

Д. Н. Алдошкин, аспирант, aldoshkind@gmail.com, **Р. Ю. Царев**, канд. техн. наук, доц., tsarev.sfu@mail.ru, Институт космических и информационных технологий, Сибирский федеральный университет, г. Красноярск

Поиск пути мобильного робота в условиях наличия препятствий и неполноты информации о среде

Рассматривается задача поиска пути мобильного робота в недетерминированной среде. Приводится описание предлагаемого вычислительно-эффективного решения задачи поиска пути в среде, которая априори неизвестна мобильному роботу. Приводится постановка задачи поиска пути мобильного робота при одновременной локализации и картировании. При этом информация о среде робот оперативно получает исключительно посредством собственных датчиков. Приводится краткий обзор существующих решений, а также их сравнительный анализ с предложенным алгоритмом, показавший превосходство в быстродействии последнего.

Ключевые слова: одновременная локализация и картирование, мобильный робот, поиск пути, построение маршрута, планирование траектории

Введение

При одновременной локализации и картировании среды, информация о которой на момент начала работы мобильного робота отсутствует, основными задачами являются задача построения карты окружающего пространства, задача определения собственного положения мобильного робота в пространстве и задача обеспечения беспрепятственного его передвижения к целевой точке. Решение этих задач предоставляет базис для решения задач более высокого уровня.

На сегодняшний день существует множество публикаций, описывающих алгоритмы решения задачи управления передвижением робота [1–4]. Данные алгоритмы можно разделить на следующие группы:

1) алгоритмы, не планирующие траекторию. В этом случае направление движения в каждый момент (такт) времени вычисляется как некоторая функция от данных, полученных от датчиков окружающего пространства [1];

2) алгоритмы, планирующие траекторию, среди которых можно выделить

2.1) алгоритмы, использующие для описания среды (картографирования) периодические структуры (сетки) [2, 3];

2.2) алгоритмы, использующие для картографирования неперiodические графовые формализации [4].

Алгоритмы первой группы применимы для первоначального сбора картографической информации (в случае начала работы, либо ориентирования в неразведанной области среды), а также для избегания столкновений с динамическими препятствиями.

Алгоритмы второй группы используют всю доступную информацию об окружающем пространстве. Применение периодических сеток требует принятия компромиссного решения: высокая плотность сетки позволяет получить большую детализацию, но приводит к чрезмерным затратам на описание пространства, и наоборот — низкая плотность сетки имеет следствием снижение детализации описания пространства. Этому недостатка лишены алгоритмы, использующие неперiodические графовые формализации: они позволяют ми-

нимизировать затраты на описание пространства, сохраняя при этом контроль над детализацией.

Наиболее общим и широко известным алгоритмом, реализующим подход неперiodических графовых формализаций, является алгоритм, описанный О'Рурком в работе [4]. Этот алгоритм в качестве абстракции данных картографирования использует граф видимости препятствий. Для решения задачи поиска пути на графе используется алгоритм Дейкстры в обобщенной формулировке, что позволяет для решения конкретной задачи с ее ограничениями выбрать подходящую модификацию алгоритма. Далее в работе алгоритм О'Рурка используется в качестве эталонного как для теоретического анализа, так и для сравнения производительности с предлагаемым алгоритмом.

Формализация задачи

В данной работе рассматривается задача в ее классической постановке, которую можно сформулировать следующим образом: в окружающем пространстве $\Omega = E \cup K$, содержащем как препятствия $E = \{E_i, i = 1, e\}$ (e — число препятствий), так и свободное пространство $K = \{K_i, i = 1, k\}$ (k — число взаимно недоступных областей свободного пространства), $E \cap K = \emptyset$, найти такой путь $M = \{M_i | M_i \in K, i = 1, m\}$ (m — число точек пути) в свободном пространстве, чтобы при движении робота из начальной точки M_1 в конечную точку M_m ни в одной точке M_i этого пути он не сталкивался с препятствиями: $M \cap E = \emptyset$ (рис. 1).

