УДК 007.52, 519.878, 519.1, 004.942

МНОГОКРИТЕРИАЛЬНЫЙ СПЛАЙН-АЛГОРИТМ ПОИСКА ПУТИ В НЕСКОЛЬКИХ ГОМОТОПИЯХ

Р.О. Лавренов

В данной работе рассказывается о разработке нового сплайн алгоритма, комбинирующего в себе методы потенциальных полей и графа Вороного, который находит оптимальный путь, в том числе в различных гомотопиях. Оптимальность будет достигаться множеством критериев оптимизации, которые возможно регулировать в режиме реального времени. Созданный алгоритм сравнивается с базовым сплайн алгоритмом и показывает лучшие результаты в быстродействии, количестве итераций и правильности нахождения пути.

***Ключевые слова*: Планирование пути, мобильный робот, граф Вороного, потенциальное поле, B-сплайн.**

Введение

На сегодняшний день разработки в сфере робототехники направлены на замену человеческого труда в случаях, когда это существенно ускорит или обезопасит какой-либо производственный процесс или проведение поисково-спасательных операций в опасных для людей условиях [1]. При этом, современные мобильные роботы должны обладать такими критическими навыками как взаимодействие с пользователями [2] и другими роботами [3, 4, 5], автономное перемещение [6], выстраивание карты окружающего пространства и локализации на карте (SLAM) [7].

Задачи планирования пути подразделяются на глобальный и локальный тип планирования. Задача планирования пути называется глобальной, когда известны все данные об окружающей среде, включая габариты робота, его начальная и целевая позиция и ориентация, а также информация о препятствиях на его пути.

Классический, но до сих популярный, метод планирования пути – метод потенциального поля [8][9]. При этом подходе со стороны целевой точки на робота действуют силы притяжения, а со стороны препятствий – силы отталкивания. Робот движется по линии градиента в направлении минимума потенциального поля [10]. Потенциалы поля могут рассчитываться как для всей карты целиком, так и только для некоторой локальной области карты. Однако, в качестве недостатков метода потенциальных полей следует указать на проблему попадания робота в места локального минимума потенциального поля (с последующим застреванием в таких местах) и возникновение колебаний в узких проходах между препятствиями под действием потенциального поля.

В предыдущем исследовании был предложен метод планирования пути для мобильного робота, основанный на сплайнах [11]. Данный алгоритм использует принципы методов потенциальных полей и дополнительные критерии оптимизации маршрута - такие как длина пути, кривизна и безопасность робота. Несмотря на то, что традиционно сглаживание пути проводят на последних этапах его построения [12], особенностью нового алгоритма является интеграция критерия гладкости маршрута с первой же итерации. Чтобы улучшить алгоритм, добавить гибкость для оптимизации пути и возможность быстрого динамического перепланирования, принято решение интегрировать подход графа Вороного и новый критерии минимального расстояния от препятствий в исходный метод.

1. Сплайн-метод планирования пути

Алгоритм, предложенный в работе [11], предлагает решение задачи планирования пути для мобильного робота в известной среде, заполненной статическими препятствиями. Препятствия представляются в виде конечной группы пересекающихся кругов разного размера. Каждая группа может содержать один или несколько кругов. Идея такого представления заключается в том, что любое невыпуклый многоугольник можно аппроксимировать группой таких кругов.

1.1. Целевая функция

Чтобы гарантировать построение пути не пересекающего препятствия, отталкивающая потенциальная функция должна обладать высоким значением внутри препятствия и на его границе, и небольшим значением вне препятствия. Таким образом, во время локальной процедуры оптимизации пути, высокое значение потенциальной функции в центре препятствия выталкивает все точки пути из него, чтобы минимизировать стоимость пути. Потенциальное поле начинает резко меняться (уменьшаться) на границе препятствия, продолжает уменьшаться с расстоянием, когда точка удаляется от границы, и довольно быстро обращается в ноль в непосредственной близости от препятствия. Определим положение робота в момент времени *t* как конфигурацию *q(t)=(x(t), y(t))*. Тогда вклад одного круга (где круг является частью препятствия) в общую функцию потенциала отталкивания определяется следующим уравнением:

$$U_{rep}(q) = 1 + \tanh(\alpha(\rho - \sqrt{(x(t)-x)^2 + (y(t)-y)^2})), \quad (1)$$

