_2 \cdot ncc \cdot n^2 \\ &= 2T(n,0) + T(n,ncc-2) + C_2 \cdot n^2 \cdot [ncc + (ncc-1)] \\ &\vdots \\ &\leqslant (ncc-1) \cdot T(n,0) + T(n,1) + C_2 \cdot n^2 \cdot [ncc + (ncc-1) + \cdots + 3 + 2] \\ &\leqslant (ncc+1) \cdot T(n,0) + C_2 \cdot n^2 \cdot \sum_{i=1}^{ncc} i \\ &= C_1 \cdot (ncc+1) \cdot n + \frac{C_2}{2} \cdot n^2 \cdot ncc^2 + \frac{C_2}{2} \cdot n^2 \cdot ncc \\ &= O(n^2 \cdot ncc^2) \end{aligned}$$

and the proposed algorithm of convex decomposition is indeed a polynomial time algorithm.

4.5. Combinations of subregions

When the proposed algorithm decomposes the polygon into unnecessary number of convex subregions, it requires redundant back and forth motion to guarantee completeness. For example, in the left hand side of Fig. 11, the UAV needs to make an additional length-way motion to cover the remaining portion of the trapezoidal subregion. This can be viewed as part of the cost in guaranteeing that the UAV exhaustively covers the entire environment [9].

To avoid unnecessary back and forth motion, two subregions, which satisfy the following conditions, will be combined:

- The widths of two subregions have the same direction.
- The two subregions are entire adjacent to each other.

In the sequel, adjacency and entire adjacency of two polygons are defined.

Definition 3. Consider two polygons P_1 with n vertices and P_2 with m vertices. If an edge of P_1, $v_{1i}v_{1(i+1)}(i \in [1,n])$, coincides with an edge of P_2 $v_{2j}v_{2(j+1)}(j \in [1,m])$,$P_1$ is adjacent to P_2.

Definition 4. P_1 and P_2 are adjacent to each other, whose coincidence edges are $v_{1i}v_{1(i+1)}$ and $v_{2j}v_{2(j+1)}$ respectively. If $v_{1i} = v_{2j}$ (or

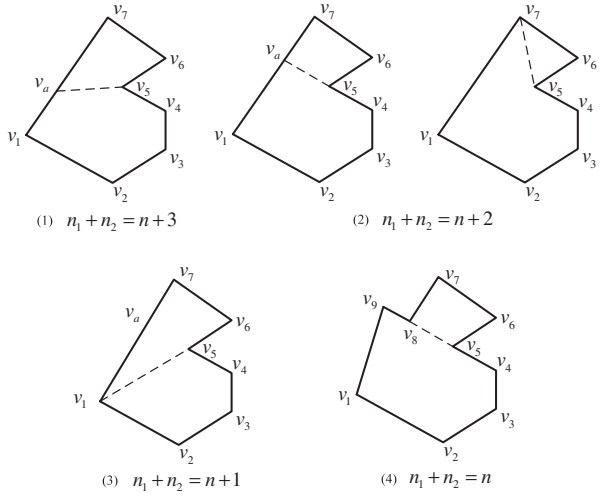

Fig. 9. Relationships between the total number of vertices of two sub-polygons $(n_1 + n_2)$ and the number of vertices of original polygon (n).

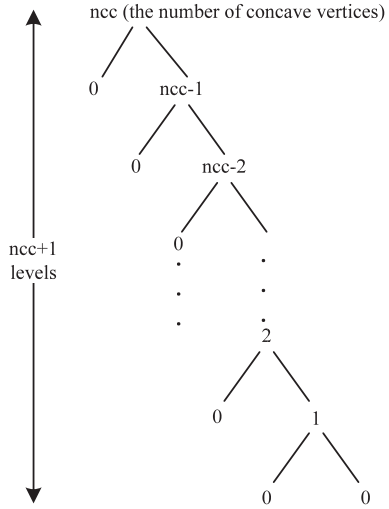

Fig. 10. The worst case of decomposition.

