

ИНФОРМАЦИОННЫЕ
ТЕХНОЛОГИИ
И ТЕЛЕКОММУНИКАЦИИ

INFORMATION
TECHNOLOGIES
AND TELECOMMUNICATIONS

УДК 004.896

DOI: 10.17212/2782-2001-2024-2-37-54

Алгоритм планирования маршрута БПЛА в условиях недостатка визуальных ориентиров*

К.Д. КОНОВАЛОВ

199178, г. Санкт-Петербург, 14-я линия В.О., 39, Санкт-Петербургский федераль-
ный исследовательский центр Российской академии наук

kodmko@mail.ru

Сегодня беспилотная авиация используется для решения самых разных задач. Одна из проблем, возникающих при использовании беспилотных летательных аппаратов (БПЛА), – это проблема навигации, в частности установление их точного местоположения в пространстве. Сегодня существуют различные методы решения этой проблемы. Перспективным способом ее решения в условиях невозможности использования спутниковой системы навигации являются системы видеонавигации, использующие для установления местоположения визуальные ориентиры. В настоящей работе проведен обзор существующих типов систем видеонавигации БПЛА, выделены особенности работы систем видеонавигации, использующих сопоставление визуальных ориентиров со спутниковыми картами, поставлена задача поиска маршрута в условиях недостатка визуальных ориентиров, при которых возможно возникновение проблемы потери опорных точек. Описан используемый графовый формат данных, хранящий значения метрики разнообразия подстилающей поверхности, которая характеризует величину вероятности возникновения проблемы потери опорных точек. Рассмотрены существующие алгоритмы поиска пути в графе и обоснована необходимость разработки нового алгоритма.

Предложено теоретическое описание алгоритма, в котором процесс поиска пути делится на два этапа: вначале производится поиск варианта решения с максимальным значением метрики разнообразия подстилающей поверхности, а затем выбор маршрута с минимальной длиной. Приведено несколько вариантов реализации предложенного алгоритма и выполнена оценка их асимптотической сложности. На основании теоретического описания разработана программная реализация предложенного алгоритма и выполнена экспериментальная проверка корректности его работы на тестовой карте с различными полетными заданиями. По результатам тестирования предложенный алгоритм показал свою работоспособность, корректно построив маршруты во всех тестовых примерах. В дальнейшем этот алгоритм может быть использован как часть системы видеонавигации БПЛА. Также в статье описаны основные преимущества и недостатки алгоритма и предложены пути для его дальнейшего совершенствования.

* Статья получена 21 марта 2024 г.

Ключевые слова: БПЛА, видеонавигация, графовый алгоритм, система технического зрения, автономная визуальная навигация, итерационный алгоритм, теория графов, вычислительная сложность, оптимизация на графах

ВВЕДЕНИЕ

Сегодня одним из наиболее перспективных типов робототехнических комплексов являются беспилотные летательные аппараты (БПЛА). Подобные летательные аппараты используются в различных отраслях народного хозяйства – от контроля качества воды в водоемах до доставки медикаментов во время экстренных происшествий. При этом любому БПЛА для работы в автономном или полуавтономном режиме вне зависимости от типа его конструкции, области применения и решаемой задачи необходима система навигации, которая должна обеспечить перемещение БПЛА в указанную точку или по указанной траектории.

Исследования в области разработки систем навигации ведутся с начала прошлого века. За это время были разработаны разнообразные типы подобных систем. Различные типы навигационных систем используют разные источники данных и сенсорные системы. Наиболее распространенными в настоящее время являются так называемые глобальные навигационные спутниковые системы, или ГНСС. Однако в некоторых случаях использование подобных систем является по тем или иным причинам невозможным. В таких случаях одним из наиболее перспективных типов систем навигации, которые можно применять в качестве замены ГНСС, являются системы видеонавигации (иначе – системы навигации по визуальным признакам).

Системы видеонавигации могут быть основаны на разных алгоритмах. Одним из подтипов подобных систем являются системы визуальной одометрии. Подобные системы основаны на сопоставлении последовательности кадров с бортовых камер. Каждый последующий кадр сравнивается с предыдущим и на основании полученной разницы определяется направление движения БПЛА, а также пройденное им расстояние. При этом, как описано в [1], системы визуальной одометрии могут использовать различные методы сравнения изображений: по особым точкам, по оптическому потоку, с использованием стереозрения и т. д. Как отмечает автор, сегодня БПЛА используют системы визуальной одометрии в качестве одного из модулей системы навигации, полноценная же автономная система навигации БПЛА на основе визуальной одометрии является предметом активных исследований.

Другим подтипом систем видеонавигации являются так называемые системы на основе одновременной локализации и построения карты, или *simultaneous localization and mapping* (SLAM). Такой тип систем видеонавигации был описан в работах [2, 3]. Системы подобного типа сопоставляют последовательность сделанных бортовой камерой снимков и на их основе строят актуальную карту местности, в рамках которой определяют текущее местоположение БПЛА. Разработка полностью автономной системы навигации типа SLAM также является предметом исследований. В настоящее время подобные системы в БПЛА используют в основном для решения вспомогательных задач, например возврата БПЛА в стартовую точку [4].

Также в настоящее время идут исследования в области разработки систем видеонавигации БПЛА, основанных на корреляционно-экстремальном анализе изображений с бортовых камер, или корреляционно-экстремальных систем навигации (КЭСН) [5]. Подобные системы сравнивают изображения с бортовых камер с некоторым эталонным изображением и выявляют корреляцию между ними. Текущим местоположением БПЛА считается та точка, в которой корреляция максимальна. Также вместо изображений может использоваться набор заданных ориентиров. Кроме того, такие системы помимо изображений с бортовых камер могут использовать радиотепловые [6], микроволновые [7] или геофизические [8] поля.

Еще одним типом систем видеонавигации, в области разработки которых сегодня ведутся исследования, являются системы, использующие сопоставление визуальных ориентиров со спутниковыми картами [9]. Системы такого типа выполняют поиск некоторых ориентиров (например, зданий, дорог, береговых линий и т. д.) на изображениях с бортовых камер, а затем производят поиск аналогичного набора ориентиров на спутниковых снимках. При этом могут быть использованы не только изображения в исходном виде, но и некоторый промежуточный формат, который содержит данные об ориентирах [10]. Далее в статье будет рассматриваться система видеонавигации такого типа.

Системы видеонавигации способны обеспечить высокую точность позиционирования, используя только бортовую камеру (или систему бортовых камер). Однако у них есть и ряд проблем, которые необходимо учитывать при разработке подобных систем. Так, для систем, использующих сопоставление визуальных ориентиров со спутниковыми картами, одной из проблем является проблема потери опорных точек. Опорными точками в данном контексте называются точки, которые имеют некоторые отличительные признаки и за счет этого могут быть использованы в качестве ориентиров при определении местоположения летательного аппарата [11]. Также может быть использовано понятие характерной или особой точки. Особой точкой называют некоторую точку на изображении, в окрестности которой содержится большое количество локальной информации. Соответственно, такая точка с большой долей вероятности будет распознана в любом другом изображении данного объекта [12].