где ρ - это радиус круга (препятствия) с центром в точке с координатами (*x, y*), а α - это эмпирически определяемый параметр, который отвечает за выталкивание пути из границ препятствия. Рисунок 1 (правое изображение) демонстрирует пример выбора параметра α=0.5 для среды с одним препятствием, которое образовано пятью пересекающимися кругами (левое изображение). Потенциальная функция имеет четко выделяющиеся пики на пересечениях кругов.

Топология *T(q)* – функция, учитывающая все *N* кругов окружающей среды и их влияние на робота на протяжении всего пути, которая определяется параметрическим уравнением на интервале [0,1]:

$$T(q) = \sum_{j=0}^{N-1} \int_{t=0}^{1} U^{j}{}_{rep}(q) \cdot \delta l(t) \cdot dt, \quad (2)$$

где *δl(t)* длина отрезка:

$$\delta l(t) = \sqrt{(x'(t))^2 + (y'(t))^2} \qquad (3)$$

Функция кривизны пути также определяется параметрическим уравнением:

$$R(q) = \sqrt{\int_{t=0}^{1} (x''(t))^2 + (y''(t))^2 \, dt} \quad (4)$$

Функция, которая учитывает длину пути (*L(q)*), суммирует длины всех сегментов пути:

$$L(q) = \int_{t=0}^{1} \delta l(t) \cdot dt \qquad (5)$$

Таким образом, итоговая стоимость пути состоит из суммы трех компонентов:

$$F(q) = \gamma_1 T(q) + \gamma_2 R(q) + \gamma_3 L(q), \qquad (6)$$

где $\gamma_{i=1..3}$ числовые значения влияния каждого компонента на общую стоимость маршрута. В частности, значение влияния потенциального поля $\gamma_{i=1} = \beta/2$, где β коррелирует с α из первого уравнения [6].

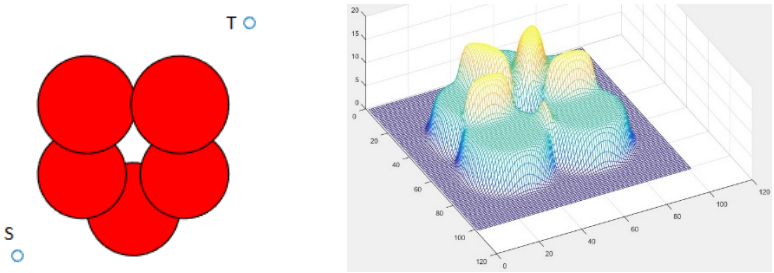

Рисунок 1. Пример среды с одним препятствием и начальная и целевая позиции (левое изображение). Потенциальная функция этого препятствия (правое изображение).

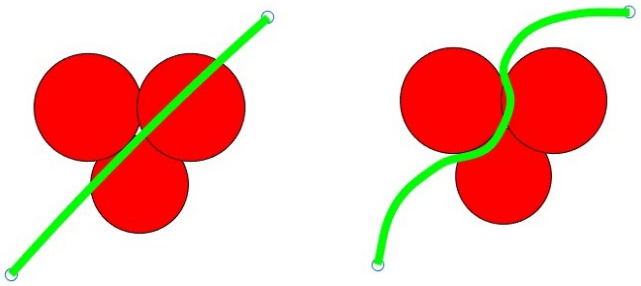

Рисунок 2. Расчет пути исходным алгоритмом: 1-ая (левое) и 18-ая итерации (правое изображение).

