

СОДЕРЖАНИЕ

А.О. Степанов

ПОКРЫТИЕ СЛОЖНОГО ПЛОЩАДНОГО МАРШ- РУТА БВС	2
--	---

ПОКРЫТИЕ СЛОЖНОГО ПЛОЩАДНОГО МАРШРУТА БВС

А.О. Степанов

Удмуртский Государственный Университет, г. Ижевск, Россия

e-mail: heimusharper@gmail.ru

В статье рассматривается процесс построения площадного маршрута для БВС самолётного типа. Рассмотрены различные варианты построения исходного маршрута и способы его оптимизации. Рассмотрена проблема образования выпуклых многоугольников и выбран алгоритм разбиения таких структур на более простые маршруты.

Ключевые слова: беспилотное воздушное судно, поддержка принятия решений, составление маршрута, оптимизация маршрута

Введение

Если представить БВС в воздухе, то фотоаппарат или другое техническое устройство будут иметь проекцию, в основном, направленную по нормали к поверхности. При этом эта точка проекции представляет центр некоторой области, которую может контролировать полезная нагрузка. Задача покрытия площади состоит в том, что-бы получить равномерный, возможно с некоторым перекрытием, набор этих областей. [1] Для многих задач требуется пролететь ограниченную область за наименьшее время, в качестве оптимизации эта область может быть сформирована по сложному маршруту с поворотами и самопересечениями.

Базовый маршрут

В большинстве случаев за нагрузку принимается фотоаппарат, который представляет на поверхности земли прямоугольник. Расстояния, между снимками с учётом перекрытия горизонтальное и вертикальное в метрах задаются соотношениями 1 и 2 соответственно. H - высота полёта над подстилающей поверхностью в метрах, m_w и m_h - ширина и высота матрицы фотоаппарата в миллиметрах соответственно, f - фокусное расстояние объектива в

миллиметрах, o_w и o_h - перекрытие горизонтальное и вертикальное в % соответственно.

$$b_w = H \cdot \frac{m_w}{f} \cdot \left(1 - \frac{o_w}{100}\right) \quad (1)$$

$$b_h = H \cdot \frac{m_h}{f} \cdot \left(1 - \frac{o_h}{100}\right) \quad (2)$$

Тогда если S_F - площадь целевой области, а $S_C = \{s_0, s_1, \dots, s_n\}$ - площади сделанных снимков, то нам необходимо стремиться к созданию такого маршрута, который бы соответствовал соотношению 3.

$$\frac{\sum_{i=0}^n s_i}{S_F} \simeq 1 \quad (3)$$

В простейшем случае заданная площадь – выпуклый многоугольник и основным подходом является провести параллельные прямые на расстоянии ширины необходимого перекрытия. Пересечения этих прямых с границей площади и будет является узлами будущего маршрута, далее запустив на таком маршруте соответствующую вертикальному расстоянию между снимками. Такой не оптимизированный подход даёт хорошие результаты если мы добавляем вокруг буферную зону на несколько снимков как на рисунке 1. [2] Это решает проблему не отснятых зон и выносит за пределы целевой площади фотографии, сделанные на поворотах, которые из-за сильных крена и тангажа БВС самолётного типа имеют сильные искажения и не могут использоваться для последующей обработки.

В случае же, если мы имеем более сложно очерченную площадь, может оказаться оптимальнее разделить её и рассматривать каждый по отдельности.

Разбиение маршрута

Простейшим случаем является маршрут, имеющий самопересечения, которые необходимо обнаружить. Есть несколько путей для поиска, можно просто перебрать пары отрезков, данный алгоритм обойдется нам в сложность $O(n^2)$. Можно пойти другим путём и использовать алгоритмы, использующие заметающие прямые, они дают уже логарифмическую сложность $O(\log n)$, что гораздо эффективнее [3].

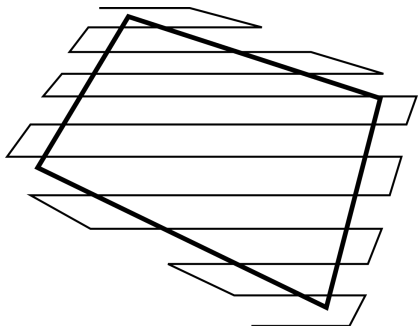


Рисунок 1 — Маршрут с буфером для выпуклого многоугольника

Могут, так же, встретиться не выпуклые площади, в которых один или несколько проходов рисуются большей частью впустую, как например на рисунке 2.

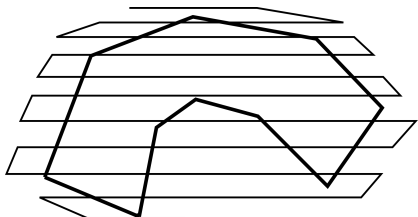


Рисунок 2 — Сложный но не оптимальный маршрут

Если такой маршрут построен, значит для этого были причины: оптимизация времени, слишком большая площадь, которую необходимо пролететь за один раз, запретные зоны или физические препятствия, источники магнитных помех и прочее.