При определении местоположения БПЛА системы видеонавигации, сопоставляющие визуальные ориентиры со спутниковыми картами, используют найденные на изображении с бортовых камер опорные точки. Однако такие методы требуют нахождения на изображении подобных точек. В случаях же, когда изображения с бортовых камер БПЛА содержат участки с однородной подстилающей поверхностью (например, лесом), подобных точек может быть крайне мало или вообще не быть. Пример подобного случая представлен на рис. 1. В кадре с бортовой камеры БПЛА виден только сплошной лес. В таких условиях даже оператору будет крайне сложно либо вообще невозможно определить точное местоположение БПЛА. Этот случай иллюстрирует ситуацию, когда на изображении лесного массива можно выделить множество ключевых точек (позволяющих осуществлять локальную навигацию в пределах небольшой территории), но при этом ни одна из них не будет являться характерной точкой для однозначной идентификации глобального положения объекта в пространстве.

Примеров подобных поверхностей существует довольно много, и при разработке системы видеонавигации необходимо учитывать подобную возможность. Поэтому возникает задача разработки алгоритма, позволяющего на этапе планирования маршрута прокладывать его таким образом, чтобы избегать зон, где возможна потеря опорных точек. В настоящей статье будет предложен вариант подобного алгоритма, а также приведены результаты экспериментальной проверки написанной с его использованием программы.

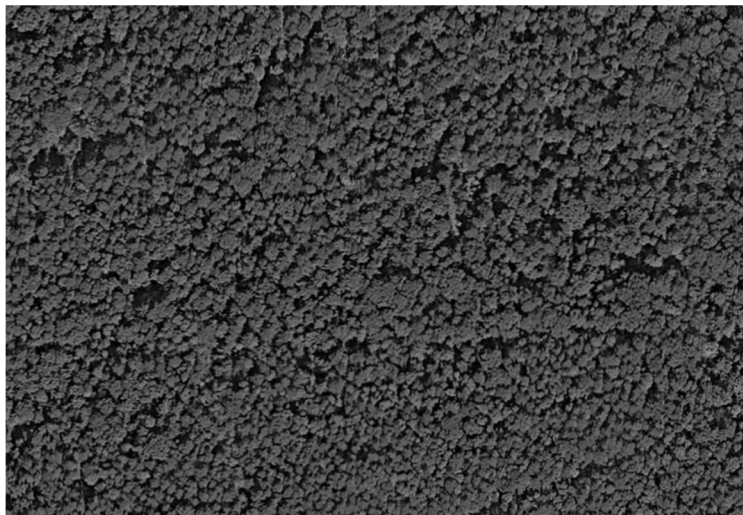


Рис. 1. Пример кадра с бортовой камеры БПЛА с однородной подстилающей поверхностью

Fig. 1. An example of an image from an on-board UAV camera with a homogeneous underlying surface

Статья состоит из трех разделов. В первом разделе рассматривается постановка задачи поиска маршрута БПЛА, а также анализируются существующие алгоритмы, которые могут быть использованы для ее решения. Во втором разделе представлено теоретическое описание предлагаемого алгоритма поиска маршрута и оценка его асимптотической сложности. В третьем разделе представлены результаты экспериментальных проверок разработанного алгоритма на тестовой карте.

1. ИСХОДНЫЕ УСЛОВИЯ И АНАЛИЗ СУЩЕСТВУЮЩИХ АЛГОРИТМОВ

Алгоритм планирования маршрута, описываемый в настоящей работе, является одним из модулей разрабатываемой системы видеонавигации БПЛА. Поэтому в алгоритме в качестве входных данных используются не снимки той местности, в которой происходит полет БПЛА, а полученные на их основе данные в графовом формате. Такой формат позволяет для каждой точки местности получить оценку вероятности потери опорных точек. Для этого используется так называемая метрика разнообразия подстилающей поверхности. Чем выше ее значение, тем ниже вероятность потери опорных точек. Метрика зависит от типов поверхностей, попавших в кадр, а также от длины границ между ними.

Значение метрики для участка с центром в точке с координатами (x_1, y_1) и величиной $(m \times n)$ точек вычисляется по формуле

$$\sum_{x=x_1 - \left\lfloor \frac{m}{2} \right\rfloor}^{x_1 + \left\lfloor \frac{m}{2} \right\rfloor} \sum_{y=y_1 - \left\lfloor \frac{n}{2} \right\rfloor}^{y_1 + \left\lfloor \frac{n}{2} \right\rfloor} (W_c(x, y) + W_b(x, y)), \quad (1)$$

где x, y – координаты точки, находящейся в кадре; $W_c(x, y)$ – значение веса класса, к которому относится точка с координатами x, y . Конкретные значения весов классов зависят от реализации и условий использования конкретной системы видеонавигации. В зависимости от того, к какому из классов относится точка, эта функция имеет значение соответствующего весового коэффициента. В процессе разработки системы видеонавигации автор использовал 6 классов: «водоемы», «здания», «дороги», «лес, деревья», «земля, песок» и «иное», каждому из которых соответствует собственный весовой коэффициент; $W_b(x, y)$ – весовая функция границ, которая описывается следующей формулой:

$$W_b(x, y) = \begin{cases} w_b & |(x, y) \in b, \\ 0 & |(x, y) \notin b, \end{cases} \quad (2)$$

где b – дополнительный класс, к которому принадлежат все точки, являющиеся граничными между областями, относящимися к разным классам; w_b – значение весового коэффициента для границ.

В общем виде входные данные представлены в виде графа, который при его визуализации будет иметь решетчатую структуру. В качестве узлов выступают точки на местности, маршрут по которой необходимо проложить. Каждый узел в качестве уникального идентификатора имеет координаты соответствующей ему точки. При практическом использовании размеры разбивки координатной сетки зависят от характеристик используемой бортовой камеры БПЛА, а также от диапазона истинных высот, на которых выполняется полет. Будем считать, что координаты всех узлов графа являются целочисленными и разность между координатами соседних вершин составляет единицу по соответствующей оси.

Ребрами являются возможные пути перемещения БПЛА из точки, соответствующей определенному узлу графа. Возможны два подхода к созданию ребер в графе.

Первый вариант подразумевает соединение каждого узла графа с соседними по горизонтали, вертикали и диагонали. В этом случае у любого неграничного узла будет 8 ребер, связывающих его с соседними узлами. Также в этом случае необходимо задавать веса ребер, соответствующие расстоянию между точками местности. Вес конкретного ребра может быть вычислен по следующей формуле:

$$W_e(e_i) = \begin{cases} 1 & |e_i \notin diag, \\ \sqrt{2} & |e_i \in diag, \end{cases} \quad (3)$$

где e_i – рассматриваемое ребро графа; $diag$ – множество тех ребер, которые являются диагональными.

Второй вариант предполагает соединение узлов только по горизонтали и вертикали. В этом случае все ребра будут одинаковой длины, и их вес можно положить равным единице. При выборе этого варианта путь, являющийся результатом работы алгоритма, будет менее качественным с точки зрения минимизации его длительности, однако алгоритм будет работать с большей скоростью из-за меньшего числа ребер, а также из-за возможности использования более простых алгоритмов поиска кратчайшего пути, например, простого обхода в ширину. Такой вариант может быть полезен в условиях использования данного алгоритма на бортовом оборудовании БПЛА с ограниченными вычислительными ресурсами.