1.2. Алгоритм

Исходный сплайн-алгоритм планирования пути работает итеративно. Путь на первой итерации представляет из себя прямую линию между точками старта *S* и цели *T*. Сплайн определяется тремя точками: *S*, *T* и равноудаленной точкой, лежащей на прямом отрезке, соединяющем точки *S* и *T*. Дальнейшие итерации пути используют уравнение (6) и оптимизируется с помощью метода Нелдера-Мида [13] (симплекс-метод). Лучший результат (т.е. маршрут) каждой итерации служит начальным значением (маршрута) для следующей итерации.

Процедура оптимизации работает только с точками пути, которые определяют его сплайн, в то время как оценка стоимости пути учитывает все точки маршрута. Маршрут перестраивается на каждой итерации, используя информацию с предыдущего этапа, увеличивая количество точек сплайна на единицу и корректируя параметры целевой функции стоимости. Алгоритм завершается, если количество итераций превышает заданный пользователем предел, или если новая итерация не улучшает предыдущую, при этом увеличивается сложность сплайна.

1.3. Недостатки оригинального алгоритма

Исходный сплайн-метод позволяет получить гладкий путь без столкновений с препятствиями, если каждое препятствие аппроксимируется одним кругом или линия начальной итерации не проходит через пересечение кругов. Однако, когда препятствия аппроксимированы группой пересекающихся кругов, пересечения образуют локальные максимумы потенциального поля (Рисунок 1). Когда в такой среде начальный сплайн проходит через пересечение нескольких кругов, то целевая функция *F(q)* выталкивает сплайн из области пересечения и далее сплайн «застревает» в локальных минимумах, а следующий итерационный сплайн не может «перепрыгнуть» некоторые компоненты препятствия из-за локального характера процесса оптимизации. Рисунок 2 демонстрирует пример с одним сложным препятствием. Из-за множества пересекающихся кругов, которые генерируют собственные локальные отталкивающие потенциалы, соответствующее глобальное поле скрывает проход и позволяет совершить только локальную оптимизацию пути. Локальная оптимизация успешно устраняет локальные максимумы, но даже после 18 итераций, продолжавшихся в течение 23 минут, результирующий путь все еще пересекает препятствия и, следовательно, бесполезен для движения робота.

2. Дополнительный параметр безопасности пути

Исходный сплайн-алгоритм использует только три критерия в целевой функции (6): топология, гладкость и длина пути. В этом разделе вводится новый дополнительный критерий - минимальное расстояние от препятствий - и исследуем его влияние на результирующий путь. Подробнее, критерии, влияющие на путь, были разобраны ранее в статье [14]

Во время движения робота по карте с множеством препятствий, путь отдаляется от препятствий благодаря использованию критерия топологии *T(q)*. Однако, в некоторых ситуациях (например, при опасности обрушения стен во время поисково-спасательной операции после землетрясения) роботу требуется держаться на некотором безопасном расстоянии от препятствий. Если робот не может безопасно пройти между препятствиями, не нарушив границы безопасной зоны, он должен выбрать другой маршрут. Минимальное расстояние пути от препятствий рассчитывается по формуле:

$$M(C) = \min_{\forall t \in [0,1]} q(t), \qquad (7)$$

где *q(t)* это минимальное расстояние до препятствий в точке *t*:

$$q(t) = \min_{\forall c \in C} \sqrt{(x(t) - x(c))^2 + (y(t) - y(c))^2} - \tau(c) \qquad (8)$$

В уравнении (8) используются все круги из множества C, где *x(c)* и *y(c)* – координаты центра круга *c*, а *τ(c)* – его радиус. Следует указывать необходимое минимально приемлемое расстояние *d* до препятствий. Уравнение для этого критерия будет следующим:

$$X(d) = \omega^{d - M(C)}, \quad (9)$$

ω является эмпирически вычисляемым числом, которое предотвращает наложение пути на препятствия. Значение функции (9) равно единице, когда минимальное расстояние до препятствий *M(C)* равно безопасному значению *d*. Функция возвращает большое положительное значение, если путь проходит близко к препятствиям, и возвращает небольшое положительное значение близкое к нулю в случаях отдаления на безопасное расстояние от препятствий.