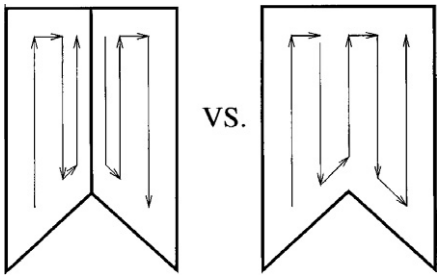

Fig. 11. Fewer subregions is better.

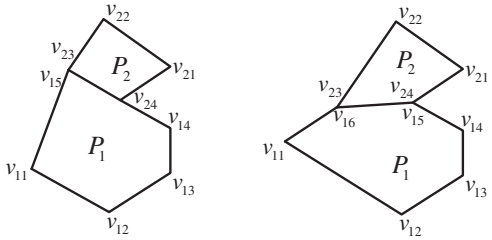

Fig. 12. The adjacency and entire adjacency of two polygons.

$v_{1i} = v_{2(j+1)}$) and $v_{1(i+1)} = v_{2(j+1)}$ (or $v_{1(i+1)} = v_{2j}$), P_1 is entirely adjacent to P_2.

Illustration of Definitions 3 and 4 are shown in Fig. 12.

5. Subregions connection based on minimum traversal of weighted undirected graph

After the coverage of each convex subregion, the connection of subregions has to be performed.

The connection of subregions is converted to the minimum traversal of weighted undirected graph in this paper. The distance of two joint-points of subregions is the weight of undirected graph. However, compared with the general weighted undirected graph, the weights of undirected graph proposed in this paper will vary with differences of joint-points.

As shown in Fig. 13, the connecting approach in Fig. 13-(2) is better than the approach in Fig. 13-(1).

If the width of a convex subregion is determined by V–E style, there are three joint-points in the subregion. If the width of a convex subregion is determined by E–E style, there are four joint-points in the subregion. In this paper, we assume that there are four joint-points in all subregions. The case of three joint-points can be regarded as the coincidence of two joint-points.

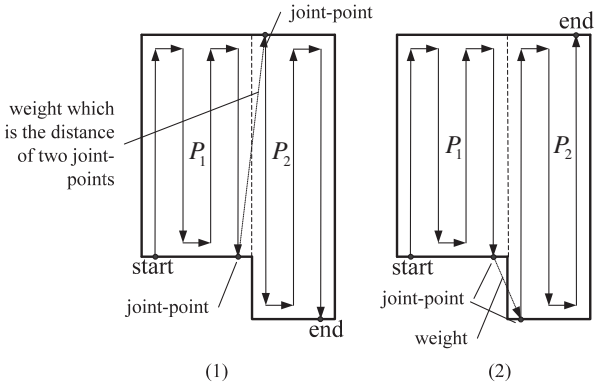

Fig. 13. The weights will vary with differences of joint-points.

The matrix of joint-points can be constructed as follows:

$$C(P_i) = \begin{bmatrix} c_{i,11} & c_{i,12} \\ c_{i,21} & c_{i,22} \end{bmatrix} \tag{29}$$

where $c_{i,11}$, $c_{i,12}$, $c_{i,21}$ and $c_{i,22}$ are joint-points of subregion P_i in a counter-clockwise order and they satisfy

$$\begin{cases} \overline{c_{i,11}c_{i,21}} \| l_{width}, & c_{i,12} = c_{i,22} \\ \overline{c_{i,12}c_{i,22}} \| l_{width}, & c_{i,11} = c_{i,21} \\ \overline{c_{i,11}c_{i,21}} \| \overline{c_{i,12}c_{i,22}} \| l_{width}, & (c_{i,11} \neq c_{i,21}) \text{ and } (c_{i,12} \neq c_{i,22}) \end{cases} \tag{30}$$

where l_{width} is the support line along the vertical direction of the width.

According to the parity of n_{turn}, the incoming joint-points and outgoing joint-point of a subregion have different relationships:

$$\begin{cases} c_{i,11} \rightleftarrows c_{i,12}, c_{i,21} \rightleftarrows c_{i,22}, & n_{turn} \text{ is odd number} \\ c_{i,11} \rightleftarrows c_{i,22}, c_{i,21} \rightleftarrows c_{i,12}, & n_{turn} \text{ is even number} \end{cases} \tag{31}$$

An incoming joint-point and an outgoing joint-point are located on the opposite sides of $\rightleftarrows$. If the incoming joint-point is fixed, the outgoing joint-point is fixed accordingly.