Также в узлах графа хранятся числовые значения метрики разнообразия подстилающей поверхности. Как было описано ранее, основной задачей алгоритма является планирование маршрута таким образом, чтобы при его прохождении БПЛА избегал зон, в которых возможна потеря опорных точек. Такая метрика характеризует качество подстилающей поверхности, то есть вероятность потери опорных точек, в зоне карты, центром которой является точка с координатами, равными координатам узла графа. При этом размеры зоны могут быть сконфигурированы в зависимости от высоты полета и угла обзора бортовой камеры БПЛА. Чем выше значение метрики, тем более низкой является вероятность потери опорных точек и тем более предпочтительной является данная точка для проложения через нее маршрута. Следовательно, при планировании маршрута необходимо максимизировать минимальное значение метрики на всем маршруте. В формальном виде это можно выразить следующим образом:

$$\min_{v \in P, p \in P} W(v) \rightarrow \max_P, \quad (4)$$

где P – множество возможных путей от начальной вершины к конечной; p – рассматриваемый вариант маршрута; v – узел графа, входящий в маршрут p ; $W(v)$ – значение метрики разнообразия подстилающей поверхности для узла v . Вторым критерием, который необходимо учитывать при планировании маршрута, является его протяженность. В настоящей работе не ставится оптимизационной задачи по минимизации длины траектории, а используются лишь эвристические правила по отсечению заведомо некорректных решений (например, если в качестве ответа может быть получен путь, проходящий через некоторую область не напрямую, а «змейкой», проходя полностью все ее точки).

На сегодняшний день существует большое количество алгоритмов, позволяющих находить путь от одного узла графа до другой. Рассмотрим наиболее популярные из них и обоснуем невозможность их использования при решении поставленной задачи. В исходной постановке эта задача схожа с задачей поиска кратчайшего пути в графе. Отличиями от стандартной постановки является сосредоточение весов не в ребрах, а в узлах; необходимость максимизации, а не минимизации веса пути; использование в качестве веса не общего веса пути, а минимальное его значение. Эти отличия не являются достаточно существенными, поэтому после сравнительно небольших модификаций в решении задачи могут быть использованы стандартные алгоритмы поиска кратчайшего пути в графе.

Наиболее популярными из таких алгоритмов являются алгоритмы Дейкстры [13] и Беллмана – Форда [14]. Первый имеет более высокую производительность, однако не допускает наличия в графе ребер с отрицательными весами. Второй допускает подобные ребра, однако накладывает ограничение в виде отсутствия циклов с отрицательным суммарным весом [15]. Асимптотическую сложность алгоритма Дейкстры можно описать следующим выражением:

$$O(n \log(n) + m), \quad (5)$$

где n – число вершин графа, а m – число его ребер. Асимптотическую сложность алгоритма Беллмана – Форда можно описать выражением

$$O(n \cdot m). \quad (6)$$

При этом алгоритмы с ограничением в виде отсутствия циклов с отрицательным суммарным весом не могут быть использованы для решения поставленной задачи. Причина этого заключается в особенностях входных данных. Рассмотрим некоторую карту, на которой расположен входящий в нее водоем (озеро или пруд). В формуле расчета метрики разнообразия подстилающей поверхности учитывается качество ориентиров, которые могут быть использованы для определения местоположения БПЛА. Береговая линия водоемов является хорошим ориентиром, следовательно, в той области графа, которая соответствует расположению водоема на карте, появится цикл из узлов с большими значениями метрики. На таких входных данных алгоритмы, подобные алгоритму Беллмана – Форда, будут заикливаться, поскольку подобный цикл с учетом постановки задачи аналогичен циклу с отрицательным суммарным весом.

Для предотвращения такой проблемы разработаны различные модификации алгоритмов Беллмана – Форда и Дейкстры (например, алгоритм Голдберга – Радзика или алгоритм топологического упорядочения, описанный в [16]). Данный алгоритм перед началом работы производит сканирование топологии с целью разрыва циклов с отрицательным суммарным весом. Асимптотическая сложность алгоритма описывается выражением

$$O(nm \log(n)). \quad (7)$$

Однако подобный алгоритм, хотя и сможет найти путь, удовлетворяющий исходному условию, на практике также не является лучшим решением. Это также связано с особенностями исходных данных. В большинстве случаев расстояние между соседними зонами при построении графа выбирается сравнительно небольшим. Соответственно, в рассмотренном выше примере элементы береговой линии будут попадать сразу в несколько узлов графа. Таким образом, хотя цикл будет разорван, однако возможность определить, какой именно путь из возможных выберет алгоритм, априори отсутствует, поскольку все эти узлы для алгоритма будут являться одинаково предпочтительными. Поэтому может возникнуть ситуация, когда результатом работы алгоритма будет являться путь, несколько раз проходящий вдоль всей береговой линии водоема и лишь затем отправляющийся к финишной точке. Такое решение будет являться некорректным относительно критерия длины пути.

В качестве возможного решения этой проблемы может быть предложено ограничение количества ребер. Например, можно поставить следующее условие: ребро является «проходимым» (то есть по нему возможно движение) тогда и только тогда, когда расстояние от текущей точки до конечной не уменьшается хотя бы по одной из осей координат. В этом случае при формате графа с диагональными ребрами вместо восьми ребер, выходящих из узла, «проходимыми» будут только пять. Такое ограничение позволит избежать как зацикливаний, так и чрезмерной длины итогового пути. Однако такое решение также не является допустимым. Рассмотрим вышеописанный пример с водоемом с одним дополнением: пусть на одном из берегов этого водоема находится мыс, окруженный с трех сторон водой. Пусть начальная точка маршрута расположена на этом мысу, а финишная – на противоположном берегу водоема. Если размеры водоема достаточно велики, то оптимальным маршрутом в данном случае будет маршрут, идущий вдоль береговой линии водоема. Однако вышеописанное решение не позволит проложить подобный маршрут, поскольку для его реализации необходимо переместиться с мыса в направлении, противоположном направлению к финишной точке. В случае же попытки перемещаться только по пяти допустимым направлениям маршрут неизбежно приведет к участку, полностью покрытому водой, на котором вероятность потери опорных точек будет крайне высокой.

На сегодняшний день исследования в области алгоритмов поиска кратчайшего пути в графе продолжают, хотя и не в столь быстрых темпах, как несколько десятилетий назад. Например, одним из последних предложенных вариантов алгоритма поиска кратчайшего пути является так называемый *Jump point search*, описанный в 2011 году в работе [17]. Его асимптотическая сложность описывается выражением

$$O(m \log(n)). \quad (8)$$

Однако все подобные алгоритмы являются улучшенными вариациями ранее разработанных алгоритмов, основной целью которых является улучшение вычислительной сложности и, соответственно, производительности процедуры поиска кратчайшего пути. Например, вышеупомянутый алгоритм *Jump point search* является улучшенной модификацией алгоритма A^* , который, в свою очередь, является улучшенным вариантом алгоритма Дейкстры и был описан в работе [18]. При этом подобные алгоритмы сохраняют ограничение в виде необходимости отсутствия циклов с отрицательным суммарным весом. Соответственно, они также не подходят для решения поставленной задачи.