Целевая функция, которая объединяет все четыре критерия, определяется следующим образом:

$$F(q) = \gamma_1 T(q) + \gamma_2 R(q) + \gamma_3 L(q) + \gamma_4 X(d), \qquad (10)$$

Где γ_4 - весовой коэффициент, который устанавливает влияние критерия минимального расстояния на общую стоимость пути.

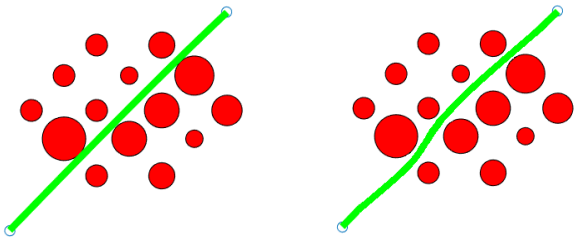

Рисунок 3. Построенный путь исходным алгоритмом (слева) и с новым критерием безопасности (справа).

Рисунок 3 демонстрирует влияние критерия на путь: в то время как при $\gamma_4 = 0$ этот критерии не вносят влияния (левое изображение), но с $\gamma_4 = 1$, *d=3* и *ω = 20* путь изменяется и становится более удаленным от препятствий (правое изображение); другие параметры получают значения $\gamma_1 = 1, \gamma_2 = 1$ и $\gamma_3 = 0.5$.

Однако, так как оптимизация маршрута выполняется только локально, влияние дополнительного параметра также локально. Путь остается в той же гомотопической группе даже в том случае, когда можно потенциально обойти какое-либо препятствие с другой его стороны. Далее в статье будет проанализирован метод умного обхода препятствий.

3. Метод, основанный на графе Вороного

Для создания хорошего начального сплайна, применяется алгоритм создания графа Вороного. Это помогает избежать недостатков оригинального подхода. Для создания этого графа требуется подготовить информацию о препятствиях; процедура выполняется в два следующих шага:

1) Сбор препятствий из пересекающихся кругов. Первоначально каждый круг отмечен как неиспользованный и имеет собственный индекс *i=1...M*, где *M* – число кругов в среде. Начнем с произвольного круга, запишем его к препятствию O_1 и пометим как использованный. Затем итеративно увеличиваем O_1, определяем все неиспользованные круги, которые пересекаются с кругами O_1, записываем их к O_1 и отмечаем как использованные. Итеративный рост O_1 продолжается до тех пор, пока в среде не останется больше неиспользованных кругов, которые пересекаются с O_1. Когда итеративный рост O_1 завершен, но остаются еще неиспользованные круги, произвольно выбирается другой свободный круг, записываем его к препятствию O_2 и повторяем процедуру роста. Объединение и распределение

препятствий завершается, когда не остается неиспользованных кругов. Например, на Рисунке 4 есть четыре препятствия, которые образованы группами кругов, а на Рисунке 5 в окружающей среде находятся три препятствия. При сборке препятствий каждая пара пересекающихся кругов (i, j) образует одну точку пересечения θ_{ij} в случае граничного касания и пару точек θ_{ij}, θ_{ji} в общем случае пересечения кругов.

2) Поиск внешних контуров препятствий множества $Obst = \{O_1, O_2, ..., O_k\}$, где k - количество препятствий в окружающей среде. Исходя из радиуса самого маленького круга внутри препятствия O_1, границы всех кругов, принадлежащих O_1, разбиваются на короткие отрезки длиной σ, соединенные через точки пересечения θ_{ij}, θ_{ji}. Таким образом, во время процедуры два сегмента соединяются, если между ними существует непрерывный путь, который построен из граничных сегментов. Эта процедура повторяется для каждого препятствия, и по ее завершению получим группу контуров. Если размер последнего множества превышает k, это указывает на наличие внутренних контуров. Чтобы избавиться от таких контуров, для каждого конкретного препятствия O_i вычисляется выпуклая оболочка; если у i-того препятствия имеется несколько контуров, удаляются контуры, которые находятся внутри выпуклой оболочки. К примеру, такая процедура успешно удалила три внутренних контура внутри сложных препятствий среды на Рисунке 4. Таким образом, получатся несколько невыпуклых многоугольников, по одному для каждого препятствия из *Obst,* и инкапсулируем их в квадратную карту.

