

СОДЕРЖАНИЕ

А.О. Степанов

ПОКРЫТИЕ СЛОЖНОГО ПЛОЩАДНОГО МАРШ-
РУТА БЕСПИЛОТНЫМ ВОЗДУШНЫМ СУДНОМ . . . 2

ПОКРЫТИЕ СЛОЖНОГО ПЛОЩАДНОГО МАРШРУТА БЕСПИЛОТНЫМ ВОЗДУШНЫМ СУДНОМ

А.О. Степанов

Удмуртский Государственный Университет, г. Ижевск, Россия

e-mail: *heimusharper@gmail.ru*

В статье рассматривается процесс построения площадного маршрута для беспилотного воздушного судна самолётного типа. Рассмотрены различные варианты построения и способы оптимизации. Рассмотрена проблема образования выпуклых многоугольников и выбран алгоритм разбиения таких структур на более простые фигуры.

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

Введение

Если представить беспилотный самолёт в воздухе, то фотоаппарат или другое техническое устройство будут иметь проекцию, в основном, направленную по нормали к поверхности. При этом полезная нагрузка определяет точку проекции, являющейся центром некоторой области. И получение равномерного, возможно с некоторым перекрытием, их набора составляет задачу покрытия площади. [1]

Для многих задач требуется пролететь ограниченную область за наименьшее время, и в большинстве случаев она представляет собой прямоугольник, но попадают ситуации, когда эта область должна быть сформирована по сложному маршруту с поворотами и, возможно, самопересечениями. Такое может произойти в связи с оптимизацией по какому-либо критерию, наличие запретных зон или физические препятствия, несущие опасность для самолёта: горные массивы, области задымления лесных пожаров, источники помех, высокие техногенные объекты, обеспечение безопасности людей и прочее.

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, тогда 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. Построение маршрута для такой фигуры может вестись построением сетки [1] – мы проходим только по тем узлам, которые оказались внутри. Точность и производительность метода сильно зависит от размерности, которая берётся из ширины и высоты перекрытия. Более точным же подходом является составление пути через последовательное построение параллельных прямых на расстоянии ширины необходимого перекрытия. Их пересечения с границей площади является узлами будущего маршрута [2]. Оба подхода дают хорошие результаты, мы же используем второй, для обеспечения более постоянной длины поворотов, которые обязательно появятся за пределами границ площади.

Сразу добавим вокруг буферную зону на несколько снимков как на рисунке 1. Это решает проблему не отснятых зон и выносит за пределы целевой площади фотографии, сделанные при разворачивании борта. Такие снимки из-за сильных крена и тангажа самолёта имеют сильные искажения и не могут использоваться для последующей обработки. Стоит отметить, что для беспилотников

вертолётного типа такой буфер теряет свою актуальность в связи с большей стабильностью таких аппаратов.

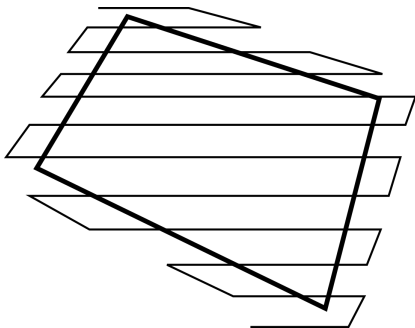


Рисунок 1 — Маршрут с буфером для выпуклого многоугольника

Стоит так же учесть, что нельзя допускать длительного полёта по или против ветра и угол проходов строится с учётом использования бокового ветра в большей части.

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 обойдётся 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].

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

В большинстве решений такие области строятся, как и обычные выпуклые, с той лишь разницей, что путь не строится через запрещённую область и оставшаяся не обработанной часть проходит тот же алгоритм, рекурсивно перебирая их. [4] Такой подход порождает проблему — могут появиться множество небольших

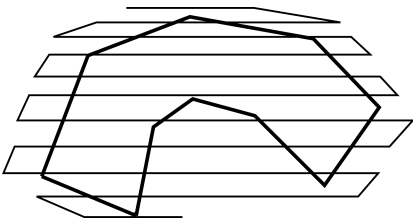


Рисунок 2 — Сложный не оптимально построенный маршрут

геометрий, которые содержат в себе короткие пути, они вносят хаос и, в целом, не оптимальны. В данном случае мы можем лишь объединить такие области с большими (есть риск выйти в запретную зону) или проигнорировать (возможно, будут области, в которых нет фотографий). Но есть предложение объединять области [5], метод подходит для решения нашей задачи, правда в большинстве вариантов предполагаемый угол прямых на разных блоках отличается, это может помешать самолёту из-за сильного воздействия на него ветра, как уже было указано выше, не стоит давать возможности лететь долгое время по или против ветра.

Исходя из этого, мы хотим разбить область на минимальное количество блоков, размеры которых, желательно, были бы близки или не были слишком маленькими. С похожей 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. Он представил и сам алгоритм, но его реализация столкнулась с некоторыми проблемами и едва ли может обогнать по производительности алгоритмы представленные ниже. [6]

Алгоритм Грeене на выходе уже выдаёт набор оптимальных многоугольников и выполняется за время $O(n \log n)$ [7]. Реализация алгоритма уже присутствует в пакете CGAL (Computational Geometry Algorithms Library) – специализированная библиотека

содержащая реализации эффективных алгоритмов, относящихся к обработке геометрий.

Если бы мы использовали например алгоритм Хертела и Мехлхорна, выполняющийся за $O(n + r \log r)$, где r – количество вогнутых полигонов [8], 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

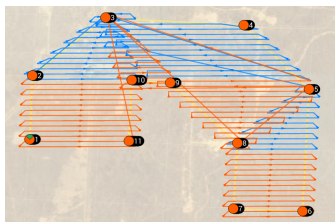
 Удалить t_m из T

 Если в предыдущем цикле t_j не изменился

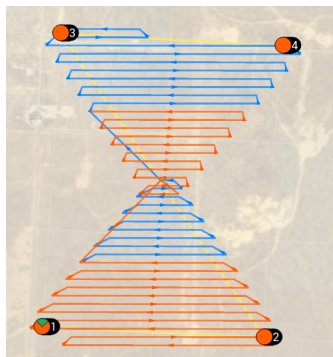
 Переместить t_j из T к результатам R

 Выход из цикла

Вернуть результат R



(a) Пример маршрута 1



(b) Пример маршрута 2

Рисунок 3 — Пример работы алгоритма

Заключение

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

Список литературы

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. Taua M. Cabreira, Lisane B. Brisolara, Paulo R. Ferreira Jr. Survey on Coverage Path Planning with Unmanned Aerial Vehicles
5. Coombes, M.; Fletcher, T.; Chen, W.H.; Liu, C. Optimal polygon decomposition for UAV survey coverage – path planning in wind. Sensors 2018
6. Cabreira, Tauã M. and Brisolara, Lisane B. and Ferreira Jr., Paulo R. Survey on Coverage Path Planning with Unmanned Aerial Vehicles – <https://www.mdpi.com/2504-446X/3/1/4>
7. 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.
8. Hertel, S., & Mehlhorn, K. Fast triangulation of simple polygons. – Lecture Notes in Computer Science, 207–218., 1983

COVERAGE OF COMPLEX AREAL ROUTE OF UNMANNED AIRCRAFT VEHICLE

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.