2. ОПИСАНИЕ АЛГОРИТМА ПОИСКА ПУТИ В ГРАФЕ

Ввиду вышеописанных недостатков существующих алгоритмов возникла необходимость разработать новый алгоритм поиска пути в графе, учитывающий особенности входных данных. На этапе анализа рассматривались две основные концепции работы алгоритма. Первая подразумевала добавление возможности разрывов циклов с отрицательным суммарным весом, как в алгоритме Голдберга – Радзика, с учетом общей топологии графа. Вторая предполагала разделение алгоритма на две стадии, в каждой из которых проводится

оптимизация по одному критерию – наибольшему значению метрики разнообразия подстилающей поверхности и протяженности маршрута. По результатам анализа предпочтение было отдано второй концепции по следующим причинам: во-первых, программная реализация первой концепции является более ресурсоемкой. Как отмечалось выше, при разрыве цикла основная сложность состоит в определении близкой к оптимальной или близкой к ней точки разрыва. Поскольку априорные данные о структуре графа перед началом работы алгоритма отсутствуют, разработка универсального условия разрыва не представляется возможной. Таким образом, при использовании такого подхода необходимо проводить анализ всего графа или отдельного его подграфа и на основании результатов анализа определять точку разрыва. Во-вторых, концепция разделения оптимизаций является более предпочтительной для дальнейшего внедрения полноценной многокритериальной оптимизации.

На основе выполненного анализа был разработан и реализован алгоритм, позволяющий решить поставленную задачу. Высокоуровневое описание разработанного алгоритма в виде блок-схемы представлено на рис. 2.

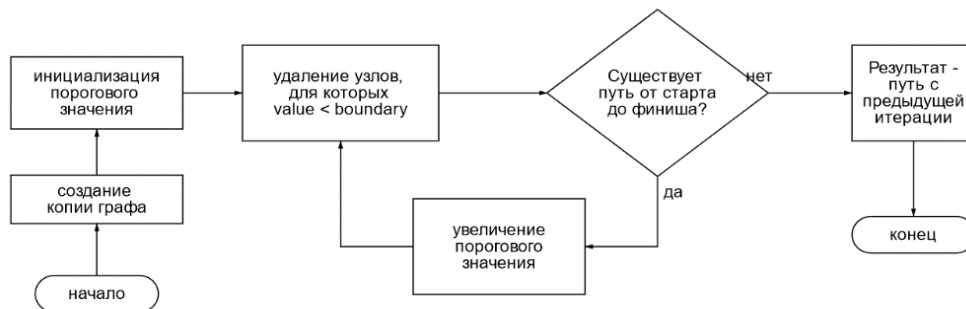


Рис. 2. Блок-схема разработанного алгоритма

Fig. 2. Block diagram of the developed algorithm

Общая идея алгоритма состоит в следующем. Вначале производится ряд предварительных действий: создается копия графа во избежание потери данных, а также проверяется наличие путей от стартовой точки до финишной без учета значений метрик в узлах (на этом этапе пороговое значение устанавливается равным некоторому числу, меньшему минимального значения метрики разнообразия подстилающей поверхности в графе; оптимальным вариантом является выбор некоторого отрицательного значения, поскольку формула вычисления метрики устроена таким образом, что итоговое значение всегда будет больше либо равно нулю). Далее, как отмечалось ранее, алгоритм разделен на две стадии. На первой стадии производится увеличение порогового значения, после чего из графа удаляются все узлы, значение метрики разнообразия подстилающей поверхности для которых меньше, чем пороговое значение. Также удаляются все связанные с ними ребра. На этом этапе итеративно производится выбор маршрута с наибольшим значением метрики.

На втором этапе, напротив, производится выбор варианта по наименьшей длительности итогового пути. После удаления узлов и ребер производится поиск кратчайшего пути от стартовой точки до финишной в полученном разреженном графе. Для этого может быть использован любой из существующих

алгоритмов, например алгоритм Дейкстры, поскольку все ребра в графе обозначают расстояния и, следовательно, имеют неотрицательный вес. Если такой путь существует, то он сохраняется в качестве промежуточного оптимального решения. Если же путь отсутствует, то промежуточное оптимальное решение с предыдущего шага считается окончательным оптимальным решением и возвращается в качестве результата работы алгоритма.

Также в зависимости от реализации графа и асимптотической сложности операций с ним возможен другой вариант реализации. Общая идея этого варианта совпадает с вышеописанным, однако он отличается реализацией этапа оптимизации по протяженности маршрута и предполагает отказ от вычисления оптимального по длительности пути на каждом этапе в пользу простой проверки существования пути. Для этого могут быть использованы алгоритмы с меньшей асимптотической сложностью (например, поиск в ширину). Однако в этом случае на каждой итерации необходимо не только удалять на каждой итерации определенный набор ребер и узлов, но и сохранять их до следующей итерации, поскольку в противном случае для получения итогового пути необходимо будет создавать новую копию исходного графа с последующим удалением аналогичного набора узлов и ребер из нее. При таком варианте реализации алгоритм будет работать следующим образом: после завершения операции удаления производится проверка на предмет наличия пути. Если такой путь существует, происходит переход к следующей итерации. Если же путь отсутствует, то в граф заново добавляются все узлы и ребра, удаленные в ходе текущей итерации, а затем в полученном графе производится поиск кратчайшего пути с помощью, например, алгоритма Дейкстры.

Оценим асимптотическую сложность обоих вариантов. Обозначим количество узлов в графе как n , а количество ребер – как m . С учетом особенностей структуры графа можно положить $m = 8n$. Также асимптотическая сложность зависит от количества итераций, которое, в свою очередь, зависит от выбора шага алгоритма. Положим количество итераций равным f . Также положим, что асимптотическая сложность удаления одного узла и связанных с ним ребер из графа равна $O(d)$, а сложность вставки одного узла и связанных с ним ребер – $O(i)$. Кроме того, необходимо оценить сложность промежуточных операций. При поиске в ширину оценка достаточно тривиальна: в худшем случае данный алгоритм проверит все узлы и все ребра по одному разу, следовательно, его асимптотическая сложность равна $O(n + m)$, или $O(n + 8n)$, или с учетом свойств оценки асимптотической сложности $O(n)$. Для поиска кратчайшего пути оценка несколько сложнее. Из всех известных алгоритмов одну из наименьших асимптотических сложностей имеет алгоритм Дейкстры, однако и в его случае сложность может изменяться в зависимости от особенностей реализации. Наиболее оптимальной является его реализация с использованием фиббоначиевой кучи, описанная в работе [19]. Для такой реализации асимптотическая сложность составляет $O(m + n \log(n))$, или с учетом сокращений $O(n \log(n))$.

С учетом вышеуказанных оценок произведем оценку общей асимптотической сложности обоих вариантов разработанного алгоритма. Оценки операций копирования графа, а также работы со счетчиками в итоговую оценку включены не будут ввиду их слабого влияния на результат. Для первого варианта асимп-

тотическая сложность после соответствующих упрощений может быть оценена по выражению (9), для второго – по выражению (10):

$$O(fn(d + \log(n))). \quad (9)$$

$$O(n(fd + i + \log(n))). \quad (10)$$

Если удалить все общие члены из полученных выражений, то можно получить выражение $f \log(n)$ для первого варианта и $i + \log(n)$ – для второго. Сравнивая эти выражения, можно определить, какой из вариантов является более предпочтительным с точки зрения асимптотической сложности. Так, если операция вставки узла в граф занимает константное время, то предпочтительнее будет второй вариант.