The basic idea of subregion connection based on different weights of joint-points is shown as follows:

Convert the subregion $P_k(k = 1,2,\cdots m)$ to a node. If $P_i \cap P_j \neq \emptyset (i, j \in [1,m])$, P_i is connected to P_j with an edge whose weight is d_{ij}. If $P_i \cap P_j = \emptyset$, there is no connection. Consequently, a weighted undirected graph G is constructed. We can select the starting point and the end point to traverse graph G. The traversal with minimum weights is the optimal subregion connection.

The weight between P_i and P_j is determined by

$$d_{ij} = \begin{cases} \min(distance(c_{i\_out}, c_{j,uv})), & P_i \cap P_j \neq \emptyset; i,j \in [1,m]; u, v \in [1,2] \\ \infty, & (P_i \cap P_j = \emptyset) \cup (P_i = P_j) \end{cases} \tag{32}$$

where c_{i_out} is the outgoing joint-point of subregion P_i, $c_{j,uv}$ is a joint-point of subregion P_j and $distance(c_{i\_out},c_{j,uv})$ is the Euclidean distance between c_{i_out} and $c_{j,uv}$.

Let the sequence number of the starting point and the end point be 1 and m respectively.

Define

$$N_i = \{2, 3, \cdots, i-1, i+1, \cdots, m\} \tag{33}$$

is a set of middle subregions from subregion 1 to i.

Let S be the set of passed subregions before the UAV get to the subregion i. Then, $S \subseteq N_i$.

Choose (i,S) as the state variable of the traversal. The optimal function $f_k(i,S)$ is the minimum weight from the subregion 1 to i through S which contains the k middle subregions. Accordingly, we have the following recursive relation of dynamic programming:

$$\begin{aligned} &f_k(i,S) = \min_{i \in S}[f_{k-1}(j, S \setminus \{j\}) + d_{ij}] \\ &(k = 1,2,\cdots,m-1; i = 2,3,\cdots,m; j \in S; S \subseteq N_i) \end{aligned} \tag{34}$$

where $f_0(i,\emptyset) = d_{1i}$ is terminal condition.

In this paper, the optimal function $f_{m-2}(m,\{2,3,\ldots,m-1\})$ means that the minimum weights from the subregion 1 to m through the middle subregions $\{2,3,\ldots,m-1\}$.

From above recursive relation function, the connection path of subregions with minimum weights can be obtained.

6. Simulation results

Fig. 14 shows an un-covered polygon area $P(n = 13)$ whose vertices (in counter-clockwise order) are:

$$V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}, v_{11}, v_{12}, v_{13}\} = \left\{ \begin{array}{l} (0.45\ 0.75), (2.37\ 1.49), (4.15\ 0.32), \\ (3.63\ 1.48), (5.58\ 1.78), (7.45\ 3.21), \\ (6.12\ 3.58), (4.75\ 6.15), (4.32\ 5.94), \\ (3.75\ 4.55), (2.45\ 6.44), \\ (1.55\ 5.45), (2.51\ 3.67) \end{array} \right\} (km)$$

The sweep breadth w is 0.16 km. Simulation steps using the proposed method are as follows:

(A) Judge concave vertices of P and get its number ($ncc = 5$) and index ($\{2,4,7,10,13\}$).
(B) Decompose the polygon area P into some subregions with the algorithm of minimum width sum based on the greedy recursive method. The result is shown in Fig. 15.
(C) Combine the subregions which satisfy the conditions of Section 4.5. The result is shown in Fig. 16.
(D) Calculate the width and the joint-points of each subregion.
(E) Build the weighted undirected graph $G(m = 7)$ according to Fig. 16. Connect the subregions based on minimum traversal of G. The shortest path of subregion nodes is the bold line in Fig. 17.
(F) Plan the converge path in each subregion and connect them. The final coverage path is shown in Fig. 18.