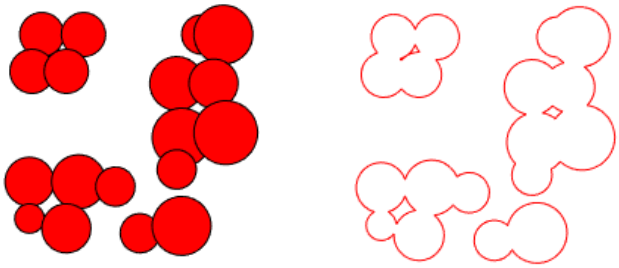

Рисунок 4. Среда с препятствиями и найденные контуры препятствий.

Далее, имея карту препятствий в виде многоугольников, строится граф Вороного, используя классический подход [15]:

1) От границ препятствий и карты с равными промежутками строятся лучи. На Рисунке 5 (изображение посередине) демонстрируется пример построения этих лучей в виде тонких синих линий, а толстые синие линии отображают границы препятствий и карты.

2) Вычисляются точки пересечения лучей от разных препятствий. Точки пересечения соединены отрезками. Соединяя отрезки, собирается граф Вороного (тонкие красные линии на рисунке 5, изображения посередине и справа)

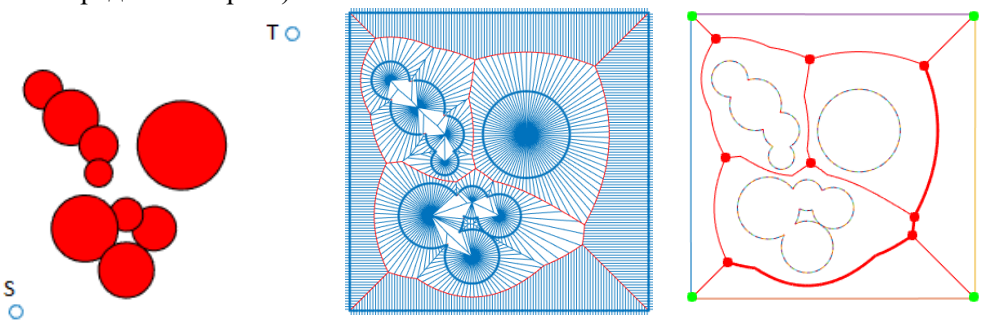

Рисунок 5. Среда с препятствиями (слева), процедура построения графа Вороного (посередине), путь на графе Вороного (справа).

После построения графа Вороного (VG), выбираются точки S' и T', ближайшие к начальной позиции S и целевой позиции T, соответственно. Отрезки $[S, S']$ и $[T, T']$ должны не пересекать препятствия. Далее, кратчайший путь между S' и T' находится внутри VG, используя алгоритм Дейкстры [16]. Так, построенный путь изображен толстыми красными линиями на Рисунке 5 (правое изображение). Любой

путь между точками (*S, T*) на графе Вороного *VG* гарантированно не пересекает препятствия и максимально безопасен в отношении расстояния от границ препятствий. Таким образом, он может обеспечить хороший начальный сплайн для исходного сплайн-метода [11].

Однако, нужно правильно выбрать начальные точки будущего сплайна, которые надлежащим образом отражают характеристики пути, но при этом позволяют избежать избыточной сложности сплайна.

На первом шаге выбирается точка *S* в качестве активной точки и находится точка VP_1 на пути, таким образом, что следующая за VP_1 точка на графе Вороного уже не видна из *S* из-за окклюзии препятствиями. *S* добавляется к множеству *L*, которое хранит точки графа Вороного, характеризующие данный маршрут и достаточные для его описания. VP_1 получает статус следующей активной точки. Затем снова ищем на пути самую дальнюю точку, видимую из VP_1 – точку VP_2. VP_1 добавляется в *L*, VP_2 становится следующей активной точкой, и процесс итеративно продолжается до тех пор, пока не станет видимой целевая точка *T*. Определим, что точка VP_{i+1} видна из точки VP_i, если они могут быть соединены прямым отрезком, который не пересекает ни одно препятствие окружающей среды. После завершения процесса точки множества *L* используются как точки для начального сплайна при запуске сплайн-метода [11].