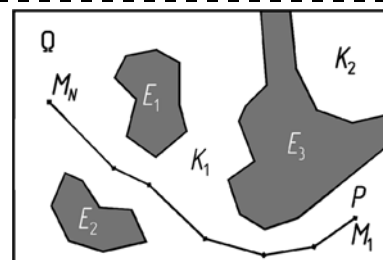


Рис. 1. Схема пространства с препятствиями и возможная траектория движения робота

Формальное описание алгоритма

Целью работы алгоритма является построение траектории движения M . Исходными данными являются данные картографирования, содержащие информацию о геометрии известного на данный момент окружающего пространства, а также информация о геометрии известных препятствий.

В предлагаемом алгоритме, равно как и в эталонном, можно выделить два этапа:

- 1) преобразование исходных данных к графовому представлению;
- 2) поиск пути на графе.

В соответствующих разделах оба этих этапа рассматриваются подробно, описываются ограничения на исходные данные, приводятся теоретические и экспериментальные оценки быстродействия их возможных реализаций.

Предлагаемый алгоритм разделяется на следующие шаги:

- 1) построение многоугольника свободного пространства $P(K)$:

- 1.1) инициализация пустого $P(K)$;
- 1.2) построение контура окружающего пространства $C(\Omega)$ в виде ломаной кривой;
- 1.3) дополнение $P(K)$ ломаной $C(\Omega)$ как ограничивающей многоугольник кривой;
- 1.4) для каждого препятствия E_i в E :
 - 1.4.1) построение контура препятствия $C(E_i)$;
 - 1.4.2) дополнение $P(K)$ ломаной $C(E_i)$ как дыркой;

- 2) построение триангуляции многоугольника свободного пространства $T = T(P(K))$;

- 3) построение дуального графа триангуляции $G(T)$;
- 4) определение вершин графа $G(T)$, соответствующих текущему положению и целевому положению;

- 5) определение пути W из текущего положения в целевое через поиск пути на графе $W = \text{Dijkstra}(G(T))$;

- 6) преобразование графового представления пути W в геометрическое представление M .

Шаги 1 и 6 зависят от конкретной постановки задачи, и их выполнение в данной статье не рассматривается. Шаги 2 и 3 составляют первый этап работы алгоритма, шаги 4 и 5 — второй.

Преобразование исходных данных к графовому представлению

Без потери общности можно сказать, что наземные мобильные роботы имеют две степени свободы по перемещению, поэтому для решения задачи построения траектории достаточным является анализ двумерной карты среды.

Исходными данными для процедуры построения траектории движения мобильного робота являются текущие данные картографирования окружающей среды, положение мобильного робота в ней (начальная точка), координаты конечной точки и геометрические параметры робота.

Для того чтобы данные картографирования были пригодными для работы предлагаемого алгоритма,

они должны быть предварительно обработаны и представлены в виде совокупности контуров объектов окружающего пространства (с учетом координат объектов).

В рамках предлагаемого алгоритма на форму представления этих данных налагается следующее ограничение: всякий контур i -го препятствия $C(E_i)$ должен быть аппроксимирован ломаной кривой, т. е. представлять собой совокупность двух множеств: множества вершин $D_i = \{D_{ij}, j = \overline{1, d_i}\}$ (d_i — число вершин контура i -го препятствия) и множества соединяющих вершины ребер $F_i = \{F_{ij}, j = \overline{1, f_i}\}$ (f_i — число ребер контура i -го препятствия): $C(E_i) = \{D_i, F_i\}$. Контур не должен иметь самопересечений: $F_{ia} \cup F_{ib} = \emptyset$; $a, b = \overline{1, f_i}$; $a \neq b$. Самопересечение контуров не допускается по двум причинам: а) контуры с самопересечениями не имеют физического смысла и свидетельствуют об ошибочной работе алгоритмов обработки данных датчиков пространства; б) контуры с самопересечениями не поддерживаются большим числом алгоритмов триангуляции по причине неоднозначности трактовки подобных контуров.

Геометрия робота может быть представлена в виде:

- точечного объекта;
- ограничивающей окружности ненулевого радиуса;
- ограничивающего выпуклого многогранника;
- многогранника, детально описывающего форму робота.