Above coverage path is the path of the camera footprint center which can be used as the reference path of the UAV.

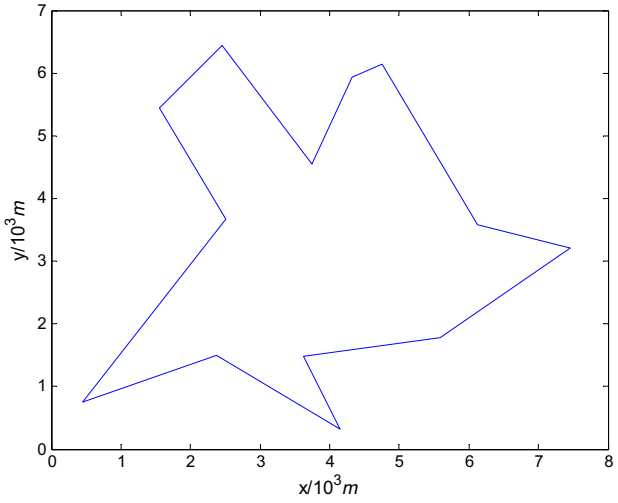

Fig. 14. The uncovered polygon area.

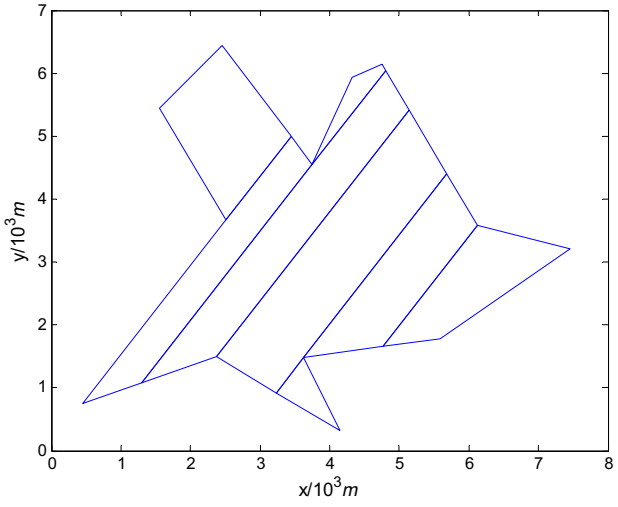

Fig. 15. The result of convex decomposition.

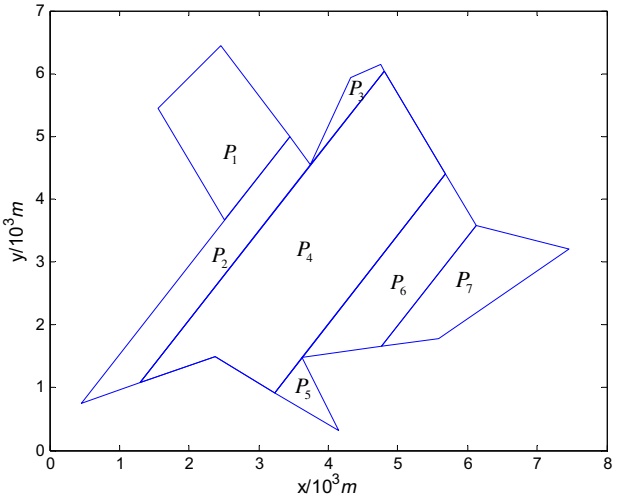

Fig. 16. The combination of subregions.

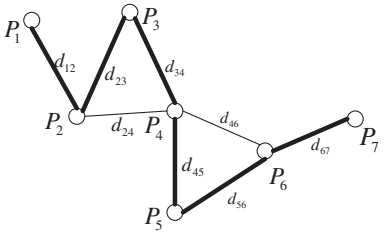

Fig. 17. The undirected graph of subregions.