3. ЭКСПЕРИМЕНТАЛЬНАЯ ПРОВЕРКА АЛГОРИТМА

На базе теоретического описания алгоритма была создана программная реализация на языке программирования Java и проведена проверка корректности его работы.

Программная реализация представляет собой веб-сервис. В практических задачах данная программа должна получать на вход заполненный граф и координаты стартовой и финишной точек, а на выходе возвращать полученный маршрут в виде последовательности точек. Однако для наглядности результатов тестирования интерфейс программы был изменен. На вход программа принимает изображение карты некоторой местности, а также координаты стартовой и финишной точек. Внутри производится преобразование карты в граф с последующим вычислением маршрута. В качестве результата программа возвращает исходное изображение с отрисованным на нем маршрутом. В качестве входных параметров программа принимает начальное значение используемого в алгоритме граничного значения, а также шаг его изменения, которые подбираются вручную исходя из набора значений метрики разнообразия подстилающей поверхности в графе.

Для проверки работы алгоритма была создана тестовая карта, изображенная на рис. 3. Карта размечена таким образом, чтобы было возможно однозначно соотнести каждый пиксель с определенным классом объектов. На приведенных далее изображениях цифрой 1 отмечены области, принадлежащие к классу «водоем», цифрой 2 – к классу «здания», цифрой 3 – к классу «дороги», цифрой 4 – к классу «деревья», цифрой 5 – к классу «земля, песок», цифрой 6 – к классу «иное». Размер области, используемый для расчета метрики разнообразия подстилающей поверхности, был задан 50×50 пикселей. Общий размер изображения карты – 1024×1024 пикселя. Далее будут приведены описания основных из проведенных тестов, а также представлены их результаты. Для каждого из тестов будет приведено два варианта маршрута – кратчайший и являющийся результатом работы разработанного алгоритма. Стартовая точка отмечена как А, финишная – как В. Маршрут отображен линией, на рисунках отмеченной цифрой 7.

Первый тест: стартовая и финишная точки расположены таким образом, что при движении по кратчайшему пути между ними на пути встретятся две области, полностью помеченные как «лес», в которых вероятность потери опорных точек велика. При планировании маршрута необходимо их обогнуть. Результаты теста приведены на рис. 4. Как видно, предложенный разработанным алгоритмом маршрут соответствует требованиям и проходит вне указанных областей.

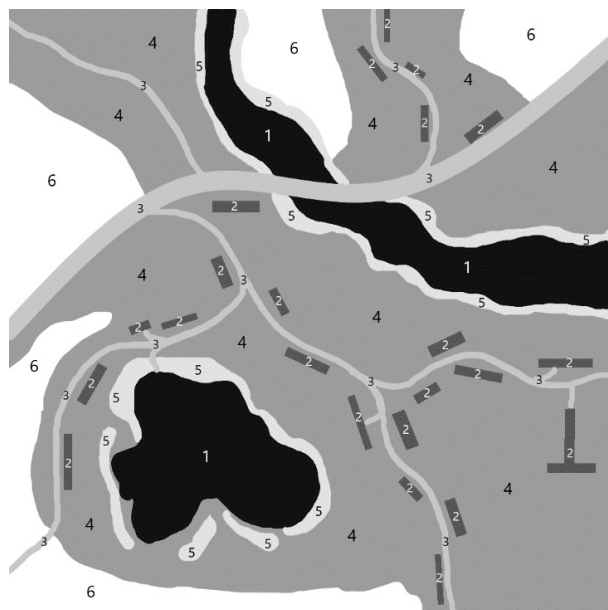


Рис. 3. Карта, используемая при тестировании

Fig. 3. The map used for testing

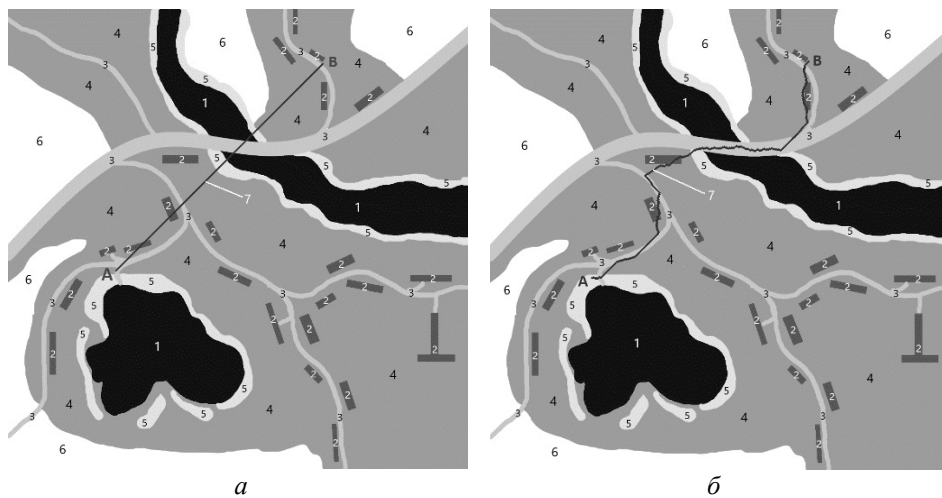


Рис. 4. Первый тест, две зоны с большой вероятностью потери опорных точек:

а – кратчайший маршрут; *б* – результат работы предложенного алгоритма

Fig. 4. The first test, two zones with a high probability of losing pivot points:

a is the shortest route; *b* is the result of the proposed algorithm

Второй тест: стартовая и финишная точки расположены на близком расстоянии друг от друга, но на разных берегах реки, которую невозможно пересечь напрямую без риска возникновения проблемы потери опорных точек. Предлагаемый маршрут должен пройти до моста, пересечь реку около него и затем дойти до финишной точки. Результаты теста приведены на рис. 5. Как видно, предложенный разработанным алгоритмом маршрут соответствует

требованиям и пересекает реку в районе моста с минимизацией риска потери опорных точек.

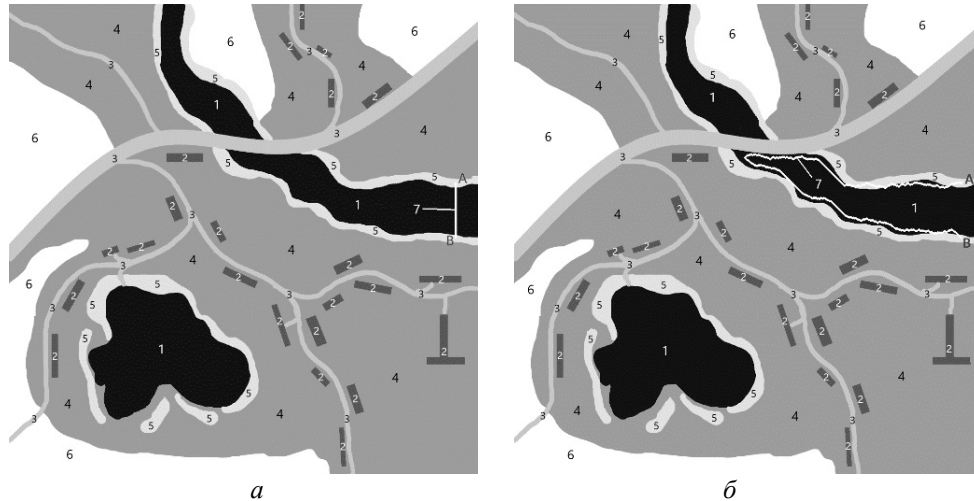


Рис. 5. Второй тест, пересечение реки:

a – кратчайший маршрут; *б* – результат работы предложенного алгоритма

Fig. 5. The second test, crossing the river:

a is the shortest route; *b* is the result of the proposed algorithm

Третий тест проверяет возможность работы алгоритма при увеличении расстояния между стартовой и финишной точками. Для этого стартовая и финишная точки были расположены на противоположных концах карты. Кратчайший маршрут проходит сразу через несколько областей с высокой вероятностью потери опорных точек. Предлагаемый маршрут должен избежать их. Результаты теста приведены на рис. 6. Как видно, предложенный разработанным алгоритмом маршрут соответствует требованиям и проходит вне областей с высокой вероятностью потери опорных точек. Маршрут проходит вдоль дорог и зданий, которые считаются наиболее качественными ориентирами и для которых вероятность потери опорных точек является минимальной.

Четвертый тест проверяет возможность работы алгоритма при необходимости на одном из участков двигаться в направлении, противоположном направлению кратчайшего пути от стартовой точки к финишной. Для этого был использован описанный в предыдущем разделе пример: стартовая точка была расположена на мысу, с трех сторон окруженном водой, а финишная – на противоположном берегу озера. Кратчайший маршрут БПЛА должен вначале пойти в обратном направлении, чтобы покинуть мыс, а затем пройти вдоль береговой линии к финишной точке. Результаты теста приведены на рис. 7. Как видно, предложенный разработанным алгоритмом маршрут соответствует требованиям и проходит по спрогнозированному плану.

Как показало тестирование, разработанный алгоритм корректно проложил маршрут во всех проведенных тестах. Ни в одном из случаев маршрут не проходил по тем областям, где велика вероятность потери опорных точек. Также ни в одном из случаев не было отмечено чрезмерного превышения длительности маршрута или его движения по замкнутому циклу.

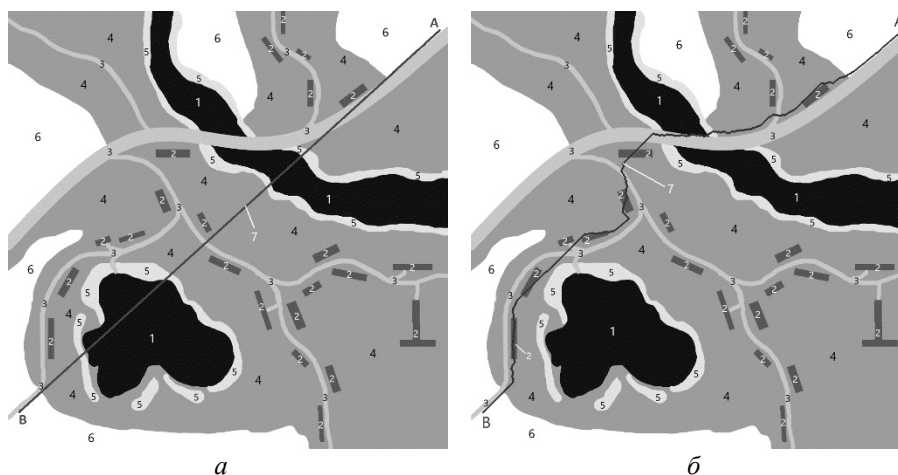


Рис. 6. Третий тест, пересечение всей карты:

a – кратчайший маршрут; *b* – результат работы предложенного алгоритма

Fig. 6. The third test, the intersection of the entire map:
a is the shortest route; *b* is the result of the proposed algorithm

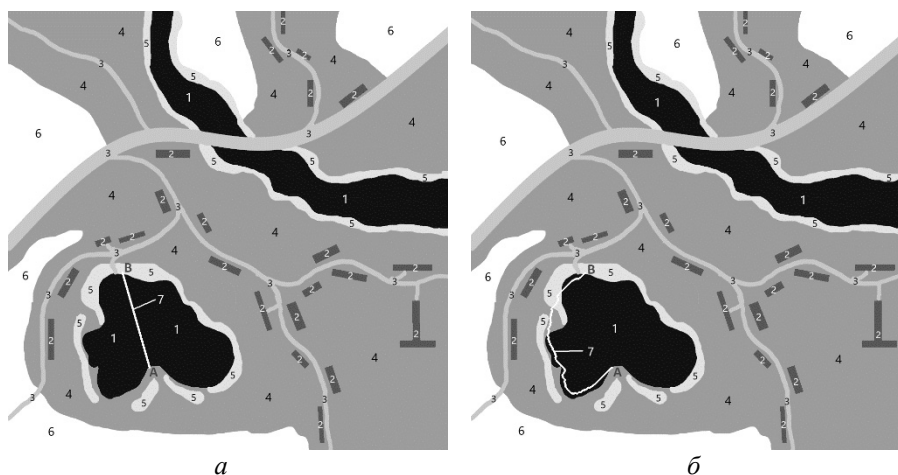


Рис. 7. Четвертый тест, огибание водоема:

a – кратчайший маршрут; *b* – результат работы предложенного алгоритма

Fig. 7. The fourth test, the circumference of the reservoir:
a is the shortest route; *b* is the result of the proposed algorithm

ЗАКЛЮЧЕНИЕ

В настоящей статье была рассмотрена задача планирования маршрута БПЛА, использующего для ориентирования в пространстве систему видеонавигации. Был описан графовый формат данных, используемый для хранения данных о местности, в которой проводится полет. Были проанализированы существующие алгоритмы поиска пути в графе, а также обоснована невозможность их применения в рассматриваемой задаче.

Далее был предложен алгоритм поиска пути в графе, разработанный с учетом особенностей структуры графа, а также используемого критерия оптимальности. На основе предложенного алгоритма была написана и протестирована программа, позволяющая отобразить полученный маршрут в графическом виде. По результатам тестирования был сделан вывод о том, что разработанный алгоритм соответствует заявленным требованиям, корректно планирует маршрут в различных условиях.

Также стоит отметить, что алгоритм имеет потенциал для развития и улучшения. Во-первых, на данный момент начальное значение и шаг изменения граничного значения задаются пользователем вручную. В дальнейшем планируется разработка алгоритма адаптивного подбора этих параметров с учетом содержащихся в графе данных. Во-вторых, на данный момент параметры при оптимизации имеют строго заданный приоритет. Вначале осуществляется оптимизация по минимальному значению метрики разнообразия подстилающей поверхности, а затем по длительности пути. В дальнейшем в алгоритм будет добавлена полноценная многокритериальная оптимизация с возможностью задания коэффициентов, определяющих важность каждого из двух критериев.