Наиболее часто рассматриваемый случай — ограничивающая окружность. Это обусловлено тем, что в этом случае можно построить траекторию, исключая столкновения при сравнительно высокой производительности. Решение задачи с описанием робота в виде точечного объекта в чистом виде на практике не применимо, однако к этой задаче несложными преобразованиями сводится задача с описанием робота в виде окружности [4]. Полноценное же решение задачи с применением представлений в виде ограничивающего выпуклого многогранника и многогранника, детально описывающего форму робота, является существенно более сложной в алгоритмическом смысле задачей, поэтому не применяется на практике в силу таких причин, как ограниченность вычислительных ресурсов мобильного робота и избыточная точность результата.

Таким образом, и алгоритм О'Рурка, и предлагаемый алгоритм используют точечное представление геометрии мобильного робота.

Предлагаемый алгоритм для решения задачи поиска пути использует графовую абстракцию исследуемого пространства. Этот подход к описанию среды выбран по той причине, что на данный момент чрезвычайно хорошо разработан и исследован аппарат анализа графов, в том числе поиска путей.

Как было сказано выше, вершинами графа могут выступать как узлы периодической сетки, так и непериодические элементы, характеризующие среду с достаточной точностью.

Периодические сетки были исключены из рассмотрения на ранних этапах анализа поставленной задачи по следующим причинам:

- для построения сетки необходимо выбрать шаг, что является нетривиальной задачей;
- периодические сетки обладают существенной избыточностью: для описания одного физически непрерывного сегмента пространства могут использоваться несколько ячеек, которые при переходе к графовому представлению увеличат объем графа.

В эталонном алгоритме предлагается в качестве узлов выбирать вершины контуров препятствий, а в качестве ребер графа использовать отношения видимости между вершинами (т. е. все множество диагоналей полигонов свободного пространства). Такая формализация позволяет обнаруживать путь из начальной точки в целевую. При этом данный путь будет являться оптимальным в смысле геометрической длины. К минусам алгоритма можно отнести:

- большой объем графа: число его вершин равно числу N всех вершин всех контуров препятствий, а число ребер графа в худшем случае достигает $\frac{N^2 - 3N}{2}$ (случай выпуклого многоугольника), т. е. можно считать, что число ребер графа (диагоналей) имеет оценку [4]

$$R = O(N^2); \quad (1)$$

- большое число циклов на графе, что налагает ограничения на алгоритмы обхода графа и поиска путей.

Следствием большого объема информации, на основании которой строится граф, является большая алгоритмическая сложность его построения. Наилучшими оценками обладают алгоритм Велзля ($O(N^2)$) [5] и алгоритм Гоша и Маунта ($O(N \log N + R)$) [4]. Сложность последнего зависит от числа ребер, имеющего оценку $R = O(N^2)$. С учетом данной оценки сложность алгоритма построения графа видимости в худшем случае составляет $O(N \log N + N^2)$. По этой причине далее рассматривается алгоритм Велзля.

Как будет показано ниже, поставленную задачу поиска пути без ограничений на его оптимальность можно решить существенно меньшими вычислительными затратами. Чтобы достигнуть этого, необходимо выбрать такую абстракцию свободного пространства, которая бы описывала его исчерпывающим образом и имела существенно меньший объем, нежели в эталонном алгоритме. В качестве такой абстракции выберем триангуляцию полигонов свободного пространства. Для всякой триангуляции существует дуальный граф, который и будет в дальнейшем использован для поиска пути. Дуальный граф представляет собой граф, вершины которого соответствуют треугольникам триангуляции, а ребрами выступают отношения смежности между треугольниками (наличия общего ребра).

Дуальный граф триангуляции, с одной стороны, полностью описывает структуру свободного пространства, а с другой, имеет объем, меньший объема графа, получаемого при использовании эталонного алгоритма. Число вершин дуального графа является величиной, зависящей от входных данных (в первую очередь от числа препятствий), и составляет $N + 2e - 2$, где e — число препятствий. Оценка числа вершин составляет $O(N)$, что совпадает с таковой для вышеописанного алгоритма. Число ребер также зависит от входных данных и составляет $N + 3e - 3$ (с оценкой $O(N)$), что существенно превосходит оценку (1).