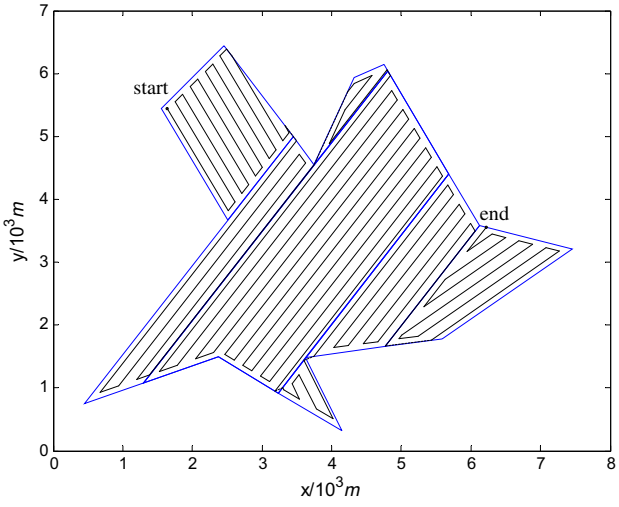

Fig. 18. The final simulation result of coverage path.

7. Conclusions

In this paper, an enhanced exact cellular decomposition method which plans the coverage path in a polygon area for UAVs is proposed. Firstly, the turning motion is proved to be less efficient from the viewpoints of the route length, duration and energy. Next, the optimal criterion for CPP which minimize the number of turns is proposed. A convex decomposition algorithm for minimum width sum based on the greedy recursive method is developed. It is proved that the algorithm is of polynomial time complexity. To obtain the least number of turns, the strategy of flying along the vertical direction of the width is proposed for CPP in the convex subregion. Finally, a minimum traversal algorithm of weighted undirected graph is given to minimize the repeated path of UAVs in the connection process. Simulation results show that the proposed method is feasible and effective.

References

[1] Office of the Secretary of Defense. Unmanned aircraft systems roadmap 2005–2030. Washington DC; 2005.
[2] Latombe JC. Robot motion planning. CA: Kluwer Aeademie; 1991.
[3] Choest H. Coverage for robotics: a survey of recent results. Ann Math Artif Intell 2001;31(1–4):113–26.
[4] Gabriely Y, Rimon E. Spanning-tree based coverage of continuous areas by a mobile robot. In: Proceedings of the 2001 IEEE international conference on robotics & automation, Seoul, Korea, 2001. p. 1927–33.
[5] Gabriely Y, Rimon E. Competitive on-line coverage of grid environments by a mobile robot. Comput. Geom. 2003;24:197–224.
[6] Zelinsky A, Jarvis RA, Byrne JC, Yuta S. Planning paths of complete coverage of an unstructured environment by a mobile robot. In: Proceedings of international conference on advanced robotics, Tokyo, Japan; 1993. p. 533–8.
[7] Carvalho R, Vidai H, Vieira P, et al. Complete coverage path planning and guidance for cleaning robots. In: Proceedings of IEEE international symposium on industrial electronics, Guimaraes, Portugal; 1997. p. 677–82.
[8] huang W. Optimal line-sweep-based decompositions for coverage algorithms. Technical Report 00-3, Rensselaer Polytechnic Institute, Department of Computer Science, Troy (NY); 2000.
[9] Choset H. Coverage of known spaces: the boustrophedon cellular decomposition. Autonom Robots 2000;9(3):247–53.
[10] Butler ZJ, Rizzi AA, Hollis RL. Simulation and experimental evaluation of complete sensor-based coverage in rectilinear environments. In: Proceedings of international symposium of experimental robotics; 2001. p. 417–26.
[11] Blakelock JH. Automatic control of aircraft and missiles. 2nd ed. New York: John wiley & sons, Inc; 1991.
[12] Pamadi Bandu. Performance, stability, dynamics, and control of airplanes. 2nd ed. American Institute of Aeronautics and Astronautics; 2004.
[13] Pirzadeh Hormoz. Computational geometry with the rotating calipers. Master thesis, School of Computer Science, McGill University; 1999.
[14] Chan Timothy M. A fully dynamic algorithm for planar width. Discrete Comput Geom 2003;30(1):17–24.
[15] de Berg M, Cheong O, van Kreveld M, Overmars M. Computational geometry: algorithms and applications. Third ed. Heidelberg, New York: Springer-Verlag; 2008.