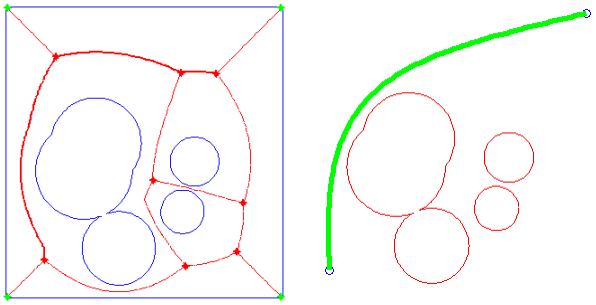

Рисунок 6. Путь на графе Вороного и результат итерационного алгоритма по этому пути.

4. Тестирование нового алгоритма и применение его на нескольких гомотопиях

Для проверки нового алгоритма, он был реализован в среде Matlab, и было проведено множество симуляций. Особое внимание было уделено случаям, когда исходный алгоритм [11] терпел неудачу. В тестах использовалась целевая функция (10) со следующим выбором параметров: $\gamma_1 = 1, \gamma_2 = 1, \gamma_3 = 0.5, \gamma_4 = 1$, безопасное расстояние $d = 3$ и параметр $\omega = 20$. Алгоритм смог успешно рассчитать маршрут во всех случаях, что было естественным следствием применения начального пути, найденного по графу Вороного, в качестве первого сплайна для итерационного алгоритма.

На Рисунке 6 изображена среда, в которой исходный сплайн-алгоритм не смог построить маршрут. Граф Вороного предоставляет безопасный путь без столкновений с препятствиями. Следовательно, этот путь гарантирует, что модифицированный сплайн-алгоритм вычислит конечный путь меньшим числом итераций. К тому же, моделирование эмпирически показало, что вычисления графа Вороного занимают приемлемо малое время. Например, для среды, показанной на Рисунке 6, расчет пути на основе графа Вороного занял всего 2 секунды. Общее время работы нового алгоритма уменьшилось в среднем в три раза по сравнению с исходным алгоритмом. Таким образом, окончательный путь в этом примере был рассчитан всего за 2 итерации в течение 2,5 минут в Matlab, в то время как исходный алгоритм на основе сплайнов провел 18 итераций и 38 минут, но не смог найти не пересекающий препятствия путь от начальной до целевой точки. Подобным образом, исходному алгоритму потребовалось 9 итераций и 15 минут, чтобы найти путь в среде, показанной на Рисунке 3, а новому алгоритму потребовалось только 4 итерации и 5 минут времени. В другом случае (Рисунок 4) исходному алгоритму не удалось найти путь, в то время как алгоритм на основе графа Вороного успешно выполнил задачу за 3 итерации и 2 минуты.

Граф Вороного *VG* содержит несколько гомотопических классов путей, и их многообразие зависит от сложности карты: чем больше отдельных препятствий есть на карте, тем больше количество гомотопических классов. В экспериментах было решено ограничить алгоритм не более чем 5-7 гомотопиями в расчетах. Затем пути в этих отобранных гомотопиях служили в качестве начального сплайна для нового сплайн-метода. Эта стратегия поиска была протестирована в нескольких средах. На каждом рассчитанном пути в каждой гомотопии вычисляется значение минимального расстояния (Уравнение (7)). Таким образом, отбрасываются гомотопии, в которых путь проходит в опасной близости от препятствий. Расчет всех подходящих гомотопий занимает в среднем от одной до трех минут, в зависимости от сложности окружающей среды. На Рисунке 7 (изображение слева) безопасные участки пути (соответствующие ребрам графа Вороного) отображены зеленым цветом – на этих участках робот способен находиться на более удаленном расстоянии от препятствий, чем разрешенная дистанция $d = 3$. Красные линии графа не являются безопасными и не включены в безопасные гомотопии. Вычисление пути (правое изображение) занимает всего три итерации и две минуты времени.