Таким образом, объем графа в рамках предлагаемого алгоритма существенно меньше в смысле наилучших оценок, нежели в эталонном алгоритме. Этот факт имеет значимость в рамках анализа алгоритмов поиска путей на графах, который будет приведен далее.

Не менее существенным вопросом является вопрос алгоритмической сложности построения триангуляции. Существует множество алгоритмов построения триангуляции с различными оценками сложности [4]. Оценка сложности лучшего алгоритма составляет $O(N)$, что существенно превосходит оценку $O(N^2)$ в алгоритме О'Рурка для решения задачи поиска пути. Однако на практике реализация такого алгоритма является чрезвычайно сложной и медленной, поэтому применение находят алгоритмы с теоретически незначительно худшей производительностью. В рамках компьютерной апробации, результаты которой описаны далее, применяется алгоритм Зейделя с оценкой сложности $O(N \log^* N + k \log N)$ [6] (здесь и далее $\log^* N$ — итерированный логарифм).

Поиск пути на графе

На сегодняшний день задача поиска пути на графах изучена чрезвычайно хорошо. Среди предлагаемых алгоритмов в первую очередь следует отметить алгоритм Дейкстры, предназначенный для обнаружения кратчайшего пути на графе. Другим широко известным алгоритмом является алгоритм A^* — модификация алгоритма Дейкстры, основывающаяся на применении эвристик [7]. A^* в общем случае превосходит алгоритм Дейкстры, однако в худшем случае оценки их быстродействия совпадают. Как существенный минус A^* можно указать стремление его к выходу в локальные минимумы с исчерпывающим там пребыванием, что приводит к существенному снижению производительности в задачах поиска пути в условиях наличия большого числа препятствий сложной невыпуклой формы. В силу своей эвристической природы алгоритм A^* имеет множество модификаций и разновидностей: взвешенный WA^* [8], декомпозирующий HGA^* [8], R^* [9].

В рамках предлагаемого алгоритма применяется алгоритм Дейкстры по той причине, что он не обладает склонностью к длительному пребыванию в локальном минимуме. При этом оценка сложности

данного алгоритма, реализованного с использованием кучи Фибоначчи, составляет [10]

$$O(V \log V + R), \quad (2)$$

где V — число вершин графа, R — число ребер.

Переход от алгоритма Дейкстры к алгоритму A^* (или его модификациям) не составляет труда, однако требует построения эвристик, специфичных для ограничений конкретной постановки задачи. Вопрос построения таких эвристик не является предметом описания данной работы.

Как было указано ранее, число вершин для графа видимости $V = N$ (где N — число вершин всех контуров препятствий), число ребер графа видимости оценивается как $R = O(N^2)$. В этом случае оценка сложности (2) работы алгоритма Дейкстры будет составлять $O(N \log N + N^2)$.

Для дуального графа триангуляции оценка числа вершин составляет $V = O(N)$, оценка числа ребер $R = O(N)$, что в итоге дает оценку быстродействия $O(N \log N + N)$.

Алгоритм поиска пути состоит из двух этапов: построения графа свободного пространства и поиска пути на этом графе. Сложность данного алгоритма можно подсчитать как сумму сложностей его этапов. Таким образом, очевидно, что сложность алгоритма, описанного О'Рурком, составляет $O(N \log N + N^2)$. Сложность же предлагаемого алгоритма оценивается как $O(N \log N + N)$. Данная теоретическая разница подтверждается результатами компьютерной апробации, результаты которой приводятся ниже.

Компьютерная апробация

Целью компьютерной апробации является подтверждение работоспособности предлагаемого алгоритма, а также демонстрация превосходства его в производительности в сравнении с известными алгоритмами.

В рамках апробации проведено сравнение функционирования двух алгоритмов: алгоритма, основывающегося на графе видимости [4] (эталонного), и предлагаемого алгоритма.

В качестве алгоритма построения графа видимости, как уже было упомянуто, применяется алгоритм Велзля, обладающий оценкой сложности $O(N^2)$. В силу отсутствия открытой реализации данного алгоритма он был реализован автором с применением языка программирования C++.