С этой проблемой часто сталкиваются в отрасли трёхмерной графики и тут разработано достаточно много алгоритмов, позволяющих разбить многоугольник на более мелкие. Классической в данном случае является задача триангуляции, но это не всегда оптимально и предпочтение мы дадим алгоритмам, способным разбивать на выпуклые многоугольники, или способные объединить треугольники для достижения того же эффекта.

В 1990 году Бернард Чазелле доказал, что любой простой многоугольник может быть триангулирован за линейное время. От представил и сам алгоритм, но его реализация столкнулась с некоторыми проблемами в архитектуре современных компьютеров и оперируя огромными числами едва ли может обогнать по производительности следующие алгоритмы. [4]

Алгоритм Греена на выходе уже выдаёт набор оптимальных многоугольников и выполняется за время $O(n \log n)$ [5]. Реализация алгоритма уже присутствует в пакете CGAL.

Если бы мы использовали например алгоритм Хертела и Мехлхорна, выполняющийся за $O(n + r \log r)$, где r – количество вогнутых полигонов [6], то получили бы множество треугольников, которые необходимо объединить.

Определим для объединения алгоритм, выполняющийся за время $O(n^2)$. Пусть $T = \{t_0, t_1, \dots, t_k\}$ – множество треугольников, которое мы получили после алгоритма триангуляции.

Пусть R – результат

Пока T не пуст

– Выбираем случайный элемент t_j из T

– Пока у t_j есть соседи

– – Для каждого соседа t_m t_j

– – – Если объединение t_j с t_m , t_j осталось выпуклым

– – – – Объединить t_j с t_m

– – – – Удалить t_m из T

– – Если в предыдущем цикле t_j не изменился

– – – Переместить t_j из T к результатам R

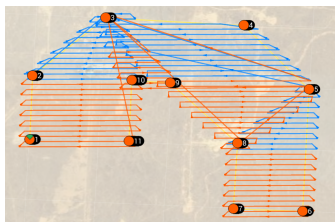
– – – Выход из цикла

Вернуть результат R

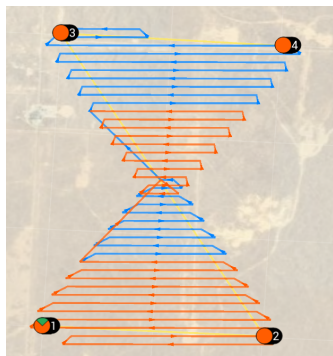
Заключение

В зависимости от сложности исходной геометрии сложно выбрать между алгоритмами Греена или Хартела, к тому же Хартела имеет более случайную природу, поэтому на данный момент в реализован вариант, использующий алгоритм Греена из пакета GDAL. Получающиеся маршруты обеспечивают полное покрытие площади на достаточно сложных геометриях, примеры представлены на рисунке 3.

Список литературы



(a) Пример маршрута 1



(b) Пример маршрута 2

Рисунок 3 — Пример работы алгоритма

1. L. H. Nam, L. Huang, X. J. Li and J. F. Xu. An approach for coverage path planning for UAVs. – IEEE 14th International Workshop on Advanced Motion Control (AMC), 2016.
2. Franco, C. D., & Buttazzo, G. Energy-aware coverage path planning of uavs. – IEEE international conference on autonomous robot systems and competitions (ICARSC) (pp. 111–117), 2015.
3. Diane Souvaine, Kevin Keating. Line Segment Intersection Using a Sweep LineAlgorithm. – Computational Geometry Tufts University, 2005
4. Chazelle, B. Triangulating a simple polygon in linear time. – Annual Symposium on Foundations of Computer Science, 1990
5. Daniel H. Greene. The decomposition of polygons into convex parts – In Franco P. Preparata, editor, Computational Geometry, volume 1 of Adv. Comput. Res., pages 235-259. JAI Press, Greenwich, Conn., 1983.
6. Hertel, S., & Mehlhorn, K. Fast triangulation of simple polygons. – Lecture Notes in Computer Science, 207–218., 1983

COVERAGE OF COMPLEX AREAL ROUTE OF UAV

A.O. Stepanov

Udmurt State University, 426034 Russia, Izhevsk, Universitetskaya, 1
e-mail: *heimusharper@gmail.ru*

The article discusses the process of building an areal route for an plane-type UAV. Various options for constructing an initial route and ways to optimize it are considered. The problem of the formation of convex polygons is considered and an algorithm for dividing such structures into simpler routes is chosen.

Keywords:unmanned aircraft vehicle, decision support system, route planning, route optimization.