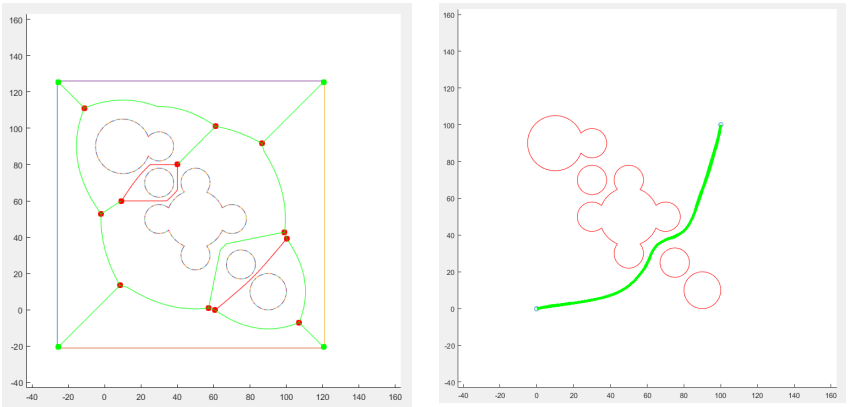

Рисунок 7. Опасные участки графа Вороного и построенный безопасный путь.

Исходный сплайн-метод не смог найти путь на карте, показанной на Рисунке 8. Алгоритм затратил 32 минуты на вычисления и завершил 18 итераций, прежде чем сообщить об ошибке пользователю. В то же время, модифицированный алгоритм смог рассчитать путь на двух гомотопиях на этой карте и сообщить, что один из вариантов пути (с правого изображения) более безопасен, чем другой (с левого изображения). Расчет пяти гомотопий проводился три минуты, и алгоритм выбирает самый безопасный путь, используя новый критерий.

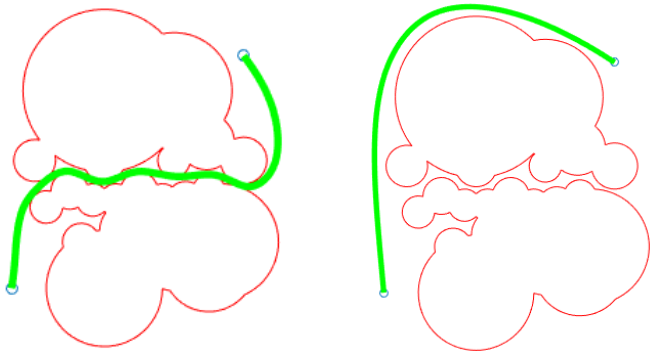

Рисунок 8. Построенные пути без учета и с учетом критерия минимального расстояния.

Заключение

В этой статье был представлен усовершенствованный метод с применением графа Вороного для навигации мобильных роботов, что позволило избежать проблемы локальных минимумов в статической плоской среде. Было проведено сравнение результатов нового умного сплайн-метода с оригинальным алгоритмом. Новый подход требует меньше итераций оптимизации, чем исходный алгоритм, благодаря умному выбору начального сплайна. На картах, где исходный алгоритм не мог найти существующий

путь между начальной и целевой позицией, его интеллектуальная версия оказалась успешной во всех симулированных тестах благодаря использованию графа Вороного.

В рамках будущей работы алгоритм будет проверен в средах большого размера, чтобы проверить приемлемость времени построения графа Вороного для более сложных случаев. Планируется расширить алгоритм для 3D-среды и добавить новые параметры целевой функции, связанных с 3D-поверхностями. Алгоритм будет реализован на языке C++ и собран в пакет ROS, после чего дополнительно проверен в экспериментальной работе для навигации гетерогенной группы роботов, выполняющей поисково-спасательные задачи.