В качестве реализации алгоритма триангуляции применяется алгоритм Зейделя, имеющий сложность $O(N \log^* N + k \log N)$, которая слабо отличается от линейной при постоянном k в силу чрезвычайно медленного роста итерированного логарифма. Реализация данного алгоритма предоставляется библиотекой *poly2tri*.

Алгоритм Дейкстры также реализован авторами по той причине, что существующие открытые реализации либо чрезвычайно обобщены, либо сложно стыкуются с реализациями описанных выше алгоритмов.

Апробацию проводили на синтетических данных. Причиной этому является большая трудоемкость подготовки выборок реальных данных с контролируемым шагом размерности выборок. Синтетические же данные генерируются процедурно, вследствие чего контроль над размерностью и иными параметрами выборок не составляет проблемы.

В апробационном эксперименте использовалась сцена следующего вида: окружающее пространство Ω прямоугольной формы, в котором в строках и столбцах располагаются препятствия E_i шестиугольной формы (рис. 2). Минимальная выборка содержит два столбца и одну строку. Число препятствий в каждой следующей выборке увеличивается на один столбец и одну строку. Такой подход, с одной стороны, обеспечивает шаг размерности выборок данных, позволяющий продемонстрировать закономерность роста времени работы того или иного алгоритма, а с другой стороны, позволяет не затрачивать большое количество времени на обработку выборок, различающихся по размеру не существенно, и, тем самым, быстро перейти к анализу выборок большего размера.

В результате апробации получен ряд зависимостей, демонстрирующих существенные характеристики работы алгоритмов, а именно:

- время работы первого этапа (построение графа) (рис. 3);
- объем результирующего графа в вершинах и ребрах (рис. 4);
- время работы второго этапа (поиск пути на графе) (рис. 5);
- общее время работы (рис. 6).

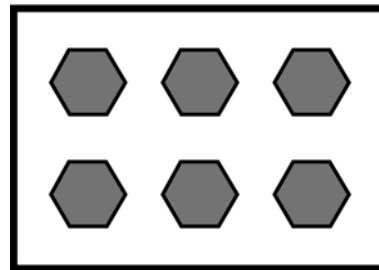


Рис. 2. Образец экспериментальной сцены. Серые области соответствуют препятствиям E_i , белая область — окружающему пространству Ω

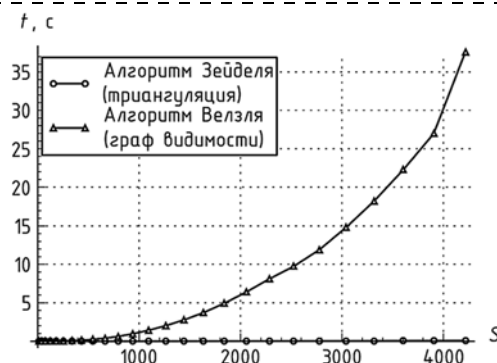


Рис. 3. Сравнение времени работы первого этапа (построение графа)

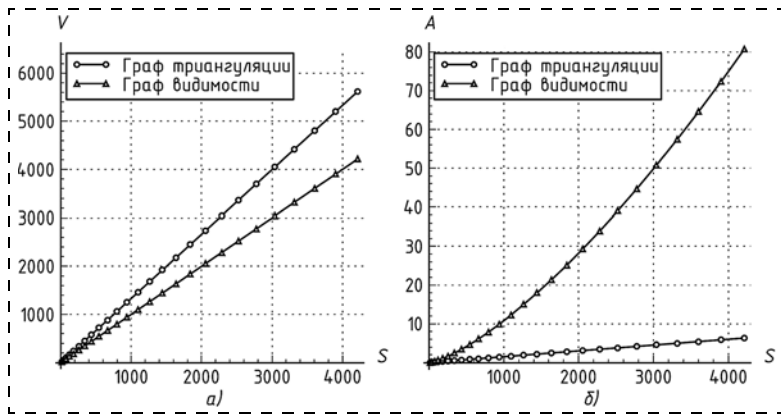


Рис. 4. Сравнение объема графа: a — вершины; b — ребра

Здесь на рисунках введены следующие обозначения: S — объем выборки (в точках сцены); t — время (в секундах); V — объем графа (в вершинах); A — объем графа (в тысячах ребер).