СПИСОК ЛИТЕРАТУРЫ

1. *Khachatryan T.B.* A review of visual odometry for UAV autonomous navigation // Вестник Национального политехнического университета Армении. Информационные технологии, электроника, радиотехника. – 2023. – № 1. – С. 9–17. – DOI: 10.53297/18293336-2023.1-9.
2. *Durrant-Whyte H., Bailey T.* Simultaneous localization and mapping (SLAM): Part I // IEEE Robotics and Automation Magazine. – 2006. – Vol. 13 (2). – P. 99–110. – DOI: 10.1109/MRA.2006.1638022.
3. *Durrant-Whyte H., Bailey T.* Simultaneous localization and mapping (SLAM): Part II // IEEE Robotics and Automation Magazine. – 2006. – Vol. 13 (3). – P. 108–117. – DOI: 10.1109/MRA.2006.1678144.
4. *Жук Р.С., Залесский Б.А., Троцкий Ф.С.* Визуальная навигация автономно летящего БПЛА с целью его возвращения в точку старта // Информатика. – 2020. – Т. 17, № 2. – С. 17–24. – DOI: 10.37661/1816-0301-2020-17-2-17-24.
5. *Антюфеев В.И., Быков В.Н.* Сравнительный анализ алгоритмов совмещения изображений в корреляционно-экстремальных системах навигации летательных аппаратов // Авиационно-космическая техника и технология. – 2008. – № 1 (48). – С. 70–74.
6. *Троцкий В.И.* Критерии информативности эталонных радиотепловых полей, используемых в КЭСН // Известия высших учебных заведений. Геодезия и аэрофотосъемка. – 2013. – № 3. – С. 61–64.
7. *Троцкий В.И.* Анализ влияния основных параметров и условий функционирования КЭСН летательных аппаратов по микроволновому излучению покровов земной поверхности на эффективность их использования // Научный вестник Московского государственного технического университета гражданской авиации. – 2018. – Т. 21, № 2. – С. 171–180. – DOI: 10.26467/2079-0619-2018-21-2-171-180.
8. *Титаренко А.А., Козлов В.А., Белинский А.В.* Результаты экспериментальных исследований точностных характеристик корреляционно-экстремальной системы навигации по малоинформативным поверхностным геофизическим полям // Шарыгинские чтения: Всероссийская научная конференция ведущих научных школ в области радиолокации, радионавигации и радиоэлектронных систем передачи информации. – 2019. – № 1 (1). – С. 117–122.
9. *Кулешов С.В., Зайцева А.А.* Варианты реализации системы технического зрения для автовзлета и автопосадки БАС // Известия ЮФУ. Технические науки. – 2018. – № 1 (195). – С. 284–293. – DOI: 10.23683/2311-3103-2018-1-284-293.
10. *Коновалов К.Д., Кулешов С.В.* Общая структура системы визуальной навигации БПЛА по видеоданным // Экстремальная робототехника. – 2024. – № 1 (34). – С. 456–460.

11. Донцов А.А., Тонконогов О.Ю., Нагалин Д.А. Оптимизация расположения опорных точек для повышения точности определения координат летательного аппарата // Информатика: проблемы, методология, технологии: сборник материалов XVIII Международной научно-методической конференции. – Воронеж, 2018. – Т. 2. – С. 134–139.
12. Kaehler A., Bradski G. Learning OpenCV 3. – USA: O'Reilly Media, 2016. – 1024 p.
13. Dijkstra E.W. A note on two problems in connexion with graphs // *Numerische Mathematik*. – 1959. – Vol. 1 (1). – P. 269–271. – DOI: 10.1007/BF01386390.
14. Bellman R. On a routing problem // *Quarterly of Applied Mathematics*. – 1958. – Vol. 16 (1). – P. 87–90. – DOI: 10.1090/QAM/102435.
15. Близнякова Е.А., Куликов А.А., Куликов А.В. Сравнительный анализ методов поиска кратчайшего пути в графе // *Архитектура, строительство, транспорт*. – 2022. – № 1 (99). – С. 80–87. – DOI: 10.31660/2782-232X-2022-1-80-87.
16. Goldberg A.V., Radzik T. A heuristic improvement of the Bellman-Ford algorithm // *Applied Mathematics Letters*. – 1993. – Vol. 6 (3). – P. 3–6. – DOI: 10.1016/0893-9659(93)90022-F.
17. Harabor D., Grastien A. Online graph pruning for pathfinding on grid maps // *Proceedings of the AAAI Conference on Artificial Intelligence*. – 2011. – Vol. 25 (1). – P. 1114–1119. – DOI: 10.1609/aaai.v25i1.7994.
18. Hart P.E., Nilsson N.J., Raphael B. A formal basis for the heuristic determination of minimum cost paths // *IEEE Transactions on Systems Science and Cybernetics*. – 1968. – Vol. 4 (2). – P. 100–107. – DOI: 10.1109/TSSC.1968.300136.
19. Lawrence M.F., Tarjan R.E. Fibonacci heaps and their uses in improved network optimization algorithms // *Journal of the Association for Computing Machinery*. – 1987. – Vol. 34 (3). – P. 596–615. – DOI: 10.1145/28869.28874.

Коновалов Константин Дмитриевич, аспирант лаборатории автоматизации научных исследований Федерального государственного бюджетного учреждения науки «Санкт-Петербургский федеральный исследовательский центр Российской академии наук». Основное направление научных исследований – системы видеонавигации летательных аппаратов, методы компьютерного зрения. Имеет 6 печатных работ. E-mail: kodmko@mail.ru

Konovalov Konstantin D., a PhD student at the laboratory of research automation in the St. Petersburg Federal Research Center of the Russian Academy of Sciences. The main field of his scientific research is aircraft video navigation systems and methods of computer vision. He has 6 publications. E-mail: kodmko@mail.ru

DOI: 10.17212/2782-2001-2024-2-37-54

An algorithm for planning the UAV route in conditions of a lack of visual landmarks*

K.D. KONOVALOV

St. Petersburg Federal Research Center of the Russian Academy of Sciences, 39, V.O. 14th line, 199178, Saint-Petersburg, Russian Federation

kodmko@mail.ru

Abstract

Today, unmanned aircraft is used to solve a variety of tasks. One of the problems that arise when using unmanned aerial vehicles (UAVs) is the problem of navigation, in particular, determining their exact location in space. Today, there are various methods to solve this problem. A promising way to solve it in conditions where it is impossible to use a satellite navigation system is video navigation systems that use visual landmarks to determine location. In this paper,

* Received 21 March 2024.

an overview of the existing types of UAV video navigation systems was carried out. The peculiarities of the operation of video navigation systems using the comparison of visual landmarks with satellite maps were also highlighted. The task was set to find a route in conditions of a lack of visual landmarks, in which the problem of losing reference points may arise. The graph data format used is described, which stores the values of the metric of the diversity of the underlying surface, which characterizes the magnitude of the probability of the problem of losing reference points. The existing algorithms for finding a path in a graph are considered and the need to develop a new algorithm is justified.

A theoretical description of the algorithm was proposed, in which the path search process is divided into two stages: first, a solution option is searched for with the maximum value of the metric of diversity of the underlying surface, and then a route with the minimum length is selected. Several options of the implementation of the proposed algorithm were also presented and their asymptotic complexity was estimated. Based on the theoretical description, a software implementation of the proposed algorithm was developed and an experimental verification of the correctness of its operation on a test card with various flight tasks was performed. According to the test results, the proposed algorithm has shown its effectiveness by constructing routes correctly in all test examples. In the future, this algorithm can be used as part of the UAV video navigation system. The main advantages and disadvantages of the proposed algorithm were also described and ways for its further improvement were proposed.