СПИСОК ЛИТЕРАТУРЫ:

1. *E. Magid, T. Tsubouchi, E. Koyanagi, and T. Yoshida*. Building a search tree for a pilot system of a rescue search robot in a discretized random step environment. Journal of Robotics and Mechatronics, T. 23(4), 2011, c. 567.
2. *A. Pipe, F. Dailami, and C. Melhuish*. Crucial challenges and groundbreaking opportunities for advanced HRI. IEEE/SICE International Symposium on System Integration (SII), 2014, c. 12–15.
3. *A. Rosenfeld, N. Agmon, O. Maksimov, A. Azaria, and S. Kraus*. Intelligent agent supporting human-multi-robot team collaboration. Proceedings of the 24th International Conference on Artificial Intelligence, 2015, c. 1902-1908.
4. *Ronzhin, I. Vatamaniuk, and N. Pavluk*. Automatic control of robotic swarm during convex shape generation. International Conference and Exposition on Electrical and Power Engineering, 2016, c. 675–680.
5. *A. I. Panov and K. Yakovlev*. Behavior and Path Planning for the Coalition of Cognitive Robots in Smart Relocation Tasks, 2017, c. 3–20.
6. *E. Magid*. Sensor-based robot navigation. Master's thesis, Technion - Israel Institute of Technology, 2006.
7. *A. Buyval, I. Afanasyev, and E. Magid*. Comparative analysis of ROS-based monocular SLAM methods for indoor navigation. 9th International Conference on Machine Vision (ICMV), Nice, France, 2016.
8. *L. Tang, S. Dian, G. Gu, K. Zhou, S. Wang, and X. Feng*. A novel potential field method for obstacle avoidance and path planning of mobile robot. 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT), T. 9, 2010, c. 633-637.
9. *O. Khatib and B. Siciliano*, Springer handbook of robotics. Springer, 2016.
10. J. R. Andrews and N. Hogan. Impedance control as a framework for implementing obstacle avoidance in a manipulator. Master's thesis, M. I. T., Dept. of Mechanical Engineering, 1983.
11. *E. Magid, D. Keren, E. Rivlin, and I. Yavneh*. Spline-based robot navigation. IEEE/RSJ International Conference on Intelligent Robots and Systems, 2006. c. 2296-2301.
12. *S. Fleury, P. Soueres, J.-P. Laumond, and R. Chatila*. Primitives for smoothing mobile robot trajectories. IEEE transactions on robotics and automation, T. 11(3), 1995, c. 441-448.
13. *J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright*. Convergence properties of the nelder-mead simplex method in low dimensions. SIAM Journal on optimization, T. 9(1), 1998, c. 112-147.
14. Лавренов, Р. О., Афанасьев, И. М., & Магид, Е. А. (2016). Планирование маршрута для беспилотного наземного робота с учетом множества критериев оптимизации. *Третий Всероссийский научно-практический семинар «Беспилотные транспортные средства с элементами искусственного интеллекта*, 10-20.
15. *H. M. Choset*, Principles of robot motion: theory, algorithms, and implementation. MIT press, 2005.
16. *E. W. Dijkstra*. A note on two problems in connexion with graphs. Numerische mathematik, vol. 1(1), 1959, c. 269-271.

01.11.2018

MULTI-CRITERIA SPLINE-BASED ALGORITHM OF PATH PLANNING IN SEVERAL HOMOTOPIES

R.O. Lavrenov

This paper describes the development of a new combined spline algorithm that combines the methods of potential fields and the Voronoi graph (diagram), which finds the best way, including various homotopies. The optimality is achieved by a variety of optimization criteria that can be adjusted in real time. The created algorithm is compared with the basic spline

algorithm and shows the best results in speed, the number of iterations and the correctness of finding the path.

Keywords: **the support system of the generation of the solution, subject domain formalization, adaptation, case-based reasoning, neural network.**

Лавренов Роман Олегович – младший научный сотрудник, ассистент кафедры интеллектуальной робототехники ВШ ИТИС КФУ, e-mail: lavrenov@it.kfu.ru