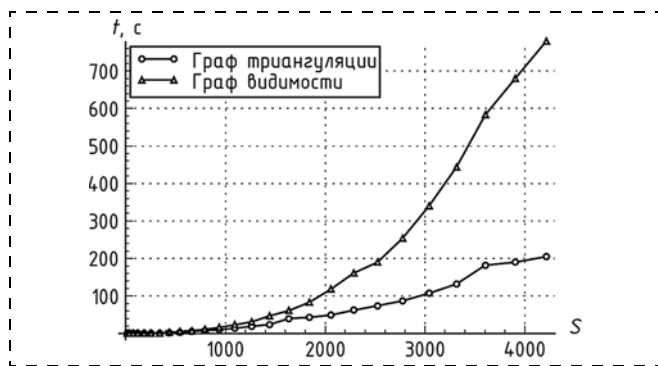


Рис. 5. Сравнение времени работы второго этапа (поиск на графе)

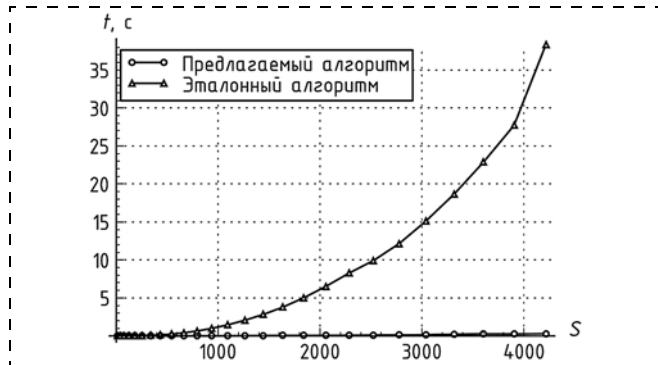


Рис. 6. Сравнение общего времени работы алгоритмов

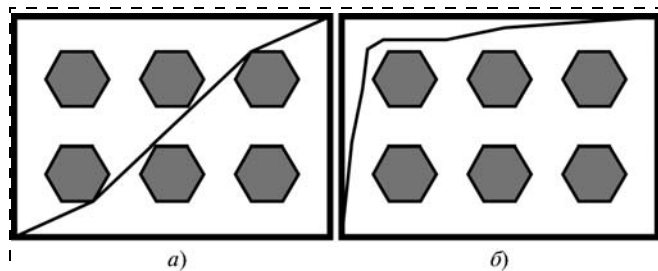


Рис. 7. Траектории, построенные согласно: a — эталонного алгоритма; b — предлагаемого алгоритма

На основании данных, приведенных на рис. 6, можно сделать вывод о существенном преимуществе в быстродействии предлагаемого алгоритма.

Быстродействие предлагаемого алгоритма обусловлено тем, что первый его этап выполняется быстрее соответствующего этапа эталонного алгоритма и вносит наибольший вклад в относительное повышение быстродействия (см. рис. 3). Кроме этого, второй этап также выполняется с меньшими затратами времени (см. рис. 5). Несмотря на то что на втором этапе используется тот же алгоритм (алгоритм Дейкстры), что и в эталонном алгоритме, он выполняется за меньшее время. Это обусловливается тем, что число вершин графа влияет на время работы алгоритма существенно, нежели число ребер

(о чем можно сделать вывод из оценки сложности алгоритма Дейкстры): число вершин анализируемого графа существенно ниже, нежели в эталонном алгоритме (см. рис. 4, a), хотя число ребер и превосходит число ребер в случае эталонного алгоритма (см. рис. 4, b), что в результате приводит к меньшему времени работы (см. рис. 5).

Рис. 7, a , b позволяет оценить качество конечного решения. Предлагаемый алгоритм не дает решения, оптимального в смысле геометрической длины траектории, он позволяет построить траекторию, длина которой оптимальна в смысле метрики графовой абстракции.