Keywords: UAV, video navigation, graph algorithm, technical vision system, autonomous visual navigation, iterative algorithm, graph theory, computational complexity, optimization on graphs

REFERENCES

1. Khachatryan T.B. A review of visual odometry for UAV autonomous navigation. Vestnik Natsional'nogo politekhnicheskogo universiteta Armenii. Informatsionnye tekhnologii, elektronika, radiotekhnika. = Proceedings of NPUA. Information technologies, Electronics, Radio engineering, 2023, no. 1, pp. 9–17. DOI: 10.53297/18293336-2023.1-9.
2. Durrant-Whyte H., Bailey T. Simultaneous localization and mapping (SLAM): Part I. IEEE Robotics and Automation Magazine, 2006, vol. 13 (2), pp. 99–110. DOI: 10.1109/MRA.2006.1638022.
3. Durrant-Whyte H., Bailey T. Simultaneous localization and mapping (SLAM): Part II. IEEE Robotics and Automation Magazine, 2006, vol. 13 (3), pp. 108–117. DOI: 10.1109/MRA.2006.1678144.
4. Zhuk R.S., Zalesky B.A., Trotski P.S. Vizual'naya navigatsiya avtonomno letyashchego BPLA s tsel'yu ego vozvrashcheniya v tochku starta [Solution of problem of returning to the starting point of autonomously flying UAV by visual navigation]. Informatika = Informatics, 2020, Vol. 17, no. 2, pp. 17–24. DOI: 10.37661/1816-0301-2020-17-2-17-24.
5. Antyufeev V.I., Bykov V.N. Sravnitel'nyi analiz algoritmov sovmeshcheniya izobrazhenii v korrelyatsionno-ekstremal'nykh sistemakh navigatsii letatel'nykh apparatov [Comparative analysis of image combination algorithms in correlation-extremal aircraft navigation systems]. Aviatsionno-kosmicheskaya tekhnika i tekhnologiya = Aerospace engineering and technology, 2008, no. 1 (48), pp. 70–74.
6. Troitsky V.I. Kriterii informativnosti etalonnnykh radioteplovykh polei, ispol'zuemykh v KESN [Criteria for the information content of reference radio-thermal fields used in CESN]. Izvestiya vysshikh uchebnykh zavedenii. Geodeziya i aerofotos"emka = Izvestia vuzov. Geodesy and aerophotosurveying, 2013, no. 3, pp. 61–64.
7. Troitsky V.I. Analiz vliyaniya osnovnykh parametrov i uslovii funktsionirovaniya KESN letatel'nykh apparatov po mikrovolnovomu izlucheniyu pokrovov zemnoi poverkhnosti na effektivnost' ikh ispol'zovaniya [Analysis of the influence of the main parameters and condition of work of aircraft CESN on the efficiency of their use methods of mathematical modeling]. Nauchnyi Vestnik Moskovskogo gosudarstvennogo tekhnicheskogo universiteta grazhdanskoi aviatsii = Civil Aviation High Technologies, 2018, vol. 21, no. 2, pp. 171–180. DOI: 10.26467/2079-0619-2018-21-2-171-180.
8. Titarenko A.A., Kozlov V.A., Belinskii A.V. Rezul'taty eksperimental'nykh issledovaniy tochnostnykh kharakteristik korrelyatsionno-ekstremal'noi sistemy navigatsii po maloinformativnym poverkhnostnym geofizicheskim polyam [Results of experimental studies of the accuracy characteristics of a correlation-extremal navigation system on low-information surface geophysical fields]. Sharyginskie chteniya = Sharygin readings, 2019, no. 1 (1), pp. 117–122.

9. Kuleshov S.V., Zaitseva A.A. Variants realizatsii sistemy tekhnicheskogo zreniya dlya avtovzleta i avtoposadki BAS [The variants of implementation of computer vision system for UAS auto-takeoff and auto-landing procedure]. *Izvestiya YFU. Tekhnicheskie nauki = Izvestiya SFedU. Engineering sciences*, 2018, no. 1 (195), pp. 284–293. DOI: 10.23683/2311-3103-2018-1-284-293.

10. Kononov K.D., Kuleshov S.V. Obshchaya struktura sistemy vizual'noi navigatsii BPLA po videodannym [General structure of the UAV visual navigation system based on video data]. *Ekstremal'naya robototekhnika = Extreme Robotics*, 2024, no. 1 (34), pp. 456–460.

11. Dontsov A.A., Tonkonogov O.Yu., Nagalin D.A. [Optimization of the location of control points to improve the accuracy of determining the coordinates of the aircraft]. *Informatika: problemy, metodologiya, tekhnologii [Informatics: problems, methodology, technologies]*. Collection of materials of the XVIII International scientific and methodological conference, Voronezh, 2018, vol. 2, pp. 134–139. (In Russian).

12. Kaehler A., Bradski G. *Learning OpenCV 3*. USA, O'Reilly Media, 2016. 1024 p.

13. Dijkstra E.W. A note on two problems in connexion with graphs. *Numerische Mathematic*, 1959, vol. 1 (1), pp. 269–271. DOI: 10.1007/BF01386390.

14. Bellman R. On a routing problem. *Quarterly of Applied Mathematics*, 1958, vol. 16 (1), pp. 87–90. DOI: 10.1090/QAM/102435.

15. Bliznyakova E.A., Kulikov A.A., Kulikov A.V. Sravnitel'nyi analiz metodov poiska krachaishego puti v grafe [Comparative analysis of methods for finding the shortest distance in a graph]. *Arkhitektura, stroitel'stvo, transport = Architecture, construction, transport*, 2022, no. 1 (99), pp. 80–87. DOI: 10.31660/2782-232X-2022-1-80-87.

16. Goldberg A.V., Radzik T. A heuristic improvement of the Bellman-Ford algorithm. *Applied Mathematics Letters*, 1993, vol. 6 (3), pp. 3–6. DOI: 10.1016/0893-9659(93)90022-F.

17. Harabor D., Grastien A. Online graph pruning for pathfinding on grid maps. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2011, vol. 25 (1), pp. 1114–1119. DOI: 10.1609/aaai.v25i1.7994.

18. Hart P.E., Nilsson N.J., Raphael B. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 1968, vol. 4 (2), pp. 100–107. DOI: 10.1109/TSSC.1968.300136.

19. Lawrence M.F., Tarjan R.E. Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the Association for Computing Machinery*, 1987, vol. 34 (3), pp. 596–615. DOI: 10.1145/28869.28874.

Для цитирования:

Коновалов К.Д. Алгоритм планирования маршрута БПЛА в условиях недостатка визуальных ориентиров // Системы анализа и обработки данных. – 2024. – № 2 (94). – С. 37–54. – DOI: 10.17212/2782-2001-2024-2-37-54.

For citation:

Kononov K.D. Algorithm planirovaniya marshruta BPLA v usloviyakh nedostatka vizual'nykh orientirov [An algorithm for planning the UAV route in conditions of a lack of visual landmarks]. *Sistemy analiza i obrabotki dannykh = Analysis and Data Processing Systems*, 2024, no. 2 (94), pp. 37–54. DOI: 10.17212/2782-2001-2024-2-37-54.