В общем случае такая траектория будет существенно отличаться от оптимальной, однако разница может быть минимизирована рядом средств: за счет выбора в качестве вершин графа других геометрических элементов триангуляционной сетки; детектирования выпуклых многоугольников и объединения соответствующих треугольников в графе в единую вершину; геометрической оптимизации результирующей траектории.

Заключение

В данной работе предложен алгоритм решения задачи поиска пути мобильного робота в двумерном пространстве с препятствиями. Показано, что предлагаемый алгоритм обладает существенным преимуществом в быстродействии в сравнении с эталонным алгоритмом. Данное теоретическое преимущество подтверждено на практике вычислительным экспериментом, в рамках которого проведено сравнение отдельных этапов предлагаемого и эталонного алгоритмов.

Однако эксперимент показал, что, несмотря на преимущество в быстродействии, предлагаемый алгоритм не позволяет достичь качества решения, получаемого при использовании эталонного алгоритма.

Таким образом, предлагаемый алгоритм не способен полностью вытеснить эталонный подход, однако его применение целесообразно в ряде случаев: для решения задачи определения существования пути между двумя точками пространства;

для вычисления траектории движения мобильного робота, когда временные затраты на построение решения являются более критичными, нежели оптимальность траектории.

Список литературы

1. Чернухин Ю. В., Бутов П. А. Синтез тормозных квазиполей препятствий для бортовой системы автономного планирования траектории движения малогабаритных мобильных роботов // Инженерный вестник Дона. 2014. № 2. Т. 29. С. 66–76.
2. Михайлов Б. Б. Техническое зрение мобильных роботов // Техническое зрение в системах управления мобильными объектами. Тр. науч.-техн. конф.-семинара. М.: Книжный дом "Университет", 2011. С. 191–201.
3. Хрущ А. В. Управление мобильным роботом с бортовой системой объемного зрения // Техническое зрение в системах управления. Сб. тр. науч.-техн. конф. 2012. С. 62–67.

4. O'Rourke J. Computational Geometry in C. Cambridge University Press, 1998.
5. O'Rourke J. Art gallery theorems and algorithms. New York: Oxford university press, 1987.
6. Seidel R. A simple and fast incremental randomized algorithm for computing trapezoidal decompositions and for triangulating polygons // Computational Geometry: Theory and Applications. 1991. Vol. 1. N. 1. P. 51–64.
7. Бутов П. А. Разработка и исследование элементов систем управления, реализующих автономные режимы навигации для малогабаритных мобильных роботов: дис. ... канд. техн. наук: 05.13.05. Таганрог, 2014.
8. Яковлев К. С., Сарафанов В. Ю., Храмоин И. В. Об одной проблеме, возникающей при планировании траектории на плоскости // Тринадцатая национальная конф. по искусственному интеллекту с междунар. участием КИИ-2012. Тр. конф. Т. 3. Белгород: Изд-во БГТУ, 2012. С. 256–267.
9. Likhachev M., Stentz A. R* Search // Proc. of the National Conference on Artificial Intelligence, 2008.
10. Левитин А. Алгоритмы: введение в разработку и анализ. М.: Вильямс, 2006.

Mobile Robot Path Planning in the Presence of Obstacles and Lack of Information about the Environment

D. Aldoshkin, Postgraduate Student, aldoshkind@gmail.com✉,

R. Tsarev, Ph.D., Associate Professor, tsarev.sfu@mail.ru,

Institute of Space and Information Technologies, Siberian Federal University,
Krasnoyarsk, 660100, Russian Federation

Corresponding author: Aldoshkin Dmitry N., Postgraduate Student,
Institute of Space and Information Technologies, Siberian Federal University,
Krasnoyarsk, 660100, Russian Federation, e-mail: aldoshkind@gmail.com

Received on March 10, 2016

Accepted on March 24, 2016

Mobile robot motion is a complex field of research intensively studied during the recent decades. In this article the authors consider the problem of path planning for a mobile robot with two degrees of freedom (ground vehicle) in a priori undetermined environment. The authors formulate the path planning problem and propose a computationally effective algorithm for solving the problem of movement path construction in a priori unknown environment. The proposed algorithm decomposes the path planning problem in two main phases: construction of a graph-based presentation of a free space and graph search. The authors propose a dual graph of a free space polygon triangulation as a means of a free space presentation. This approach allows us to minimize the graph size without an uncontrollable loss of accuracy of environment representation. This approach uses Dijkstra algorithm for the conduct path search on a free space graph, but it is possible to replace this algorithm by another one with account of the particular constraints. The proposed algorithm envisages a continuous collection of the environment information by a robot using its sensors. A brief review of the existing approaches to the problem is presented, as well as a theoretical comparison of these approaches with the proposed one in terms of the computational complexity theory. The proposed algorithm's computational efficiency is demonstrated via a computational experiment, where the proposed and reference algorithms operate on a set of synthetic environment models of different geometric size, but with a similar structure. An experiment proved the algorithm's supremacy in speed.

Keywords: simultaneous localization and mapping, mobile robot, path finding, trajectory planning, path planning

For citation:

Aldoshkin D., Tsarev R. Mobile Robot Path Planning in the Presence of Obstacles and Lack of Information about the Environment, *Mekhatronika, Avtomatizatsiya, Upravlenie*, 2016, vol. 17, no. 7, pp. 465–470.

DOI: 10.17587/mau.17.465-470

References

1. Chernuhin Ju. V., Butov P. A. Sintez tormoznykh kvazipolej prepyatstvij dlja bortovoj sistemy avtonomnogo planirovaniya traektorii dvizhenija malogabaritnyh mobil'nyh robotov (Synthesis of brake obstacle quazifields for onboard autonomous movement trajectory planning system of compact mobile robot), *Inzhenernyj Vestnik Dona*, 2014, vol. 29, no. 2, pp. 66–76 (in Russian).
2. Mihajlov B. B. Tehnicheskoe zrenie mobil'nyh robotov (Mobile robot technical vision), *Tehnicheskoe Zrenie v Sistemah Upravlenija Mobil'nymi Obektami: Trudy Nauchno-Tehnicheskoj Konferencii-Seminara*, Knizhnyj dom "Universitet", 2011, pp. 191–201 (in Russian).
3. Hrushh A. V., Mihajlov B. B. Upravlenie mobilnym robotom s bortovoj sistemoj obemnogo zrenija (Controlling mobile robot with onboard volumetric vision system), *Tehnicheskoe Zrenie v Sistemah Upravlenija: Sbornik Trudov Nauchno-Tehnicheskoj Konferencii*, 2012, pp. 62–67 (in Russian).

4. O'Rourke J. Computational Geometry in C. 2 ed. Cambridge University Press, 1998.
5. O'Rourke J. Art gallery theorems and algorithms, New York, Oxford university press, 1987.
6. Seidel R. A simple and fast incremental randomized algorithm for computing trapezoidal decompositions and for triangulating polygons, *Computational Geometry: Theory and Applications*, 1991, vol. 1, no. 1, pp. 51–64.
7. Butov P. A. Razrabotka i issledovanie jelementov sistem upravlenija, realizujushhijh avtonomnye rezhimy navigacii dlja malogabaritnyh mobil'nyh robotov (Development and analysis of control system elements, which implement autonomous navigation modes for compact mobile robots): dis. ... kand. tehn. nauk: 05.13.05, Taganrog, 2014 (in Russian).
8. Jakovlev K. S., Sarafanov V. Ju., Hramoin I. V. Ob odnoj probleme, vznikajushhej pri planirovanii traektorii na ploskosti (On the problem of trajectory planning on the plane), *Proc. of the 13th National Conf. on Artificial Intelligence with International Participation KII-2012* (16–20 oktjabrja 2012 g., Belgorod, Rossija), vol. 3, Belgorod: Izd-vo BGTU, 2012, pp. 256–267 (in Russian).
9. Likhachev M., Stentz A. R* Search, *Proc. of the National Conference on Artificial Intelligence*, 2008.
10. Levitin A. Algoritmy: vvedenie v razrabotku i analiz (Algorithms: introduction and analysis), Moscow, Vil'jams, 2006 (in Russian).